Security and Communication Networks

Distributed Intelligence Empowered 6G Internet of Vehicles: Safety, Security and Applications

Lead Guest Editor: Chen Chen
Guest Editors: Bodong Shang, Ming Xiao, Licheng Wang, and Qingqi Pei

Distributed Intelligence Empowered 6G Internet of Vehicles: Safety, Security and Applications

Security and Communication Networks

Distributed Intelligence Empowered 6G Internet of Vehicles: Safety, Security and Applications

Lead Guest Editor: Chen Chen
Guest Editors: Bodong Shang, Ming Xiao, Licheng Wang, and Qingqi Pei

Contents

Hindawi
Security and Communication Networks
Volume 2023, Article ID 1416999, 11 pages
https://doi.org/10.1155/2023/1416999

Research Article

The Design of Vehicle Profile Based on Multivehicle Collaboration for Autonomous Vehicles in Roundabouts

Dongfa Cao,[1] **Zhandong Liu**,[1] **Chuangye Hu**,[1] **and Nan Ding**[1,2]

[1]*College of Computer Science and Technology, Xinjiang Normal University, Urumqi, China*
[2]*Key Laboratory of Intelligent Control and Optimization for Industrial Equipment, Dalian University of Technology, Dalian, China*

Correspondence should be addressed to Nan Ding; dingnan@dlut.edu.cn

Received 27 September 2022; Revised 22 October 2022; Accepted 2 February 2023; Published 8 April 2023

Academic Editor: Chen Chen

The collaborative strategy of vehicle-road-environment based on intelligent and connected vehicles (ICVs) to assist in driving vehicles safely and relieve traffic congestion has become an effective solution. This paper proposed a strategy for vehicle lane change and roundabouts traffic based on vehicle profile (VP) in combination with the driving characteristics of roundabouts. Initially, in order to solve the confusion problem of multisource heterogeneous data of ICVs in roundabouts, this paper defines VP to describe and characterize the multidimensional data of ICVs, so the data in ICVs can be further applied. Furthermore, the weights of relevant parameters in the VP are updated based on the random forest algorithm. In addition, the payoff function is designed for the lane change decision at the exit of roundabouts based on the VP and dynamic weights. Finally, the performance of the proposed algorithm is compared with other algorithms through the SUMO platform and three scenarios are used in the simulation verification, including traffic congestion, normal, and sparse. The experimental results verify the optimization effect of vehicle profile on roundabout traffic strategy and also show that this algorithm can effectively improve the efficiency of vehicle traffic in roundabouts. In particular, the efficiency and comfort of vehicles in roundabouts are effectively improved in normal traffic scenarios.

1. Introduction

As a kind of traffic facility that can effectively solve the problem of urban traffic congestion, traffic roundabout has been widely used in the urban traffic system. However, the roundabout is limited by the capacity of the road, and as the volume of traffic increases, traffic congestion worsens. Ensuring the efficiency and safety of the roundabout has become an important problem in the traffic system. Traffic lights have been added to control the movement of vehicles in large roundabout areas. Although it can optimize the problem, signal lights can also cause vehicles to wait and even block [1]. With the continuous development of intelligent connected vehicles [2, 3], the application of a multivehicle coordination strategy to roundabouts has become a new solution to solve the congestion problem of roundabouts and improve traffic efficiency.

Intelligent and connected vehicles (ICVs) use the internet of vehicles communication technology [4] to introduce more data information, such as vehicle information and road condition information to form an organic whole of vehicle-road-environment and realize the coordination of the three. How to effectively use the above-mentioned multiple heterogeneous data have become the basis and key to effective vehicle-road-environmental coordination.

The key to solving the congestion and the safety of traffic around the roundabouts is to optimize the strategies of lane change and traffic order. Therefore, the coordination problem of roundabouts can be detailed as the vehicle lane change decision at the intersection. Based on the defined payoff function, Nilsson et al. [5] proposed changing lanes if the lane change time and location are appropriate. Based on reduplicated game theory, Cheng et al. [6] defined a payoff function consisting of safety, rapidity, and control

indicators. Various factors related to vehicles are involved in lane change strategies, which caused the current situation of complexity when considering lane change decisions.

Combining the above-given problems, this paper researches the traffic efficiency optimization of roundabout scenes and the status quo in the multivehicle coordinated lane change, then proposes a method for cooperative control of vehicles under roundabouts based on the construction of VP. The main contributions of this paper are summarized as follows:

(1) Based on the idea of the user profile, the vehicle profile is constructed. Combining the data of vehicle, driver, and driving environment, this paper provides a basis for the further application of intelligent connected vehicle data in multivehicle cooperation problems.

(2) The dynamic weight of the vehicle lane changing payoff function is defined, which makes the weight of the function dynamically updated with the vehicle driving scene and driving state, so as to realize the scene adaptive optimization of the lane changing strategy.

(3) According to the revenue function of the vehicle profile and dynamic weight, a lane-changing decision algorithm (UPC) based on the user profile is established, and the overall traffic efficiency of the roundabout is optimized by the UPC algorithm.

2. Related Works

2.1. Vehicle Collaborative Decision Making. For collaborative decision-making of vehicles, Jin et al. proposed a lane-changing behavior decision model based on the Gaussian mixture hidden Markov model (GM-HMM) for the characteristics of drivers' lane-changing behavior [7], which can effectively simulate driving behavior. So et al. [8] proposed an emergency vehicle control strategy that achieved advantages in mobility and safety, and the advantages of the emergency vehicle control strategy can be maximized when signal preemption and autonomous driving control operate cooperatively. Bai et al. [9] established lane change models with different degrees of cooperation with the following vehicle in the target lane based on the characteristics of accelerated lane change, combined with vehicle kinematics and comfort requirements. It can achieve a safe accelerated lane change trajectory and meet the requirements of vehicle kinematics and comfort control. Ni et al. [10] established the feasibility of the cooperative lane change operation by establishing the gain function based on the excitation model. By comparing the lane change gain and lane keeping gain, we can judge whether the cooperation is feasible under the current conditions. The lane change process is divided into the lane change stage and the longitudinal vehicle distance adjustment stage.

Song et al. analyzed the game characteristics and game models existing in traffic signal control at intersections [11], analyzed the game characteristics in multiphase signal control at single intersections in detail, studied the multiperson cooperative game method in multiphase signal control at single intersections, and established the corresponding game model and solved it. Dewangan and Sahu [12] designed finite state machine models for straight and turning intersections, combined with safety judgment rules, and realized the safe passage of intelligent vehicles at intersections. Guo et al. combined with trajectory prediction [13], proposed a decision-making process (model) and multifactor driving behavior selection method for intelligent driving vehicles based on conflict resolution. Ali et al. developed a forced lane change model based on game theory (AZHW model) that can effectively capture forced lane change decisions with high accuracy [14]. The game-based lane change behavior modeling under incomplete information proposed by Yu et al. [15], whose model parameters can be learned and updated during the lane change. Leon Calvo and Mathar designed a cooperative formation scheme using the joint paradigm to increase traffic flow and stability [16], in which platoon formation is based on the method of alliance game theory. Jing et al. formulated the lane-changing problem as a Markov game between active and passive vehicles [17]. Ding derived the global optimal merging model based on a cooperative game to minimize the global revenue and achieve the optimal MS and trajectory. The fuel consumption, passenger comfort, and travel time in the merged control area were used as the revenue conditions [18].

2.2. Collaborative Decision-Making in Roundabouts. Since the appearance of roundabouts in the 1960s, researchers from many countries have tried to study and optimize the capacity of the roundabout [19]. Due to the right-of-way problem of vehicles and insufficient data [20], the efficiency of the roundabout will decrease with the increase in traffic volume [21] and other problems. Therefore, it has not been able to play its capacity advantage in the actual scene of high traffic flow [22]. But the further development of technologies such as the Internet of vehicles now offers a great opportunity to improve transportation efficiency.

Since deceleration, lane merging and lane changing at roundabouts are the main causes of congestion, the current research mainly focuses on route planning inside the roundabout and lane merging at the junction. And, they achieve this by controlling vehicle speed, traffic flow, etc. Silva and Grassi [23] make path planning by clothoid, circular arcs, and straight lines, whose curvature is piecewise linear and continuous as well. A continuous and smooth driving line is planned based on the linear variation of curvature relative to distance. Hidalgo et al. [24] proposed a method to solve the roundabout merging considering a nominal trajectory generated through Bézier curves combined with a model predictive control (MPC) to assure a safe future state.

For coordinated decision-making at the intersection, Hang et al. [25] designed and optimized a motion prediction module through model predictive control (MPC), and the payoff function of decision-making was defined with the consideration of vehicle safety, ride comfort, and travel

efficiency. Stackelberg game and grand coalition game approaches are adopted to address the decision-making of CAVs at an unsignalized roundabout. Tian et al. [26] proposed an algorithm based on a game-theoretic model; the algorithm shows the interactions between the ego vehicle and an opponent vehicle and adapts to an online estimated driver type of the opponent vehicle. Similar to the problems of Nilsson et al. [5] and Cheng et al. [6], their construction of the payoff function lacks the definition of the weights of the relevant factors, which makes the constructed payoff function not accurately characterize the vehicle payoff.

At the same time, Ye et al. explored the influence of different parameter values on high-precisiondecision-making in complex scenes [27], and Yu et al. [15] and Xu et al. [28] also verified that model parameters can be learned and updated in the process of lane change to bring better decision-making effects, and the decision optimization effect combined with different road weights in different road scenarios [29]. Therefore, in order to better characterize the vehicle state and optimize the decision-making performance, it is necessary to update the weight of the payoff function.

3. System Model and Problem Formulation

3.1. Vehicle Profile. In this paper, the method of the vehicle profile (VP) is used to construct (design) vehicle tags from five aspects, including driver information, vehicle information, vehicle driving status, driving behavior, and external environment, as shown in Figure 1. And, the VP is used to characterize the vehicle status, the feature types are shown in Table 1.

Based on the environment of Intelligent and connected vehicles (ICVs) built by roadside unit (RSU), this paper builds a vehicle profile. A roadside unit (RSU) is set in the roundabout where vehicles can obtain traffic information about themselves and surrounding vehicles and sets the data transfer to the ideal case: no delay no packet loss. When the vehicle enters the communication range, the vehicle immediately connects with the roadside unit and accesses the network. Data in vehicle driving are gathered and managed by the roadside unit, then the vehicle completes the construction of its vehicle profile by the data; in addition, the VP is used for the problems of collaborative decision-making or others, as shown in Figure 2.

Autopilot mentioned in the paper intelligent snatched automotive vehicle data not only included in the basic data, including infrastructure, environment, traffic data, road lane around size, location of the vehicle, road and the direction of motion, weather conditions, traffic intensity), the identity of the owner (driving experience, age), the state data (gestures, Eye position changes, etc.), and behavioral data (abnormal lane change frequency, driving style, etc.). The driver information and vehicle information are inherent information, and the external environmental data are collected by RSU and distributed to each node. The driving behavior can be obtained by visual collaborative analysis and other methods, and the vehicle running state can be obtained by onboard sensors and dynamically divided by combining the definition of safe driving in different scenarios in relevant laws and regulations.

Except for driving behavior, the data used for VP are all inherent factors related to the vehicle, and all information can be obtained directly through RSU. As for the representation of driving behavior, Murphey et al. [30] proposed the driver style identification coefficient Rdriver by taking advantage of vehicle acceleration and its standard deviation and proved that the proposed style coefficient could accurately describe the driver's driving style through experimental verification.

Based on this idea and the timeliness of the VP, this paper presents a simple representation of driving behavior style through vehicle spacing.

Definition 1. Driving behavior identification parameters:

$$U_{\mathrm{sp}} = \begin{cases} 1, & x \geq x_{\mathrm{safe}}, \\ \dfrac{x}{x_{\mathrm{safe}}}, & x < x_{\mathrm{safe}}, \end{cases} \tag{1}$$

where x is the vertical distance between two cars, x_{safe} is the safe distance between two cars at the current speed.

The construction of the VP provides a unified basis for the consideration of the relevant factors in the multivehicle cooperation problems so that the research on multivehicle cooperation problems can be taken and used on demand.

3.2. Label Weight Calculation. Breiman [31] proposed the random forest algorithm based on combining the Bagging method with the random subspace method. Random forest algorithm is an algorithm for classification and prediction, which uses the bootstrap resampling method to draw multiple samples from the original sample model decision trees for each bootstrap sample and then combine the predictions of multiple decision trees to arrive at the final prediction result by voting and has high prediction accuracy, good tolerance for outliers and noise, and is not prone to overfitting [32]. Applications of random forest in the field of assisted driving include the detection of trains ahead to avoid collisions [33] and the monitoring of driver emotions [34]. Considering that the process of random forest algorithm implementation is to set up different weights for different decision trees to complete the voting to arrive at the final result. Random forest can be used for the calculation and selection of feature weights for the dataset, although the random forest algorithm appears to classify and predict the data.

The common decision trees are divided into ID3, C4.5, and CART. ID3 divides attributes by information gain (IG) and recursively constructs decision trees, C4.5 constructs decision trees by gain rate, and CART constructs decision trees by Gini coefficient as a criterion. In this paper, the calculation of different label weights in the vehicle profile is completed by the random forest algorithm constructed with IG. Information gain is a feature selection method based on the information theory proposed by Harrington [35]; in other words, information gain is the change resulting from

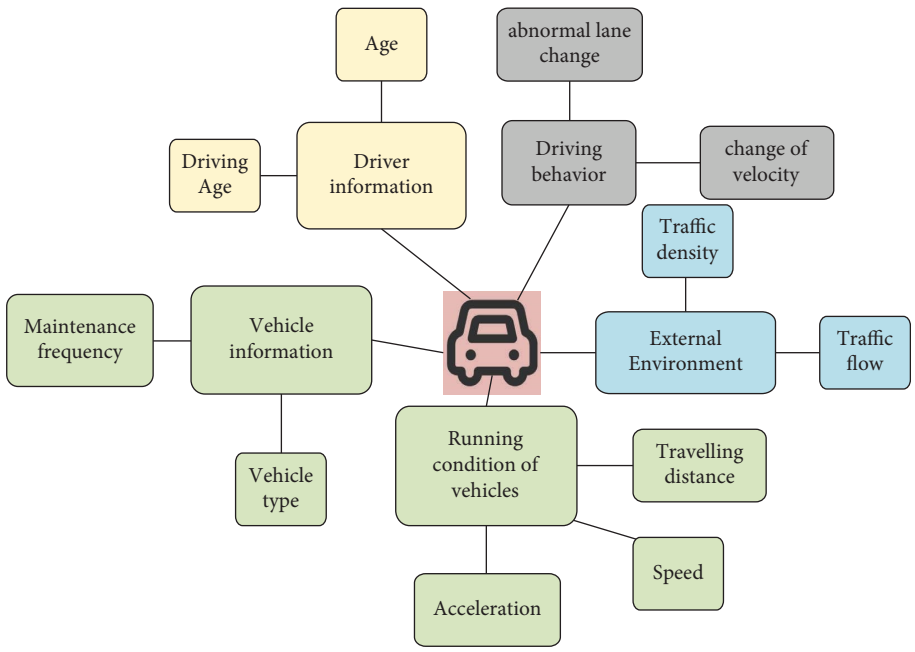

Figure 1: The label of vehicles.

Table 1: Feature type of vehicle profile.

Feature	Feature type
Traffic flow (A)	(1) Fewer vehicles (2) More vehicles (3) Traffic congestion
Light conditions (B)	(1) Morning (2) Daytime (3) Nighttime
Traffic control (D)	(1) No signal light (2) Signal light (3) Signal light damage
Driving distance (E)	(1) Close (2) Moderation (3) Far away
Change of driving angle (F)	(1) Little (2) Moderation (3) Large
...	...
Changes in acceleration (G)	(1) Gentle (2) Normal (3) Great
Speed (H)	(1) Slow (2) Normal (3) Fast
Space headway (J)	(1) Little (2) Moderation (3) Large
Age of driver (K)	(1) Young (2) Middle-aged (3) Old
Driving style (L)	(1) Gentle (2) Normal (3) Radical
...	...

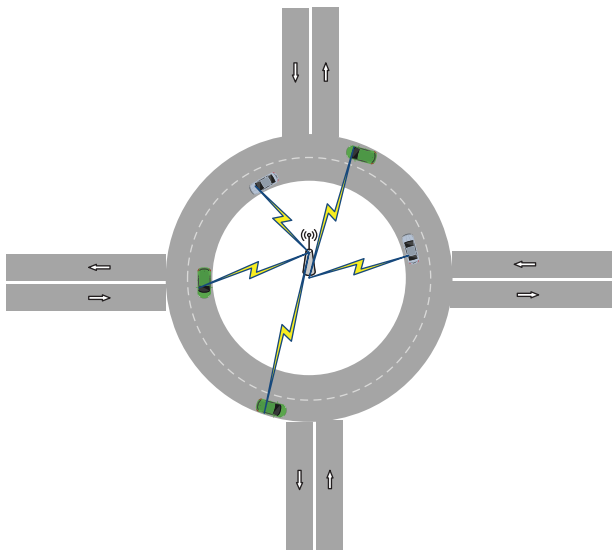

Figure 2: The roundabout scene.

the regularization of a data set. IG is calculated via entropy, which is the expectation of information. Then, the empirical entropy for the traffic dataset T is

$$H(T) = -\sum_{k=1}^{k} \frac{|c_k|}{|T|} \log_2 \frac{|c_k|}{|T|}. \tag{2}$$

The data types of dataset T are C_k, and the total number of datatypes is k. The subsets $(\{T_1, T_2, T_3, \ldots, T_i\})$ can be divided from the traffic dataset T based on feature A. Then, the empirical conditional entropy of A is

$$\begin{aligned} H(T|A) &= \sum_{i=1}^{n} \frac{|T_i|}{T}, \\ H(T_i) &= -\sum_{i=1}^{n} \frac{|T_i|}{T} \sum_{k=1}^{k} \frac{|T_{\mathrm{ik}}|}{|T_i|} \log_2 \frac{|T_{\mathrm{ik}}|}{|T_i|}. \end{aligned} \tag{3}$$

The information gain of A is

$$\mathrm{IG}(T, A) = H(T) - H(T|A). \tag{4}$$

Traditional user profiles generally complete personalized recommendations through user profile tag data, the correlation between tags, and the tag weight value. This paper mainly defines the weight of each component in the payoff function through the label weight value of the VP. That means we get the weight from the calculation of information gain.

If three factors that speed, acceleration, and travel time in the vehicle profile need to be taken into the construction of the payoff function, the weights of speed, acceleration, and travel time in the payoff function are

$$\begin{aligned} P_v &= \frac{\mathrm{IG}(T, v_{sv})}{\mathrm{IG}(T, a_{\mathrm{sv}}) + \mathrm{IG}(T, v_{\mathrm{sv}}) + \mathrm{IG}(T, \mathrm{time}_{\mathrm{sv}})}, \\ P_{\mathrm{acc}} &= \frac{\mathrm{IG}(T, a_{\mathrm{sv}})}{\mathrm{IG}(T, a_{\mathrm{sv}}) + \mathrm{IG}(T, v_{\mathrm{sv}}) + \mathrm{IG}(T, \mathrm{time}_{\mathrm{sv}})}, \\ P_{\mathrm{time}} &= \frac{\mathrm{IG}(T, \mathrm{time}_{\mathrm{sv}})}{\mathrm{IG}(T, a_{sv}) + \mathrm{IG}(T, v_{\mathrm{sv}}) + \mathrm{IG}(T, \mathrm{time}_{\mathrm{sv}})}. \end{aligned} \tag{5}$$

The weight of each part of the payoff function involved in the game of multivehicle coordination is usually used as a parameter in the construction process, then constantly adjusting the parameter to optimize the experimental results. However, in the process of continuous debugging, the importance degree of each part reflected by the weight value proportion is not accurate enough, and the importance degree of all kinds of income of the vehicle in its running process should be constantly changing.

Therefore, on the basis of constructing the vehicle profile, this paper introduces the weight value of each label into the construction process of the payoff function, so that the weight value of the payoff function can change dynamically during the driving process and describe the vehicle driving state more accurately. In this case, the weight in the payoff function is no longer a customizable parameter but participates as a variable calculated from the vehicle data.

3.3. Lane Change Scenarios. The vehicle lane change decision in the roundabouts is similar to the vehicle lane change decision at the highway intersection, in that they both have fixed exits, and the vehicle must complete its lane change action before the fixed exit. In general, the lane-changing behavior of vehicles is only related to the changing vehicle and its surrounding vehicles. However, the VP also has the problem of a cold start caused by insufficient behavioral data at the early stage of construction like traditional user profiles, making it difficult to accurately portray user characteristics. In the new scenario, the data before the vehicle enters the new decision point is processed and divided according to relevant standards, which serves as the basis for constructing the initial VP. The decision module can obtain the information of the VP label only after the VP is built. VP is constantly improved in the process of driving in new road conditions, including the update of weights. As shown in Figure 3, the dynamic update of the vehicle profile starts from the vehicle distance X from the start of the intersection and provides decision help for an intersection lane change.

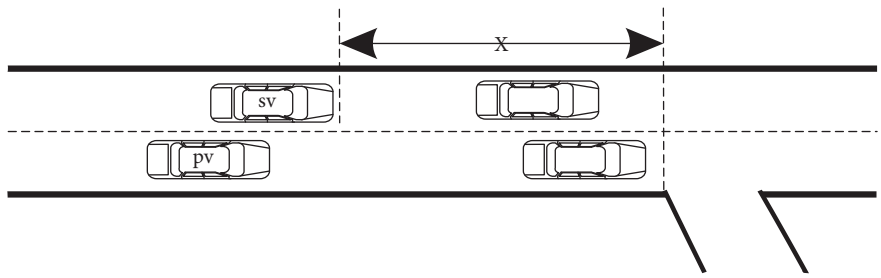

Figure 3: The traffic scene.

4. Lane Change Strategy Based on Vehicle Profile

4.1. Vehicle Payoff Function. In this paper, the payoff functions for the lane change vehicle SV and the yield vehicle SRV are defined as follows:

$$\begin{aligned} C_{\mathrm{sv}} &= P_v U_v(k) + P_{\mathrm{acc(sv)}} U_{\mathrm{acc(sv)}}(k) + P_x x(k), \\ C_{\mathrm{srv}} &= \alpha\left(P_x U_{\mathrm{sp}}(k)\right) + P_v U_v(k) + P_{\mathrm{acc}} U_{\mathrm{acc(srv)}}(k), \end{aligned} \tag{6}$$

subject to:

$$U_{\mathrm{acc(sv)}}(k) = \beta \frac{\left|a_{\mathrm{sv}}(k) - a_{\max}\right|}{a_{\max}}, \tag{7}$$

$$U_{\mathrm{acc(srv)}}(k) = \beta \frac{\left|a_{\mathrm{srv}}(k) - a_{\max}\right|}{a_{\max}}, \tag{8}$$

$$U_v(k) = \frac{\left|\left(v_{\mathrm{srv}}(k) - v_{\mathrm{sv}}(k)\right)\right|}{v_{\max}}, \tag{9}$$

$$\begin{aligned} &0 < k \le N \\ &N = \frac{x_{\mathrm{des}} - x_{\mathrm{pos}}}{v_{\mathrm{pos}}}, \end{aligned} \tag{10}$$

where x_{des} is the location of vehicle target intersections, x_{pos} is the current position of the vehicle, and the expected vehicle travel period N is obtained from the distance and the current speed.

The payoff function includes three parts. The first part (9) describes the benefits of security. When the speed difference between the two vehicles is larger, the safety benefit is higher. The second parts (8) and (10) describe the benefits of comfort when the acceleration of the vehicle is less, that is the difference between the vehicle acceleration ($a_{\mathrm{srv}}(k)$, $a_{\mathrm{sv}}(k)$) and the upper limit of the road acceleration ($a_{\max}$) tends to level off, the benefits of comfort are higher. For the lane change vehicle SV, the third part is the benefits of efficiency. The smaller the vehicle distance ($x(k)$) from the exit, the lower the probability that the vehicle will successfully change lanes, and the lower the benefit it will bring. And, for the yield vehicle SRV, the third part is the benefits of aggressive ($U_{\mathrm{sp}}(k)$). When the distance between two vehicles is greater than the safe distance, the greater the willingness of SRV vehicles to give way, and when the vehicle distance is smaller, the lower the willingness of vehicles to give way. α, β are all set to 1. And, variables such as $P_v, P_{\mathrm{acc(sv)}}, P_x$ are all obtained by calculating the weights in formula (5).

4.2. The UPC Algorithm. In the previous subsection, the payoff function of the vehicle was defined. Therefore, the multivehicle cooperation algorithm based on vehicle profile is shown in Algorithm 1.

For the inner lane vehicles exiting the roundabout, this algorithm obtains the weight coefficients of each part of the vehicle payoff function based on the vehicle profile and builds the vehicle payoff function based on this. The estimated vehicle payoff function value is obtained by the vehicle driving state, and the threshold value is set as its average value. For the inner lane vehicles whose payoff function value exceeds the threshold, it is forced to change lanes to the outer lane of the roundabout at the driving position beyond the threshold. The vehicles, namely, the inner lane vehicles, change lanes at the position where the impact on the surrounding vehicles is low and the revenue is high, so as to improve the lane-changing efficiency and exit efficiency of the vehicles at the exit of the roundabout, so as to improve the overall operational efficiency of the roundabout.

The state machine of the algorithm in this paper is shown in Figure 4, and the corresponding execution process is as follows:

S1: complete the initialization work, and collect vehicle data, when the vehicle reaches the critical position ($x == 50$), enter S2, if S5 exists, transmit vehicle data to S5.

S2: normalize the vehicle data set, and start the vehicle user profile update work after completion (enter S3).

S3: select the desired vehicle user portrait label, and enter S4.

S4: calculate the weight of the selected label, and enter S5.

S5: construct the vehicle income function and calculate the income value based on the vehicle data (S1) and the label weight (S4), enter S6.

S6: it is judged whether the income value satisfies the lane-changing condition. If the condition is met, the lane change is performed, and enter S2 to make the lane change decision of the next vehicle. If the condition is

```
Input: The data of vehicles
Output: The lane-changing position of the vehicle
Initializing data;
VP (data);
WHILE (TRUE):
  IF (0 < X < 50)
    P = CalculateWeight();
    Payoff (k) = GetPayoff (P, VP (data));
    IF (Payoff (k)>Payoff)
      ChangeLane (position);
    ELSE
    Payoff (k + 1) = GetPayoff (P, VP (data));
      IF (Payoff (k + 1) > Payoff)
        ChangeLane (position);
      END
END
```

Algorithm 1: (UPC).

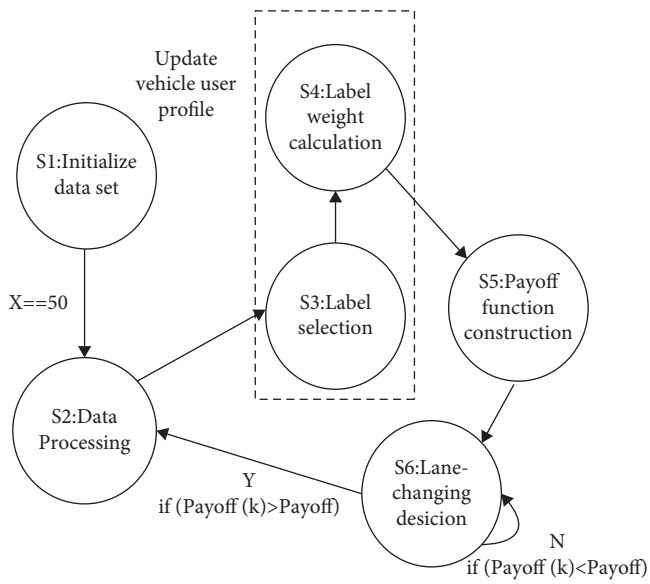

Figure 4: Finite state machine.

not met, enter S6 to make a decision at the next moment.

5. Evaluation

5.1. Experiment Setup. In this paper, we use SUMO as the simulation platform to simulate and compare the method proposed. The simulation scene is shown in Figure 5, and the data related to the scenario are shown in Table 2.

According to the vehicle generation probability of SUMO, this section is divided into three roundabout traffic scenarios. According to the provisions of the old and new traffic laws on the safe driving distance, when the vehicle generation probability of SUMO is 0.05 (maximum 5 vehicles per 100 meters) [36], the safe driving distance limit has been reached. Considering the safe driving distance, lane change has little impact on surrounding vehicles. Therefore, this paper divides SUMO vehicle generation probability into three round-island traffic scenarios based on vehicle driving safety spacing. They are a congestion scenario with a generation probability of 0.1 (up to 10 cars per 100 meters), a normal scenario with a generation probability of 0.08 (up to 8 cars per 100 meters), and a sparse scenario with a generation probability of 0.05 (up to 5 cars per 100 meters).

To compare the performance analysis, in addition to the UPC algorithm proposed in this paper, three algorithms are also introduced: (1) the built-in algorithm of the SUMO platform, (2) the lane-changing algorithm with the uniform weight of the revenue function (UW) [37], (3) the algorithm that focuses on driving comfort (HA), which is mainly based on the acceleration parameters to construct the vehicle payoff function [38], and (4) the algorithm (CRP) that

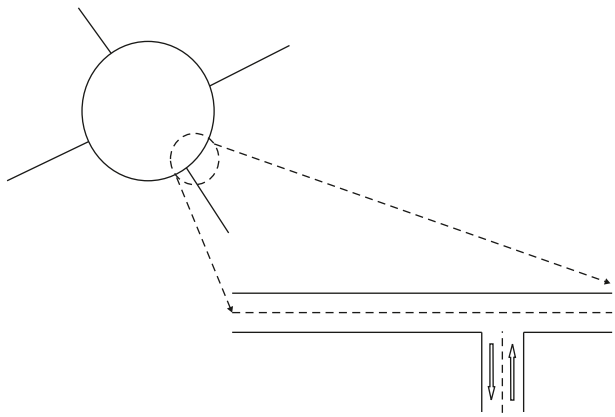

Figure 5: The simulation scene.

Table 2: Explanation of experimental parameters.

Parameter	Value
Accel (maximum acceleration: m/s^2)	2.5
Decel (maximum deceleration: m/s^2)	2.5
X (vehicle distance from intersection: m)	50
Speed (maximum speed: m/s)	13.89
Radius (m)	200

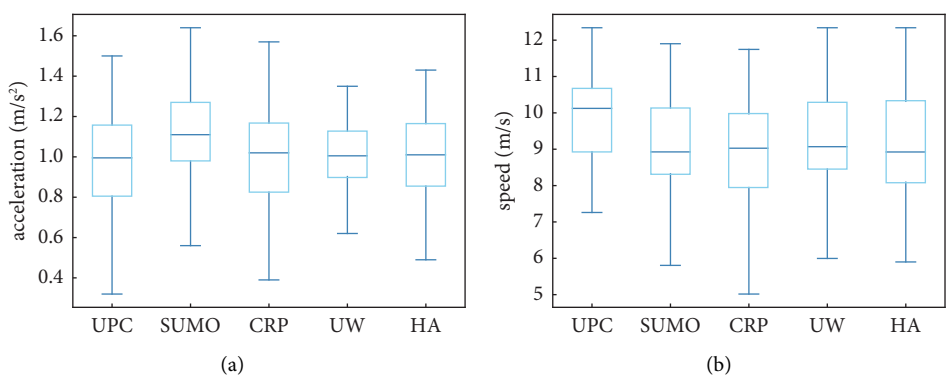

Figure 6: Main parameter analysis in normal scenarios. (a) Trend of acceleration. (b) Velocity distribution.

optimizes the overall revenue of multiple vehicles on the basis of the cooperative game [39].

5.2. Results Analysis in Normal Traffic Scenarios. Figures 6(a) and 6(b) and Figures 7(a) and 7(b) are the acceleration distribution diagram, the vehicle speed distribution diagram, and the vehicle traveling time distribution diagram of some selected vehicles, respectively. It can be seen from Figures 6(a) and 6(b) and Figures 7(a) and 7(b) that the lower and upper quartiles of the UPC algorithm are lower than those of the SUMO algorithm, and the acceleration distribution interval of the UPC algorithm is smaller in normal traffic scenarios. That means the vehicle driving stability of the UPC algorithm is better.

The lower quartile and the minimum and median values of the velocity distribution of the UPC algorithm in Figure 6(b) are also higher than those of the SUMO algorithm. In Figure 7(a), the vehicle travel time of the UPC algorithm is less than that of other algorithms. And, the vehicle speed of the UPC algorithm in Figure 7(b) is mostly distributed in the high range. Therefore, the vehicle traffic efficiency of the UPC algorithm is higher than that of the SUMO algorithm.

And, the distribution of acceleration, speed, and vehicle travel time of the UPC algorithm all show that the vehicle driving stability and vehicle traffic efficiency of the UPC algorithm are better than the CRP algorithm, the UW algorithm, and the HA algorithm.

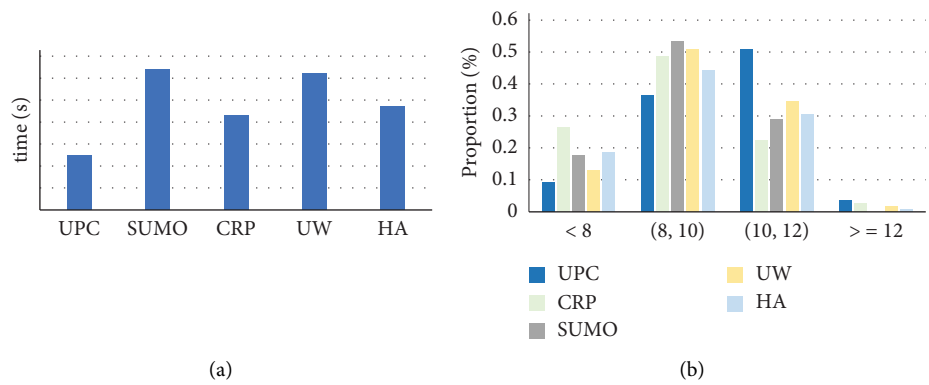

Figure 7: The results in normal scenarios. (a) Vehicles' travel time. (b) Velocity distribution interval.

Table 3: The results in different situations.

Parameters	Scene		
	Sparse scene	Normal scene	Congestion scene
Average speed (m/s)	10.6813	9.5963	8.5698
Travel time (s)	4334	7578	10051

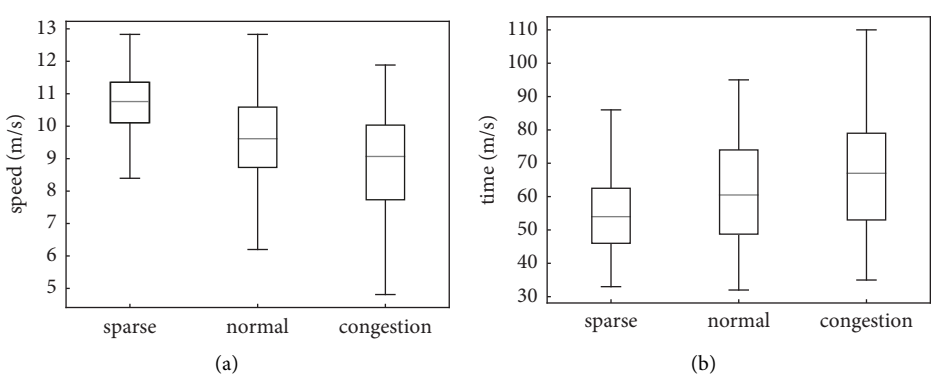

Figure 8: The results in different scenarios. (a) Velocity distribution. (b) Vehicles' travel time.

5.3. Results Analysis in Different Traffic Scenarios. The vehicle traffic efficiency optimization ability of the UPC algorithm proposed in this paper has been verified in the normal traffic situation of the roundabout. Therefore, we verify the difference in the traffic efficiency optimization ability of the UPC algorithm in different scenarios by changing the vehicle generation probability as shown in Table 3.

Figures 8(a) and 8(b) show the UPC algorithm's ability to optimize vehicle traffic efficiency in three different scenarios. Due to the difference in vehicle generation probability, the number of final vehicles is proportional to the generation probability, and the speed and travel time are inversely proportional to the generation probability.

6. Conclusion

For the problem of vehicle cooperation at the exit of roundabouts, this paper constructs the vehicle profile based on the idea of a user profile and designs the vehicle payoff function according to the characteristics of roundabout traffic scenarios and establishes the vehicle lane-changing cooperative strategy model. SUMO software is used to simulate the model, and the main conclusions of this paper are as follows.

The vehicle profile (VP) can objectively describe the driving state of the vehicle, and the label weight of the VP obtained by the random forest algorithm can solve the problem of manual debugging of the weight in the payoff function, making the weight as a variable rather than a parameter.

The vehicle payoff function is constructed based on the dynamic weight, and the UPC algorithm designed can effectively improve the traffic efficiency of vehicles in the roundabout scene.

However, the experimental scenario in this paper assumes that there is no delay and no packet loss and does not

consider the problems that may exist in real-time communication limitation and data packet loss in practical applications. Therefore, the actual situation of communication limitations and so on should be considered in subsequent studies.

Data Availability

The data used to support the findings of this study have not been made available because the data are relevant for followup studies.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant nos. 2021D01E20 and 2020D01A73 and the National Science Foundation of China under Grant nos. 62072071 and 62162061.

References

[1] Y. Cai, Z. Lv, J. Chen, and L. Wu, "An intelligent control for crossroads traffic light," in *Proceedings of the 2011 8th international conference on fuzzy systems and knowledge discovery (FSKD)*, vol. 1, pp. 494–498, Shanghai, China, January, 2011.

[2] Z. Q. Wei, H. Ma, Q. X. Zhang, and N. Ding, "Challenge and trend of intelligent vehicle networking based on apperceive-communication-computing fusion," *ZTE Technology Journal*, vol. 26, no. 1, pp. 45–49, 2020.

[3] C. Chen, C. Wang, T. Qiu, M. Atiquzzaman, and D. O. Wu, "Caching in vehicular named data networking: architecture, schemes and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2378–2407, 2020.

[4] N. Chen, T. Qiu, Z. Lu, and D. O. Wu, "An adaptive robustness evolution algorithm with self-competition and its 3D deployment for Internet of things," *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, pp. 368–381, 2022.

[5] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, and J. Fredriksson, "If, when, and how to perform lane change maneuvers on highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 4, pp. 68–78, 2016.

[6] C. Cheng, Z. Yang, and D. Yao, "A speed guide model for collision avoidance in non-signalized intersections based on reduplicate game theory," *IEEE Intelligent Vehicles Symposium*, vol. 4, pp. 1614–1619, 2018.

[7] H. Jin, C. Duan, Y. Liu, and P. Lu, "Gauss mixture hidden Markov model to characterise and model discretionary lane-change behaviours for autonomous vehicles," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 401–411, 2020.

[8] J. J. So, J. Kang, S. Park, I. Park, and J. Lee, "Automated emergency vehicle control strategy based on automated driving controls," *Journal of Advanced Transportation*, vol. 2020, Article ID 3867921, 11 pages, 2020.

[9] H. Bai, J. Shen, L. Wei, and Z. Feng, "Accelerated lane-changing trajectory planning of automated vehicles with vehicle-to-vehicle collaboration," *Journal of Advanced Transportation*, vol. 2017, Article ID 8132769, 11 pages, 2017.

[10] J. Ni, J. Han, and F. Dong, "Multivehicle cooperative lane change control strategy for intelligent connected vehicle," *Journal of Advanced Transportation*, vol. 2020, Article ID 8672928, 10 pages, 2020.

[11] H. Song, J. Zhu, and Y. Jiang, "Two-stage merging network for describing traffic scenes in intelligent vehicle driving system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25509–25520, 2022.

[12] D. K. Dewangan and S. P. Sahu, "Driving behavior analysis of intelligent vehicle system for lane detection using vision-sensor," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6367–6375, 2021.

[13] Z. Guo, D. Sun, and L. Zhou, "Game algorithm of intelligent driving vehicle based on left-turn scene of crossroad traffic flow," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 9318475, 9 pages, 2022.

[14] Y. Ali, Z. Zheng, M. M. Haque, and M. Wang, "A game theory-based approach for modelling mandatory lane-changing behaviour in a connected environment," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 220–242, 2019.

[15] H. Yu, H. E. Tseng, and R. Langari, "A human-like game theory-based controller for automatic lane changing," *Transportation Research Part C: Emerging Technologies*, vol. 88, pp. 140–158, 2018.

[16] J. A. Leon Calvo and R. Mathar, "Connected vehicles coordination: a coalitional game-theory approach," in *Proceedings of the 2018 European Conference on Networks and Communications (EuCNC)*, pp. 1–6, Ljubljana, Slovenia, June 2018.

[17] S. Jing, F. Hui, X. Zhao, J. Rios-Torres, and A. J. Khattak, "Cooperative game approach to optimal merging sequence and on-ramp merging control of connected and automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4234–4244, Nov, 2019.

[18] O. Boualam, A. Borsos, C. Koren, and V. Nagy, "Impact of autonomous vehicles on roundabout capacity," *Sustainability*, vol. 14, no. 4, p. 2203, 2022.

[19] H. K. An and D. S. Kim, "A review of roundabout capacity model," *KSCE Journal of Civil and Environmental Engineering Research*, vol. 41, no. 2, pp. 143–150, 2021.

[20] R. Wu, H. Jia, L. Yang, H. Miao, Y. Lin, and Y. Zhang, "A distributed trajectory control strategy for the connected automated vehicle in an isolated roundabout," *IET Intelligent Transport Systems*, vol. 16, no. 2, pp. 232–251, 2022.

[21] A. Danesh, W. Ma, C. Yu, R. Hao, and X. Ma, "Optimal roundabout control under fully connected and automated vehicle environment," *IET Intelligent Transport Systems*, vol. 15, no. 11, pp. 1440–1453, 2021.

[22] G. Ding, S. Aghli, C. Heckman, and L. Chen, "Game-theoretic cooperative lane changing using data-driven models," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3640–3647, Madrid, Spain, October 2018.

[23] J. A. R Silva and V. Grassi, "Path planning at roundabouts using piecewise linear continuous curvature curves," in *Proceedings of the 2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, pp. 1–6, Parana, Brazil, November 2017.

[24] C. Hidalgo, R. Lattarulo, J. Pérez, and E. Asua, "Hybrid trajectory planning approach for roundabout merging scenarios," in *Proceedings of the 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 1–6, Graz, Austria, November 2019.

[25] P. Hang, C. Huang, Z. Hu, Y. Xing, and C. Lv, "Decision making of connected automated vehicles at an unsignalized roundabout considering personalized driving behaviours," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 4051–4064, 2021.

[26] R. Tian, S. Li, and N. Li, "Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts," in *Proceedings of the 2018 IEEE Conference on Decision and Control (CDC)*, pp. 321–326, Miami Beach, FL, USA, December 2018.

[27] M. Ye, P. Li, Z. Yang, and Y. Liu, "Research on lane changing game and behavioral decision making based on driving styles and micro-interaction behaviors," *Sensors*, vol. 22, no. 18, p. 6729, 2022.

[28] C. Xu, W. Zhao, J. Liu, and F. Chen, "Decision making for highway complex scenario by improved safety field with learning process," *Proceedings of the Institution of Mechanical Engineers - Part D: Journal of Automobile Engineering*, vol. 236, no. 9, pp. 2012–2024, 2022.

[29] C. Chen, L. Liu, T. Qiu, D. O. Wu, and Z. Ren, "Delay-awaregrid-based geographic routing in urban VANETs: a backbone approach," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2324–2337, 2019.

[30] Y. L. Murphey, R. Milton, and L. Kiliaris, "Driver's style classification using jerk analysis," in *Proceedings of the 2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*, pp. 23–28, IEEE, Graz, Austria, November 2009.

[31] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] F. Kuang-Nan, J.-B. Wu, and J.-P. Zhu, "A review of technologies random forests," *Statistics and Information Forum*, vol. 26, no. 3, pp. 32–38, 2011.

[33] L. Wang, X. Yang, and W. Siu, "Learning approach with random forests on vehicle detection," in *Proceedings of the IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018*, pp. 1–5, Shanghai, China, January 2018.

[34] M. Jeong, J. Nam, and B. C. Ko, "Lightweight multilayer random forests for monitoring driver emotional status," *IEEE Access*, vol. 8, pp. 60344–60354, 2020.

[35] P. Harrington, *Machine Learning in action*, Simon & Schuster, New York, NY, USA, 2012.

[36] N. Ding, X. Meng, W. Xia, D. Wu, L. Xu, and B. Chen, "Multivehicle coordinated lane change strategy in the roundabout under Internet of vehicles based on game theory and cognitive computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5435–5443, 2020.

[37] Z. Q. Wang, *Research on Lane-Changing Strategy Based on Game Theory in the Mixed Driving Environment*, Chang'an University, Xi'an, China, 2020.

[38] D. W. Fu, *Research and Implementation of Intersection Multi-Vehicle Cooperative Driving Decision-Making Method Based on Swarm Intelligence*, Beijing University of Posts and Telecommunications, Beijing, China, 2020.

[39] Y. Cheng, Y. Zhao, R. Zhang, and L. Gao, "Conflict resolution model of automated vehicles based on multi-vehicle cooperative optimization at intersections," *Sustainability*, vol. 14, no. 7, p. 3838, 2022.

Hindawi
Security and Communication Networks
Volume 2023, Article ID 1363639, 13 pages
https://doi.org/10.1155/2023/1363639

Research Article

Traffic Safety Oriented Multi-Intersection Flow Prediction Based on Transformer and CNN

Tingting Fu,[1] **Qianwen Yu**,[2] **Haksrun Lao**,[3] **Peng Liu**,[1] **and Shaohua Wan**[4]

[1]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China*
[2]*HDU-ITMO Joint Institute, Hangzhou Dianzi University, Hangzhou, China*
[3]*Center of Engineering and Design, Chong Cheng Chinese School, Phnom Penh, Cambodia*
[4]*Department of Information Systems, King Abdulaziz University, Jeddah, Saudi Arabia*

Correspondence should be addressed to Haksrun Lao; haksrunlao@hotmail.com

Received 1 July 2022; Revised 22 August 2022; Accepted 31 January 2023; Published 15 February 2023

Academic Editor: Licheng Wang

Intelligent traffic signal control is one of the important means to ensure traffic safety. Effective signal control can make traffic flow fast and smooth, which first needs current and future traffic information. As the control of one intersection may affect adjacent intersections, this paper proposes to predict future traffic flow with consideration of multi-intersections. It can dynamically improve traffic conditions and reduce traffic congestion. Based on various nonlinear spatial relationships at correlated multi-intersections and the potential time-dependent relationship in traffic volume, a traffic flow prediction method named CNNformer which combines transformer with CNN, is proposed. The convolution neural network (CNN) and transformer are used to extract the spatial and temporal features of correlated multiple intersections. The learnable time code is embedded into transformer's location code, and the location information and time information are injected into the model to help it learn the time characteristics of traffic volume. This study also considers the impact of cyclical traffic flow pattern and proposes CNNformer$^+$. In the method, all of the data from the previous time window, as well as the data from the prior week and month, are correspondingly entered into the network. Finally, the output is generated through average pooling, realizing the learning of cyclical traffic flow characteristics. In the experiment, CNNformer$^+$ and the state-of-the-art traffic flow prediction methods are compared using simulated data. The results show that the proposed model outperforms the existing models in prediction accuracy and efficiency.

1. Introduction

Urban traffic is an important factor of urban function layout, which seriously affects the development of the social economy and the improvement of people's living standards. Due to the increase in consumers' purchasing power, there are more and more private cars, and the road density is increasing as well as the traffic safety concerns. Thus, in-depth research on traffic congestion and effective measures to improve traffic efficiency has become a research highlight.

According to the National Urban Car PARC Report, by the first half of 2019, as of June 2019, there were 250 million cars in China, and 66 cities across the country had more than 1 million cars. As the number of cars on the road rises, morning and evening rush hour overcrowding and minor traffic accidents become more and more common. Traffic congestion and accidents are very detrimental to urban development. They will not only increase the time needed for people's travel activities but also adversely affect people's work efficiency and life experience. Moreover, the congestion will lead to increased vehicle exhaust emissions and damage the environment.

In the face of a complex traffic environment, the prediction of traffic flow can improve the utilization rate of urban road resources and reduce the possibility of car accidents. It can also provide accurate traffic guidance information for urban traffic signal control [1], and the design of data dissemination techniques based on the travel of traffic participants [2]. Shortly, 6G will be crucial for communication, resource allocation, and compute

offloading [3, 4]. It will also help to collect data for traffic prediction. Through extracting characteristics from the obtained traffic data, traffic prediction methods can improve road safety and intelligent transportation constructions.

In the past decades, many scholars have put forward various traffic flow prediction models and achieved a series of theoretical and applied research results [5]. Most of these research methods are mainly based on statistical models or shallow machine learning methods to describe the evolution of traffic network flow, such as ARIMA [6], ANN [7], and SVR [8]. However, these methods can only be used when the data are relatively stable and linear. The actual traffic flow is extremely variable and will be affected by weather, date, traffic accidents, and other factors. Because the aforementioned elements have an impact on traffic flow, traffic-related time series data typically exhibit nonlinear or rapid change characteristics and are interdependent. In addition, due to the complex traffic network and the increasing number of vehicles, the spatiotemporal sequence traffic data collected based on the Internet of vehicles technology is large in scale and high in latitude, as well as lots of security threatens [9]. Therefore, the traditional methods are difficult to mine the deep relationship between traffic spatiotemporal series data and face a huge bottleneck when being applied in practice. In recent years, deep learning has been proven to be able to effectively extract depth features and has made breakthroughs in image processing, speech recognition, natural language processing, and other fields [10]. Due to the complex nonlinear spatiotemporal correlation between different traffic time series data, the deep learning method is a good choice for traffic flow forecasting tasks.

Because intersections are often interrelated, especially in cities with large traffic flow and short intersection spacing, the congestion of an intersection may affect the traffic distribution and capacity of the whole region. At the same time, the improvement of traffic congestion at a single intersection may aggravate the congestion at adjacent intersections and cannot accurately improve the overall traffic efficiency. Therefore, it is necessary to predict the traffic flow of multiple intersections. There are not many studies on multi-intersection traffic flow prediction, though, and most of the outcomes are not particularly good.

This paper presents a traffic flow prediction method based on transformer and CNN, called CNNformer. In addition, we have made improvements to CNNformer, added the learning of the periodic characteristics of traffic flow, and proposed CNNformer$^+$. The main contributions are as follows:

(i) The traffic flow prediction in this paper is aimed at multiple intersections. Many existing studies are conducted for a single intersection. The improvement of traffic congestion at a single intersection may aggravate the congestion at adjacent intersections, and cannot accurately improve the overall traffic efficiency.

(ii) This work proposes CNNformer, a new multi-intersection traffic flow prediction approach based on transformer and CNN. The flow data of correlated multiple intersections are constructed into a two-dimensional matrix with the shape of (number of intersections × number of lanes), and CNN is used to extract the spatial features of correlated multiple intersections. The transformer model is innovatively used in intersection flow prediction. Compared with LSTM, transformer can avoid recursion, which allows parallel computing, reduces training time, reduces performance degradation due to long-term dependence, and has better performance in prediction accuracy. In this paper, the learnable time code is embedded into the transformer's location code, and the location information and time information are injected into the model to help the model better learn the time characteristics of traffic volume.

(iii) In addition, this study also considers the weekly and monthly cycle trend of traffic volume, makes improvements on CNNformer, and proposes CNNformer$^+$. CNNformer$^+$ can learn the periodic characteristics of traffic flow.

2. Related Work

With the development of the intelligent transportation systems, many cameras, sensors, and other information collection equipments are deployed on the road. These equipments have accumulated a large number of traffic time series data with spatial information such as traffic flow, vehicle speed, and lane occupancy rate, providing a good data foundation for traffic flow prediction.

2.1. Shallow Machine Learning Methods. For a long time, to improve the congestion analysis and management decision-making ability of intelligent transportation, researchers have proposed a large number of traffic flow prediction models. Willams and Hoel used ARIMA [6] to model the traffic flow. This method is to model the single variable traffic flow sequence data as an autoregressive moving average process, to predict the traffic flow. Chan et al. proposed an ANN model using a mixed exponential smoothing strategy and Levenberg–Marquardt optimization to support short-term traffic flow prediction [7]. Alkheder et al. proposed to use a Bayesian joint neural network for short-term traffic flow prediction of adjacent intersections [11]. Alajali et al. proposed to use gradient enhanced regression tree (GBRT) and random forest (RF) to realize traffic flow prediction and suggested to use extreme gradient enhanced tree (XGBoost) algorithm to process traffic flow big data [12]. However, these methods have not achieved a completely satisfying result.

2.2. Deep Learning Methods. Due to the complex traffic network and the increasing number of vehicles, the usually collected spatiotemporal sequence data related to traffic flow has the characteristics of large-scale, high-dimensional, dynamic, and abrupt. Traditional methods are facing great challenges. The capacity of deep learning to capture

nonlinear depth characteristics has raised significant concerns [13]. It can also be used to automatically extract and learn deep-seated features in traffic time series data. Therefore, more and more scholars began to study the traffic flow prediction model based on deep learning [14]. For example, Tu et al. proposed a traffic congestion prediction model SG-CNN [15]. By analyzing the characteristics of traffic data, the model groups road segments. According to the correlation characteristics of road segments in the road network, the CNN model is used to extract the characteristics of road segment data, to realize the information sharing between road segments. Ren et al. proposed a new global-local time convolution network (GL-TCN) to predict traffic flow [16]. This new local time convolution mechanism can effectively capture the local characteristics of long-term traffic flow. At the same time, the influence of the periodicity of the global traffic flow on the local traffic flow law is considered.

Ma et al. proposed a traffic flow prediction model based on a bidirectional LSTM network. By improving the LSTM model, combining the characteristics of sequence data and the long-term dependence of BiLSTM, the bidirectional long-term memory network (BiLSTM) is integrated into the prediction model [17]. Du et al. proposed a deep irregular convolution residual LSTM network model (DST-ICRL) for traffic flow prediction [18]. To learn the spatiotemporal feature representation, the traffic flow between various roads in the road network is modeled as a multichannel matrix, which is comparable to the RGB pixel matrix of the image. Furthermore, deep learning methods, such as Deep Q-learning Network (DQN), can also be used to find optimal offloading strategies in intelligent-connected vehicles [19].

The intersection is the most complex part of the road network because it involves a variety of different objects, such as vehicles and pedestrians. With the increase in traffic demand, the problem of traffic congestion at urban intersections is becoming more and more serious. The short-term traffic flow forecast of intersections has also been the subject of numerous corresponding studies. For example, Qu et al. established a two-layer superposition model based on intersection short-term traffic flow prediction by integrating k-nearest neighbor (KNN) and Elman neural network modeling methods [20]. Kim and Jeong proposed a collaborative traffic signal control method based on multi-intersection traffic flow prediction (TFP-CTSC) [21]. Li et al. proposed a new deep intersection spatiotemporal network (DISTN) for traffic flow prediction. Considering the spatial and temporal characteristics of the convolutional neural network (CNN) and long-term and short-term memory (LSTM), the depth learning method was applied to intersection traffic volume prediction [22]. Furthermore, digital twins have been used to facilitate the design, evaluation, and deployment of IoV-based systems [23, 24]. However, the research is still in an initial stage.

3. Methodology

In terms of traffic flow prediction, the camera on the road is usually used to count the number of cars passing by. If multiple intersections in a certain area are considered, data from different cameras will contain geolocation and time information. Therefore, we can regard the traffic flow prediction problem as a spatiotemporal sequence problem, namely, we can use the time and space information contained in the data to predict the traffic flow of different intersections. The structure of the model is shown in Figure 1. The input of CNNformer $^{+}$ contains the traffic flow data of three time windows, which are the traffic flow data of the previous time window $(X_{t-H}, X_{t-H+1}, \ldots, X_t)$, the simultaneous data of the week before the previous time window $(X_{t-H-\text{week}}, X_{t-H-\text{week}+1}, \ldots, X_{t-\text{week}})$, and the simultaneous data of the previous month in the previous time window $(X_{t-H-\text{month}}, X_{t-H-\text{month}+1}, \ldots, X_{t-\text{month}})$. Each time window contains H time steps, and the traffic flow data of each time step can be described as a two-dimensional matrix. The three input time windows are processed separately, that is, to stack the H two-dimensional matrices in the data of each time window and input them to CNN. After using CNN to extract the spatial features of data, the convoluted data are input into transformer. After using transformer to extract the time characteristics of data, the data of all time steps will be output. Then, it stacks the outputs $(Z_{\text{now}}, Z_{\text{week}}, Z_{\text{month}})$ of the three time windows and puts them into the average pooling layer. The final output of the model is the predicted traffic flow in the next time window.

3.1. Extracting Spatial Features Using CNN. Because the intersections are often interrelated, the upstream and downstream intersections may affect the traffic flow prediction of the target intersection. In this paper, CNN is used to extract the spatial features of associated intersections. The input data has the following shape: $[H, N, D]$, where H stands for the number of time steps, N for the number of intersections, and D for the quantity of traffic flow directions at each intersection, where D is equal to 12.

With the great success of convolutional neural network in the field of image processing, other fields are also trying to use the method of deep learning to solve practical application problems. In the field of traffic flow prediction, because the traffic flow based on region or station can be organized into a two-dimensional vector or a one-dimensional vector, it is considered as an effective method to mine the spatial characteristics of traffic volume data using the convolution neural network.

For instance, in time step t, the historical flow data of a given road network can be described as a matrix as follows:

$$X_t = \begin{bmatrix} X_t^{1-1} & X_t^{1-2} & \cdots & X_t^{1-12} \\ X_t^{2-1} & X_t^{2-2} & \cdots & X_t^{2-12} \\ \vdots & \vdots & \ddots & \vdots \\ X_t^{N-1} & X_t^{N-2} & \cdots & X_t^{N-12} \end{bmatrix}. \quad (1)$$

For each element in the matrix, the superscript format is (intersection number - traffic flow direction number), and the subscript represents the time step t. Each row of the matrix represents the traffic flow of all traffic flow directions

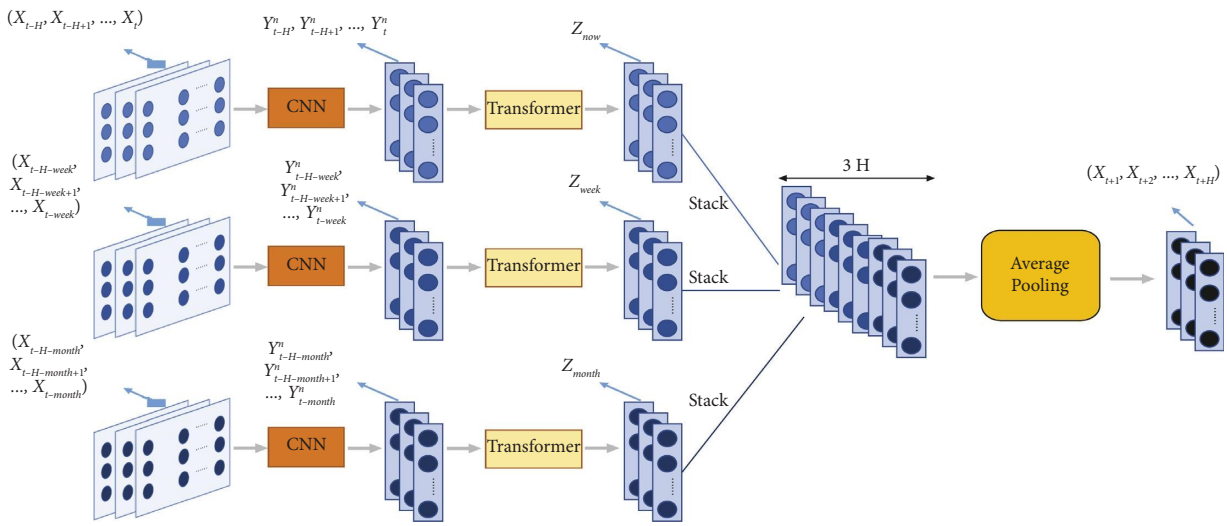

Figure 1: The CNNformer$^+$ model structure.

at time t at the nth target intersection. Each intersection has 12 traffic flow directions. Therefore, there are 12 columns of traffic flow data, and each column of the matrix represents the traffic volume in a certain traffic flow direction from intersection 1 to intersection N.

When the data of a time step can be described as a matrix, it is easy to think that the matrix can be used as the input of CNN. The convolution model in this paper is shown in Figure 2. The spatial features of associated intersections are extracted by using two-dimensional convolution layers with a convolution kernel size of (2, 2) and padding size of (2, 1). After convolution, a ReLU and Dropout layer are added.

The output X of the Nth convolution layer at time t is X_t^N, it will then pass through a residual connection. Finally, through the full connection layer, it is transformed into a one-dimensional spatial eigenvector Y_t. This vector is used as the input of the transformer network to capture the time correlation.

The output shape of CNN is $[H, M]$, where M represents the sum of traffic flow directions at each time step and all relevant intersections, and H represents the number of time steps.

3.2. Extracting Time Characteristics Using Transformer. The task of predicting traffic flow is a typical time series prediction challenge that uses historical observation data to forecast future traffic flow data. Since transformer is an excellent sequence model, this paper takes the output of CNN as the input of transformer and uses transformer to extract the time characteristics of traffic flow.

Currently, the majority of tasks involving traffic flow forecasting uses RNN and its derivatives, LSTM and GRU. RNN and its variants must process data in sequence during training. The calculation of time step t depends on the calculation result at time $t-1$, so parallel training is not possible. In addition, the coding of the traffic flow by RNN and its variants is only retained in the next time step, which

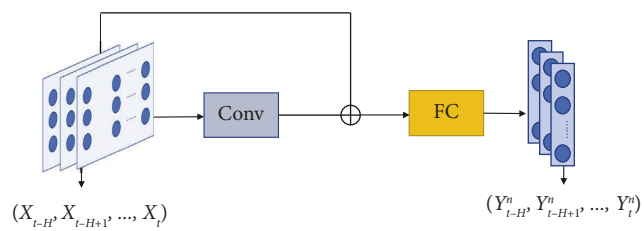

Figure 2: Convolution model structure.

means that the coding of the current time step only strongly affects the representation of the next time step, and its influence will disappear soon after a few time steps. Although the structure of gate mechanisms such as LSTM alleviates the problem of long-term dependence to some extent, LSTM is still powerless for particularly long dependencies. The transformer model can avoid recursion, allows parallel computing to reduce training time, and reduces performance degradation caused by long-term dependency. Compared with RNN and variants, the transformer model has stronger structural flexibility and versatility, and can capture a wider range of information relevance. In addition, in the NLP field, the transformer model processes sentences in a nonsequential manner, and sentences are processed as a whole rather than word by word.

The transformer does not rely on past hidden states to capture dependencies on previous words but processes a sentence as a whole, so there is no risk of losing or forgetting past information. Based on the abovementioned advantages, this paper attempts to apply transformer to the task of traffic flow prediction.

The input of transformer is a sequence of spatial eigenvectors containing H time steps, expressed as $(Y_{t-H}, Y_{t-H+1}, \ldots, Y_t)$, where Y_t is the spatial eigenvector output from the flow data of time step t after n convolution layers, where $t-H$ to t is the historical time step. The network is trained to predict the traffic flow of all associated intersections in the next H time steps.

Transformer is a seq2seq model. The encoder layer receives input and the decoder layer obtains output.

3.2.1. Encoder. The encoder layer of transformer includes two sublayers:

(i) The first sublayer is a multihead attention, which is used to calculate the input Self-Attention.

(ii) The second sublayer is feed forward, which is a simple fully connected network.

After each sublayer, the residual network is simulated, and the results of each sublayer are displayed as follows:

$$\text{LayerNorm}(x + \text{Sublayer}(x)), \tag{2}$$

where $\text{Sublayer}(x)$ represents the mapping of the sublayer to the input X. To ensure full connection, the dimensions of the output of all sublayers and embedded layers are the same.

The structure of the encoder layer is shown in Figure 3. The encoder input consists of the following three parts:

(i) Input embedding: In the original transformer model, the input of the model is a high-dimensional eigenvector. The feature vector is obtained by converting the input text through word embedding method such as Word2Vec [25], which is called an embedded vector. This paper uses the full join layer to replace the word embedding method to encode the input data. After the full join layer, the shape of the input data becomes $[H, E]$, where H represents the number of input time steps and E represents the feature size of the input data.

(ii) Position encoding: Transformer adds an additional vector positional encoding to the input of the encoder layer. The dimension of this vector is the same as that of the embedded vector, which is used to provide relative position information. This vector can determine the position of the current time step in the time window, and the transformer can learn the position information of the time step through this vector. The formula of the position code is shown as follows:

$$PE(pos, 2) = \sin\left(pos/10000^{2i}/d_{\text{model}}\right), \tag{3}$$

$$PE(pos, 2i + 1) = \cos\left(pos/10000^{2i}/d_{\text{model}}\right), \tag{4}$$

where pos refers to the position of the current time step in the time window, i refers to the subscript of each value in the vector, and d_{model} refers to the size of the input dimension. When pos is an even number, Sine coding is used; when pos is an odd number, Cosine coding is used.

(iii) Global time encoding: Based on the transformer model, this paper not only uses location coding for local location embedding but also takes into account the effectiveness of timestamp information in practical applications. The location codes are extracted from the timestamp corresponding to time series data.

The calculation of global time encoding is shown as follows:

$$GTE = FC\left(X_{\text{mon}}, X_{\text{dow}}, X_d, X_h, X_{\text{min}}\right), \tag{5}$$

where X_{mon} refers to the month location embedding, X_{dow} refers to the day of week location embedding, X_d refers to the day location embedding, X_h refers to the hour location embedding, and X_{min} refers to the minute location embedding. These five vectors are combined and input into the full connection layer for coding to generate a learnable embedding.

Finally, the model adds the abovementioned three embedded vectors and sends them to the next layer as input.

A multihead attention is equivalent to the integration of M Self-Attention. The specific process of Self-Attention is as follows:

(i) Self-Attention will use the input embedded vector to calculate three new vectors. The dimension of the vector is the same as that of the embedded vector. These three vectors are named as Query, Key, and Value, respectively. These three vectors are obtained by multiplying the embedded vector with a matrix, which is randomly initialized. The dimension of the matrix is $[64, E]$, and E represents the characteristic size of the input data.

(ii) Calculate the score of Self-Attention, which determines the degree of attention paid to the input data of other time steps when the model encodes one-time step data at a certain position. The fractional value is calculated by point multiplication of Query and Key.

(iii) Next, divide the result of point multiplication by a constant. The constant value selected in this paper is 8, which is the root of the first dimension of the matrix. Then, do a Softmax calculation on the obtained results. The result is the correlation between each time step data and the time step data at the current location.

(iv) Finally, use the result to multiply the value to get the Self-Attention Value.

This method of determining the weight distribution of values through the similarity between Query and Key is called scaled dot product attention. The calculation formula is shown as follows:

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{6}$$

where d_k represents the dimensions of Query, Key, and Value vectors.

A multihead attention is to perform the process of scaled dot-product attention M times, in which not only one group of Q, K, and V matrices is initialized, but M groups are initialized, and then M matrices are output.

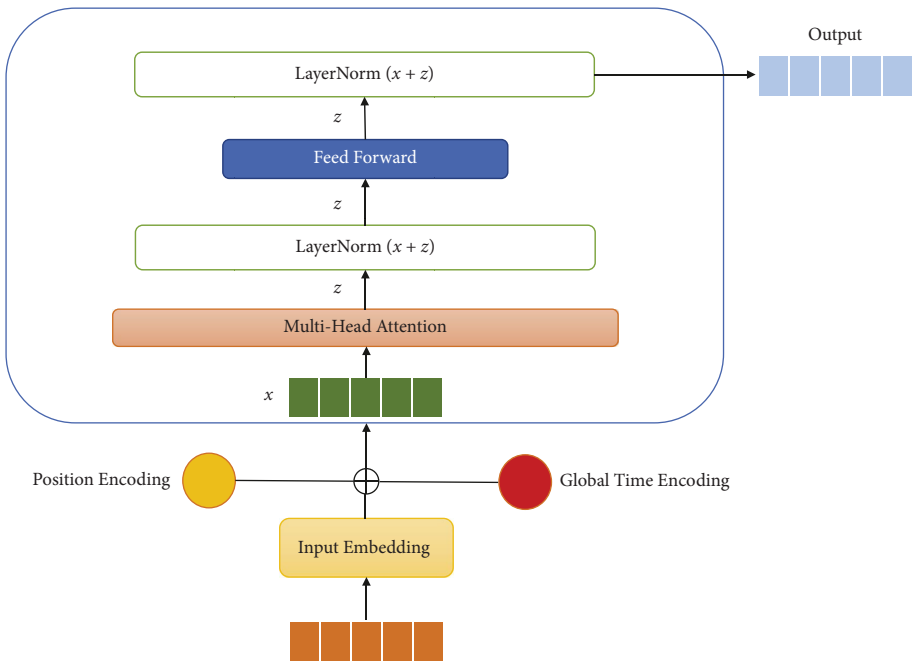

Figure 3: The transformer encoder layer.

However, the feed forward neural network cannot input multiple matrices. Therefore, M matrices need to be reduced to one. The precise method entails joining M matrices to create a large matrix, multiplying this large matrix by a weight matrix with the random initialization, and then obtaining the final matrix.

In the transformer, each sublayer will be followed by an incomplete module, and there is a layer normalization. There are many normalization methods, but the purpose of each method is to normalize the input data to achieve the effect that the mean value is 0 and the variance is 1. The data should be normalized before entering the activation function so that the input data do not fall in the saturation region of the activation function.

Unlike batch normalization, which calculates the mean and variance in the batch direction, layer normalization calculates the mean and variance on each sample. Therefore, layer normalization is usually used to normalize the sequence model.

3.2.2. Decoder. The transformer decoder layer includes three sublayers.

(i) The first sublayer is masked multihead attention, which is also the Self-Attention of calculation input. However, since future information cannot be known at the time of generation, it is necessary to mask future information. For a sequence, suppose the time step is t, the decoding output should only depend on the output before t, not after t. Therefore, mask operation is required.

(ii) The second sublayer is encoder-decoder attention. The output of the encoder layer and the output of the masked multihead attention sublayer are used for attention calculation.

(iii) The third sublayer is feed forward, which is the same as the encoder layer.

The structure of the decoder layer is shown in Figure 4. The traffic flow of the input decoder layer is composed of a part of the historical data that is close to the predicted data and an empty vector. The length of the empty vector is the length of the data to be predicted. The encoder layer uses the same coding technique for input traffic volume.

The masked multihead attention sublayer of the decoder layer needs to use a mask so that the decoder cannot see future information. The specific method is to generate an upper triangular matrix, the values of which are all 0 s, and apply this matrix to each sequence to achieve the purpose of covering.

The encoder-decoder attention sublayer of the decoder layer uses the output information of the encoder to calculate the content of the current decoded output. The difference between this part and Self-Attention lies in the three vectors of Q, K, and V. Q is the attribute of the decoder, while K and V are the last output K and V of the encoder layer. The calculation method of attention is the same as that of Self-Attention. Through this method, the encoder can capture the output information of the encoder.

3.3. Learn the Periodicity of Traffic Flow Using Average Pooling. When the decoder layer is completely executed, the final output of the three time windows is Z_{now}, Z_0, and Z_{month}. Then, we stack these three vectors and input them to the average pooling layer. The calculation process of average pooling is as follows:

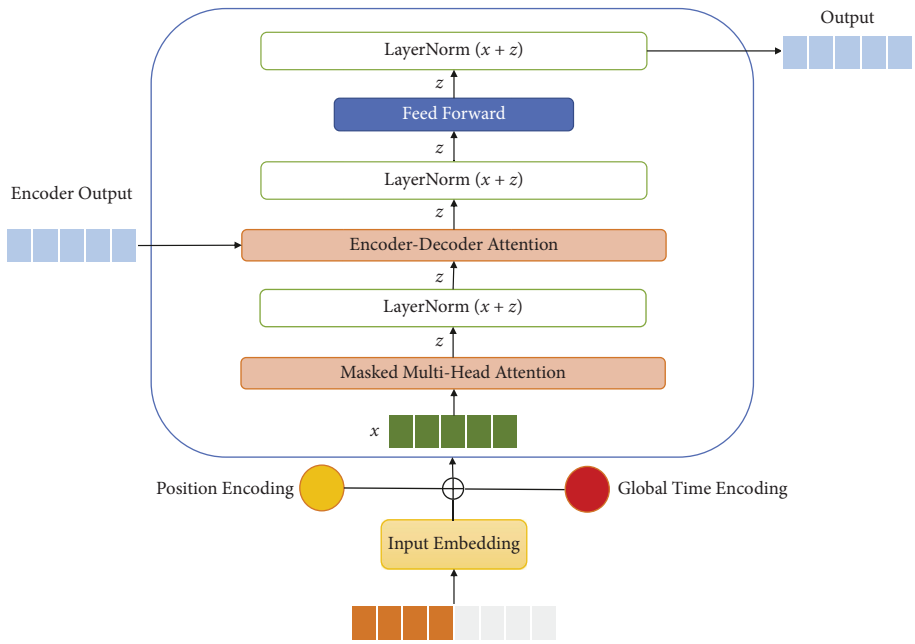

FIGURE 4: The transformer decoder layer.

$$\hat{q}_{t+1} = \text{AvgPooling}(Z_{\text{now}}, Z_{\text{week}}, Z_{\text{month}}), \tag{7}$$

where $\hat{q}_{t+1}$ represents the predicted flow data. As shown in Figure 5, average pooling involves combining feature points from different neighborhoods and averaging their values to create new features. Compared with the full connection layer, the average pooling can greatly reduce the network parameters, thus reducing the overfitting phenomenon.

The final output of the average pooling layer is the predicted traffic volume of the next time window. Gaussian error linear element (GELU) is used as the activation function of the average pooling layer. It is a high-performance neural network activation function because the nonlinear change of GELU is a random regular transformation mode that meets the expectation, and the formula is as follows:

$$xP(X \leq x) = x\Phi(x), \tag{8}$$

where $\Phi(x)$ refers to the cumulative distribution of the Gaussian normal distribution of x. GELU introduces the idea of random regularity in activation, which is a probabilistic description of neuron input, and is more intuitive and natural.

3.4. Loss Function. The loss function, also known as the error function, is used to measure the operation of the algorithm. The loss function is shown as follows:

$$L(\alpha) = Loss(\hat{q}_{t+1} - q_{t+1}), \tag{9}$$

where α represents the learning rate, Loss() represents the loss function, $\hat{q}_{t+1}$ represents the predicted flow data, and q_{t+1} represents the actual flow data. The error between the anticipated traffic flow in the following time window and the actual traffic flow in that time window is measured using the loss function to determine how closely the predicted output value is to the actual value.

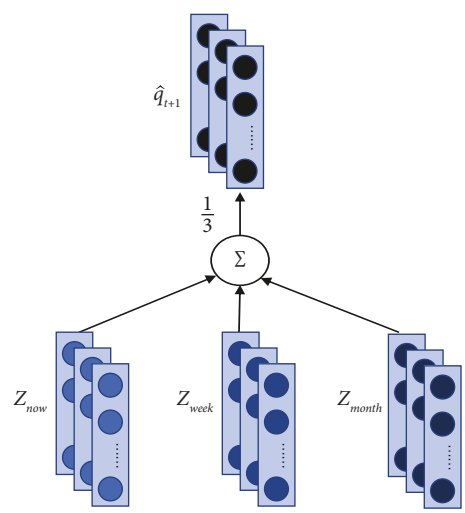

FIGURE 5: Architecture of average pooling.

3.5. Optimization Algorithm. The application of machine learning is a process highly dependent on experience. With a large number of iterations, many models need to be trained to find the right one. When training a neural network, we frequently employ a large data collection, which will cause the training time to be extremely slow. Therefore, using an appropriate optimization algorithm can effectively improve the speed of the training model. Gradient descent is a method to find the objective function, that is, to minimize the loss function. It uses gradient information to find the appropriate objective value by iteratively adjusting parameters. It is one of the most widely used optimization

algorithms in neural networks. This paper uses Adam as the optimization algorithm of the model. The reason is that it is essentially the combination of momentum and RMSprop algorithms and then corrects its deviation. The momentum algorithm uses momentum similar to physics to accumulate gradients, and the RMSprop algorithm can make convergence faster while making fluctuations smaller. Therefore, the performance achieved by combining these two algorithms is assumed to be better. Adam fully utilizes the second moment mean of the gradient in addition to computing the adaptive parameter learning rate based on the first moment mean, as does the RMSProp algorithm. Specifically, the algorithm computes exponential moving averages of the gradients, using hyperparameters beta1 and beta2 to control the decay rate of these moving averages. Because the initial moving average, beta1 and beta2 values are all close to 1, the moment estimate's deviation is close to 0. By first computing the deviated estimate, and then, the deviate-corrected estimate, the deviation is optimized.

4. Simulation Experiment of Regional Traffic Flow Prediction Based on AnyLogic

AnyLogic is a professional virtual prototyping environment for designing complex systems with discrete, continuous, and mixed behaviors. Using AnyLogic, one may easily create a simulation model of the intended system and the system's surrounding environment, including its physical equipment and operators. The road traffic Library in AnyLogic allows users to model, simulate, and visualize vehicle traffic. The library supports detailed and efficient physical hierarchical modeling of vehicle motion. AnyLogic can be applied to model vehicles, roads, and lanes of highway traffic, street traffic, production site transportation, parking lot, or any other system.

4.1. Data Description. In the experiment, AnyLogic is used to build a regional road network micro model to simulate the actual road conditions for the statistics of intersection traffic flow data. This area is a real region composed of three associated intersections, and each intersection has 12 lanes, as shown in Figure 6. The simulation data includes three months' traffic flow data. The statistical interval is 15 minutes, and the traffic flow data of all intersections are collected every 15 minutes. Each model data represents the number of vehicles passing in the direction of traffic flow within 15 minutes.

In the simulation, external factors such as morning peak, weekends, and holidays, are considered to enhance the randomness, making the simulation data tend to the real data.

4.2. Data Preprocessing. Before inputting the data into the model, it is necessary to standardize the data to scale the attributes of a sample to a specified range. It is necessary to eliminate the influence of different attributes of samples with different orders of magnitude because

(i) The difference in orders of magnitude will lead to the dominant position of attributes with larger orders of magnitude;

(ii) The difference of orders of magnitude will cause the convergence speed of iteration to slow down;

(iii) Algorithms that depend on sample distance are very sensitive to the order of magnitude of data.

In this paper, min-max standardization, also known as normalization, is used as the method of data standardization. The specific method is as follows: after the data (x) are centered according to the minimum value, it is scaled according to the range (maximum value-minimum value), and the data are converged to [0, 1]. After normalization, the range of the optimization process becomes smaller, the optimization process becomes gentle, and it is easier to correctly converge to the optimal solution. The calculation formula is shown as follows:

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{10}$$

4.3. Evaluation Metrics. This paper measures the prediction effect of the model using the mean square error (MSE) and mean absolute error (MAE) loss functions to evaluate the prediction performance of the algorithm more thoroughly.

4.4. Experimental Setup. This paper uses the Python3.7 simulation environment and the deep learning framework PyTorch to build the model. The CPU model used is Intel (R) Xeon (R) w-2133 CPU @ 3.60 GHz, the memory is 32 GB, the GPU model is NVIDIA GeForce GTX 1080 Ti, and the operating system is Ubuntu.

4.5. Simulation Results and Analysis. The proposed CNNformer$^+$ is compared with several baseline models, including CNN, LSTM, DISTN [22], CNNformer, transformer, and informer [26]. Table 1 compares the performance of the baseline model and CNNformer$^+$ in the traffic flow prediction task at the associated intersections. Table 2 shows the hyperparameter settings of the experimental model.

The following phenomena were observed during the experiment:

(i) Compared with the traditional time series model LSTM, the convolution model combined with CNN and LSTM is more suitable for traffic flow prediction tasks. This is because CNN can better extract the spatial features of associated intersections than LSTM. Compared with CNN, LSTM has better performance in traffic flow prediction tasks.

(ii) Compared with the model combining CNN and LSTM (DISTN). Transformer is more suitable for traffic flow prediction tasks, thanks to its ability to better establish long-distance dependencies, and

Figure 6: The road network in the simulation network contains three intersections (Yinzhou district, Ningbo city, China).

Table 1: Comparison of simulation results.

Models	MAE	MSE
CNN	0.12205	0.02899
LSTM	0.09840	0.01720
DISTN	0.09778	0.01683
Transformer	0.09599	0.01656
Informer	0.09736	0.01681
CNNformer	0.09499	0.01604
CNNformer$^+$	0.09220	0.01515

Table 2: Experimental model hyperparameter settings.

Parameters	Parameters size
Input dimensions	512
Batch size	128
Learning rate	0.0001
Epoch	120
Time window size	96
Number of encoder layers	1
Number of decoder layers	1

unlike LSTM, which depends on the calculation at the previous moment, it can be well parallel.

(iii) Informer has achieved good results in time series prediction tasks in many fields, which proves that the improvement made by informer in transformer is effective. However, in this experiment, the accuracy of the forecast is lower than that of transformer, which might be because informer has a difficult time capturing the details of traffic flow data.

(iv) Compared with transformer, CNNformer has higher prediction accuracy, thanks to CNN's ability to extract the spatial features of traffic flow data at associated intersections.

(v) The prediction accuracy of CNNformer$^+$ is higher than that of CNNformer, which verifies that learning the periodic characteristics of traffic flow is helpful to improve the prediction accuracy.

(vi) The model proposed in this paper achieves the best results in the traffic flow prediction task, which shows that the model is superior to some of the most advanced traffic flow prediction methods in the literature.

It can be seen from Figure 7 that the dimension size of the hidden layer inside the model will also affect the performance to a certain extent. The richer hidden layers can play a positive role. However, when the number of hidden layer units is greater than 512, the model performance begins to decline.

Figure 8 shows the comparison between the real traffic volume and the traffic volume predicted by CNNformer$^+$ at a single time step, i.e., 10 a.m., 2 p.m., and 5 p.m. Each time step includes $36\,(12 \times 3)$ traffic movements. As can be observed, the model successfully captures the changing trend of the actual traffic volume in the majority of traffic flow directions where the predicted value is near the real value.

Figure 9 shows the comparison of real traffic volume at 10 a.m. with traffic volume predicted by informer and transformer. It can be seen from the marks in the figure that CNNformer$^+$, informer, and transformer have a huge deviation when predicting the traffic volume with movement number 3. However, when predicting the traffic volume with movement numbers 23–28, CNNformer$^+$ can better fit the real traffic volume than informer and transformer, which reflects the superiority of the algorithm used in this model.

After introducing the overall performance of the proposed model, the prediction accuracy of single vehicle flow motion is now given. Table 3 provides the prediction accuracy for each movement at the second intersection. The MSE of the traffic movement from east and west is better than that from north and south. This is because intersection 2 is located in the middle of the three intersections. Since the volume of traffic leaving from the north and south is lower than that leaving from the east to west, the trends of the traffic flow are more varied, which makes it more difficult to predict the direction of the traffic flow.

Table 4 provides the prediction accuracy of each of the three intersections. It can be seen that intersection 2 has the lowest MSE. Since intersection 2 is located in the center of the main road, the flow data of this intersection is also related to the traffic conditions of intersection 1 and intersection 3. Intersection 1 and intersection 3 are located at the boundary of the main road. There is only one upstream or downstream intersection, which is less affected. Therefore,

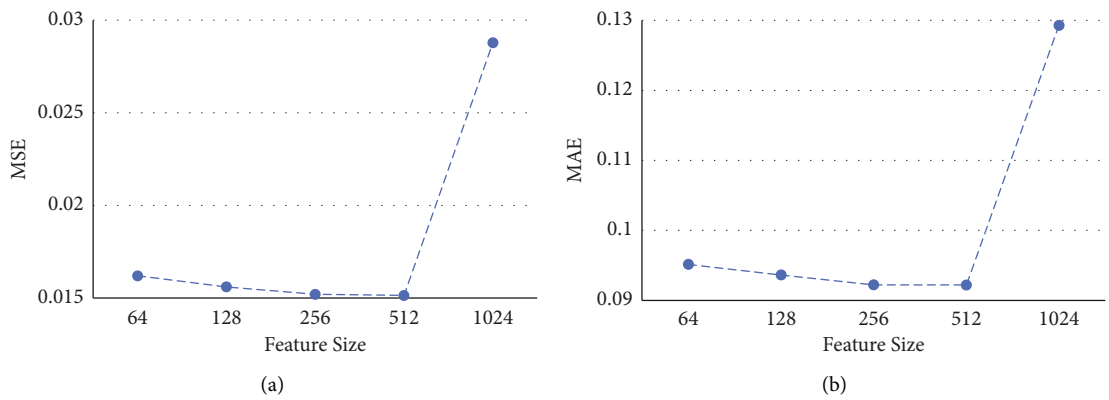

Figure 7: Effect of a hidden layer's dimension on the effectiveness of the task of predicting traffic flow at related intersections. (a) MSE. (b) MAE.

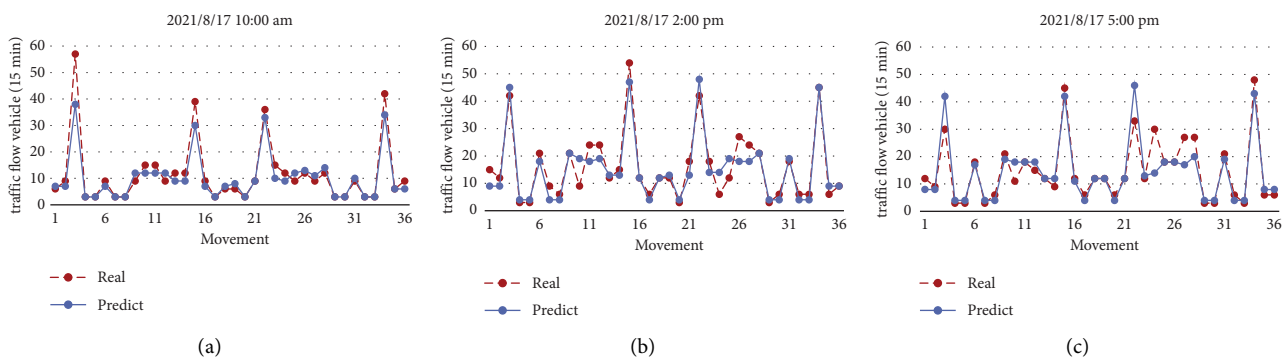

Figure 8: The comparison between the real traffic volume and the traffic volume predicted by CNNformer$^+$ at a single time step, i.e., 10 a.m., 2 p.m., and 5 p.m. Each time step includes 36 (12×3) traffic movements. (a) CNNformer$^+$ at 10 a.m. (b) CNNformer$^+$ at 2 p.m. (c) CNNformer$^+$ at 5 p.m.

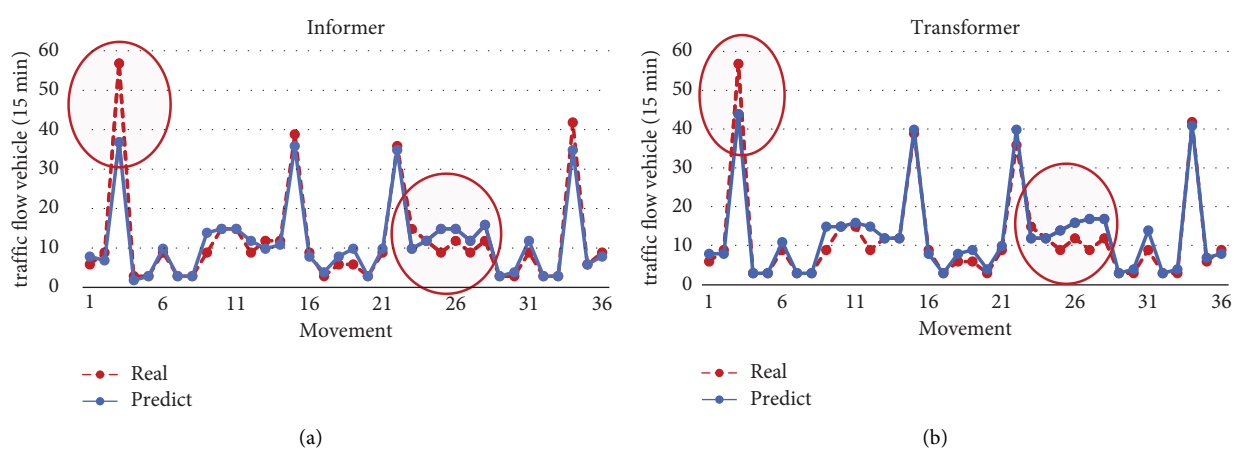

Figure 9: The comparison of real traffic volume at 10 a.m. with traffic volume predicted by informer and transformer. Each time step includes 36 (12×3) traffic movements. (a) Informer at 10 a.m. (b) Transformer at 10 a.m.

Table 3: Volume prediction results for each intersection.

Intersections	MSE
Intersection 1	0.01483
Intersection 2	0.01563
Intersection 3	0.01500

the change in traffic volume is more regular, reducing the difficulty of prediction.

Figure 10 shows the forecast results of traffic volume in different time step sizes. It can be seen that MSE and MAE also begin to decrease significantly with the increase of time step size. This is because the transformer requires a large

Table 4: Volume predication results for each movement of intersection 2.

Movement of intersection 2	MSE
West -> North	0.01387
West -> South	0.01154
West -> East	0.01067
North -> West	0.01974
North -> South	0.01824
North -> East	0.01812
South -> West	0.01797
South -> North	0.02021
South -> East	0.01972
East -> West	0.01276
East -> North	0.01341
East -> South	0.01137

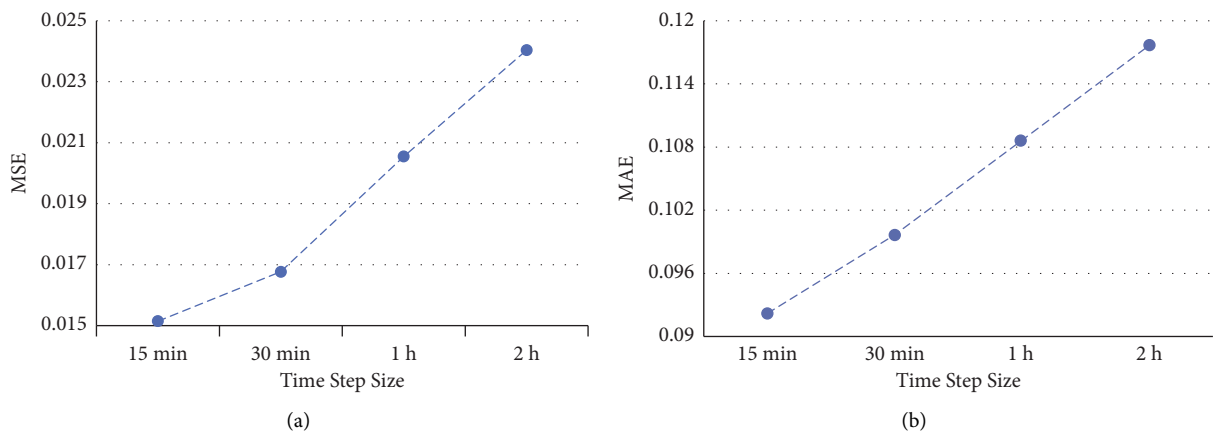

Figure 10: Volume prediction results using different time step sizes. (a) MSE. (b) MAE.

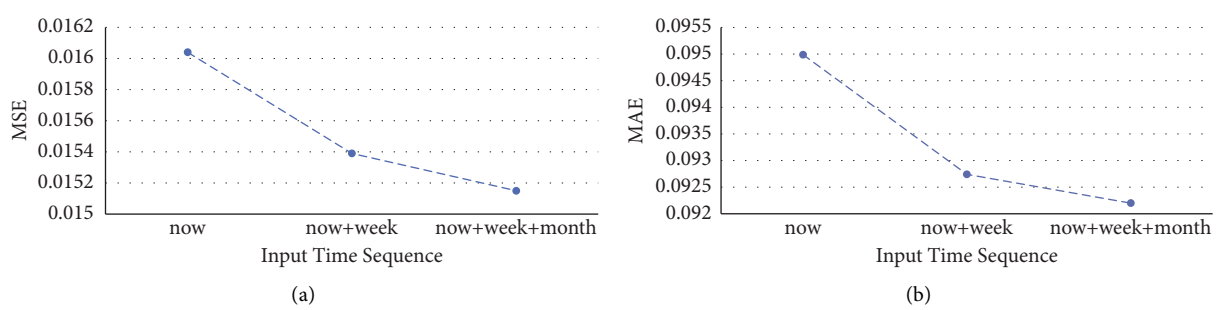

Figure 11: Volume prediction results using different time sequences. (a) MSE. (b) MAE.

amount of data for training. With the increase of time step size, the number of input samples of the model decreases, and the number of learned traffic volume features decreases, this increases the difficulty of prediction. The results show that a small time step size should be selected as far as possible in traffic flow prediction.

Figure 11 shows the influence of different sequences on prediction accuracy. "now" refers to the input only using the traffic flow data of the previous time window $(X_{t-H}, X_{t-H+1}, \ldots, X_t)$. "now + week" refers to the input contains the traffic flow data of the previous time window and the simultaneous data of the week before the previous time window $(X_{t-H-\text{week}}, X_{t-H-\text{week}+1}, \ldots, X_{t-\text{week}})$. "now + week + month" refers to the input contains the traffic flow data of the previous time window, the simultaneous data of the week before the previous time window, and the simultaneous data of the previous month in the previous time window $(X_{t-H-\text{month}}, X_{t-H-\text{month}+1}, \ldots, X_{t-\text{month}})$. The findings demonstrate that the minimal MSE and MAE are reached by taking into account all three time windows.

5. Conclusion

Transformer has advantages in dealing with time series tasks. Many current research works are based on the transformer architecture to establish models for various series tasks and have achieved good results beyond the traditional models in many application fields. To tackle the problem of traffic safety oriented multi-intersection flow prediction, in this research, a new architecture integrating CNN and transformer is proposed from the viewpoint of accuracy improvement, making it more suitable for the traffic flow prediction task of associated intersections. The comparative experiment with informer and other baseline models proves the superiority of the new architecture.

In the research work of this paper, the following results have been achieved:

(i) A new intersection traffic flow prediction model CNNformer $^{+}$ is proposed, which considers that the traffic flow data at the associated intersection is a group of spatiotemporal sequences, using CNN to extract the spatial features of the data can significantly improve the prediction accuracy of the transformer model.

(ii) The average pooling layer successfully learns the periodicity of the traffic flow data, increasing the model's forecast accuracy. Experiments on the simulated network dataset demonstrate the superiority of the proposed method.

Data Availability

The data used to support the findings of the study are available on request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Tingting Fu and Peng Liu contribute to problem formulation and solution design. Qianwen Yu contributes to the implementation of the algorithms and writing. Haksrun Lao is responsible for the simulation experiment and part of the writing. Shaohua Wan helps revise the paper.

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grant no. 62172134.

References

[1] L. Cheng, J. Liu, G. Xu et al., "SCTSC: a semicentralized traffic signal control mode with attribute-based blockchain in IoVs," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1373–1385, 2019.

[2] A. H. Abdul Hanan, M. Yazid Idris, O. Kaiwartya, M. Prasad, and R. Ratn Shah, "Real traffic-data based evaluation of vehicular traffic environment and state-of-the-art with future issues in location-centric data dissemination for vanets," *Digital Communications and Networks*, vol. 3, no. 3, pp. 195–210, 2017.

[3] S. Liu, J. Yu, X. Deng, and S. Wan, "Fedcpf: an efficient-communication federated learning approach for vehicular edge computing in 6G communication networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1616–1629, 2022.

[4] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25536–25545, 2022.

[5] C. Chen, L. Liu, S. Wan, X. Hui, and Q. Pei, "Data dissemination for industry 4.0 applications in Internet of vehicles based on short-term traffic prediction," *ACM Transactions on Internet Technology*, vol. 22, no. 1, pp. 1–18, 2021.

[6] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[7] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg–marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, 2012.

[8] M. Castro-Neto, Y. S. Jeong, M. K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[9] T. Alladi, V. Kohli, V. Chamola, and F. R. Yu, "A deep learning based misbehavior classification scheme for intrusion detection in cooperative intelligent transportation systems," *Digital Communications And Networks*, vol. 8, 2022, https://www.sciencedirect.com/science/article/pii/S2352864822001407.

[10] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[11] S. Alkheder, W. Alkhamees, R. Almutairi, and M. Alkhedher, "Bayesian combined neural network for traffic volume short-term forecasting at adjacent intersections," *Neural Computing & Applications*, vol. 33, no. 6, pp. 1785–1836, 2020.

[12] W. Alajali, W. Zhou, and W. Sheng, "Traffic flow prediction for road intersection safety," in *Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Guangzhou, China, October 2018.

[13] K. Tan, D. Bremner, J. Le Kernec, L. Zhang, and M. Imran, "Machine learning in vehicular networking: an overview," *Digital Communications and Networks*, vol. 8, no. 1, pp. 18–24, 2022.

[14] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 1–13, 2020.

[15] Y. Tu, S. Lin, J. Qiao, and B. Liu, "Deep traffic congestion prediction model based on road segment grouping," *Applied Intelligence*, vol. 51, no. 11, pp. 8519–8541, 2021.

[16] Y. Ren, D. Zhao, D. Luo, H. Ma, and P. Duan, "Global-local temporal convolutional network for traffic flow prediction,"

IEEE Transactions on Intelligent Transportation Systems, vol. 23, pp. 1–7, 2020.

[17] C. Ma, G. Dai, and J. Zhou, "Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 1–10, 2021.

[18] B. Du, H. Peng, S. Wang et al., "Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2019.

[19] C. Chen, Y. Zhang, Z. Wang, S. Wan, and Q. Pei, "Distributed computation offloading method based on deep reinforcement learning in ICV," *Applied Soft Computing*, vol. 103, no. 7, pp. 107-108, 2021.

[20] W. Qu, J. Li, L. Yang et al., "Short-term intersection traffic flow forecasting," *Sustainability*, vol. 12, no. 19, p. 8158, 2020.

[21] D. Kim and O. Jeong, "Cooperative traffic signal control with traffic flow prediction in multi-intersection," *Sensors*, vol. 20, no. 1, p. 137, 2019.

[22] W. Li, X. J. Ban, J. Zheng, H. X. Liu, C. Gong, and Y. Li, "Real-time movement-based traffic volume prediction at signalized intersections," *Journal of Transportation Engineering Part A Systems*, vol. 146, no. 8, Article ID 40, 2020.

[23] J. Guo, M. Bilal, Y. Qiu, C. Qian, X. Xu, and K.-K. Raymond Choo, "Survey on digital twins for internet of vehicles: fundamentals, challenges, and opportunities," *Digital Communications And Networks*, vol. 12, 2022, https://www.sciencedirect.com/science/article/pii/S235286482200116X.

[24] B. Fan, Z. Su, Y. Chen, Y. Wu, C. Xu, and T. Q. S. Quek, "Ubiquitous control over heterogeneous vehicles: a digital twin empowered edge AI approach," *IEEE Wireless Communications*, pp. 1–8, 2022.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computer Science*, vol. 1, 2013.

[26] H. Zhou, S. Zhang, J. Peng, S. Zhang, and W. Zhang, "Informer: beyond efficient transformer for long sequence time-series forecasting," 2020, https://arxiv.org/abs/2012.07436.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 3288308, 18 pages
https://doi.org/10.1155/2022/3288308

Research Article

Formal Model and Analysis for the Random Event in the Intelligent Car with Stochastic Petri Nets and *Z*

Yang Liu, Yingqi Fan, Darong Huang, Bo Mi, and Liyuan Huang

Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China

Correspondence should be addressed to Yingqi Fan; fanyingqi@mails.cqjtu.edu.cn

Received 24 June 2022; Revised 15 September 2022; Accepted 20 September 2022; Published 13 October 2022

Academic Editor: Chen Chen

With the continuous development of science and technology, people's lifestyle becomes more and more intelligent, especially in intelligent transportation. However, running in a random environment, safety can be affected by various factors during operation. For an intelligent car, guaranteeing its safety in operation is important to the passengers in the vehicle. So, it is vital to verify the safety of the system of the smart car. This study proposes an integration formal method with stochastic Petri nets and Z (SPZN). Stochastic Petri nets can better simulate the occurrence of random events in the driving process of intelligent cars. With the advantages of the frame structure of Z language, the concurrent process and state before and after the system at different times can be better described. In addition, the frame structure of Z language can solve the problem of state explosion in Petri nets. Using this method, the random events that may occur during the operation can be formally modeled, and the subsequent behavior of the vehicle can be analyzed and predicted effectively. Using the reinforcement learning, the parameter λ in the stochastic Petri nets can be optimized, which can reduce the probability of bad states and ensure the stability and security of the system. Moreover, a case study of the intelligent car modeled by stochastic Petri nets and Z is given. The results show that it can improve the safety and effectiveness of the smart vehicle driving system.

1. Introduction

With the rise of the Internet of vehicle technology [1–5], smart cars are gradually integrated into people's lives. However, smart cars always operate in an environment full of random factors, such as pedestrians, traffic signs on the road, other driving vehicles, and changeable weather, and its safety is particularly important. Liu et al. [6] proposed an efficient communication method FedCPF. Compared with traditional federated learning, this method achieves more efficient communication, faster convergence, and higher test accuracy. In terms of communication, the security of the Internet of vehicles system is improved. Zhao et al. [7] proposed a digital twin-assisted storage strategy for satellite-terrestrial networks (INTERLINK), which leverages the digital twins (DTs) to map the satellite networks to virtual space for better communication. By enabling more reliable communication, the safety of smart cars is guaranteed. Soni et al. [8] developed a novel low-cost sensor system that improves safety in intelligent transportation systems by improving vehicle sensor systems. Martinez et al. [9] found that driving style plays an important role in driving safety and then used different algorithms to characterize and identify the driver's driving style. Through the analysis of the driver's driving style, the safety performance during the driving process can be improved. Colombo et al. [10] proposed a strategy to dynamically decompose the formal verification problem of a large road network, which can effectively prevent vehicle collisions and improve the safety of autonomous vehicles. To sum up, the researchers have improved the security of the Internet of vehicles from different aspects, but the above methods have not effectively solved the random events that occur in the driving process of smart cars.

The formal method is an effective method to prove system security, accessibility, and effectiveness, and it is not only widely used in computer hardware technology [11], software requirement verification [12–14], industrial

engineering [15], medicine [16], communication [17], and so on. In recent years, it has been increasingly used in the field of transportation. Qi et al. [18–20] used Petri nets to model intersections and designed strategies to solve the congestion problem. In addition, they also used time-delay Petri nets (TPNs) for intersection modeling, designed corresponding strategies to solve the accident-induced congestion problem, and proposed a method to classify driving behavior at intersections in congestion. Their work effectively improves the congestion at intersections and enhances the management and safety of urban traffic. In the study by Labadi et al. [21], stochastic Petri net (SPN) was used to model and analyze public bike sharing systems, and the results showed that SPN is very suitable for analyzing and simulating discrete-time systems. Moreover, there exist some researchers who have applied formal modeling to Internet of vehicles (IoVs). Liu et al. [22] proposed a formal model based on integration time Petri nets and Z (TPZN). By applying TPZN to IoVs, the behavior of IoVs can be accurately and formally described, and the model is validated with a case study, and the results show that the proposed approach can effectively improve the safety and intelligence of IoVs.

Although the above work has improved the security and intelligence of IoV, it still suffers from the problem of insufficient ability to describe random events. SPN has the ability to describe random events, but it still exists the problem of state explosion. To solve this problem, this study proposes a formal modeling approach combining stochastic Petri nets and Z (SPZN). SPZN consists of two parts, SPZN-SPN and SPZN-Z. SPZN-SPN defines the structure and flow of the overall model, and SPZN-Z abstracts the structure and describes the related constraints. Through the abstraction of SPZN-Z, the complexity of SPN can be effectively reduced, the number of states can be reduced, and the state explosion can be avoided. In addition, each transition in SPN has a transition implementation rate λ. λ can affect the stability and security of the system, so setting the value of λ so that the system has high stability and high security is an urgent problem to be solved. However, setting λ in practical problems is not an easy task, and Song and Sun et al. [23, 24] used assumed λ in their work and did not give a method to determine the values of λ. Therefore, in this study, we propose an optimization method for the transition implementation rate λ. We use the actor-critic algorithm in reinforcement learning to optimize λ for SPZN-SPN, which improves the stability and security of the system.

The rest of this study is organized as follows. Section 2 presents some of the basics required for this study. Section 3 introduces a formal modeling approach based on SPZN, refines the method, and proposes a transition implementation rate optimization method. Section 4 analyzes the model in terms of reachability, boundedness, and safety, respectively, and lists the advantages of the model. An example of a SPZN-based smart connected car system with random events is given in Section 5 to verify the effectiveness of the method in this study. Section 6 concludes the study and discusses future work.

2. Preliminaries

In this section, we review smart net cars, stochastic Petri nets, the relationship between stochastic Petri nets and Markov chains, Z language structures, and reinforcement learning.

2.1. Intelligent Connected Vehicle. With the rapid development of computer technology and artificial intelligence, traditional traffic system is gradually being replaced by the intelligent connected vehicle system that integrates people, vehicles, roads, and clouds, that is, "smart transportation" [25, 26]. As shown in Figure 1, this figure is a conceptual diagram of the vehicle-road-human-cloud integrated system. The relationship between the intelligent connected vehicle (ICV) and its various components is shown in Figure 2. It is not difficult to see that the ICV is mainly divided into autonomous vehicles and the Internet of vehicles. For the yellow part where smart transportation and the Internet of vehicles intersect in the figure, the part to which the smart connected car belongs. There is a technical architecture diagram, as shown in Figure 3. In the infrastructure submodule, sensors play an essential role. Usually, sensors are divided into three categories: distance, speed, and image. The distance sensor is mainly radar, and the image sensor is mainly a surveillance camera.

The related research on autonomous vehicles can be summarized into three aspects: autonomous driving in high-speed environments, autonomous driving in urban environments, and autonomous driving in special environments. Autonomous driving in high-speed environments is mainly applied to highways, and the difficulty lies in the safety of vehicles and passengers under high-speed driving. A lot of research is still needed to break through this difficulty. In the urban environment, the automatic driving speed is slow, the vehicle safety performance is high, the application is more extensive, and the prospect is better. However, it still has a complex urban environment and requires a more sensitive perception control algorithm. For autonomous driving in special environments, it is mainly used in harsh environments such as military operations. In this application scenario, we should first consider the reliability of vehicles in harsh environments.

For the Internet of vehicles, Ji et al. [26] proposed a new Internet of vehicle architecture. The purpose is to promote the rapid development of intelligent connected vehicles, improve the ability of road condition perception and traffic management and control, and lay the foundation for intelligent transportation. As shown in Figure 4, the architecture mainly consists of four layers: security authentication layer, data acquisition layer, edge layer, and cloud platform layer.

2.2. Stochastic Petri Net Extension. A stochastic Petri net is a kind of advanced Petri net that includes time factors and probability [27]. The main difference between the stochastic Petri net and the time Petri net is randomness. So, a

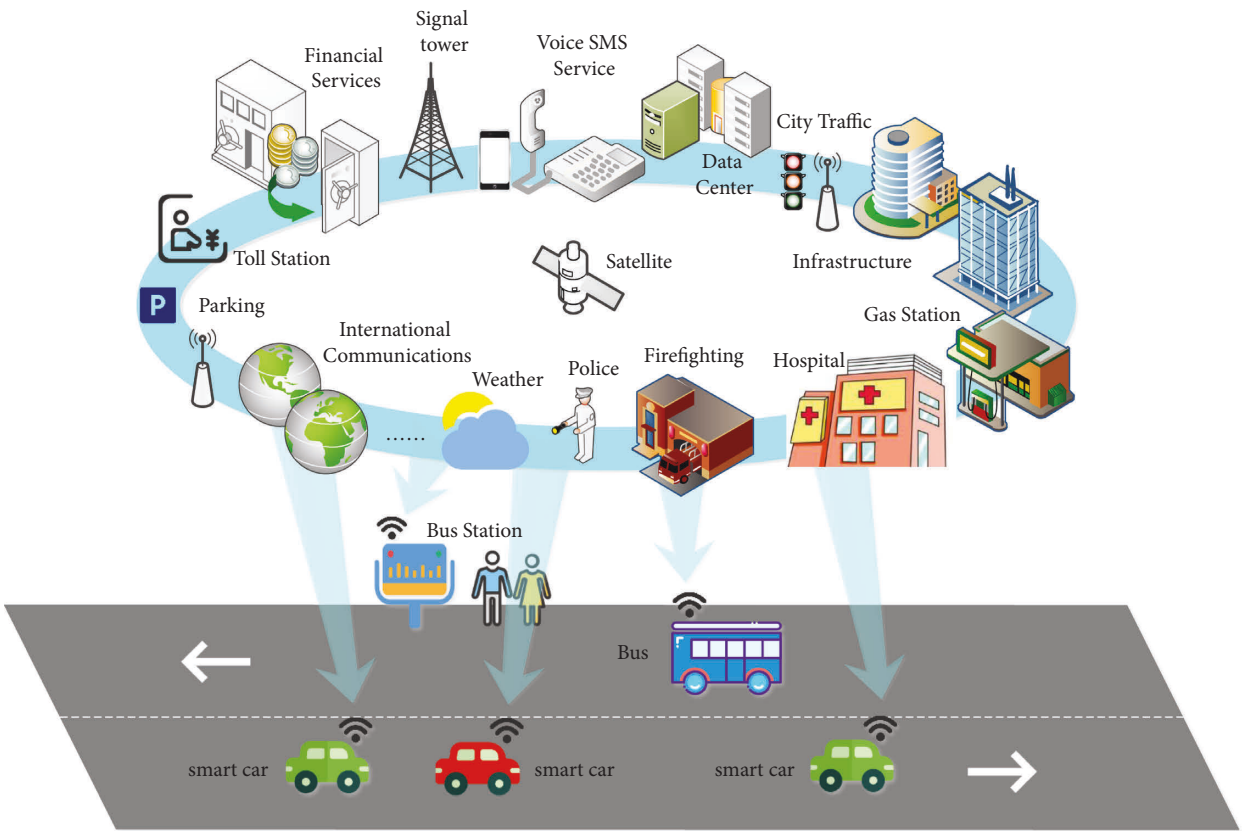

Figure 1: Conceptual diagram of the vehicle-road-human-cloud integrated system.

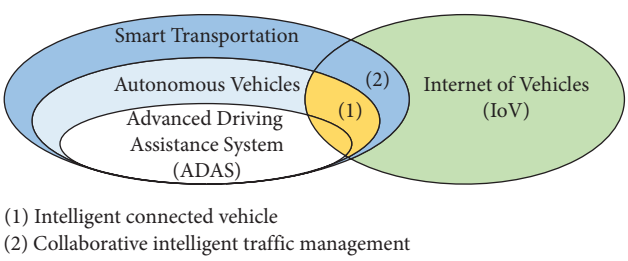

Figure 2: Relationship among ICV, IV, and IoV.

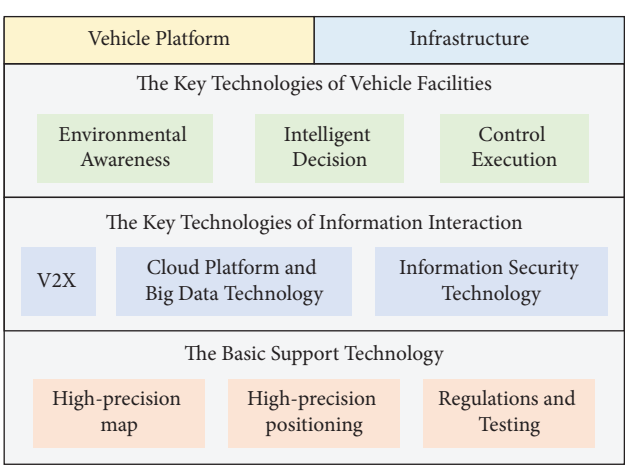

Figure 3: Framework of ICV technology.

stochastic Petri net is more suitable to describe the uncertain system. Furthermore, reachable marking graphs of stochastic Petri nets with finite places and transitions are isomorphic to one-dimensional continuous-time Markov [28, 29].

Stochastic Petri net is defined as 5-tuple $N = (P, T, F, M_0, \lambda)$:

(1) $P = \{p_1, p_2, \ldots, p_n\}$ is a finite and non-empty set of places

(2) $T = \{t_1, t_2, \ldots, t_n\}$ is a finite and non-empty set of transitions

(3) $F = \{P \times T\} \cup \{T \times P\}$ represents places to transitions and transitions to places

(4) M_0 is the initial state vector and represents the number of tokens in each place of the model

(5) λ represents the set of implementation rate of transition

Here, for $\forall t_i \in T$, $\lambda(t_i) = \lambda_i$ is a nonnegative real number, which represents the rate of occurrence when the transition t_i satisfies the occurrence conditions, and $1/\lambda_i$ represents the average firing delay or average service time.

$\forall t \in T$, $F_t(x) = 1 - e^{-\lambda_t x}$, represents the probability of t happen in x-time [30].

If there is $[M|t_j > M'$, then the transition t_j is triggered and the state is changed from M to M'. The token transfer rule is shown as follows:

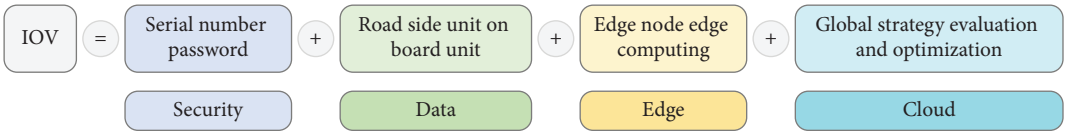

Figure 4: Main elements of the proposed architecture.

$$\forall p_i \in P, M'(p_i) = M(p_i) - \mathrm{Pre}(p_i, t_j) + \mathrm{Post}(p_i, t_j). \quad (1)$$

2.3. SPN to Markov Chain. For a stochastic Petri net, it has the rate of occurrence of transition, $\lambda_i \in \lambda$, where λ_i is a nonnegative real number; then, the time delay d_i of t_i is a random variable that is related to time and obeys exponential distribution. Because the exponential distribution has memoryless properties, if there is a stochastic Petri net, the reachable marking graph of the stochastic Petri net is isomorphic to a finite Markov chain and satisfies the following definition [30].

Definition 1. For a Markov chain that is isomorphic to a stochastic Petri net, where the Markov chain has n states, we define a transition matrix Q of $n \times n$ order. When i is not equal to j, if there is $t_k \in T$ that makes $M_i[t_k > M_j$, then we have

$$q_{ij} = \frac{d(1 - e^{-\lambda_k \tau})}{d\tau}\Big|_{\tau=0} = \lambda_k, \quad (2)$$

else

$$q_{ij} = 0. \quad (3)$$

When i is equal to j, we have

$$q_{ij} = \frac{d\prod_k (1 - (1 - e^{-\lambda_k \tau}))}{\mathrm{d}\tau}\Big|_{\tau=0} = \frac{d(e^{-\tau}\sum_k \lambda_k)}{\mathrm{d}\tau}\Big|_{\tau=0} = -\sum_k \lambda_k, \quad (4)$$

where λ_k is the average implementation rate of transition t_k and $\exists M' \in [M_0 >$ and $\exists t_k \in T$ that makes $M_i[t_k > M_j$.

Figure 5 is a simple case about the reachable marking graph isomorphic to the Markov chain. From the Petri net of Figure 5(a), the reachable marking graph of Figure 5(b) can be obtained, and the reachable marking graph is isomorphic to the Markov chain, so we can get the Markov chain shown in Figure 5(c). There are 5 states in the reachable marking graph, so there are also 5 states in the corresponding Markov chain.

By constructing the reachable marking graph of the stochastic Petri net, we can obtain that the reachable marking graph and its isomorphic Markov chain have n states, and the steady-state probability from state M_0 to state M_{n-1} is an n-dimensional vector P, and then, $P = (P(M_0), P(M_1) \cdots P(M_{n-1}))$, where $P(M_i)$ is the steady-state probability of marking M_i. According to the Markov process, there are the following equations. By solving this system of equations, we can obtain the steady-state probability $P(M_i)$ for each reachable state M_i.

$$\begin{cases} P \times Q = 0, \\ \sum_{i=0}^{n-1} P(M_i) = 1. \end{cases} \quad (5)$$

2.4. Z Frame Structure. Z language is a specification language mainly based on first-order predicate calculus. It is a functional language, which can easily express the state and behavior of things in mathematical symbols. As shown in Figure 6(a), in the Z language this structure is called a schema. It is the basic description unit and the basic structure of Z language. A pattern includes the name of the pattern (S), the declaration part (D), and the assertion part (P), and the assertion part is divided into pre-assertion and post-assertion. The framework is similar to a class in the C++ language, as shown in Figure 6(b). A class has a class name, attributes, and functions, which correspond to the name, declaration part, and assertion part of the schema, respectively.

In fact, Z language is just a set of prescribed mathematical symbols, and the "program" written in Z language is an abstract design of computer software or hardware system. Therefore, the content written in the Z language is not a computer program, nor is it a code that can be compiled to generate and can be run on a computer. The content written in Z language is not for the computer to run, but for human understanding and analysis. Using Z language, users can understand the modules, data types, and processes of the system so that the system can be analyzed, optimized, verified, tested, etc. In terms of data abstraction, Z language has a stronger descriptive ability than Petri net [31]. Using the Z language, Petri net state can be reduced and state explosion can be avoided.

2.5. Reinforcement Learning. In reinforcement learning, when the agent makes an action a according to the policy function $\pi(a|s)$ in the state s, the environment rewards the agent according to the action. We usually have two methods for artificial intelligence to intelligently control the agent: the first method is policy learning, and the second method is value learning.

Policy learning uses a neural network $\pi(a|s;\theta)$ to approximate the policy function $\pi(a|s)$, so the neural network is also called a policy network. The parameters θ in the policy network represent the weights, and they are updating and optimizing through the actions of the agent and the rewards given by the environment so that the policy network can achieve the desired result. It is represented by the policy gradient algorithm in policy learning, and the Monte Carlo method is the simplest among the many gradient algorithms.

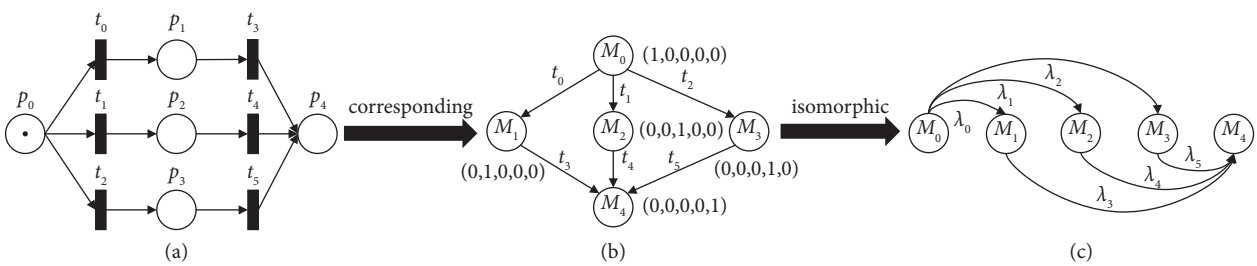

Figure 5: Reachable marking graph isomorphic Markov chain. (a) Petri net. (b) Reachable marking graph. (c) Markov chain.

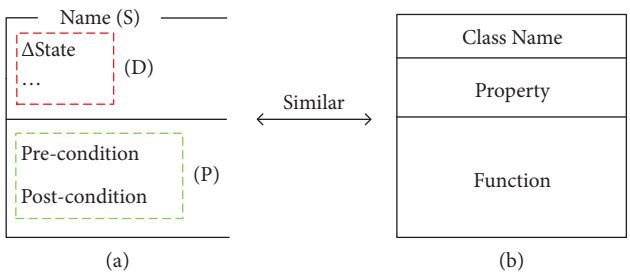

Figure 6: Structure of the Z language is similar to the structure of the C++. (a) The framework of Z language. (b) The C++ language class framework.

Value learning is also a method of approximating the value function $Q^*(s_t, a_t)$ using a neural network $Q(s, a; w)$ called a value network. s_t and a_t represent the state and action at time t, respectively, and s, a, and w represent the state, action, and weight of the value network, respectively. The value network is used to predict the score of the action in the current state, and then, the actual reward of the action is given according to the environment, and the gradient descent method is used to correct the future reward prediction. Common value learning algorithms include Q-learning algorithm and deep Q-network algorithm (DQN). Based on the Q-learning algorithm, the DQN algorithm combines the deep neural network, uses the neural network to approximate the value function, and uses the experience playback to train the entire process of reinforcement learning.

It should be noted that there are many learning methods in reinforcement learning. In this study, we use the actor-critic method for reinforcement learning. The method approximates the policy function and the value function using two neural networks, the policy network (actor) is equivalent to a gymnast, and the value network (critic) is equivalent to a gymnastic referee. In the initial state, athletes can only make random actions, and the referee can only make random evaluations based on the athletes' actions and current state. After the referee makes an action, the environment gives corresponding rewards, and the referee gradually becomes professional by catering to the environment. After the athlete makes an action, the referee evaluates it. By catering to the referee, the athlete's action gradually becomes standardized to achieve the purpose of optimizing the strategy function. Figure 7 depicts the working principle of the actor-critic method.

In the actor-critic method, a state value function $V_\pi(s) \approx \sum_a \pi(a|s) \times Q_\pi(s, a)$ is defined, where $\pi(a|s)$ is the policy function and $Q_\pi(s, a)$ is the value function, because two neural networks are used to approximate the policy function and the value function, respectively, so the policy network is $\pi(a|s; \theta)$, the value network is $q(s, a; w)$, and the state value function is as follows:

$$V_\pi(s) \approx \sum_a \pi(a|s; \theta) \times q(s, a; w). \tag{6}$$

By summing the products of the probability and value of each action, we can easily obtain the pros and cons of executing the policy function π in the current state s. The main steps of the method are shown in Figure 8. Among them, the temporal difference (TD) algorithm is an algorithm that stages the entire model training process, which is suitable for the idea of discounted returns in reinforcement learning and can be applied to value learning.

Comparing the above three reinforcement learning algorithms, their respective advantages and disadvantages are shown in Table 1 [32–34]. Although the representative algorithm DQN in value learning can store previous data through the experience pool and break the relationship between information, it can effectively solve the problem of complex state and action and the existence of a correlation between data. However, this method still has problems such as overfitting, low sample utilization, and unstable evaluation in the solution process. Compared with value learning, policy learning is simpler and has better convergence, but it still has shortcomings such as high algorithm variance, slow convergence speed, and difficulty in determining the learning step size. To further reduce the variance, the actor-critic algorithm is proposed, and the state value function is used as the baseline to predict the value of the bootstrapping method, which greatly reduces the variance, but it introduces deviation, making it a major drawback of the actor-critic algorithm.

3. Modeling with SPZN

To improve the abstraction ability and randomness ability in the intelligent networked automobile system, this study integrates the stochastic Petri net and the Z framework and proposes SPZN. Using the randomness of SPN and the abstraction ability of Z, the shortcomings of the system can be effectively improved. Compared with TPN, SPN, and PZN, SPZN can define and describe the system more efficiently.

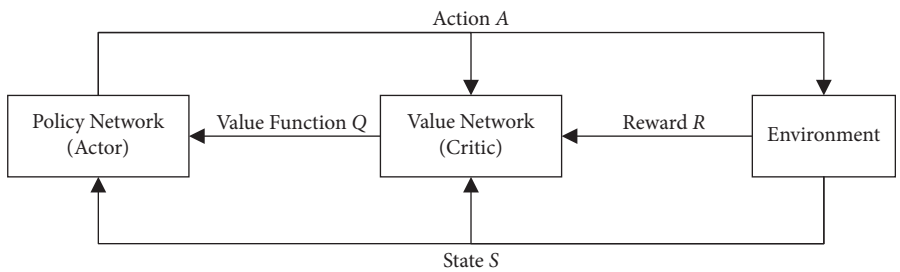

Figure 7: Principle of the actor-critic method.

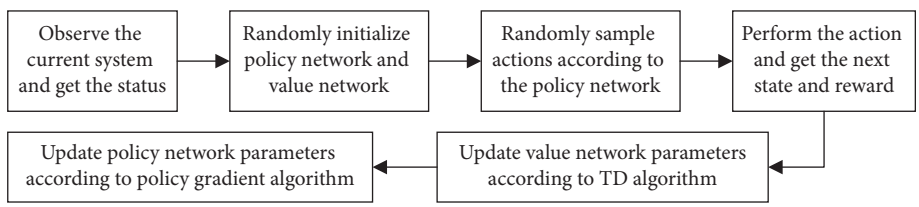

Figure 8: Main steps of the actor-critic method.

Table 1: Advantages and disadvantages of three reinforcement learning algorithms.

Category	Representative algorithm	Advantage	Disadvantage
Value learning	DQN	It can solve high-dimensional complex problems and is not easy to fall into local optimum	Overfitting, low sample utilization, instability, poor convergence
Policy learning	Monte Carlo method	High stability and strong convergence	Large variance, slow convergence, the easy local optimal solution
Value policy learning	Actor-critic	Small variance, fast training, and can solve continuous problems	Low learning efficiency, large deviation, and poor stability

3.1. SPZN

Definition 2. A SPZN is a tuple $(P, T, F, M_0, \lambda, Z_P, Z_T, S, C)$, where

(1) P is a set of the places, $P = \{p_1, p_2, \ldots, p_n\}$
(2) T is a set of the transitions, $T = \{t_1, t_2, \ldots, t_n\}$
(3) F is a set of the arcs, which links a place and a transition
(4) M_0 is the initial marking, which describes the initial state of the system
(5) λ is a set of the transition implementation rate, which is a concept with λ in SPN
(6) PN = (P, T, F, M_0) is a basic Petri net
(7) SPN = (P, T, F, M_0, λ) is a stochastic Petri net
(8) PZN = $(P, T, F, Z_P, Z_T, S, C)$ is a PZN
(9) Z_P is a set of the place based on Z
(10) Z_T is a set of the transition based on Z
(11) S: $P \longrightarrow Z_P$ is a set of the one-to-one map relationship between P and Z_P
(12) C: $T \longrightarrow Z_T$ is a set of the one-to-one map relationship between T and Z_T

To better express SPZN, the corresponding relationship between SPN and Z is shown in Figure 9. Among them, the implementation rate of transition in SPN can be used as a precondition for the occurrence of transition T, so corresponding to the pre-assertion in Z language, the implementation rate of transition can be reflected in the assertion. In Petri nets, the place is treated as an entity that does not have any action, so it has only attributes, corresponding to the declaration part in the Z language, whereas the transition is treated as an action, and the place is triggered by the transition, so the transition has properties and functions, corresponding to the declaration part and the assertion part in the Z language.

3.2. Model Refining. The modeling process of an intelligent networked vehicle system with SPZN is shown in Figure 10. First, the vehicle's driving data, node device information, and other data from the smart car are obtained. While initializing the intelligent networked car system, the obtained node device information is abstracted, and the Z framework and node device information are established for the system node. The preconditions, post-conditions, input and output parameters are set, and an SPN model is established for the system information transmission process.

According to the above flowchart, the modeling of the system can be realized and the corresponding Markov chain can be constructed to calculate the steady-state probability. Taking measures to protect the vulnerable parts of the

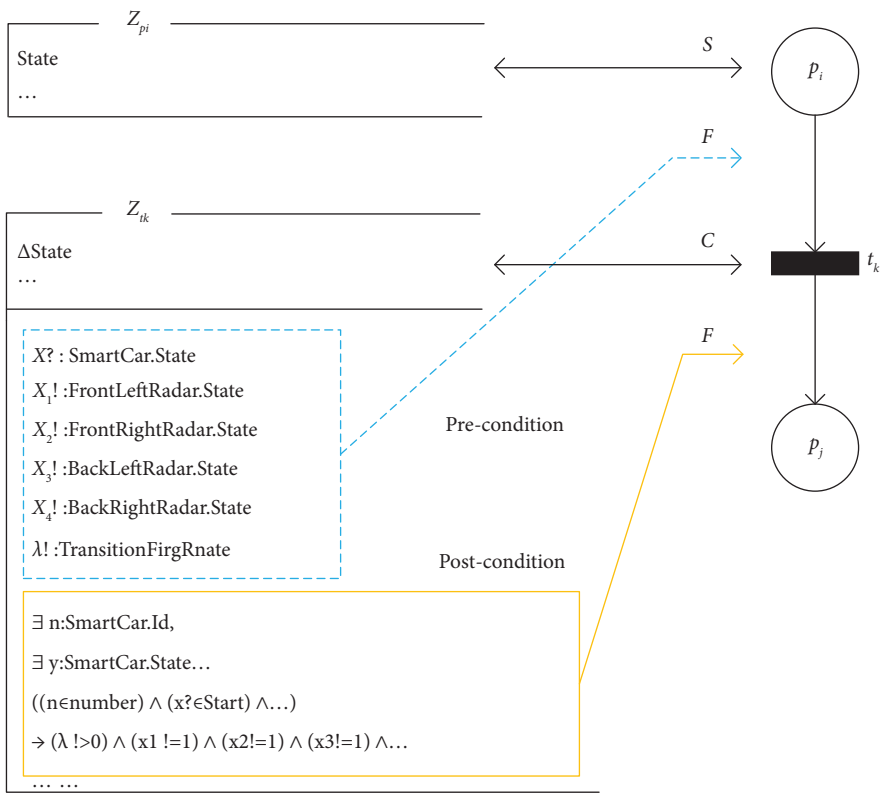

Figure 9: Relationship between SPN and Z in SPZN.

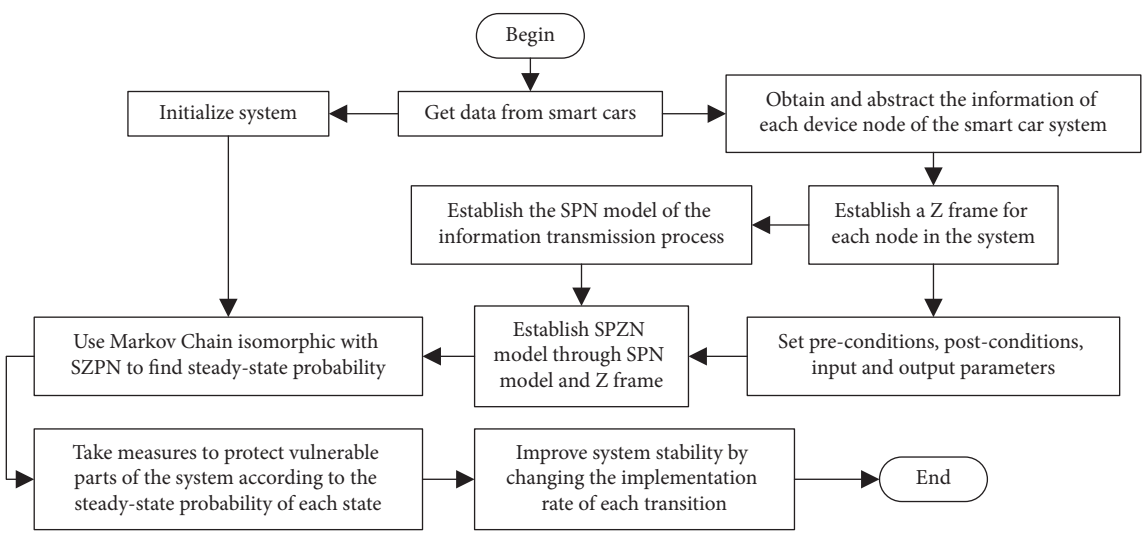

Figure 10: Flow chart of SPZN model establishment.

system based on the steady-state probability, the stability of the system can also be improved by changing the implementation rate of each transition.

3.3. Optimization Method of Parameter λ. In stochastic Petri nets, the transition implementation rate λ needs to be set for each transition. It is known through equation (5) in the Markov process that the transition implementation rate can affect the system steady-state probability. In practical problems, however, it is not easy to make reasonable assumptions about the rate of transition implementation based on improving system stability. In the papers of Song et al. [24] and Sun and Li [23], only the assumed transition implementation rate is used, and no method for determining the transition implementation rate is given. To this end, we propose an optimization method for the transition implementation rate λ, which uses the actor-critic algorithm to

Input: SPN, Environment, the Reward and Punishment Rules.
Output: λ.
Let the initial global time be t and the SPN Contains I Places and m transactions.
(1) The reachable marking graph and Markov Chain with n states Can be derived from the SPN.
(2) Random initialization transactions implementation rate $\lambda_t = \{\lambda_{t,1}, \lambda_{t,2} \ldots \lambda_{t,m}\}$ and let Action $= \{\lambda_{1^{++}}, \lambda_{1^{--}}, \cdots \lambda_{m^{++}}, \lambda_{m^{--}}\}$.
(3) The nth-order square matrix Q is introduced based on the Markov Chain obtained in Step 1 and Definition 1.
(4) According to the Markov Chain in Step 1, the steady-state probability vector $P_t = (P_t(M_0), \ldots P_t(M_{n-1}))$, P_t is an n-dimensional vector.
(5) According to equation (5), the vector P_t at the current moment t is calculated.
(6) Set the set of states $GS = \{g_1, g_2, \ldots, g_t\}$, the set of bad states $BS = \{b_1, b_2, \ldots, b_t\}$. By P_t then the steady-state probability vector of good states $P_{GS,t} = (p(g_{1,t}), p(g_{2,t}), \ldots, p(g_{x,t}))$, the steady-state probability vector for the bad state $P_{BS,t} = (p(b_{1,t}), p(b_{2,t}), \ldots, p(b_{y,t}))$.
(7) iteration = 0;
(8) While (iteration ≥ 100)
(1) Observe the current state s_t,i.e., the current steady-state probability P_t. Random initialize the policy network $\pi(\cdot|s_t;\theta_t)$ and the value network $q(s,a;w)$ and randomly sample an action a_t according to the policy network.
(2) Execute action a_t. The environment generates the next states $s_{t+1} = P_{t+1}$, according to action a_t. The reward R_t is calculated according to the reward and punishment rules.
(3) The policy network $\pi(\cdot|s_{t+1};\theta_t)$ randomly samples an action $a_{t+1}\prime$ according to the current state s_{t+1}, but does not execute the action $a_{t+1}\prime$.
(4) From the value network $q(s,a;w), q_t = (s_t, a_t; w_t), q_{t+1} = (s_{t+1}, a_{t+1}\prime; w_t)$.
(5) Calculate TD error according to TD algorithm $\delta_t = q_t - (R_t + \gamma \cdot q_{t+1})$, and γ is the discount rate.
(6) Calculate the value network gradient $d_{w,t} = \partial q(s_t, a_t, w)/\partial w | w = w_t$.
(7) Update the value network, $w_{t+1} = w_t - \alpha \cdot \delta_t \cdot d_{w,t}$.
(8) Calculate the policy network gradient, $d_{\theta,t} = \partial \log \ \pi(a_t|s_t,\theta)/\partial\theta|\theta = \theta_t$.
(9).Update the policy network, $\theta_{t+1} = \theta_t + \beta \cdot q_t \cdot d_{\theta,t}$.
(10) t++ iteration++,
where α and β correspond to the learning rates in the value network and policy network, respectively.

Algorithm 1: Parameter λ optimization algorithm based on actor-critic.

train and optimize the transition implementation rate λ in SPZN-SPN to improve the stability of the system, as shown in Algorithm 1.

We initialize the global time t. If there exists an SPN containing l places and m transitions, then the implementation rate $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ is randomly initialized. The corresponding reachable marking graph and Markov chain containing n states can be obtained from the SPN. From Definition 1, equation (5), and the Markov chain, the steady-state probability vector of each state at the current moment t can be calculated, $P_t = (P_t(M_0), \ldots, P_t(M_{n-1}))$.

In the actor-critic algorithm, there are four elements: state, environment, action, and reward. The environment corresponds to equation (5), and calculating equation (5) yields the current state, i.e., the steady-state probability P_t of each state at the current moment t. We take changing the rate of transition implementation as the action in the algorithm, and for each transition implementation rate λ, there exist two actions, i.e., increasing and decreasing by an order of magnitude, so that for a stochastic Petri net with m transitions there are a total of $2 \times m$ actions. Action $= \lambda\{1++, \lambda_1--, \ldots, \lambda_m++, \lambda_m--\}$, where λ_i++ represents the action that adds one to the value of λ_i and λ_i-- represents the action that subtracts one from the value of λ_i. The reward should be used as part of the environment, but equation (5) does not have its function. Therefore, we redefined the reward and punishment rules of the environment as shown in equations (7) and (8).

The purpose of this algorithm is to increase the probability of stable states and decrease the probability of nonideal states, so we group the stable states into the set of good states $GS = \{g_1, g_2, \ldots, g_x\}$, and the undesirable states are grouped into the set of bad states $BS = \{b_1, b_2, \ldots, b_y\}$, and it should be noted that $GS \cap BS = \varnothing$ and $x + y \leq n$. By finding the steady-state probabilities of the corresponding states from P_t, we have the good state probability vector $P_{GS,t} = (P(g_{1,t}), \ldots, P(g_{x,t}))$ and the bad state probability vector $P_{BS,t} = (P(b_{1,t}), \ldots, P(b_{y,t}))$.

Because changing the rate of transition implementation may cause the probability of good states and bad states to increase or decrease simultaneously, it is stipulated that the probability of a good state at time $t+1$ is higher than the probability of time t, and then, the positive feedback score R_t^+ given by the environment is greater and vice versa. The greater the probability of a bad state, the greater the negative feedback score R_t^- given by the environment and vice versa. If the probabilities of good states and bad states are the same, then the positive and negative feedback scores remain the same. At time t, the score R_t given by the environment is the difference between the positive feedback score and the

negative feedback score. The above is an explanation of equations (7) and (8) and the reward and punishment rules.

$$\begin{cases} P_{GS,t+1} - P_{GS,t} = (P_t(g_1), \ldots, P_t(g_x)), \\ P_{BS,t+1} - P_{BS,t} = \left(P_t(b_1), \ldots, P_t\left(b_y\right)\right), \end{cases} \tag{7}$$

$$\begin{cases} R_t^+ = P_t(g_1) + \cdots + P_t(g_x), \\ R_t^- = P_t(b_1) + \cdots + P_t\left(b_y\right), \\ R_t = R_t^+ - R_t^-. \end{cases} \tag{8}$$

4. Modeling Analysis

4.1. Reachability. Reachability is the most basic dynamic property of Petri nets, and all other properties must be defined by reachability, so whether a Petri net is reachable is crucial.

Definition 3. Let $\Sigma = (P, T, F, M_0)$ be a Petri net; if there is $\exists t \in T$ that makes $M[t > M'$, then M' is called directly reachable from M. If there are transition sequences $t_0, t_1, \ldots, t_{k-1}$ and marking sequences $M_0, M_1, \ldots, M_k$, such that $M_0[t_0 > M_1[t_1 > M_2 \cdots M_{k-1}[t_{k-1} > M_k]$, then M_k is said to be reachable from M_0. The set of all markings reachable from M_0 is denoted as $R(M_0)$.

By abstracting the above definitions, the algorithm for verifying the reachability of Petri nets as shown in Algorithm 2 can be obtained.

Generally speaking, to verify that a Petri net is reachable, it can be judged by constructing a reachable marking graph. As shown in the reachable marking graph in Figure 5(b), it can be seen that each state of the reachable marking graph is reachable, and there is no isolated state, so the Petri net has good reachability.

4.2. Boundedness and Safety. For any Petri net, we have Definition 4 and Definition 5 to verify its safety and boundedness.

Definition 4. Let $\Sigma = (P, T, F, M_0)$ be a Petri net; if there is a positive integer B such that $\forall M \in R(M_0)$: $M(p) \leq B$, then the place p is called bounded, and the smallest positive integer B that satisfies this condition is called the bound of place p, denoted as $B(p)$.

$$B(p) = \min\{B | \forall M \in R(M_0)\colon M(p) \leq B\}. \tag{9}$$

When $B(p) = 1$, the place p is said to be safe.

Definition 5. Let $\Sigma = (P, T, F, M_0)$ be a Petri net; if any place is bounded, it is called a bounded Petri net.

$$B(\Sigma) = \max\{B(p) | p \in P\}. \tag{10}$$

We call $B(\Sigma)$ the bounds of Σ. When $B(\Sigma) = 1$, we call Σ is safe.

By abstracting Definition 4 and Definition 5, the algorithm for verifying the boundedness and security of Petri nets as shown in Algorithm 3 can be obtained. Among them, $S(p)$ and $S(\Sigma)$ are the security of the place p and Petri net, respectively. When $S = 0$, the object is unsafe, and when $S = 1$, the object is safe. For the Petri net shown in Figure 5(a), by observing its operation, it is not difficult to see that the places p_0 to p_4 are bounded, and their bounds are 1, so they are all safe. According to Definition 4 and Definition 5, the Petri net is safe and bound.

4.3. Advantage. For the recent research on the security of intelligent connected vehicles, Abu et al. [35] discussed what attack methods are available when attacking intelligent connected vehicles, what solutions can be adopted to defend against attacks, and the efficiency comparison of different solutions. Vijayarangam et al. [36] conducted a more in-depth discussion on the "data fascination attack," an attack method in intelligent connected vehicles, and proposed a model to detect data fascination attacks to enhance the security of connected vehicles. Additionally, the authors propose a model to reduce time in the case of traffic jams. With the above two models, ICVs can be prevented from data fascination attacks and the throughput efficiency can be improved.

The above two research studies are all proposed models and methods to improve the security of intelligent connected vehicles when they are attacked. They have not studied the security of intelligent vehicles during operation, but the SPZN model we propose can study the safety of smart car during operation. Since SPN has randomness that TPN does not have, it can vividly describe some random events. It is unavoidable that various random events occur during the vehicle's operation. Therefore, describing random events is the most crucial thing for smart cars. Notably, the best solution to random events is to prevent them in advance.

By adjusting the transiton implementation rate λ in SPN, in this study, an optimization method for λ is also proposed. Through reinforcement learning, the optimal λ value can be quickly found, which improves the security of the system.

Compared with SPN, SPZN also has a Z framework structure that SPN does not have, which can better describe an abstract system. Because the Z framework restricts the places and transitions, it can reduce the number of states in the system. The problem of state explosion in traditional Petri nets is effectively avoided. As shown in Table 2, the advantages of SPZN are more intuitively displayed.

5. A Case Study

To verify the effectiveness of our modeling method for the analysis and verification algorithm, in this section, we simplify the sensor control system of the smart car and only consider the situation of the speed control subsystem when the smart car is driving on a straight road with random events, such as shown in Figure 11. Suppose a smart car has four radar sensors, two video monitors, an acceleration controller, a deceleration controller, a brake, a front sensor subsystem, a rear sensor subsystem, an acceleration system, a deceleration system, and a braking system.

The first step of model building is to obtain the Z frame of each node. Due to the limited space, only the Z frame in part of the system model is given.

Input: $\sum$.
Output: $R(M_0)$.
if $\exists t \in T, M_0[t > M$
$M \in R(M_0), R(M_0) = \{M\}$
else if $\exists t_0, t_1, \cdots t_2 \in T, M_0[t_0 > M_1[t_1 > M_2 \cdots M_{k-1}[t_{k-1} > M_k$
$M_1, M_2, \ldots M_k \in R(M_0), R(M_0) = \{M_1, M_2, \ldots M_k\}$
else
$R(M_0) = \{\varnothing\}$.

Algorithm 2: Algorithm for verifying reachability of Petri nets.

Input: $\sum$.
Output: $B(p), B(\sum), S(\mathrm{p})$ and $S(\sum)$.
if $\forall M \in R(M_0)$: $M(p) \leq B$
$B(p) = \min\{B | \forall M \in R(M_0) M(p) \leq B\}$
if $B(p) = 1$
$S(p) = 1$
else
$S(p) = 0$
if $\forall p \in P, \exists B(p)$
$B(\sum) = \max\{B(p) | p \in P\}$
if $B(\sum) = 1$
$S(\sum) = 1$
else
$S(\sum) = 0$
else
$B(\sum) = 0$
else
$B(p) = 0$

Algorithm 3: Algorithms for verifying the boundedness and security of Petri nets.

Table 2: Difference between Z, PZN, TPZN, SPN, and SPZN.

	Framework	Dynamic	Randomness
Z	✓	×	×
PZN	✓	✓	×
TPZN	✓	✓	×
SPN	×	✓	✓
SPZN	✓	✓	✓

SmartCar
Id: number
Brand: T oyota, Honda, Ford, BYD, ...
Type: small, middle, large,
Version: VersionNumber
AnticollisionRadar: FrontLeftRa, FrontRightRa, BackLeftRa, BackRightRa, LeftRa, RightRa
Monitor: FrontMo, BackMo
System:RadarSystem, MonitorSystem, AcceleratingSystem, BrakeSystem, ...
... ...
State: Start, Stop, Acceleration, Deceleration, ...

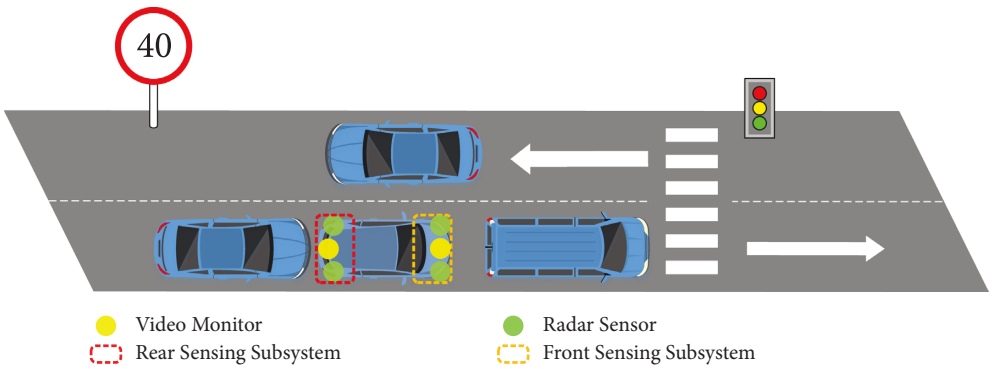

Figure 11: Illustration of smart car running in a case environment.

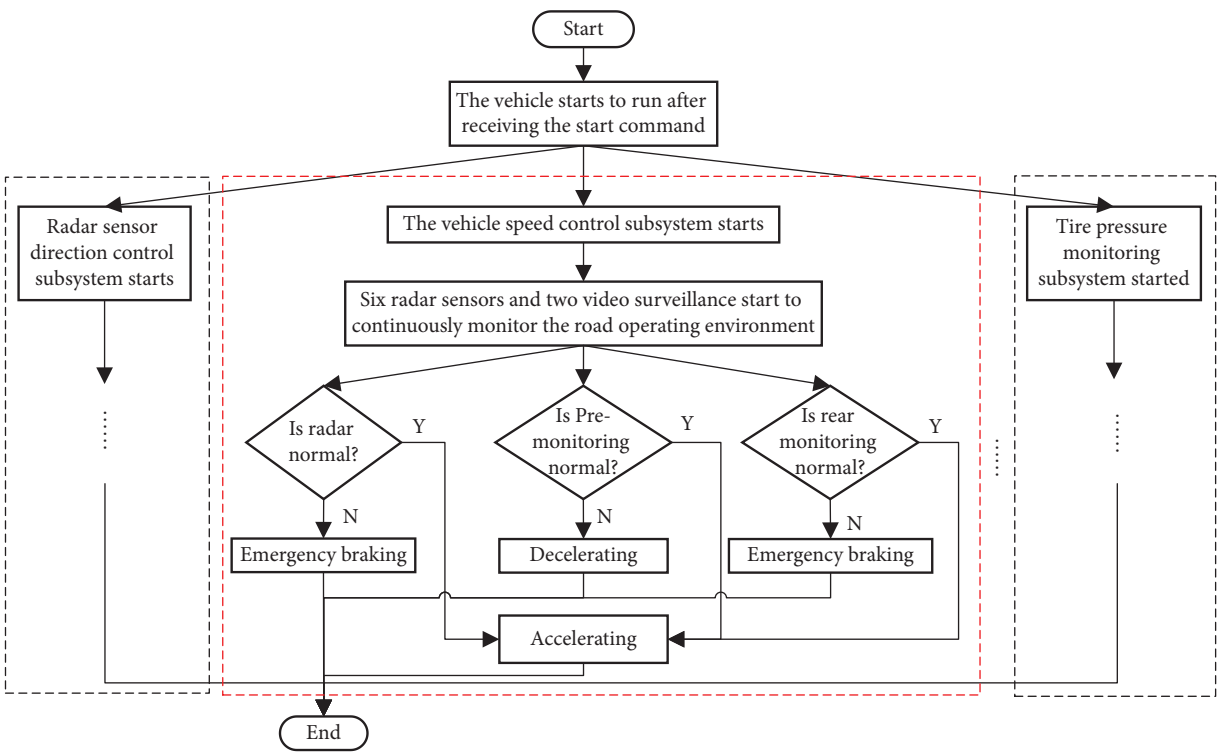

Figure 12: Flowchart in the case study.

The frame above shows the definition of the Z frame for smart cars, which is represented by Z_p in the system, and the blue dashed box in the picture defines the Z frame for the related equipment in the system. The following picture shows the definition of the Z frame in system. A certain node device is defined, and it is also an element in Z_p.

FrontLeftRa

Name: FrontLeftRadar
Type: AnticollisionRadar
Time: LocalTime, GlobalTime
Distance: SaftDistance, DangerDistance, LimitDistance
State: Close, Error, Activation

In the frame below, we have defined action in the system. It is also an element in Z_t, corresponding to a transition in SPZN.

START

ΔSmartCar, ΔFrontLeftRa, ΔFrontRightRa
ΔBackLeftRa, ΔBackRightRa, ΔFrontMo, ΔBackMo
. . . . . .
x?: SmartCar.State
x1!: FrontLeftRadar.State
x2!: FrontRightRadar.State
x3!: BackLeftRadar.State
x4!: BackRightRadar.State
λ!: StateTransitionProbility
. . . . . .

∃ n:SmartCar.Id, ∃ y:SmartCar.State . . .
((n ∈number)^(x?∈Start)^. . .)
→(λ!>0)^(x1!=1)^(x2!=1)^(x3!=1)^
. . .

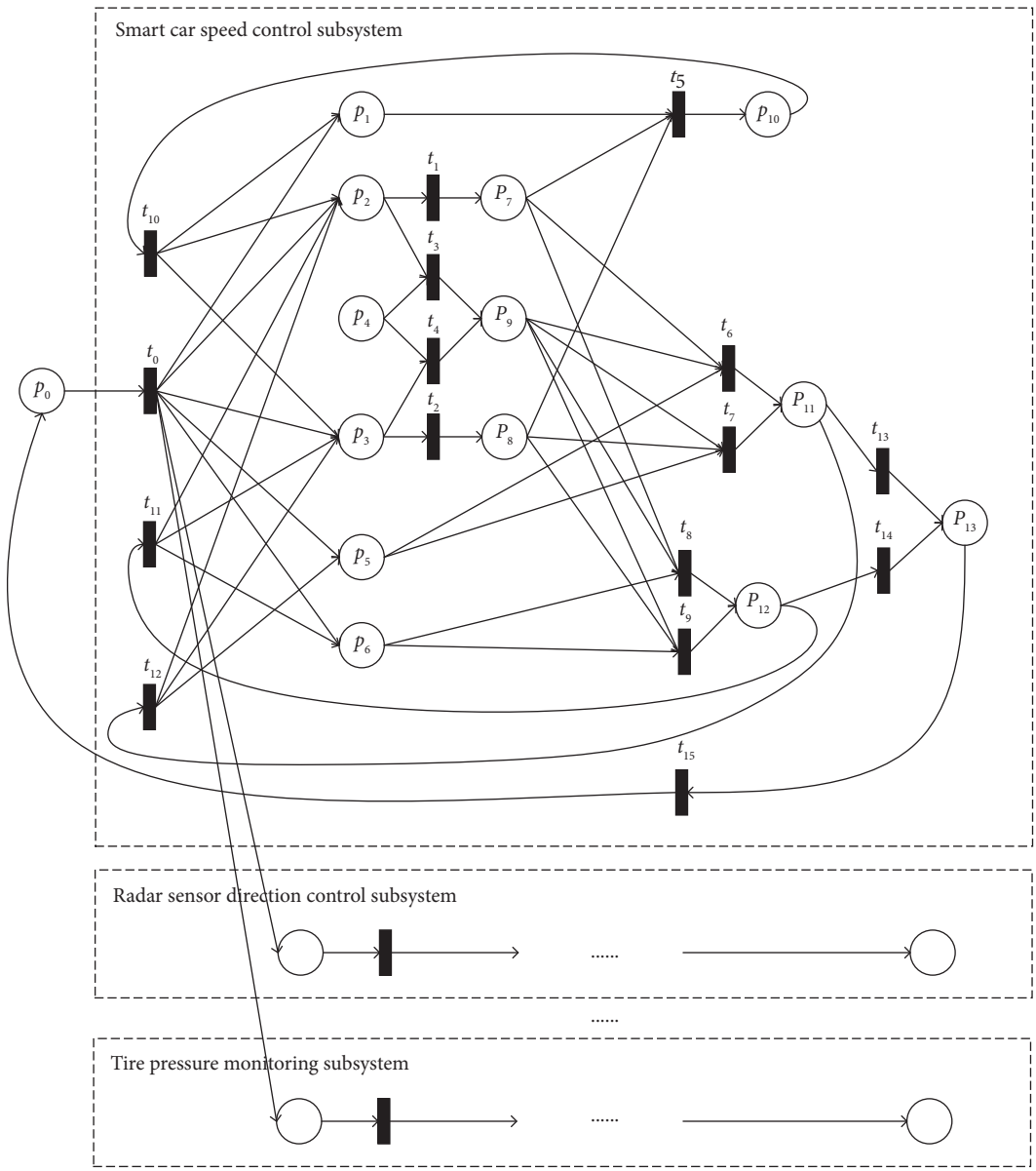

Figure 13: SPN model of the case study.

In the first step of model building, we obtained the Z frame structure of each node device, place, and transition. The SPN model of the intelligent vehicle speed control subsystem will be constructed in the next second step. Before constructing the SPN model, we analyzed the process of this subsystem and got the flow chart shown in Figure 12. The part of the constructed SPN model is shown in Figure 13.

In the intelligent vehicle speed control subsystem shown in Figure 13, the transition t_0 is triggered from the initial place p_0, indicating that the vehicle has been started. Then, the various sensors of the vehicle are monitored when the vehicle is started. If a random event occurs, corresponding processing will be performed according to the type of random event. There are usually three processing modes in the vehicle speed control subsystem: acceleration, deceleration, and braking. After processing random events, the system continues to monitor individual sensors. The meaning of each place and transition is shown in Table 3.

It can be seen from Figure 13 and Table 3 that there are mainly two situations in this case: the normal driving of the vehicle and the random event of the vehicle. Therefore, we will discuss the above two situations separately.

5.1. The Normal Driving Condition of the Vehicle. In this subsection, only the normal driving of the smart car is considered, and there will be no random events before and after the vehicle. Therefore, it is very easy to obtain the SPN model of the normal driving of the car as shown in Figure 14, and the meanings of the corresponding places and transitions in this figure are shown in Table 4.

Table 3: Meaning of each place and transition.

Place	Meaning
p_0	Initial state of the vehicle
p_1	Acceleration controller
p_2	Front sensor subsystem
p_3	Rear sensor subsystem
p_4	Random events
p_5	Deceleration controller
p_6	Automobile brake
p_7	Front sensor subsystem is normal
p_8	Rear sensor subsystem is normal
p_9	Warning message
p_{10}	Keep running
p_{11}	Deceleration
p_{12}	Brake
p_{13}	Vehicle damage
Transition	**Meaning**
t_0	Start
t_1	Processing normal information for front sensor subsystem
t_2	Processing normal information for rear sensor subsystem
t_3	Handling exception information for front sensor subsystem
t_4	Handling exception information for rear sensor subsystem
t_5	Accelerating
t_6	Slowdown caused by random events behind the vehicle
t_7	Slowdown caused by random events in front of the vehicle
t_8	Braking due to random events behind the vehicle
t_9	Braking due to random events in front of the vehicle
t_{10}	Continuous monitoring of vehicle status during normal driving
t_{11}	Slow down and continuously monitor vehicle status
t_{12}	Braking and continuous monitoring of vehicle status
t_{13}	Vehicle collides while decelerating
t_{14}	Vehicle collides while braking
t_{15}	Sending the vehicle to a repair shop for repair

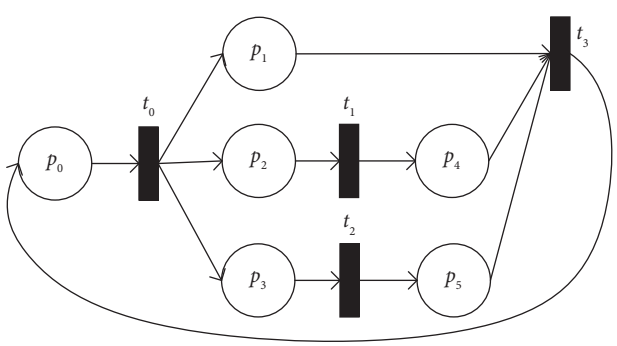

Figure 14: SPN model under normal driving of the vehicle.

It is easy to construct its reachable marking graph according to the constructed SPN model. Because the reachable marking graph of SPN is isomorphic with the Markov chain, the Markov chain can be used to calculate the steady-state probability of each possible state and use related mathematical methods to analyze the equilibrium state and change law of the system. Therefore, we constructed the isomorphic Markov chain as shown in Figure 15 from the reachable marking graph.

It can be seen from Figure 15 that the Markov chain contains 5 states, so we can obtain a 5×5-order transition matrix Q from Definition 1, as shown as follows. Assume that there is a steady-state probability vector $P = (P(M_0), P(M_1), \ldots, P(M_4))$, and the following equation can be obtained from equation (5):

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 \\ 0 & -\lambda_1 - \lambda_2 & \lambda_1 & \lambda_2 & 0 \\ 0 & 0 & -\lambda_2 & 0 & \lambda_2 \\ 0 & 0 & 0 & -\lambda_1 & \lambda_1 \\ \lambda_3 & 0 & 0 & 0 & -\lambda_3 \end{pmatrix},$$

$$\begin{cases} \lambda_0 P(M_0) = \lambda_3 P(M_4), \\ \lambda_0 P(M_0) = (\lambda_1 + \lambda_2) P(M_1), \\ \lambda_1 P(M_1) = \lambda_2 P(M_2), \\ \lambda_2 P(M_1) = \lambda_1 P(M_3), \\ \lambda_3 P(M_4) = \lambda_2 P(M_2) + \lambda_1 P(M_3), \\ \sum_0^4 P(M_i) = 1. \end{cases} \tag{11}$$

Assume that the value of the transition implementation rate λ_i is shown in Table 5; then, we can get the steady-state probability in each state, as shown in Table 6.

From Table 6, the steady-state probability $P(M_i)$ of each state M_i occurring at the corresponding implementation

Table 4: Meaning of each place and transition.

Place	Meaning
p_0	The vehicle is running normally
p_1	Acceleration controller
p_2	Front sensor subsystem
p_3	Rear sensor subsystem
p_4	Front sensor subsystem is normal
p_5	Rear sensor subsystem is normal
Transition	**Meaning**
t_0	Continuous monitoring of vehicle status during normal driving
t_1	Processing normal information for front sensor subsystem
t_2	Processing normal information for rear sensor subsystem
t_3	Accelerating

rate of transition can be obtained. The steady-state probability should be as large as possible for the state favorable to the system stability, and the steady-state probability should be as small as possible for the state unfavorable to the system stability. For a state with a small probability, if the state is a key state in the system, relevant physical means can be used to increase the probability of the state, thereby improving the stability of the system. For example, the vulnerable parts of the vehicle should be inspected and repaired regularly. In addition, the probability of the state can also be changed by changing the rate of implementation of the transitions. For the state M_0, its steady-state probability is 0.4651, which ranks first among all states. The meaning of the state M_0 is that the vehicle is running normally, so there is no need to adjust the transition implementation rate λ_i of the SPN. However, it should be noted that the meaning of state M_2 is to process the normal information of the sensor in front of the vehicle first, and the meaning of state M_3 is to process the normal information of the sensor behind the vehicle first. Although the two have the same effect, the steady-state probability is different. The reason is that the transition implementation rates λ_1 and λ_2 corresponding to the two states are different. It is not difficult to see that different transition implementation rates have a significant impact on the steady-state probability.

5.2. The Random Event of the Vehicle. In this subsection, because the random events in front of the vehicle and the random events behind the vehicle are essentially the same, only the situation where the random event occurs in front of the vehicle is considered, and corresponding measures are taken according to the different random events that occur. The constructed SPN model is shown in Figure 16, and the meanings of the places and transitions in this figure are shown in Table 7, and the corresponding Markov chain is constructed as shown in Figure 17.

It can be seen from Figure 17 that the Markov chain contains 8 states, so we can obtain a 8×8-order transition matrix Q from Definition 1, as shown as follows. Assume that there is a steady-state probability vector $P = (P(M_0), P(M_1), \ldots, P(M_7))$, and the following equation can be obtained from equation (5):

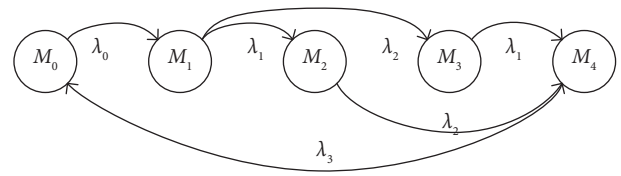

Figure 15: Markov chain corresponding to the SPN model under the normal driving of the car.

Table 5: Implementation rate of each transition.

Rate	Value
λ_0	2
λ_1	3
λ_2	5
λ_3	6

Table 6: Steady-state probability in each state.

State	Probability
$M_0\,(1,0,0,0,0,0)$	0.4651
$M_1\,(0,1,1,1,0,0)$	0.1163
$M_2\,(0,1,0,1,1,0)$	0.0698
$M_3\,(0,1,1,0,0,1)$	0.1938
$M_4\,(0,1,0,0,1,1)$	0.1550

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_1 & \lambda_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\lambda_2 & \lambda_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\lambda_3-\lambda_4 & \lambda_3 & \lambda_4 & 0 \\ 0 & 0 & \lambda_5 & 0 & -\lambda_5-\lambda_7 & 0 & \lambda_7 \\ 0 & 0 & \lambda_6 & 0 & 0 & -\lambda_6-\lambda_8 & \lambda_8 \\ \lambda_9 & 0 & 0 & 0 & 0 & 0 & -\lambda_9 \end{pmatrix}. \tag{12}$$

Assume that the transition implementation rate λ_i is shown in Table 8; according to equation (12), the steady-state probability shown in Table 8 can be obtained.

$$\begin{cases} \lambda_0 P(M_0) = \lambda_9 P(M_6), \\ \lambda_0 P(M_0) = \lambda_1 P(M_1), \\ \lambda_1 P(M_1) = \lambda_2 P(M_2), \\ \lambda_2 P(M_2) = (\lambda_3 + \lambda_4) P(M_3), \\ \lambda_3 P(M_3) = (\lambda_5 + \lambda_7) P(M_4), \\ \lambda_4 P(M_3) = (\lambda_6 + \lambda_8) P(M_5), \\ \lambda_7 P(M_4) + \lambda_8 P(M_5) = \lambda_9 P(M_6), \\ \sum_0^6 P(M_i) = 1. \end{cases} \tag{13}$$

It can be seen from Table 9 that the steady-state probability $P(M_0)$ of state M_0 is 0.2907, which is the largest

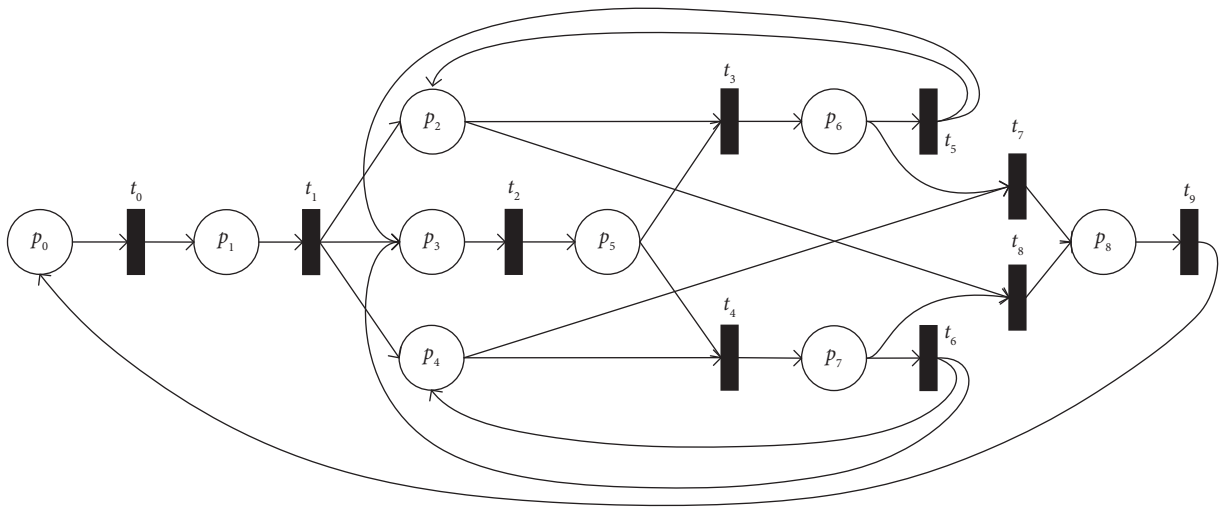

Figure 16: SPN model for random events in front of the vehicle.

among all steady-state probabilities. It can be seen from Table 10 that the meaning of the M_0 state is the vehicle initialization state. For a smart car, the initialization state is obviously not the state we want to maintain for a long time, but this state is not an unacceptable state. So, the state is neither a good state nor a bad state. For the good state in the system, the steady-state probability needs to be increased, and for the bad state in the system, the steady-state probability needs to be reduced, thereby improving the system stability.

We keep the other transition implementation rates unchanged and only change the value of the transition implementation rate λ_0, so the change trend of the steady-state probability of each state as shown in Figure 18 can be obtained. It can be seen that with the increase in λ_0, that is, the increase in the starting trigger rate of the vehicle, the steady-state probability of state M_0 will decrease rapidly, and the probability of being in other states will also increase with the increase in λ_0. Different states have different magnitudes of increase. The steady-state probability increases the most for state M_1 in this figure, since this state is directly affected by the transition implementation rate λ_0. It is not difficult to see that when $\lambda_0 > 13$, the steady-state probability of each state tends to be stable, and the steady-state probability of state M_1 is the highest. The corresponding meaning of state M_1 is that the vehicle is running normally, which is the state we want to achieve and maintain for a long time, so increasing the value of λ_0 helps to maintain the stability of the system.

We know that in normal driving, smart cars need to prevent and avoid random events as much as possible.-Whether it is a random event inside the car such as brake failure or high engine temperature, or a random event outside the car such as emergency braking of the car in front or pedestrians passing by, it can cause a huge impact on the normal driving of smart cars. For random events outside the car, we cannot predict in advance, but we can formulate different measures to prevent different random events. For random events in the car, protective facilities can be installed in the relevant parts of the car, and relevant measures can be taken to prevent them.

Table 7: Meaning of each place and transition.

Place	Meaning
p_0	Initial state of the vehicle
p_1	The vehicle is running normally
p_2	Deceleration controller
p_3	Front sensor subsystem
p_4	Automobile brake
p_5	Warning message
p_6	Deceleration
p_7	Braking
p_8	Vehicle damage
Transition	**Meaning**
t_0	Start
t_1	Continuous monitoring of vehicle status during normal driving
t_2	Handling exception information for front sensor subsystem
t_3	Slowdown caused by random events in front of the vehicle
t_4	Braking due to random events in front of the vehicle
t_5	Slow down and continuously monitor vehicle status
t_6	Braking and continuous monitoring of vehicle status
t_7	Vehicle collides while decelerating
t_8	Vehicle collides while braking
t_9	Sending the vehicle to a repair shop for repair

Through the above experiments, it is found that the transition implementation rate has a significant impact on the steady-state probability. Therefore, we use the reinforcement learning method to optimize the transition implementation rate λ and take corresponding preventive measures and solutions to the system according to the optimized transition implementation rate, which can effectively improve the security of the system in a stochastic dynamic environment.

In this case, there are a total of 7 states, in which the set of good states is only M_1, and the set of bad states is only M_6.

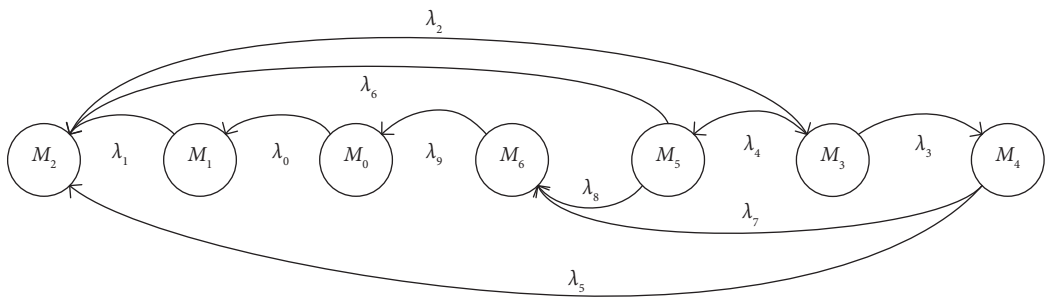

Figure 17: Markov chain corresponding to the SPN model when a random event occurs in front of the car.

Table 8: Implementation rate of each transition.

Rate	Value
λ_0	3
λ_1	4
λ_2	5
λ_3	5
λ_4	3
λ_5	7
λ_6	6
λ_7	2
λ_8	2
λ_9	7

Table 9: Steady-state probability in each state.

State	Probability
$M_0\,(1,0,0,0,0,0,0,0,0)$	0.2907
$M_1\,(0,1,0,0,0,0,0,0,0)$	0.2180
$M_2\,(0,0,1,1,1,0,0,0,0)$	0.1744
$M_3\,(0,0,1,0,1,1,0,0,0)$	0.1090
$M_4\,(0,0,0,0,1,0,1,0,0)$	0.0606
$M_5\,(0,0,1,0,0,0,0,1,0)$	0.0227
$M_6\,(0,0,0,0,0,0,0,0,1)$	0.1246

Table 10: Corresponding meaning of each state.

State	Meaning
M_0	Vehicle initialization
M_1	The vehicle is running normally
M_2	Sensors continuously monitor vehicle status
M_3	The sensor receives abnormal information
M_4	Vehicle deceleration
M_5	Vehicle braking
M_6	Vehicle collision damage

According to the previous definitions of action, environment, good state set, and bad state set, Table 11 can be obtained.

After the basic parameters are set, we still use the data in Table 8 to initialize the transition implementation rate λ. The state corresponding to the transition implementation rate, that is, the steady-state probability vector of each state, is $s = (0.2707, 0.2180, 0.1744, 0.1090, 0.0606, 0.0227, 0.1246)$. After simulation with Algorithm 1, the optimized state is obtained as

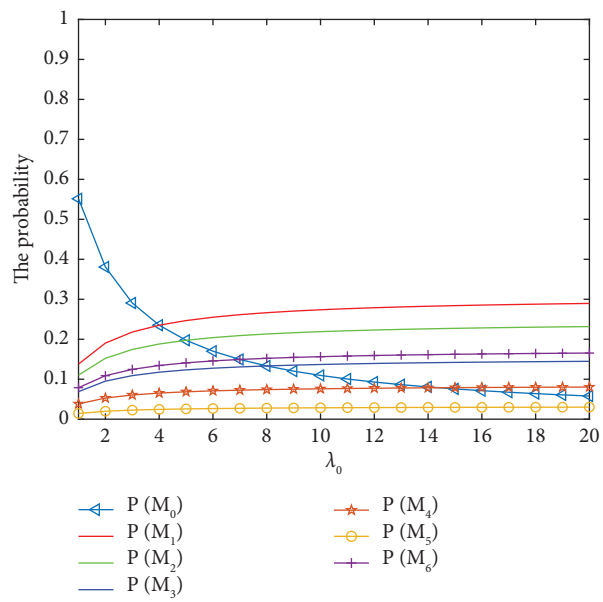

Figure 18: Steady-state probability trend of each state when only changing λ_0.

Table 11: Parameters involved in the experiment.

Parameter	Contents
Action	$\{\lambda_0++, \lambda_0--, \ldots, \lambda_9++, \lambda_9--\}$
GS	$\{M_1\}$
BS	$\{M_6\}$
s_t	$P_t = (P_t(M_0), P_t(M_1), \ldots, P_t(M_6))$
Environment	$\begin{cases} \lambda_0 P(M_0) = \lambda_9 P(M_6) \\ \lambda_0 P(M_0) = \lambda_1 P(M_1) \\ \lambda_1 P(M_1) = \lambda_2 P(M_2) \\ \lambda_2 P(M_2) = (\lambda_3 + \lambda_4) P(M_3) \\ \lambda_3 P(M_3) = (\lambda_5 + \lambda_7) P(M_4) \\ \lambda_4 P(M_3) = (\lambda_6 + \lambda_8) P(M_5) \\ \lambda_7 P(M_4) + \lambda_8 P(M_5) = \lambda_9 P(M_6) \\ \sum_0^6 P(M_i) = 1 \end{cases}$

$s^* = (0.1099, 0.3297, 0.3297, 0.1099, 0.0366, 0.0110, 0.0733)$. It is not difficult to see that the steady-state probability of state M_1 has increased from 0.2180 to 0.3297, and the steady-state probability of state M_6 has decreased from 0.1246 to 0.0733, successfully increasing the steady-state probability of good states

Table 12: Implementation rate of each transition after optimized.

Rate	Value
λ_0	6
λ_1	2
λ_2	2
λ_3	3
λ_4	3
λ_5	7
λ_6	8
λ_7	2
λ_8	2
λ_9	9

and reducing the steady-state probability of bad states. The optimized transition implementation rates are shown in Table 12.

By observing the optimized transition implementation rates, we can find that compared with the previous transition implementation rates, λ_0, λ_6, and λ_9 increase, λ_1, λ_2, and λ_3 decrease, and λ_4, λ_5, λ_7, and λ_8 remain unchanged. According to the corresponding meaning of each transition implementation rate, on the basis of the original vehicle setting, increasing the vehicle start trigger rate and the vehicle brake trigger rate will help to improve the system stability. Prompt maintenance after a car crash also contributes to the stability and safety of the vehicle.

6. Conclusions

In this study, a formal modeling and verification method for smart cars based on stochastic Petri nets and Z language framework is proposed. With this method, the simulation of stochastic events during the driving process of smart cars can be better implemented. In addition, the study uses reinforcement learning to optimize the proposed method to improve the safety of smart connected cars. Experimental cases are given to explain and study the method in detail. Although the method addresses the stochastic nature of events and the abstraction of the system well, it only considers the speed control system of a single smart car driving on a straight road and does not consider the coordinated control of multiple smart vehicles. The complexity and size of the current model will not be conducive to modeling and validation using Petri nets. How to abstract and simplify the model will be our next work.

Data Availability

The experimental data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 62273065 and 61903053 and the Science and Technology Research Program of Chongqing Municipal Education Commission in China under Grants KJZD-K201800701 and KJQN201900702.

References

[1] C. Chen, L. Liu, and S. H. Wan, "Data dissemination for industry 4.0 applications in internet of vehicles based on short-term traffic prediction," *ACM Transactions on Internet Technology*, vol. 22, no. 1, pp. 1–18, 2021.

[2] W. Wei, R. Yang, and H. Gu, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 1–10, 2021.

[3] C. Chen, Y. Zhang, and Z. Wang, "Distributed computation offloading method based on deep reinforcement learning in ICV," *Applied Soft Computing*, vol. 103, no. 7, pp. 1–11, 2021.

[4] C. Wang, C. Chen, and Q. Pei, "An information centric in-network caching scheme for 5g-enabled internet of connected vehicles," *IEEE Transactions on Mobile Computing*, vol. 69, no. 12, pp. 1–14, 2021.

[5] C. Wang, C. Chen, and Q. Pei, "Popularity incentive caching for vehicular named data networking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3640–3653, 2020.

[6] S. Liu, J. Yu, and X. Deng, "FedCPF: an efficient-communication federated learning approach for vehicular edge computing in 6G communication networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1616–1629, 2021.

[7] L. Zhao, C. Wang, and K. Zhao, "INTERLINK: a digital twin-assisted storage strategy for satellite-terrestrial networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 1, 2022.

[8] N. Soni, R. Malekian, and D. Andriukaitis, "Internet of vehicles based approach for road safety applications using sensor technologies," *Wireless Personal Communications*, vol. 105, no. 4, pp. 1257–1284, 2019.

[9] C. M. Martinez, M. Heucke, and F. Y. Wang, "Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 666–676, 2017.

[10] A. Colombo, G. R. D. Campos, and F. D. Rossa, "Control of a city road network: distributed exact verification of traffic safety," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4933–4948, 2017.

[11] F. Khalid, S. R. Hasan, and O. Hasan, "Runtime hardware trojan monitors through modeling burst mode communication using formal verification," *Integration*, vol. 61, no. 3, pp. 62–76, 2018.

[12] Y. Liu, J. Z. Wu, and R. Qiao, "Consistency verification between goal model and process model in requirements analysis of networked software," *Journal of Computational and Theoretical Nanoscience*, vol. 11, no. 5, pp. 1385–1393, 2014.

[13] Y. Liu, J. Z. Wu, and R. Qiao, "Dynamic evolution of requirements process model deployed on networked environment with PZN," *Journal of Computational Information Systems*, vol. 9, no. 8, pp. 3329–3336, 2013.

[14] Y. Liu, J. Z. Wu, and R. Zhao, "Formal verification of process layer with Petri nets and Z," *Advances in Information Sciences and Service Sciences*, vol. 5, no. 1, pp. 68–77, 2013.

[15] P. Herrmann and J. O. Blech, "Formal model-based development in industrial automation with reactive blocks," *Federation of International Conferences on Software Technologies: Applications & Foundations*, vol. 9946, no. 12, pp. 253–261, 2016.

[16] K. Rahul and S. Dutta, "Formal verification of a medical insurance system prototype: the event-B modeling approach," *Journal of Information Assurance & Security*, vol. 17, no. 1, pp. 25–34, 2022.

[17] Y. H. Wang, Q. Zhou, and Y. Zhang, "A formal modeling and verification scheme with an RNN-based attacker for CAN communication system Authenticity," *Electronics*, vol. 11, no. 11, pp. 1–21, 2022.

[18] L. Qi, M. C. Zhou, and W. J. Luan, "Emergency traffic-light control system design for intersections subject to accidents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 170–183, 2015.

[19] L. Qi, M. C. Zhou, and W. J. Luan, "Impact of driving behavior on traffic delay at a congested signalized intersection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1882–1893, 2016.

[20] L. Qi, M. C. Zhou, and W. J. Luan, "A two-level traffic light control strategy for preventing incident-based urban traffic congestion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 13–24, 2016.

[21] K. Labadi, T. Benarbia, and J. P. Barbot, "Stochastic Petri net modeling, simulation and analysis of public bicycle sharing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1380–1395, 2017.

[22] Y. Liu, L. Y. Huang, and J. W. Chen, "Formal verification on the safety of internet of vehicles based on TPN and Z," *Mathematical Problems in Engineering*, vol. 2020, no. 2020, Article ID 6618168, 11 pages, 2020.

[23] Q. Y. Sun and X. Y. Li, "Establishment of emergency coordination model across organizations based on stochastic Petri net," *Journal of Safety Science and Technology*, vol. 11, no. 9, pp. 63–69, 2015.

[24] Y. B. Song, H. B. Mou, and Z. Y. Jiang, "Safety performance analysis of urban rail transit system based on stochastic Petri net," *China Safty Science Journal*, vol. 21, no. 9, pp. 82–87, 2011.

[25] D. Nallaperuma, R. Nawaratne, and T. Bandaragoda, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019.

[26] B. Ji, X. Zhang, and S. Mumtaz, "Survey on the internet of vehicles: network architectures and applications," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 34–41, 2020.

[27] M. K. Molloy, "Discrete time stochastic Petri nets," *IEEE Transactions on Software Engineering*, vol. 11, no. 4, pp. 417–423, 1985.

[28] M. K. Molloy, "Performance analysis using stochastic Petri nets," *IEEE Transactions on Computers*, vol. 31, no. 9, pp. 913–917, 2006.

[29] A. D. Febbraro, D. Giglio, and N. Sacco, "A deterministic and stochastic Petri net model for traffic-responsive signaling control in urban areas," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 510–524, 2015.

[30] M. A. Marsan, G. Balbo, and G. Chiola, "An introduction to generalized stochastic Petri nets," *Microelectronics Reliability*, vol. 31, no. 4, pp. 699–725, 1991.

[31] Z. H. Muhamad, D. A. Abdulmonim, and B. Alathari, "An integration of uml use case diagram and activity diagram with Z language for formalization of library management system," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 4, pp. 3069–3076, 2019.

[32] P. F. Niu, X. F. Wang, and L. Lu, "Survey on vehicle reinforcement learning in routing problem," *Computer Engineering and Applications*, vol. 58, no. 1, pp. 41–55, 2022.

[33] S. M. Yang, Z. Shan, and Y. Ding, "Survey of research on deep reinforcement learning," *Computer Engineering*, vol. 47, no. 12, pp. 19–29, 2021.

[34] K. H. Du, R. Z. Song, and Q. L. Wei, "Review of reinforcement learning applications in machine games," *Control Engineering China*, vol. 28, no. 10, pp. 1998–2004, 2021.

[35] T. M. Abu, S. Abbas, and Q. Nasir, "Systematic literature review on Internet-of-Vehicles communication security," *International Journal of Distributed Sensor Networks*, vol. 14, no. 12, pp. 1–21, 2018.

[36] S. Vijayarangam, G. C. Babu, and S. Ananda Murugan, "Enhancing the security and performance of nodes in Internet of Vehicles," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 7, pp. 1–10, 2021.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7018360, 11 pages
https://doi.org/10.1155/2022/7018360

Research Article

Enhanced Multilink Single-Radio Operation for the Next-Generation IEEE 802.11 BE Wi-Fi Systems

Xiyang Lan,[1] Xinyu Zu,[2] and Jie Yang[2]

[1]*School of Information Science and Engineering, Shandong University, Qingdao 266237, China*
[2]*College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*

Correspondence should be addressed to Jie Yang; jyang@njupt.edu.cn

Received 25 July 2022; Accepted 16 September 2022; Published 6 October 2022

Academic Editor: Chen Chen

For the next-generation Wi-Fi systems, the Enhanced Multilink Single-Radio (EMLSR) operation has become a promising feature to improve the Wi-Fi system performance. The EMLSR enables dynamic switching among multiple bands with low-cost implementation and efficient power consumption. However, with realistic channelization of multiple links with 320 MHz bandwidth on 6 GHz and 160 MHz bandwidth on 5 GHz, the performance of EMLSR is jeopardized due to the crowded 5 GHz band with existing Wi-Fi systems. In addition, the unbalanced bandwidth configuration between an access point (AP) and stations (STAs) multilink devices (MLDs) may result in wasted secondary channels when STAs supporting smaller bandwidth stay on the primary channel only. We proposed the further enhanced EMLSR with a new MAC protocol and link selection mechanism, where a primary link (PL) is selected between AP and STA with the unbalanced Link/Band and one or multiple secondary links (SLs) with balanced Link/Band. The STA tunes the main radio on the PL and the scan radio to one of the SLs. The PL will be more prioritized than the SL based on the link selection algorithm to maximize the channel utilization. Compared with legacy EMLSR, the enhanced EMLSR can improve the utilization of the secondary channel. Even under heavy OBSS load on SL or both PL and SL, the further enhanced EMLSR can achieve 50% to 70% throughput gain.

1. Introduction

The development of IEEE 802.11 standards has been moving quickly in the past few years. With an ever-growing number of devices using Wi-Fi, the next-generation Wi-Fi systems are required to be capable of managing dense scenarios, increased data traffic, and a diverse mix of applications and services with differing requirements. Up to IEEE 802.11ac [1], the evolution of Wi-Fi standards and technologies was focused primarily on achieving successively higher throughput [2]. However, in the real deployments, with lots of users with varying application requirements, the next-generation Wi-Fi system is designed to deliver a satisfactory experience to all the users that are operating in the system [3–7]. The problem is not how high the throughput can be achieved but whether the Wi-Fi system has enough capacity to handle the growing demand for many users with different application requirements.

The latest Wi-Fi standard IEEE 802.11ax [8] or Wi-Fi 6 is designed to improve system performance by introducing several multiuser (MU) technologies, including orthogonal frequency-division multiple access (OFDMA) [9] and target wake time (TWT) [10]. In addition, the 802.11ax standard has enhanced the multiuser multiple input, multiple output (MU-MIMO) mechanism that was firstly introduced by the 802.11ac standard. By using the OFDMA mechanism, an AP can distribute the data transmissions to multiple users into multiple resource units (RUs) and simultaneously transmits data in the RUs to these users. Compared with the single-user transmission mechanism, OFDMA reduces the overhead of channel access to transmit data to the multiple users one by one. With the help of OFDMA, the performance of Wi-Fi systems under dense environment is dramatically

improved [11]. The MU-MIMO mechanism is enhanced as well in the 802.11ax standard to support more spatial streams. Furthermore, MU-MIMO mechanism can be used together with OFDMA to increase the number of users simultaneously in the MU transmissions. One of the existing issues of Wi-Fi system is the overhead associated with the channel access. In order to handle the negative effects of channel contention, the TWT mechanism was introduced in the IEEE 802.11ax amendment. It provides a simple but efficient solution to schedule frame transmissions to different user groups in different time period so the channel contention is limited inside the scheduled time period. In addition to the contention reductions, TWT mechanism can also contribute to the advantage of other transmission mechanisms such as MU transmissions, spatial reuse [12], and coexistence in high-density WLAN networks.

After the IEEE 802.11ax standard was recently concluded, IEEE 802.11 Working Group (WG) [13] has been established for the development of the next-generation Wi-Fi standard named as IEEE 802.11be or Wi-Fi 7. Compared with IEEE 802.11ax [14] that focused on the development of the new DL/UL MU (OFDMA and MU-MIMO) mechanisms to improve the spectrum efficiency, 802.11be spends more efforts on the wider bandwidth operation to improve the throughput and reduce latency, especially considering the limitations from the regulations that prevent Wi-Fi system from using more bandwidth. For example, according to the FCC [15], the current bandwidth on the 6 GHz band for Wi-Fi system is up to 320 MHz. In addition, there are other issues which may not enable wider bandwidth on single band. According to the existing EDCA procedure [16] of the Wi-Fi system, the transmission opportunity (TXOP) is initiated with the backoff procedure running on the primary 20 MHz channel. The channel is determined as busy if the primary 20 MHz subchannel is busy even if the rest of the other 20 MHz channels are idle. Further, the channel with broad bandwidth is determined as busy if any of the 20 MHz channels within the operating bandwidth is busy. In other words, the broader the operating bandwidth is, the less likely the broad bandwidth of channel may be available for data transmissions. Furthermore, the operation in a channel with wider bandwidth consumes more power in a single band. The power consumption behavior is crucial for mobile devices such as phones or tables. IEEE 802.11be has changed the direction of pursing more bandwidth in a single band for maximizing the spectrum utilization. The task group has defined a mechanism named multilink operation (MLO) to explore the benefits of available bandwidth from multiple bands.

Currently, the 802.11be standard is still under development. There are no Wi-Fi products following the 802.11be specification available on the market. Most of the previous studies focused on the performance of the traditional single-link Wi-Fi systems. The IEEE 802.11be task group's technical contributions [17–19] are the building blocks for MLO operation. The contribution in [17] introduces the basic MLO architecture and necessary changes from the exiting Wi-Fi system to the next-generation MLO-based Wi-Fi systems. The contribution in [18] discusses the potential performance gain that can be obtained using the multiradio-based MLO operation. The contribution in [19] introduces the concept of single-radio-based MLO operation and provides initial performance results of single-radio-based MLO system. In paper [20], the authors have studied the impact of MLO operation on the existing 802.11 channel access procedure and presented the throughput performance of a multiradio-based MLO system, where the multiradio system is assumed to support simultaneous transmitting and receiving (STR) capability. In paper [21], the authors have summarized the developing directions of 802.11 standard and discussed the benefits of both the single-radio-based and multiradio-based MLO systems. To the best of the authors' knowledge, the studies on the newly introduced MLO mechanism of 802.11be are still very limited. This contribution is to establish a system throughput model for the existing EMLSR mechanism for performance analysis and further propose the enhancement to improve the existing EMLSR mechanism.

Section 2 reviews the basic MLO mechanisms including multilink mutiradio (MLMR) mode and multilink single-radio (MLSR) mode defined in the IEEE 802.11be draft standard. Then, in Section 3, the existing EMLSR mode of the IEEE 802.11be is introduced and the related issues/problems are analyzed. In Section 4, a system throughput model of existing EMLSR mechanism is given for performance analysis. The further enhanced EMLS mode operation is proposed in Section 5 and performance evaluation based on NS3 simulator is shown to compare the enhanced EMLSR mechanism with the existing one in Section 6. Finally, the conclusion of this contribution is drawn in the last section.

2. 802.11BE Multilink Operation

Today, major AP vendors support the AP solutions with dual-band/triband operation. In these APs, the Wi-Fi MAC and PHY of multiple bands work almost independently and provide multiple independent links to Wi-Fi STAs. As shown in Figure 1, a dual-band AP may have both 5 GHz and 6 GHz enabled simultaneously. The AP may establish BSS1 with 160 MHz bandwidth on the 5 GHz band and BSS2 with 320 MHz bandwidth on the 6 GHz band. Because these two BSSs are operating independently and provide services to different STAs, the 160 MHz bandwidth on the 5 GHz band and the 320 MHz bandwidth on the 6 GHz band cannot be combined to achieve the optimal wider bandwidth operation of 160 + 320 MHz.

In order to optimize the system spectrum utilization and achieve better throughput performance, the IEEE 802.11be has defined the MLO to support sending data frames concurrently on multiple links. The MLO allows the users to enjoy the multilink benefits unavailable for a simple non-contiguous wide spectrum on a single link, such as asynchronous channel access and enhanced power save. As shown in Figure 1, the MLO can aggregate a various number of links of different widths, for example, 160 MHz + 40 MHz. The 802.11be has introduced a concept of a multilink device (MLD) [22] as illustrated on the right side of Figure 1 which

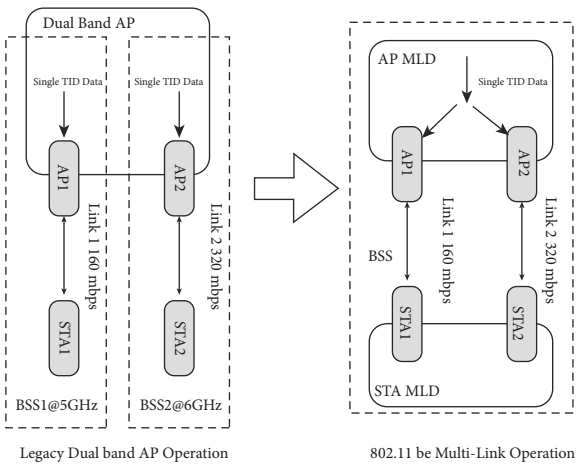

Figure 1: IEEE 802.11be MLO.

consists of multiple AP/STAs but with a single interface to the upper layer. The upper layer protocol takes the MLD as a single Wi-Fi device. Despite having multiple PHY/MAC interfaces, MLD has a single MAC address and uses this MAC address as its own identity. The sequence number space is shared by the AP/STAs within the same MLD. The MLO simplifies the procedures for frame fragmentations and frame reassembly, duplication detection, and dynamic link switching. It enables frame transmission and retransmission on any link regardless of the link of the initial transmission of the frame.

The IEEE 802.11be group has designed a new association procedure that allows AP MLD and STA MLD to establish a connection on any supported link [23]. The capabilities of all the links can be exchanged in any enabled link as well. 802.11be also enables two types of acknowledgment modes, referred to as restricted and dynamic link switch. In the restricted mode, data frames and ACKs are bound to one link [24]. Management exchanges transmitted over one link, such as relation to power save mode, security key negotiation, and Block ACK (BA) negotiation, apply only to this link. It is a simple scheme of multiple independent links with enabled aggregation. In the dynamic link switch mode, multiple links can be used for transmission of the same flow. Management information and negotiations sent over one link can apply to other links. This mode enables load balancing and congestion avoidance. It also improves peak throughput and reduces latency, overhead, and power consumption.

According the MLD types, there are different flavors [25] of MLO specified in the IEEE 802.11be draft standard. The MLD types are defined based on the capabilities of the MLD. If an MLD implements multiple radios and uses these multiple radios concurrently for the MLO, then these devices are defined as multilink multiradio (MLMR) MLD. If an MLD only implements single radio and still wants to operate multiple links, then these devices are called multilink single-radio (MLSR) MLD. After an MLSR STA MLD associates with AP MLD, it may establish MLO with the AP MLD. However, due to the limitation of having only one radio, the MLSR STA MLD cannot use multiple links concurrently. The single radio needs to switch back and forth between multiple bands in a time domain multiplex (TDM) fashion. Normally the band switch operations of a single radio require both time and extra signaling. Therefore, the MLSR operation only allows an MLSR STA MLD to switch the band in a static fashion; that is, the MLSR STA MLD may have to park the radio on one band for several minutes finishing a data transmission session and then switch to another band. To provide more flexibility, the IEEE 802.11be draft spec has defined an Enhanced Multilink Single-Radio (EMLSR) operation [26] to enable an MLSR MLD to dynamically switch band to improve both throughput and latency performance. The EMLSR provides flexibility for an EMLSR MLD STA to switch dynamically between bands to improve the opportunities to obtain a TXOP. We will discuss the details of EMLSR from the next section.

3. EMLSR Operation and Related Issues

Most of the MLSR STAs implement either a configurable radio that has the flexibility of switching between two 1×1 radios and one 2×2 radio or a scan radio in addition to the main radio to support the EMLSR operation. An example of using scan radio to support EMLSR operation is shown in Figure 2. Using the other architecture of configurable radio has a similar issue, so we skip it here to save some text. When an AP MLD intends to conduct EMLSR operation with an EMLSR STA MLD, each AP within the AP MLD tries to access the corresponding band/channel by running EDCA function independently. In this example, AP1 of the AP MLD is operating on the 6 GHz band and AP2 is operating on the 5 GHz band. If the EDCA function completes the backoff procedure, the corresponding AP starts frame exchange procedure by sending Initial Control Frame (ICF).

In this example, AP1 on the 6 GHz band completes backoff first, so AP1 sends an MU-RTS [27] frame to start EMLSR operation. MU-RTS is one of the ICF types. The STA MLD, in order to operate under the EMLSR mode, configures scan radio (STA1) on the 6 GHz band and main radio (STA2) on the 5 GHz band. When AP1 on the 6 GHz band sends out MU-RTS, the scan radio receives the MU-RTS and understands the following Downlink (DL) data transmission which will be carried out on the 6 GHz band. The scan radio has limited functionality and is not capable of receiving data frames with high MCS or NSS (e.g., above MCS4 or NSS = 2). Therefore, the main radio that stays on the 5 GHz band needs to be tuned to the 6 GHz band. The tuning procedure that involves a number of PHY/MAC operations (e.g., PLL settling, register configuration, etc.) consumes nonnegligible time. According to the IEEE 802.11be spec, the band switching time ranges from 16 μs to 256 μs depending on the implementation.

To accommodate this band switching delay and let the STA MLD be able to respond within the SIFS (16 mu s), it is defined that the MAC padding field is added to the ICF. Different from PHY padding or packet extension (PE) [28], the MAC padding is a MAC frame field with specific pattern and is added before the Frame Check Sequence (FCS). When the scan radio is in the process of receiving the ICF, it

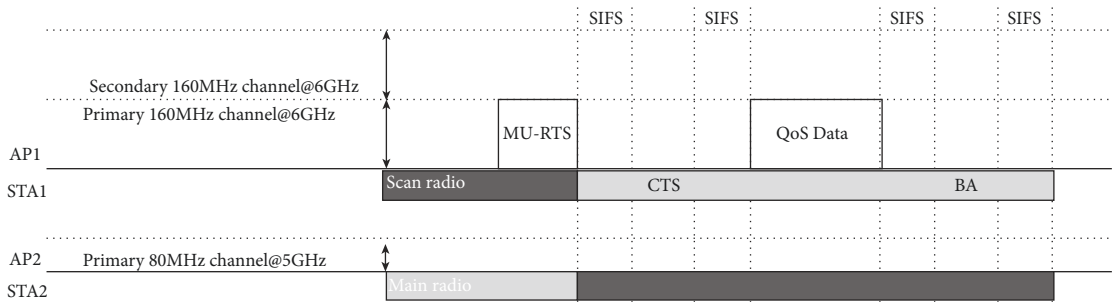

Figure 2: EMLSR operation frame sequence.

identifies the MAC padding field by matching the special pattern and starts the band switching operation before the FCS validation is conducted; that is, once the start of the MAC padding field is identified, the scan radio indicates the main radio to start the band switch procedure and the scan radio stays on the same band to complete the reception of the ICF. During this process, the scan radio performs CFO correction and other PHY related tasks and passes the obtained information to the main radio so that the main radio can get prepared to send out Trigger Based (TB) PPDU as the response. The STA when sending a TB PPDU as a response has a requirement that is more stringent than the requirement for the same STA sending out a regular Single User (SU) PPDU. It is because when an STA sends out a TB PPDU, other STAs at the same time might be triggered to send out TB PPDUs as well.

For the AP to be able to correctly decode frames in the TB PPDUs from multiple STAs, the TB PPDUs from multiple STAs are required to meet both timing and power accuracy requirements [29]. While the PHY of the scan radio is performing the above operations, the MAC of the scan radio reads the band switching information in the ICF assuming that the FCS check will pass later, as well as passing this information to main radio so that the main radio can start band switching operation. During the association procedure, the STA MLD indicates its required switching time, so when AP MLD sends out the ICF, MAC padding field with sufficient length is added to the ICF. When the scan radio in the process of the ICF reception, at the end of MAC frame, performs FCS check, if the FCS passes, then the scan radio indicates to the main radio that FCS is successful and the main radio may send out the response frame (e.g., a Clear to Send (CTS) or Buffer Status Report (BSR) frame) after SIFS. Otherwise, if the FCS check fails, then the scan radio shall indicate to the main radio to revoke the band switch operation.

Compared with other MLO options, the EMLSR has its advantages. Firstly, compared with the true MLMR MLO options, STA MLD with EMLSR does not need to implement multiple radios, so it is more cost and power consumption efficient, while it still allows end user to have the MLO operation with low-cost implementation; that is, during the EDCA procedure, both AP and STA have the flexibility of exploring multiple bands. Secondly, relative to the single-radio option, EMLSR does not compromise the capability of fast switching between multiple bands and achieves better system performance than MLSR.

As described in the above paragraphs, the EMLSR is a promising feature for the next-generation Wi-Fi systems. However, there are several issues that need to be solved to optimize the system performance. First, there is an issue caused by the bandwidth configuration of the 6 GHz and the 5 GHz BSS operation. According to the channelization [30], the 6 GHz band is cleaner and allows 320 MHz BSS operation. However, it is only allowed to operate a BSS up to 160 MHz in the 5 GHz band. In most of scenarios, it is really challenging to secure a 160 MHz PPDU transmission in the 5 GHz band due to the legacy Wi-Fi systems coexisting on the 5 GHz band. When an AP or an STA on the 5 GHz band intends to transmit a 160 MHz PPDU, it needs to complete the EDCA procedure on the whole 160 MHz channel, which means an STA needs to complete the backoff procedure on the primary 20 MHz channel and pass the ED check on the secondary 20 MHz, 40 MHz, and 80 MHz channels to finally be able to initiate the transmission. However, due to the legacy Wi-Fi systems that are already deployed, if any of the secondary channels within the 160 MHz channel is occupied by other 802.11be BSSs or a legacy BSS, the PPDU transmission is not allowed to be started. Therefore, in the crowded 5 GHz band, it is very unlikely that an AP or an STA can transmit 160 MHz PPDU in most of the typical Wi-Fi operation scenarios. As shown in Figure 3, when an AP MLD performs EMLSR operations on the 5 GHz and 6 GHz bands, each AP associated with the AP MLD runs EDCA operation on the 5 GHz and 6 GHz bands independently. Let us assume that both the 5 GHz and 6 GHz bands are equally busy and OBSSs only exist on the primary channels. Then the chance for an AP or an STA to be able to acquire the channel is almost the same. In other words, if the EDCA function is conducted independently, then the backoff on either band may be completed and the possibility of that is almost the same for either band. Now, as mentioned early, the 6 GHz is running on 320 MHz, while the 5 GHz can only run on 160 MHz or even 80 MHz. An EMLSR STA may end up with using the 6 GHz or the 5 GHz band with equal probability and switching back and forth between transmitting PPDUs with 320 MHz on 6 GHz and PPDU with 160 MHz or 80 MHz. Compared with an STA that does not support EMLSR and only operates on 6 GHz band, an EMLSR STA may have even worse performance due to the transmission spent on the narrow channel on the 5 GHz band.

Second, there is another issue caused by the bandwidth configuration difference between AP MLD and STA MLD. As shown in Figure 4, the spectrum regulation already allows

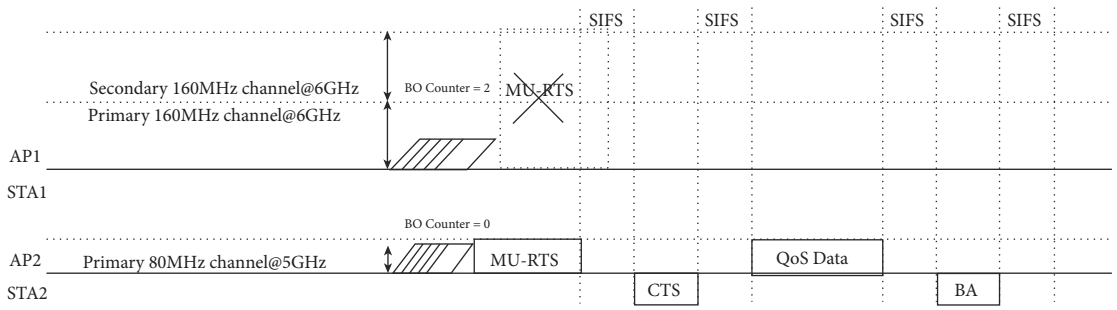

Figure 3: Unbalanced bandwidth issue of EMLSR.

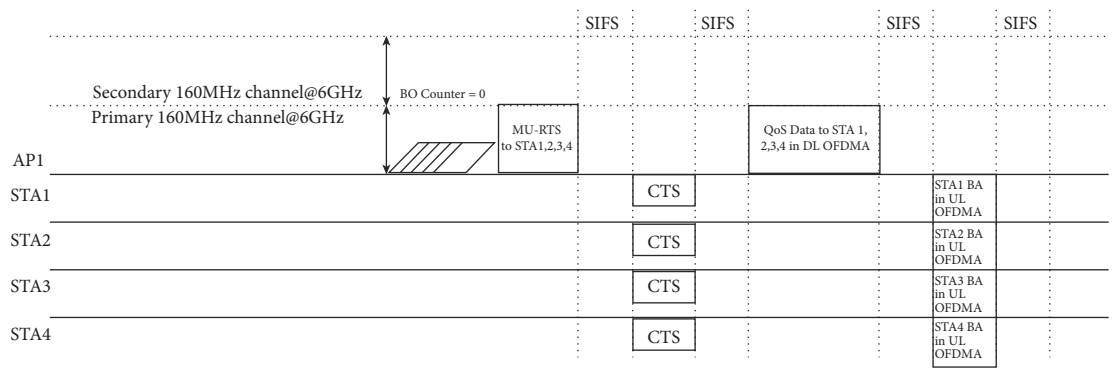

Figure 4: AP STA bandwidth configuration issue of EMLSR.

320 MHz operation in the 6 GHz band and 160 MHz operation in the 5 GHz band. So today almost all the AP implementation processes support 320 MHz BSS operation on the 6 GHz band and 160 MHz BSS operation on the 5 GHz band. On the other hand, due to the cost and power consumption consideration, most STA implementation processes support only 80 MHz in the 5 GHz band and 160 MHz in the 6 GHz band. It is because, according to the IEEE 802.11 specification, 80 MHz is mandatory for STA on the 5 GHz band and 160 MHz is mandatory for STA on the 6 GHz band. As shown in Figure 4, on the 6 GHz band, when an AP or an AP MLD has established an 320 MHz BSS, all the associated STAs need to stay on the primary channels. Only if the STAs are capable of 320 MHz operation, half of the bandwidth of the 320 MHz BSS will be wasted. Although EMLSR is a promising mechanism to improve the performance for the next-generation Wi-Fi systems, in reality there is a lack of the capabilities to utilize the secondary channels.

4. System Modeling

Let us assume that both AP MLD and STA MLD support the multilink operation MLO with only two enabled links. The concept can be extended to the multilink operation MLO with more than two links, where p_1 is the probability that link 1 is busy and p_2 is the probability that link 2 is busy. According to IEEE 802.11 specification, the Clear Channel Assessment (CCA) mechanism [1] is used to decide whether a channel is busy. On the primary 20 MHz channel of the system operating channels, the CCA checks the incoming energy by running Energy Detection (ED) and detects the incoming 802.11 packet by running Packet Detection (PD). On the other hand, on the secondary 20 MHz channels of the system operating channels, only ED is performed. Therefore, the probability that a primary 20 MHz channel is busy and the probability that a secondary 20 MHz channel is busy could be slightly different. In this paper, to simplify the analysis without loss of generality, we consider that the probability of a busy 20 MHz channel is assumed to be the same regardless of whether it is for the primary or secondary 20 MHz channels. Hereafter, we use p to represent the probability that a 20 MHz channel is busy. Then, the probability that link 1 is busy is expressed as follows:

$$p_1 = 1 - (1 - p)^m, \tag{1}$$

where m is number of 20 MHz channels of the system operating channels of link 1. If link 1 is operating on the 6 GHz band, the value of m is equal to 12 which enables 320 MHz operation. Similarly, the probability that link 2 is busy can be expressed as follows:

$$p_2 = 1 - (1 - p)^n, \tag{2}$$

where n is number of 20 MHz channels of the system operating channels of link 2. If link 2 is operating on the 5 GHz band, the value of n is equal to 6 which enables 160 MHz operation.

Now the maximum throughput c_1 that can be obtained on link 1 and the maximum throughput c_2 that can be obtained on link 2 can be calculated based on Shannon capacity. c_1 can be expressed as follows:

$$c_1 = 20m10^6\log_2(1 + \mathrm{SINR}), \tag{3}$$

and c_2 can be expressed as follows:

$$c_2 = 20m10^6\log_2(1 + \mathrm{SINR}). \tag{4}$$

The total maximum throughput of an MLMR system then can be expressed as follows:

$$T_{\mathrm{MLMR}} = (1 - p_1)c_1 + (1 - p_2)c_2. \tag{5}$$

By substituting p_1, p_2, c_2, c_2 in equation (5) with the results of equations (1)–(4), the maximum system throughput of an MLMR system with two enabled links can be derived as follows:

$$T_{\mathrm{MLMR}} = 20\left(m(1 - p)^m + n\left(1 - p^n\right)\right)10^6\log_2(1 + \mathrm{SINR}). \tag{6}$$

For an EMLSR system, there is only one radio available for frame transmissions. Therefore, link 1 can be used for transmission under one of two conditions. The first condition is that link 1 is not busy while link 2 is busy. The second condition is that both links are not busy, and then we can assume that both links can be used for transmission with the same probability. Transmission on link 2 follows similar conditions. The maximum system throughput for the EMLSR system with two enabled links can be expressed as follows:

$$T_{\mathrm{EMLMR}} = (1 - p_1)p_2c_1 + (1 - p_2)p_1c_2 + \frac{(1 - p_1)(1 - p_2)(c_1 + c_2)}{2}. \tag{7}$$

Following the same way of deriving the maximum throughput of MLMR system in equation (7), the maximum system throughput of the EMLSR system can be expressed as follows:

$$T_{\mathrm{EMLMR}} = 20\left(\frac{(1 - p)^m(1 - (1 - p)^n)m + (1 - p)^n(1 - (1 - p)^m)n + (1 - p)^m(1 - p)^n(m + n)}{2}\right) \cdot 10^6\log_2(1 + \mathrm{SINR}). \tag{8}$$

5. Further Enhanced EMSLR Operation

The following mechanism is proposed to further improve the EMLSR performance. The mechanism can be divided into two parts. In the first part, an enhanced protocol is proposed to solve the AP STA unbalanced bandwidth issue, and, in the second part, a link selection mechanism is proposed to solve the band unbalanced bandwidth issue.

As shown in Figure 5, we define the primary link (PL) and the secondary link (SL) for the EMLSR operation. The PL is the Link/Band where the AP STA unbalance issue occurs, and the SL is the Link/Band where AP STA unbalance issue does not exist. In this example, 6 GHz is defined as the PL where AP MLD on the 6 GHz band is operating on 320 MHz, while the STA MLD on the same band is operating on 160 MHz. In the other Link/Band (e.g., 5 GHz), both the AP MLD and the STA MLD are operating on 80 MHz. In this proposal, it is assumed that only one PL and one or more SLs are allowed. The reason is that, in a practical deployment, no AP MLD could support more than one 320 MHz link due to cost considerations.

When an STA MLD is under the procedure to associate with the AP MLD, AP MLD and STA MLD need to exchange per-band capabilities, for example, Operating Bandwidth Capabilities and MCSs. If the STA MLD figures out that, on the 6 GHz band, its own bandwidth capability is smaller than that of AP MLD on the same band, the STA MLD determines this link as the PL and conveys this information to the AP MLD. Other links are determined as SLs. Then STA MLD tunes the scan radio to one of the SLs and tunes the main radio to the PL. This example illustrates only the case in which STA MLD supports one PL and one SL. But the protocol can be extended to support the case of multiple SLs.

The two following scenarios are evaluated, respectively: (a) AP MLD completes the backoff procedure on the SL first; (b) AP MLD completes the backoff procedure on the PL first. Under scenario (a) where the backoff procedure is completed firstly on the SL, AP2 in the AP MLD sends out the MU-RTS with non-HT DUP PPDU of 80 MHz bandwidth on the 5 GHz band. Because both AP2 in the AP MLD and STA2 in the STA MLD are operating on 80 MHz, there is no need to explore the secondary channels. The main radio on STA2 of the STA MLD after receiving the MU-RTS from AP2 stays on the 5 GHz band and responds with a CTS after SIFS. AP2 after receiving CTS continues sending DL QoS Data to STA2. STA2 after receiving the DL QoS Data sends BlockAck back to AP2 to complete the current TXOP. Under scenario (b) where the backoff procedure is completed on the PL firstly, AP1 in the AP MLD sends out the MU-RTS in the same PHY PPDU format but with 320 MHz bandwidth

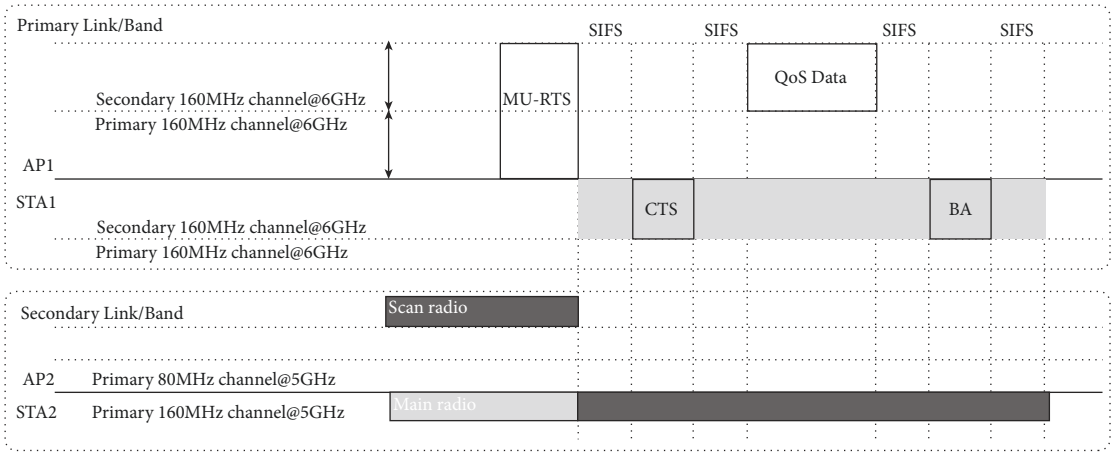

Figure 5: EMLSR optimization protocol.

on the 6 GHz band. STA1, different from STA2 in scenario (a), operates only on 160 MHz bandwidth, and, instead of the main radio, it is the scan radio that receives the MU-RTS.

Although the scan radio of STA1 is operating on 160 MHz while the MU-RTS is coming in with 320 MHz bandwidth, because the MU-RTS is sent out using the non-HT DUP PPDU format, the same information is repeated per 20 MHz channel so that the scan radio that operates on the partial bandwidth of full BSS bandwidth can still successfully receive the MU-RTS. When the scan radio is in the process of receiving the MU-RTS, following the regular EMLSR protocol, the scan radio detects the MAC padding field of the MU-RTS by matching the special pattern of the MAC padding field. Once the MAC padding field is detected, the scan radio indicates to the main radio who is staying on the SL to start the procedure of tuning to the secondary 160 MHz channel of the PL. During the reception of the MU-RTS, the scan radio collects PHY related information and passes this information to the main radio so that it can get ready to send CTS back to AP1 SIFS after the reception of the MU-RTS. The CTS is sent back using the non-HT DUP PPDU with 160 MHz bandwidth on the secondary 160 MHz channel of the 320 MHz BSS. Once the CTS is received, AP1 sends out QoS Data on the secondary 160 MHz channel of the 320 MHz BSS, and a BlockAck frame is sent out by STA1 after the QoS Data is successfully received to complete the current TXOP.

In general, the principle of the link selection algorithm is to prioritize the maximization of the broader bandwidth utilization. The SL will be used only when the PL is busy with its own transmission or occupied by transmissions from other STAs. Then the maximum system throughput of the enhanced EMLSR system with two enabled links can be expressed as follows:

$$T_{\text{EEMLMR}} = (1 - p_1)c_1 + p_1(1 - p_2)c_2, \tag{9}$$

and, following the same way through which equation (8) is derived, the maximum throughput of the enhanced EMLSR system can be expressed as follows:

$$T_{\text{EEMLMR}} = 20\left((1 - p)^m m\right) + \left((1 - p)^n \left(1 - (1 - p)^m\right)n\right) \cdot 10^6 \log_2(1 + \text{SINR}), \tag{10}$$

where p_1 is the possibility that the PL is busy and p_2 is the possibility that the SL is busy. ρ_1 is the PHY supported data rate for certain MCS when rate selection is enabled. ρ_2 is the corresponding PHY data rate of the SL. T is the instantaneous throughput given the busy possibility and selected MCSs. As mentioned earlier in the section, SL will be used only when PL is busy with its own transmission or occupied by transmissions from other STAs. Then the throughput upper bond can be defined as follows:

$$T = (1 - p_1)\rho_1 + p_1(1 - p_2)\rho_2. \tag{11}$$

In a real Wi-Fi system, the accurate ratio for a certain link to be busy is unknown. We can only estimate it based on the historical usage of certain channel. It requires long duration of sampling period to have accurate estimation, which is not suitable for TXOP-based EMLSR operation. In this paper, we propose the following link selection algorithm that is based on the cross-link signaling between PL and SL. Because EDCA function is running on both the PL and the SL, each of the PL and SL shall maintain its own backoff counters and NAV independently. We assume that there is a tunnel between the PL and SL so that the PL and SL can exchange their NAV information and backoff counter information at each slot boundary. Again, let us start with the two scenarios as we did in the previous section. But, this time, we will start with scenario (b), which is the scenario in which the PL finishes the backoff procedure firstly. Recall that the principle of the link selection

algorithm is to maximize the use of the PL. So when PL finishes the backoff earlier, it will move forward to start the frame transmission without checking the NAV and backoff counter information on the SL. The EDCA function on the PL is exactly the same as that of a non-MLO operation. On the other hand, in scenario (a), which is the scenario in which the SL completes the backoff procedure firstly, as shown in Figure 6, at the boundary of the slot, SL checks the NAV of the PL through a special tunnel. If either the physical CS or the NAV on the PL indicates that the PL is busy, then the SL link can move forward to start the frame transmission. Otherwise, if both the physical CS and the NAV on the PL indicate that the PL is idle, then the PL is either in the idle state or in the process of the EDCA. Now the SL needs to compare the remaining backoff counter number with a predefined threshold. If the number is bigger than the threshold, meaning that the PL is far away from completing the backoff procedure, then the SL can move forward to start the frame transmission. On the other hand, if the remaining backoff number is smaller than the threshold, the SL shall redraw a new backoff counter number using the same AC [26] and restart the backoff process so that the TXOP can be given to the PL. The value of the threshold depends on the network deployment and can be optimized according to the channel status on both PL and SL.

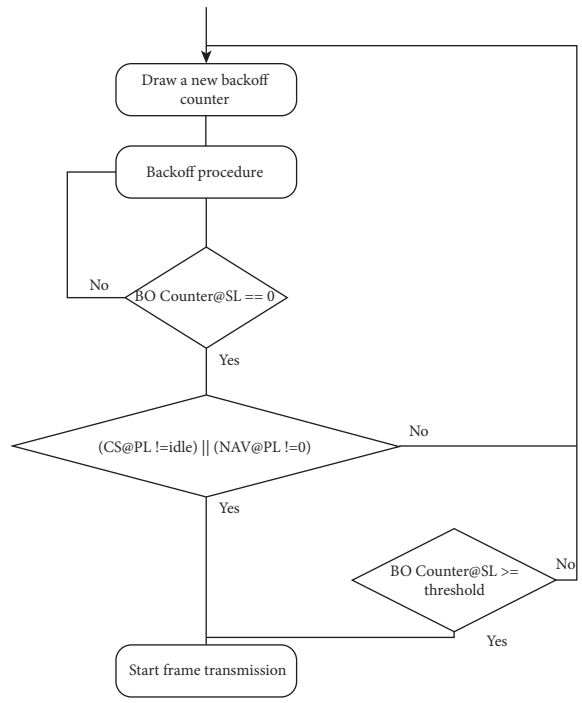

Figure 6: EMLSR link selection algorithm.

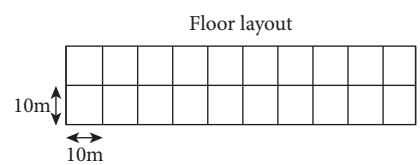

Figure 7: Simulation scenario.

6. Performance Evaluation

The performance of further enhanced EMLSR is evaluated based on the NS3 simulator [32] with following assumptions. The simulation scenario is shown in Figure 7. It is a single floor apartment build scenario with each apartment with the size of 10 m × 10 m. In each apartment, there is one AP MLD configured and two STA MLDs associated with the AP MLD. The penetration loss of the wall between each apartment is 5 dB. After the association, AP MLDs start DL SU type of transmissions with their associated STA MLDs in a round robin fashion. There are no DL MU transmissions or UL transmissions configured in any of the apartments. All the STA MLDs are configured with stationary positions with no mobility. Both AP MLD and STA MLD support 5 GHz and 6 GHz EMLSR operations. The AP MLDs are configured to use two TX/RX antennas on both the 5 GHz and 6 GHz bands. The STA MLDs are configured to use only one TX/RX antenna under the EMLSR mode. HE MCS0 to MCS11 are enabled. Depending on the configuration, the rate selection is enabled/disabled. EDCA with default parameters per each traffic class is used. For operating channels, on 5 GHz, a random channel with 80 MHz bandwidth is selected, and, on 6 GHz, a random channel with 160 MHz bandwidth is selected. We only enabled AMPDU aggregate with 64 BA window. There is no AMSDU aggregation. There is no regular RTS configured. Only MU-RTS is enabled to start the EMLSR operation sequence. Each AP MLD is independently managed.

Table 1 summarizes the channel model and path loss configurations, where $\mathrm{PL}(d) = 40.05 + 20 \times \log 10\ (f_c/2.4) =, 20 \times \log 10(\min(d,5)) + (d > 5) \times 35 \times \log(d/5) + 18.3 \times F^{(F+2)/(F+1)-0.46} + 5W$, where $d = \max(3\mathrm{D\ distance}[m], 1)$, f_c is frequency with unit GHz, F denotes number of floors traversed, and W denotes the number of walls traversed in x-direction plus number of walls traversed in y-direction.

We turn on the BSS in the first apartment and turn off the rest of the BSSs in the first set of experiment; that is, there are no OBSSs. Rate selection is turned off as well. As shown in Figure 8, the x-axis shows the MCSs variation and the y-axis shows the throughput of original EMLSR and our proposed enhanced EMLSR of the BSS under test. It is demonstrated that enhanced EMLSR provides better throughput performance on all the MCSs. With MCS0, which is the base rate for all the management frame and control frame transmission, the proposed enhanced EMLSR provides 30 Mbps more throughput on top of 26 Mbps that is achieved on the original EMLSR operation. Almost 100% throughput gain is achieved. With MCS11, we have similar observations. The throughput gain here comes from the capabilities of the enhanced EMLSR that can utilize the secondary channel of the broader bandwidth, while the original EMLSR can only utilize the primary 80 MHz channels.

Table 1: Channel model and path loss configurations.

Fading model	TGac channel D NLOS for all the links
Path loss model	PL(d)
Shadowing	Log-normal with 5 dB standard deviation, i.i.d. across all links

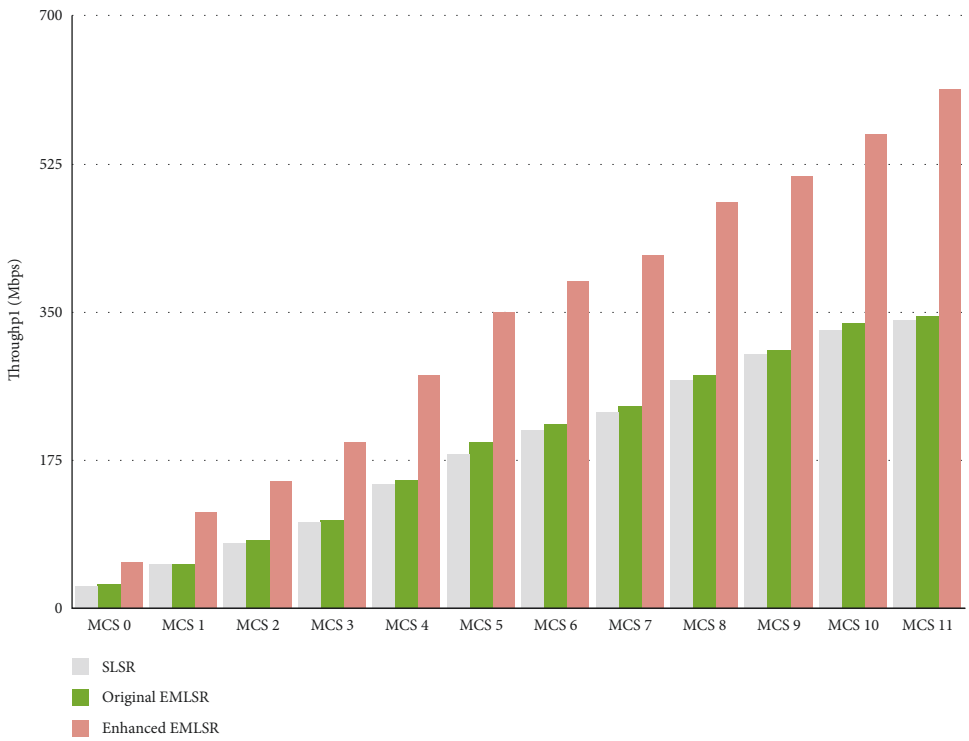

Figure 8: Throughput performance without OBSSs.

The performance is also evaluated in a scenario in which all BSSs are turned on. and, on top of that, one AP and one STA BSS in each apartment are added. The one BSS added will create OBSS traffic for each of the BSSs in the apartment. All the OBSSs added only run DL SU traffic and a global knob is implemented which can control ratio of the OBSS traffic against my BSS traffic. Assuming the same radio of OBSS traffic on both PL and SL, as shown in Figure 9, the x-axis shows the ratio of traffic of OBSS against that of my BSS and the same ratio applies to both PL and SL. With increasing of the OBSS traffic load, the average BSS throughput is decreased. However, the proposed further enhanced EMLSR still outperforms the original EMLSR. With the OBSS traffic load increased to 80%, as shown in Figure 10, the average BSS throughput using original EMLSR is 70 Mbps, while with the further enhanced EMLSR the average BSS throughput is increased to 120 Mbps. So, under heavy OBSS load on both PL and SL, the further enhanced EMLSR achieves 70% throughput gain.

Let us then look at a more practical scenario where the SL (5 GHz) is busier than the PL (6 GHz). Because the 6 GHz band has been recently opened by the regulation bodies, there are not many deployments on the 6 GHz band yet. The 5 GHz band, on the other hand, supports legacy Wi-Fi BSS down to 802.11ac or 802.11n. So, it is a reasonable assumption that the SL is busier than the PL in the real-world deployment. For simulation studies, the OBSS load on the PL stays the same at 20% and the OBSS load on the SL increases from 20% all the way up to 80%. Again, with increased OBSS traffic load, the average throughput performance is decreased. However, with the further enhanced EMLSR being adopted, the average BSS throughput is still better than that of the original EMLSR. Under the heavy BSS load of 80% on the SL, the original EMLSR provides 200 Mbps throughput, while the throughput of the further enhanced EMLSR reaches 300 MHz. The 50% throughput gain is achieved by adopting the further enhanced EMLSR. The main contribution of the throughput performance gain comes from the flexibility of the enhanced EMLSR. It distributed more TXOPs on the PL that is less busy tham the SL, and the bandwidth of the PL is broader than that of SL. It is interesting to observe that the original EMLSR has worse performance than the baseline SLSR system under heavy OBSS load on the SL. This is because the original EMLSR may try to use the smaller bandwidth of the 5 GHz band when the EDCA is completed on the 5 GHz band. But it is more efficient for the SLSR system to wait a bit long on the 6 GHz band to obtain a TXOP with bigger bandwidth.

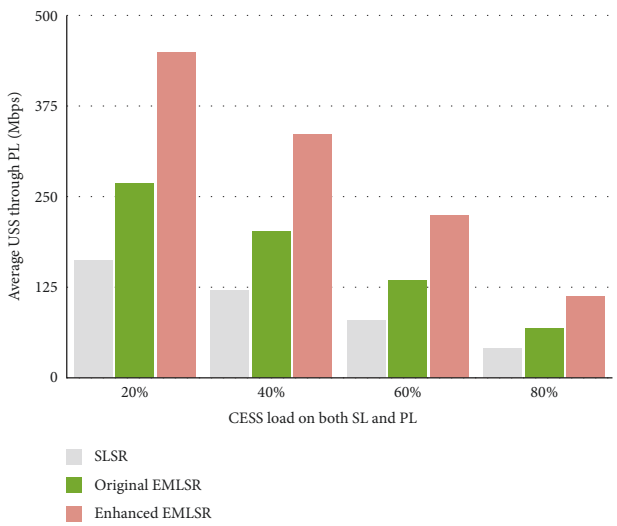

Figure 9: Throughput with OBSS on both PL and SL.

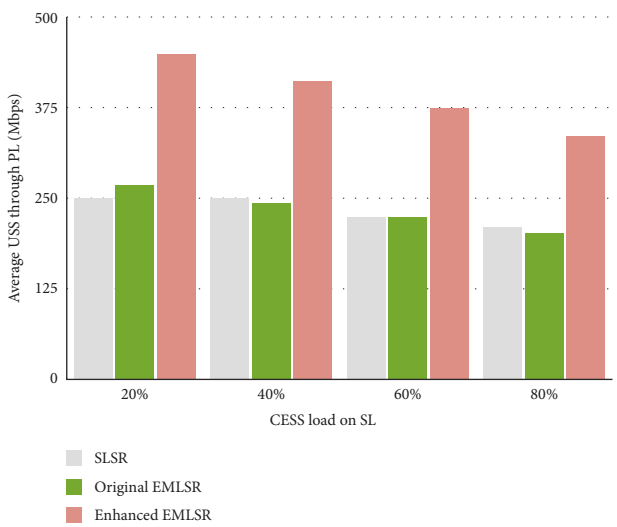

Figure 10: Throughput with OBSS on SL only.

7. Conclusion

The MLO is a new mechanism defined in the IEEE 802.11 draft specification. With the help of MLO, APs and STAs will be provided with the capabilities to transmit and receive data from the same traffic flow over multiple radio interfaces. The EMLSR is an enhanced feature of MLO enabling dynamic switching between multiple bands with low-cost implementation and efficient power consumption. In this paper, we proposed the further enhanced EMLSR with a new MAC protocol and link selection mechanism, where a primary link (PL) is selected between AP STA with the unbalanced Link/Band and one or multiple secondary links (SLs) with balanced Link/Band. The STA tunes the main radio on the PL and the scan radio to one of the SLs. The PL will be more prioritized than the SL based on the link selection algorithm to maximize the channel utilization. Compared with the legacy EMLSR, the further enhanced EMLSR can improve the utilization of the secondary channel. Even under heavy OBSS load on SL or both PL and SL, the enhanced EMLSR can achieve 50% to 70% throughput gain.

Data Availability

The data used to support the findings of the study can be obtained 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 Key Research and Development Program of China, under Grant no. 2021ZD0113003.

References

[1] in*Wireless LAN Medium Access Control and Physical layer Specification*IEEE, Piscataway, NJ, USA, 2013.

[2] F. Tian, Y. Yu, X. Yuan, B. Lyu, and G. Gui, "Predicted decoupling for coexistence between WiFi and LTE in unlicensed band," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4130–4141, 2020.

[3] C. Chen, J. Jiang, Y. Zhou, N. Lv, X. Liang, and S. Wan, "An edge intelligence empowered flooding process prediction using internet of things in smart city," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 66–78, 2022.

[4] Y. Wang, L. Guo, Y. Zhao et al., "Distributed learning for automatic modulation classification in edge devices," *IEEE Wireless Communications Letters*, vol. 9, no. 12, pp. 2177–2181, 2020.

[5] C. Chen, Y. Zhang, Z. Wang, S. Wan, and Q. Pei, "Distributed computation offloading method based on deep reinforcement learning in ICV," *Applied Soft Computing*, vol. 103, Article ID 107108, 2021.

[6] X.-X. Zhang, H.-T. Zhao, Z. Hongbo, and A. Bamidele, "NAS-AMR: neural architecture search based automatic modulation recognition method for integrating sensing and communication system," *IEEE Transactions on Cognitive Communications and Networking, early access*, vol. 8, no. 3, 2022.

[7] Y. Peng, P. Liu, Y. Wang, G. Gui, B. Adebisi, and H. Gacanin, "Radio frequency fingerprint identification based on slice integration cooperation and heat constellation trace figure," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 543–547, Mar. 2022.

[8] IEEE, "Proposed TGax draft specification," IEEE, IEEE P802.11ax/D2.0, Technical Report, 2017.

[9] G. Naik, S. Bhattarai, and J. Park, "Performance analysis of uplink multi-user OFDMA in IEEE 802.11ax," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kansas City, MO, USA, May 2018.

[10] L. Sanabria-Russo and B. Bellalta, "Traffic Differentiation in Dense Collision-free WLANs Using CSMA/ECA," *Ad-Hoc Networks*, vol. 75–76, pp. 333–351, 2018.

[11] B. Bellalta and K. Kosek-Szott, "AP-initiated multi-user transmissions in IEEE 802.11ax WLANs," *Ad Hoc Networks*, vol. 85, pp. 145–159, March 2019.

[12] Z. Shen, B. Li, M. Yang, Z. Yan, X. Li, and Y. Jin, "Research and performance evaluation of spatial reuse technology for next generation WLAN," in *Proceedings of the International Wireless Internet Conference*, pp. 41–51, Springer, Cham, January 2019.

[13] IEEE 802.11TM Wireless Local Area Netwokrs, https://www.ieee802.org/11/.

[14] N. Korolev, I. Levitsky, and E. Khorov, "Analyses of NSTR multi-link operation in the presence of legacy devices in an IEEE 802.11 be network," in *IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 94–98, Thessaloniki, Greece, December 2021.

[15] Report and Order and Further Notice of Proposed Rulemaking, https://docs.fcc.gov/public/attachments/FCC-20-51A1_Rcd.pdf.

[16] J. Sandoval and S. Cespedes, "Performance evaluation of IEEE 802.11ax for residential networks," in *Proceedings of the 2021 IEEE Latin-American Conference on Communications (LATINCOM)*, pp. 1–7, Santo Domingo, November 2021.

[17] Y. Seok, J. Yee, J. Liu, and T. Pare, "Synchronous multi-link operation," *Doc IEEE*, vol. 11-19/130Sr4, April 2020.

[18] L. Chu, Y. H. Kwon, M. Kumar, H. Zhang, Y. Zhang, and R. Cao, "Multiple link operation follow up," *Anales de Documentación: IEEE*, vol. 11-20/04B7r5, April 2020.

[19] D. Akhmetov and L. Cariou, "Performance aspects of multi-link operations," *Anales de Documentación: IEEE*, vol. 11-19/1291r3, September 2019.

[20] Y. Li, Y. Guo, G. Huang, Y. Zhou, M. Gan, and D. Liang, "Channel access in multi-band operation," *Doc IEEE*, vol. 11-19/1116r5, January 2020.

[21] E. Khorov, I. Levitsky, and I. F. Akyildiz, "Current status and directions of IEEE 802.11be, the future wi-fi 7," *Current status and directions of ieee*, vol. 8, pp. 88664–88688, 2020.

[22] S. Chauhan, A. Sharma, S. Pandey, K. N. Rao, and P. Kumar, "11be: a review on Wi-Fi 7 use cases," in *Proceedings of the International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)*, pp. 1–7, IEEE, Noida, India, September 2021.

[23] G. Naik, D. Ogbe, and J.-M. J. Park, Q. C. Montreal, Can Wi-Fi 7 support real-time applications? On the impact of multi link aggregation on latency," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 1–6, Canada, June 2021.

[24] J. Gomez-Ponce, D. Burghal, N. A. Abbasi et al., "Directional delay spread and interference quotient analysis in sub-7GHz Wi-Fi bands," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Taipei, Taiwan, December 2020.

[25] I. Levitsky, Y. Okatev, and E. Khorov, "Study on simultaneous transmission and reception on multiple links in IEEE 802.11be networks," in *Proceedings of the International Conference Engineering and Telecommunication (En&T)*, pp. 1–4, Dolgoprudny, Russia, November 2020.

[26] STR AP Sync. PPDU Transmission, https://mentor.ieee.org/802.11/dcn/20/11-20-0638-00-00be-str-ap-sync-mlo-operation.pptx.

[27] MU-RTS/CTS PHY Format, https://mentor.ieee.org/802.11/dcn/16/11-16-0648-00-00ax-mu-rts-cts-phy-format.pptx.

[28] P. Angueira, I. Val, J. Montalban et al., "A survey of physical layer techniques for secure wireless communications in industry," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 810–838, Secondquarter 2022.

[29] A. S. Gvozdarev, A. Alischyuk, and M. Kazakova, "Impact of system correlation matrix bordering on the MU-MIMO ergodic sum-rate capacity in the presence of the multipath fading channel," in *Proceedings of the 4th International Conference On Advanced Communication Technologies And Networking (CommNet)*, pp. 1–6, Rabat, Morocco, December 2021.

[30] https://mentor.ieee.org/802.11/dcn/20/11-20-0479-00-00be-240-mhz-channelization.pptx.

[31] D. Nunez, F. Wilhelmi, S. Avallone, M. Smith, and B. Bellalta, "TXOP sharing with coordinated spatial reuse in multi-AP cooperative IEEE 802.11 be WLANs," in *Proceedings of the IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 864–870, Las Vegas, NV, USA, January 2022.

[32] Network Simulator, https://www.nsnam.org/.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 1781757, 19 pages
https://doi.org/10.1155/2022/1781757

Research Article

A Novel Approach for Estimating Performance of IIoT-Based Virtual Control Train Sets under DoS Attacks

Shuomei Ma,[1] **Hongwei Wang**,[2] **Zhu Li**,[1] **and Qihe Zhang**[1]

[1]*State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China*
[2]*National Research Center of Railway Safety Assessment, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Shuomei Ma; 18111055@bjtu.edu.cn

Received 14 July 2022; Revised 2 August 2022; Accepted 22 August 2022; Published 30 September 2022

Academic Editor: Chen Chen

The virtually coupled train sets (VCTS) have been proposed to improve operational capability and passenger satisfaction and ensure punctuality, thus alleviating the rapidly worsening traffic pressure. Recently, due to the lack of reliable wireless communications and accurate perceptual information, VCTS based on industrial internet of things (IIoT) are receiving growing concerns by integrating into the IIoT, AI, and edge computing. However, denial of service (DoS) attacks are feasible for IIoT-based VCTS due to the physically exposed open electromagnetic environment. They would cause severe safety and punctuality problems, such as poor real-time capability, enormous packet dropout rates, extensive train operational delays, and disturbance in the train convoy's dynamic schedule. This paper deeply discusses the effects of DoS attacks on the performances of the IIoT-based VCTS by combing the physical layer with the cyber layer and explores the requirement of an attacker. We consider that the system is under the attack of a rational DoS attacker with limited jamming attacks, which will cause the most system state offset. In the paper, we propose a novel train status estimation approach to compensate for the losing information of the front train by the trade-off between the best gain of the DoS attacker and the punctuality of the IIoT-based VCTS. System performance includes physical dynamic indicators, train operational delay variance, and average waiting time of passengers. Taken together, these findings indicate that the established status estimation approach can effectively mitigate safety concerns and reduce train operational delays.

1. Introduction

The concept of the virtual control train set (VCTS) has received increasing attention as an indicator of the future railway signaling system within the past years. VCTS employs bidirectional wireless data communication to ensure the safety operation of the rail transport and couples two neighbor trains with relative braking distance to increase the transportation capacity and the flexibility of railway organization, rather than the current traditional communication system AND communication-based train control (CBTC) system [1–3]. The physical coupler is cancelled between adjacent trains in the VCTS for meeting the distribution of passengers by making the density of trains. Train members of VCTS have stringent requirements for reliability of wireless communications and the timeliness for supporting the autonomy of a train convoy, which is the same as the unmanned aerial vehicles (UAVs) and classified as a typical industrial Internet of things (IIoT) [4]. With the development of IIoT, an IIoT-based VCTS is proposed in the article based on the popular communication-based train control architecture [2].

In the knowledge of this new IIoT-based VCTS, all trains generate a long virtual train, known as a train convoy or train platoon [2]. Flammini et al. [5] introduced VCTS and presented requirements, which had a tight coupling relationship with the stability of the train convoy, for safety. Quaglietta et al. [6, 7] introduced the need for additional safety constraints, especially at diverging junctions, and presented the train operation's model under ETCS level 3 and virtual coupling. Members of VCTS can be recoupled and decoupled automatically according to transportation

demands and plans based on train-to-train (T2T) wireless communications [5]. However, this system is prone to several security threats, especially crucial safety requirements of the train convoy, owing to the fact that it works under an open electromagnetic environment and the common unlicensed band [8, 9]. The allowable minimum relative braking distance between adjacent trains in VCTS would be violated. If DoS attacks were not mitigated, this violation could bring packet loss and time delay and disturb the train convoy operation.

Current studies of VCTS mainly focus on principles and control strategies for improving the efficiency of a train convoy operation. Di Meo et al. [10] assume to enrich ERTMS with virtual coupling instead of defining a fully new signaling system, which is the preferred approach to ensure backward compatibility and minimize the impact on existing infrastructures since it guarantees the reuse of standard operating modes and related safety mechanisms. Felez et al. [11] proposed an MPC approach to reduce the impact of time delays on the operation performance of the train convoy, thus ensuring its safety and stability. Reference [12] has proposed the concept of VCTS and requirements of safety especially for the stability of a train convoy's achievement with a tight coupling relationship of a series of trains. In addition, a smarter and efficient railway system could be achieved by integrating with IIoT, AI, 5G, big data analysis, and edge computing [2, 13]. However, with the continuous advance of the transportation intelligent construction tide, the security of wireless communications is starting to become powerless, especially in terms of large information flows in IIoT-based VCTS [14]. Therefore, DoS attacks are feasible for IIoT-based VCTS due to the unreliability of T2T communications [15], which are ignored in the current study.

The IIoT-based VCTS can be considered a cyber-physical system (CPS) [14]. Its physical layer, which represents the train control system, ensures the safety and the efficiency of the train convoy operation, whereas the cyber layer represents the wireless communication system. Functionally, wireless communication predisposes the system to cyberattackers, who interfere with the train convoy operation schedule, while interference with the transmission of controller command poses security risks during train convoy operations. In the IIoT-based VCTS, a key feature of train convoy operation safety is that the following train can track the trajectory of the one in front while maintaining a known headway distance. Notably, trains are required to abruptly uncouple all members of a train convoy and apply emergency brakes, if this communication is tampered with, a phenomenon that has been associated with disruption of train scheduling and leaving passengers stranded. Therefore, the development of an efficient method for preventing the need for the application of emergency brakes and ensuring the ideal headway distance within unreliable wireless communications, caused by cyberattacks, is significant for the safety and efficiency of a train convoy in the IIoT-based VCTS [16].

To date, various conventional cryptography technologies and intrusion detection systems (IDS) have been developed with the aim of mitigating the impact of cyberattacks [17, 18]. Consequently, these approaches have played a significant role in the defense strategies of the conventional international system. In addition, previous studies have explored the potential for the data network [19], deep reinforcement learning [20], and the blockchain [21] in mitigation of the impact of DoS attacks. Reference [22] has proposed an online intrusion detection cloud system to detect and filter malicious attack with the new spiking neural network architecture called the NeuCube algorithm. Reference [23] has introduced context-aware security (ConSec) protocol to support internet of things applications to reduce the latency while encrypting and decrypting the applications. However, the above literature on cloud computing technology has perpetuated the huge computing flows and data circulation through the Internet; they are insufficient to meet the security challenge of VCTS system, due to the combination of the cyber layer performance with the physical dynamic.

Currently, many studies have applied analyses of the effects of cyberattacks on the network control system (NCS), a type of CPS, to explore the performance of the cyber layer in combination with the physical dynamics [24]. Reference [25] explores a min-max cost-optimal problem to guarantee the convergence rate of federated learning in terms of cost in wireless edge networks. A status estimation approach was proposed in the cyber-physical system (CPS) to ensure the stability of the vehicle platoon under unreliable wireless communication. Reference [26] proposed a linear deception attack strategy and presented the corresponding feasibility constraint on the optimal attack strategy among all linear attacks, while [27] explored the potential for remote status estimation of CPS based on the game-theoretic approach under DoS attacks.

Moreover, in contrast to the Internet and the CPS, the challenge experienced by the defense system in the IIoT-based VCTS comprises a combination of packet losses (i.e., cyber layer), train dynamic operation, and stranded passengers (i.e., physical layer). Recently, some security field studies have developed defense methods from an attacker's point of view [28], which are based on the fact that combining an attacker's strategy and defense method effectively simulates the actual subway environment and explores the performance of DoS attacks under energy limits, which are found that attacks were random and irregular, albeit with a limited sum of attack energy is limited. These methods are shedding new light on the challenge of the defense system in the IIoT-based VCTS. Results from analyses of the energy constraint indicated that an optimal attack strategy causes the most significant effect on wireless communication and packet losses, thereby causing a train to make frequent emergency braking. The study thus adopts the status estimation approach based on the optimal attack strategy to improve estimator accuracy.

In the IIoT-based VCTS, the development of an efficient method for preventing the need for application of emergency brakes is urgently needed to ease traffic jam on the railway. DoS attacks have been shown to be possible attacks that can negatively affect the physical performance of the

IIoT-based VCTS since they target wireless communications [29, 30]. In fact, an intelligent DoS attacker can reduce the signal-to-interference-plus-noise ratio (SINR) of the wireless communication channels, a phenomenon that results in a low packet arrived ratio, and ultimately cause serious train accidents [31]. DoS attacks on the IIoT-based VCTS not only significantly interfered with the wireless communication between the AP and the train but also resulted in frequent packet losses and ultimately the safety and congestion of the transportation owing to the uncoupling of the train convoy and emergency braking by trains.

In the present study, we propose a train status estimation approach for developing a defense against the IIoT-based VCTS during DoS attacks that adopts an optimal attack strategy with an evaluation, which is based on the physical layer performance (physical dynamic) and the cyber layer performance (signal-to-interference-plus-noise ratio (SINR)), is adopted. The train status estimation approach is used to compensate for the status information (i.e., position, velocity, and acceleration) of the front train under DoS attacks. Notably, this study makes the following main contributions:

(i) We propose a train status estimation approach combing the enhanced Kalman filtering with the optimization, the gain of the attacker, and the solution of a Markov stochastic process to mitigate the frequency of emergency braking in the decoupling mode of the IIoT-based VCTS and compensate for the gap of packet losses caused by DoS attacks. The analysis procedure of the enhanced Kalman filtering method provides new ideas for solving the estimation error covariance matrix during unreliable wireless communication while can be generalized to other CPS.

(ii) We consider the IIoT-based VCTS is under the attack of a rational DoS attacker with limited jamming attacks, which will cause the most system state offset. When these attacks happen, the mode of train convoy would decouple by the "fail-safe" rule, but it cannot avoid the eventual traffic jam of the urban transit and enormous packet dropout in the process of T2T communication. To simulate the actual attack environment and improve the accuracy of the train status estimation approach, we consider a trade-off between the best gain of the attacker and the punctuality of the train convey set, which has been decoupled for safety.

(iii) Criterion indicators for evaluating the performance of the status estimation approach are defined. The evaluation principle combines performances across wireless communication, physical dynamics (train speed/distance profile), and passenger satisfaction (train operational delay and passenger waiting time).

The rest of the paper is organized as follows. The framework of the IIoT-based VCTS and the impacts of DoS attacks on the IIoT-based VCTS are proposed in Section 2. Section 3 involves the system model and problem formulation, and Section 4 describes the train status estimation approach based on the optimal attack strategy In Section 5, the evaluation criterion of the effects of DoS attacks on the IIoT-based VCTS system is presented. Section 6 demonstrates the simulation results and discussions. Finally, we conclude this study in Section 7.

2. Framework of the IIoT-Based VCTS and Impacts of DoS Attacks on the IIoT-Based VCTS

In this section, VCTS is first outlined. Based on the principles of VCTS, a novel structure of the IIoT-based VCTS-based train-centric is proposed, and the effects of jamming attacks on T2T communication are analyzed.

2.1. Overview of Virtual Coupling. Virtual coupling will be a significant feature of the future railway system [6, 7] that can improve the capacity and efficiency of transportation to deal with the forecasted growth of traveling demands. Figure 1 shows the contrast between the traditional moving block (MB) and the virtual coupling. In MB mode, the zone controller (ZC) can monitor the running status of the train and generate train control commands called movement authority (MA) of trains. Generally, MA is defined as the location of the nearest obstacle, which is related to the braking headway distance of trains, including trains, turnouts, and signals.

Additionally, virtual coupling increases the density of trains, which means that the interval between adjacent trains of a formation is much smaller. When trains are coupled via T2T communications, the train movement depends on the status of adjacent trains, including their acceleration, velocity, and position, through onboard sensors and wireless communication modules. In this study, the first train in the train convoy is called the leading train. The control strategy of each following train is also optimized with the approach so that its velocity and acceleration are the same as the last known information of the leading train. When trains are within virtual coupling, IIoT-based VCTS aims to provide a controlling strategy to ensure that each position difference between the ahead train and its following train is close to the objective relative braking distance [32]. In addition, IIoT-based VCTS prefers a train-centric control system, which is different from the traditional MB. One of the challenges for the IIoT-based VCTS is how to meet the high mobility and efficiency of virtually coupled via T2T wireless communications by facility designing and ensure the safety and joint security of VCTS operations. The next subsection presents a new framework of the IIoT-based VCTS.

2.2. A Novel Framework of the IIoT-Based VCTS. This section shows a novel framework of the IIoT-based VCTS based on T2T communications, as illustrated in Figure 2. The proposed structure consists of a control center subsystem, an onboard subsystem, and a trackside subsystem. Onboard

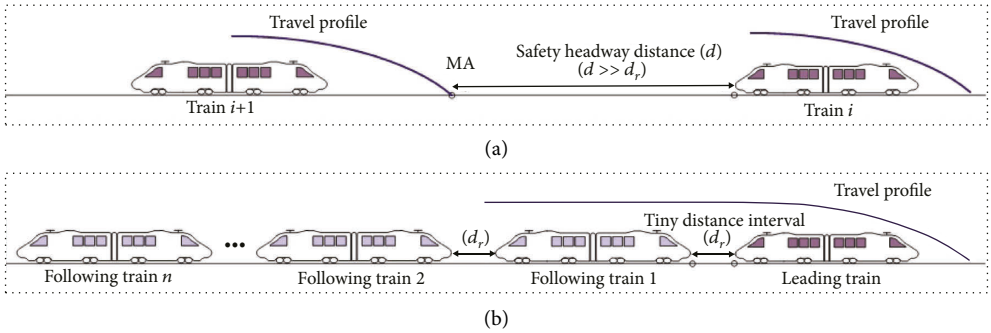

Figure 1: Moving block versus virtual coupling: (a) moving block and (b) virtual coupling.

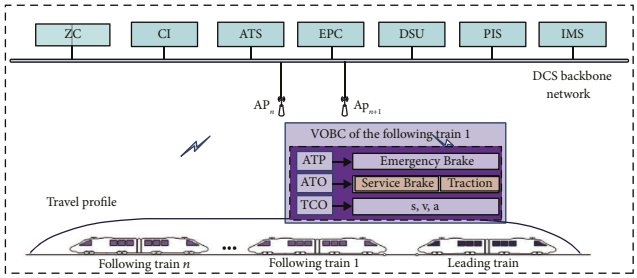

Figure 2: Structure of the IIoT-based VCTS.

subsystems include automatic train protection (ATP), automatic train operation (ATO), and train cooperative controller (TCO). Through the co-work of computer interlock (CI), data storage unit (DSU), and evolved packet core (EPC), zone controller (ZC), operation plans, and the running state of subway lines can be transmitted to trains. Moreover, a cooperative controller is designed to operate the train in the virtual coupling mode. Through T2T wireless communications, TCO can provide optimal control strategies for trains according to velocities and locations of adjacent trains. Additionally, ATP can be secured based on the overall running state of the whole line, which signifies that T2T communications are essential to provide low-latency and high-capacity information exchange among members of a train convoy in the IIoT-based VCTS. When T2T communication links fail or the velocities of trains exceed the limiting speed of ATP, an emergency train braking is executed. In this study, the access point (AP) and customer premises equipment (CPE) of the IIoT-based VCTS via T2T communication links of adjacent trains are established by long-term evolution for metros (LTE-M). The CPE exchanges the status information of train i with RRU and other trains via the T2T communications link [33]. Clearly, T2T communications play an essential role in the IIoT-based VCTS. Therefore, the mechanism is necessary to avoid collisions in T2T communications under unpredictable disturbances.

2.3. Impacts of DoS Attacks on the IIoT-Based VCTS. Generally, attackers can send enormous jamming traffics or fake bits to exhaust the frequency bandwidth, channel capacity, and legitimate communication services. Figure 3 illustrates the comparison of the IIoT-based VCTS and the effects of jamming attacks on the IIoT-based VCTS. In a train convoy, the leading train communicates with the control center via the train-ground (T2G) communications. When the interruption time caused by jamming attacks on T2G is significantly larger than the preset value, the leading train must implement emergency braking. When jamming occurs in T2T communications, the stability of members in a train convoy will be disturbed. Due to the high speed and the tiny interval, jamming may cause safety risks or running as decoupled trains belonging to MB mode.

The security of IIoT-based VCTS, as a new technology in urban rail transit, is a severe challenge because it is more vulnerable to jamming attacks than before [33]. This subsection aims to analyze the constraints of jamming attacks. Considering the distance from the attacker to the victim node, jamming attacks can be classified as constant jamming, deceptive jamming, and reactive jamming [34]. In this study, the effects of constant jamming on T2T for the IIoT-based VCTS are mitigated by the resilience control approach with the ETC condition. In the next section, we will analyze the dynamic model of the train convoy and the indicators of the safety operations of VCTS.

3. System Model and Problem Formulation

In this section, we propose a dynamic control module when trains are coupled. The physical dynamic model for the IIoT-based VCTS also is presented to improve the train operation safety if the train convoy is decoupled caused by DoS attacks. In addition, we also design a cost function to provide a theoretical basis for the train status estimation approach.

3.1. Dynamic Model of the Train Convoy. For the IIoT-based VCTS, stability means that distance intervals between adjacent trains are almost the same while suggesting that all trains are running at the same speed. An objective relative safety distance exists between adjacent trains for optimal performance. The status formulae of the leading train and other trains can be given by

$$\dot{x}_l(t) = A_c x_l(t) + C_c W(t), \tag{1a}$$

$$\dot{x}_i(t) = A_c x_i(t) + B_c u_i(t), \tag{1b}$$

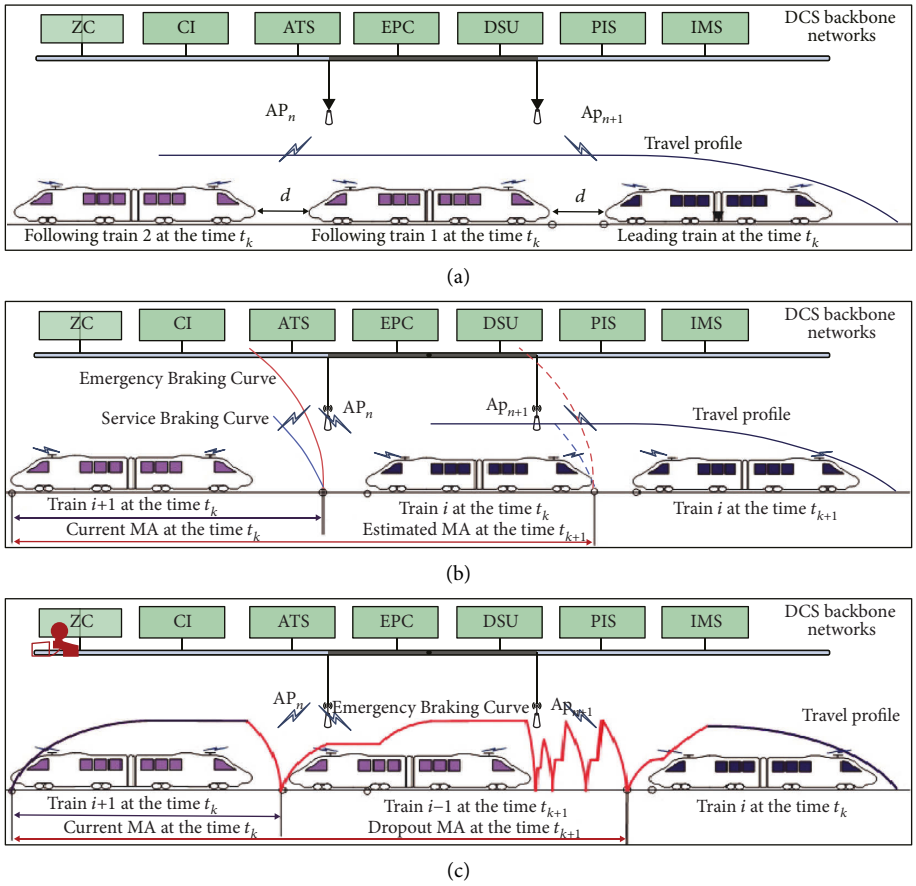

Figure 3: The impacts of jamming attacks on the IIoT-based VCTS: (a) virtual coupling mode in the IIoT-based VCTS, (b) decoupling trains in the IIoT-based VCTS, and (c) impacts of DoS attacks on the IIoT-based VCTS.

where $x_i(t) = [s_i(t), v_i(t), a_i(t)]^T$ is the train running status information; $W(t)$ denotes the matrix of the resistance force, which subtracts the sum of traction and braking force at times t; $u_i(t)$ is the control law based on ETC that will be proposed in Section 3; A_c is the status matrix; B_c is the control matrix; and C_c is the noise and disturbance matrix, which can be calculated by kinematic equations.

A dynamical model of IIoT-based VCTS is established by applying the cooperative adaptive cruise control (CACC), which can avoid collisions according to the sacrificial part of system performance to ensure the safety and stability of the VCTS system when jamming attacks happen. The details can be founded on our previous work [3]. However, concerning the unreliable wireless communications caused by DoS attacks in the IIoT-based VCTS, the stability condition of the train convoy would be violated. As a result, the train convoy is decoupled after the control command. The physical dynamic model of the IIoT-based VCTS is presented in the next subsection, where the train convoy is decoupled.

3.2. Physical Dynamic Model of the IIoT-Based VCTS on the Impacts of DoS Attacks. Due to the consensus about the "fail-safe" rule, which means that all techniques in the signaling control system of railways are needed for following the safety and avoiding collision between adjacent trains at the expense of efficiency and punctuality of the transportation, the passengers' waiting time and traffic paralysis are even. Concerning the DoS attacks, the train convoy would be decoupled [6]. The physical dynamic model of the IIoT-based VCTS is presented in the subsection if the train convoy is decoupled caused by DoS attacks.

In the physical dynamic model of the IIoT-based VCTS, the control objective such as status information plays a significant role in ensuring safety for the operation of trains if the train convoy is decoupled caused by DoS attacks. In the study, the train status information, including the location, speed, and acceleration of the train is defined as follows:

$$x(t) = \left[S(t) \;\; V(t) \;\; \widehat{A}(t) \right]^T, \tag{2a}$$

$$S(t) = \left[s_1(t) \;\; s_2(t) \;\; \ldots \;\; s_n(t) \right]^T, \tag{2b}$$

$$V(t) = \left[v_1(t) \;\; v_2(t) \;\; \ldots \;\; v_n(t) \right]^T, \tag{2c}$$

$$\widehat{A}(t) = \left[a_1(t) \;\; a_2(t) \;\; \ldots \;\; a_n(t) \right]^T, \tag{2d}$$

where $x(t)$ represents the status information matrix of trains at time t; $S(t)$ represents the position matrix of trains at time t; $V(t)$ represents the speed matrix of trains at time t; $\widehat{A}(t)$

denotes the acceleration matrix of trains at time t; $s_i(t)$ and $v_i(t)$ represent the position and speed of train i at time t, respectively; $a_i(t)$ represents the acceleration of time at time t; and n indicates the number of trains.

Moreover, according to kinematic equations, we can write the status information of train i as follows:

$$s_i(t_k) = s_{i-1}(t_{k-1}) - L_{l,i} - L_s + v_i(t_{k-1}), \tag{3a}$$

$$+\left[a_i(t_{k-1}) + \frac{\kappa_i(t_{k-1}) - w_i(t_{k-1})}{M}\right]h, \tag{3b}$$

$$v_i(t_k) = v_i(t_{k-1}) + \left[a_i(t_{k-1}) + \frac{\kappa_i(t_{k-1}) - w_i(t_{k-1})}{M}\right]h, \tag{3c}$$

$$a_i(t_k) = a_i(t_{k-1}) + \left[\frac{\kappa_i(t_{k-1}) - w_i(t_{k-1})}{M}\right]h, \tag{3d}$$

where $L_{l,i}$ indicates the length of train i, h is the T2T communication cycle of the IIoT-based VCTS, L_s denotes the minimum safe distance between adjacent trains, κ_i indicates the resistance force of train i train i at time t_{k-1}, $w_i(t_{k-1})$ indicates the sum of traction and braking force of train i at time t_{k-1}, t_k represents the beginning of the kth communication cycle, and M represents the mass of the train.

Due to the decoupling of the train convoy, the operation of train i is described using a linear system, for each member of the train convoy, as follows:

$$x(t_k) = Ax(t_{k-1}) + Bu(t_{k-1}) + w_e(t_{k-1}), \tag{4}$$

where $x(t_k) = [x_1(t_k), x_2(t_k), \ldots, x_n(t_k)]^T$, $x_i(t_k) = [s_i(t_k), v_i(t_k), a_i(t_k)]^T$ indicates the status information of the train i at time t_k, $u(t_k) = [u_1(t_k), u_2(t_k), \ldots, u_n(t_k)]^T$, $u_i(t_k)$ indicates the input of the controller, and A and B denote the known matrixes with compatible dimensions. A and B can be designed as follows, and the pair (A, B) is stabilized:

$$A = blk\ di\ ag[A_1, A_2, \ldots, A_n],$$

$$B = blk\ di\ ag[B_1, B_2, \ldots, B_n],$$

$$A_1(t) = A_2(t) = \cdots = A_n(t) = \begin{bmatrix} 1 & h & \frac{1}{2}h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

$$B_1(t) = B_2(t) = \cdots = B_n(t) = \begin{bmatrix} \frac{1}{2}h^2 \\ h \\ 1 \end{bmatrix}.$$

The observation equation is expressed as follows:

$$y(t_k) = Cx(t_k) + v_e(t_{k-1}), \tag{6}$$

where

$$\begin{aligned} y(t_k) &= [y_1(t_k), \ldots, y_n(t_k)]^T, \\ w_e(t_k) &= [w_{e,1}(t_k), \ldots, w_{e,n}(t_k)]^T, \\ v_e(t_k) &= [v_{e,1}(t_k), \ldots, v_{e,n}(t_k)]^T, \end{aligned} \tag{7}$$

where $w_{e,i}(t_k) \sim (0, R)$ and $v_{e,i}(t_k) \sim (0, N)$ represent the process noise and the measurement noise, respectively. Both parameters are independent Gaussian distributions with zero mean and error covariance. In addition, Q indicates the process noise covariance matrix, whereas R and $C = [1, 0, 0]$ denote the measurement noise covariance matrix and the observation matrix, respectively.

When the train convoy is decoupled, the mathematical expression of the control strategy for train i, under DoS attacks at time t_k, is shown as follows:

$$x_i(t_k) = \begin{cases} x_i(t_k), & \vartheta(t_k) = 1, \\ \hat{x}_i(t_k), & \vartheta(t_k) = 0, \end{cases} \tag{8}$$

where $\hat{x}_i(t_k)$ denotes the estimation status information of the front train at time t_k and $\vartheta(t_k) = 1$ implies that train i's status information is transmitted successfully under DoS attacks, whereas $\vartheta(t_k) = 0$ indicates transmission failure.

For the above analyses, the MA can be expressed as follows:

$$lm(t_k) = Dy(t_k) + L, \tag{9}$$

where $lm(t_k) = [lm_1(t_k), lm_2(t_k), \ldots, lm_n(t_k)]^T$, $L = [L_1, L_2, \ldots, L_n]^T$, $L_i = L_{l,i} + L_s$ indicates the sum of the length of train i and the safety margin, and $D = \begin{bmatrix} 0 & 0 \\ I_{n\times n} & 0 \end{bmatrix}$.

The physical dynamic model of the IIoT-based VCTS is prone to mitigate traffic paralysis and is adjusted by sacrificing part of the system performance to ensure the safety and punctuality of the IIoT-based system when DoS attacks happen. In addition, we assumed that the IDS of the IIoT-based system has high precision and detection methods for DoS attacks. Although previous studies have employed several detection methods, such as [35], none of these is discussed in the current study.

3.3. Problem Formulation. DoS attackers intend to intercept and prevent legitimate T2T communication services for legitimate APs. Generally, they achieve this by sending enormous wrong information traffic and by exhausting the wireless network bandwidth or abrogating the connection capacity [36]. DoS attacks on the wireless communication system affect the T2T and T2G communications because members of the train convoy can no longer receive accurate status information during each communication cycle. Communication delays of the IIoT-based VCTS after decoupling are randomly caused by DoS attacks. Therefore, a novel train status information estimation approach is constructed to improve the performance of the IIoT-based

VCTS system during DoS attacks. The accuracy of the estimation approach depends on the minimum estimate error, which forms the basis of the cost function reported in the current study [12]. This is expected to be circumventing the challenges associated with unreliable T2T and T2G communications. In order to improve the accuracy of the train status estimation, the IIoT-based VCTS performance requirement needs to be combined with the attack strategy from the attacker's perspective. Due to random communication delays, we define the cost function with the aim of improving the accuracy of the train status estimation approach, with a focus on the optimal attack strategy from an attacker's standpoint. The study hypothesized that DoS attacks follow a Bernoulli distribution, whereas the energy of a one-time attack follows a Poisson distribution as described by [28, 37].

Next, we analyzed the energy limits of attacks to ascertain the realistic unreliable wireless communication channel, owing to the fact that the rational attacker always looks for a strategy that can significantly compromise the wireless communication system in an IIoT-based VCTS system and is likely to employ an approach that consumes the lowest energy consumption. Summarily, DoS attacks interfere with the wireless communication channel between ZC and VOBC of its control area, thereby causing the retransmitting of the status information before the safety margin of the limited time. These situations indicate that conventional approaches cannot efficiently manage DoS attacks on the IIoT-based VCTS system, owing to the system's strict communication latency. Therefore, there is a need to improve the train status estimation approach, from the view of the energy limits of the attacker, to ensure the effective overcoming of the insufficient status information during DoS attacks. Detailed instructions are described as follows. Firstly, the error estimation covariance matrix is expressed as follows:

$$\chi_i^-(t_k) \triangleq \mathbb{E}\left[\delta_i(t_k)\delta_i^T(t_k)\right], \tag{10}$$

where $\chi_i^-(t_k)$ and $\mathbb{E}[\bullet]$ represent the error estimation covariance matrix of train i at time t_k and an expectation function, respectively, while $\delta_i(t_k)$ represents the estimate error as follows:

$$\delta_i(t_k) \triangleq x_i(t_k) - \hat{x}_i(t_k), \tag{11}$$

where $x_i(t_k)$ indicates the status information of train i at time t_k, while $\hat{x}_i(t_k)$ represents the estimation value of the estimator at time t_k.

Next, we propose a cost function for minimizing the estimation error covariance with the energy constraint of one attack, as follows:

$$\min \sup_{\Delta T \longrightarrow \infty} \frac{1}{\Delta T}\left[\sum_{t_k=T_1}^{T_2} \chi_i^-(t_k)\right], \tag{12a}$$

$$s.t. \sum_{t_k=T_1}^{T_2} \rho(t_k) \in [0, \rho_{\max}], \tag{12b}$$

where $\rho(t_k)$ represents the power of interference of DoS attacks at time t_k, $\rho_{\max}$ indicates the maximum attack energy, and T_1 and T_2 denote the start and end times of DoS attacks, respectively, whereas $\Delta T = T_2 - T_1$ denotes the duration of DoS attacks.

Thereafter, we analyze the convenience using the cost function shown as follows:

$$\min \sup_{\Delta T \longrightarrow \infty} \frac{1}{\Delta T}\left(\mathcal{J}_e - \lambda_e \mathrm{AE}_T\right), \tag{13a}$$

$$\begin{aligned} \mathcal{J}_e &= E\left[\sum_{t_k=T_1}^{T_2} \delta_i t(t_k)\delta_i^T(t_k)\right] \\ &= \sum_{t_k=T_1}^{T_2} \chi_i^-(t_k), \end{aligned} \tag{13b}$$

$$\mathrm{AE}_T = E\left[\sum_{t_k=T_1}^{T_2} \rho(t_k)\rho(t_k)^T\right], \tag{13c}$$

where $\mathcal{J}_e$ indicates the sum of error estimation covariance for DoS attacks and AE_T denotes the sum of attack energy.

It is significant that the underground environment and tunnels under a subway environment cannot provide the charging point for an attacker. One key feature of DoS attacks, which is the limited energy caused by no charging point, is that they occur randomly. DoS attacks targeting the IIoT-based VCTS system significantly interfere with the wireless communication between the ZC and the train, thereby causing frequent packet losses and congestion of the transportation owing to frequent emergency braking of trains. Notably, it is challenging for the attacker to affect the transportation of IIoT-based VCTS by significantly jamming under energy limit situations. In the current study, we explored the defense strategy from the attacker's standpoint. Therefore, the cost function described herein is based on the optimal attack strategy that the attacker is most likely to choose. In the subsections, we review studies describing the random energy distribution of DoS attacks in the IIoT-based VCTS system as well as the approaches applied to solve the cost function while ensuring the optimal energy strategy and the minimum estimation error covariance, as described in the following sections.

4. The Train Status Estimation Approach Based on the Optimal Attack Strategy

In this section, we propose the train status estimation approach, which refers to an enhanced Kalman filtering scheme based on the optimal attack strategy. The estimation approach seeks to transfer indispensable status information from the front train to the following train.

4.1. Modeling the Train Status Estimation Approach of the IIoT-Based VCTS. When packet loss occurs in the physical dynamics of the IIoT-based VCTS system caused by DoS attacks, based on the "fail-safe" requirement of the

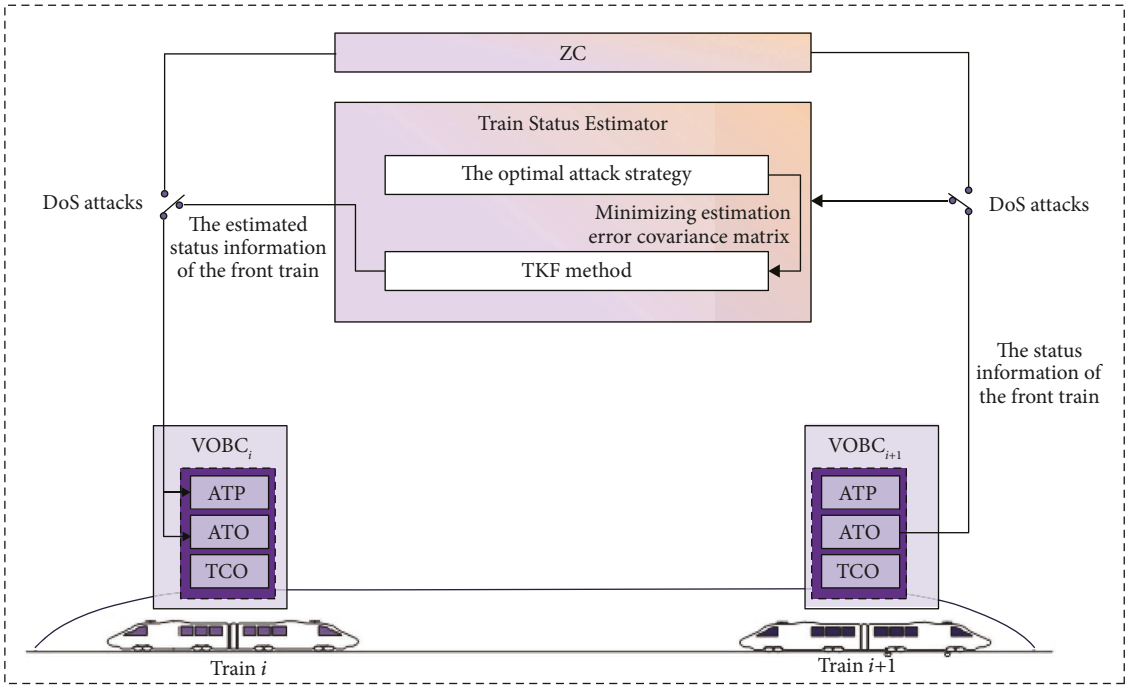

Figure 4: Structure of the train status estimation approach.

signaling control system, the train convoy would be decoupled. As a result, the virtual coupling mode is instead of the MB mode. Because ZC experiences the transmission failure of that limit MA, which is the maximum safety margin of the following train, at the start instance of each communication cycle. Then emergency braking would be executed when the speed of the following train is violated the safety margin. Moreover, after emergency braking, following trains have to stay stationary until the wireless communication is resolved; thus, it can receive an updated MA in real time when the train convoy is decoupled. Eventually, this situation may cause traffic paralysis and enormous numbers of passengers stranded. Therefore, the train status estimation approach aims to mitigate this traffic paralysis caused by DoS attacks, by combing with the enhanced Kalman filter method and the attack strategy from the attacker's perspective.

The Kalman filter, which is known as the linear quadratic estimation (LQE) algorithm, is an optimal estimator that has been extensively applied as an industry controller [38]. The feasibility of this status estimation approach is mainly constrained by the implementation of minimum error estimation covariance to approach the performance of the IIoT-based VCTS during randomly instantaneous attacks. In addition, the estimation approach is unreasonable if it meets the requirements of the conventional error covariance at each iteration. In the current section, we propose a novel solution to this problem, considering the aforementioned shortcomings of DoS attacks. To achieve a minimum unbiased estimation covariance of the status information, which members of the train platoon are decoupled, we propose combing conventional Kalman filtering with the optimal attack strategy, which is the limited attacks' energy, to obtain the train status information $x_i(t_k)$ in the unreliable communication network under DoS attacks (as shown in Figure 4). Therefore, the Kalman filter is a discrete-time controlled process, and the linear stochastic equation of the train i is expressed as follows:

$$x_i(t_k) = A_e x_i(t_{k-1}) + B_e u_i(t_{k-1}) + w_e(t_{k-1}). \tag{14}$$

Consequently, the observation equation of enhanced Kalman filtering is expressed as follows:

$$y_i(t_k) = C_e x_i(t_k) + v_e(t_{k-1}), \tag{15}$$

where $A_e = \begin{bmatrix} 1 & h & 1/2h^2 \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix}$, $B_e = \begin{bmatrix} 1/2h^2 \\ h \\ 1 \end{bmatrix}$, and $C_e = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$.

Estimation of the status information for the performance of the IIoT-based VCTS is divided into time-updated (predicting) and measurement-updated (receiving) sections. The prediction equation of enhanced Kalman filtering is thus expressed as follows:

$$\hat{x}_i^-(t_k) = A_e \hat{x}_i(t_{k-1}) + B_e \hat{u}_i(t_{k-1}), \tag{16}$$

where $\hat{x}_i^-(t_k)$ is the estimated status information of train i at time t_k using measurements up to time t_{k-1}.

The status estimation approach, which is combing the Kalman filtering with the optimal attack strategy, abrogates the effects of packet loss during the transfer of the status information of the front train, a phenomenon that alleviates

the train operational delays caused by DoS attacks. Reference [39] indicates that packet dropout in the measurement updating process can be calculated as follows:

$$K_e(t_k) = \chi_i^-(t_{k-1})C_e^T\left[C_e\chi_i^-(t_{k-1})C_e^T + R\right]^{-1}, \tag{17}$$

$$\hat{x}_i(t_k) = \hat{x}_i^-(t_k) + K_e(t_k)[y_i(t_k) - C_e\hat{x}_i^-(t_k)], \tag{18}$$

$$\chi_i(t_k) = [I - K_e(t_k)C_e]\chi_i^-(t_k), \tag{19}$$

where $K_e(t_k)$ denotes the Kalman gain at time t_k and $\chi_i(t_k)$ is the updated error estimation covariance matrix of train i at time t_k, whereas I is the identity matrix.

Furthermore, in conventional Kalman filtering, the error estimation covariance matrix at time t_k is predicted based on both the iteration procedure of the error covariance matrix at time t_{k-1} and the prediction noise covariance matrix, which are obtained using the channel noise, mainly referring to Gaussian distribution. However, when the T2T and T2G wireless communications of the IIoT-based VCTS interfere with DoS attacks, which follow the Bernoulli distribution, the traditional Kalman filtering is no longer effective, while the prediction noise covariance matrix is uncertainly due to random and uncertain DoS attacks. Therefore, we consider the regular character of the DoS attacks and energy limits to optimize the estimation error.

In the status estimation approach of the IIoT-based VCTS, formula (17) is used to accurately estimate and update the status information of the front train, which is based on the minimization error estimation covariance matrix of the front train. The cost function and the optimal attack strategy from the attacker's standpoint are provided to minimize the error estimation covariance matrix $\chi_i^-(t_k)$ of the front train. Therefore, the mathematical status estimate at time t_k is calculated as follows:

$$x_i(t_k) = \begin{cases} x_i(t_k), & \vartheta(t_k) = 1, \\ \hat{x}_i(t_k) = \hat{x}_i^-(t_k) + & \\ K_e(t_k)[y_i(t_k) - C_e\hat{x}_i^-(t_k)] & \vartheta(t_k) = 0, \end{cases} \tag{20}$$

where $\vartheta_{t_k} = 1$ indicates that metro i information has been successfully transmitted at time t_k, whereas $\vartheta_{t_k} = 0$ indicates failed transmission.

The predicting error estimation covariance matrix, which is different from the conventional Kalman filtering, significantly affects the accuracy of the train status estimation approach. In the study, we present the minimization of the error estimation covariance that depends on the optimal attack strategy. This strategy, together with the predicting error estimation covariance, is discussed in the next subsection.

4.2. Reformulation of the Optimization Model for Analysis of Attack Energy Limits. In this subsection, we are prone to probe the factor of impact on DoS attacks. Generally, the main aim of an attacker is to jam the T2T and T2G wireless communication of trains and consume the performance of the IIoT-based VCTS by interfering with the SINR and packet transmission successful rate (or packet dropout rate). These two aspects are key in evaluating the quality performance communication of the IIoT-based VCTS system. The success rate of the packet transmission in the IIoT-based VCTS system is affected by SINR as well as attack energy [27], and it can be expressed as follows:

$$f(t_k) = f(g(t_k), \rho(t_k)), \tag{21}$$

where $g(t_k)$ is the signal attenuation at time t_k.

SINR values can be determined on the basis of periodic sampling in the IIoT-based VCTS system, using a specific communication software on train i during each communication cycle. The SINR value less than the specific threshold indicates the MA packet dropout. The relationship between symbol error rate (SER) and SINR in the IIoT-based VCTS is defined by the digital communication theory [27] as follows:

$$\begin{aligned} \mathrm{SER}(t_k) &= 2S_Q\left(\sqrt{S_\alpha \hat{s}_i(t_k)}\right), \\ \hat{s}_i(t_k) &= \frac{\rho^s(t_k) - g(t_k)}{\varpi + \iota + \rho(t_k)}, \end{aligned} \tag{22}$$

where $\hat{s}_i(t_k)$ is the value of SINR at time t_k, $S_Q = 1/\sqrt{2\pi}\int_x^\infty(-\eta^2/2)\mathrm{d}\eta$, S_α is a constant parameter, $\rho^s(t_k)$ is the transmitting power of APs at time t_k, ϖ is the measurement noise power on the wireless channel, and ι is the interference power on the wireless channel. The probability of the success rate of packet transmission in the IIoT-based VCTS system $f(t_k)$ can be described as follows:

$$f(t_k) = f(g(t_k), \rho(t_k)) \triangleq 1 - 2S_Q\left(\sqrt{S_\alpha \hat{s}_i(t_k)}\right). \tag{23}$$

APs provide enough transmission power to support the stabilities of the T2T and T2G communication subsystems, thus ensuring the performance (i.e., improving punctuality, reducing the waiting time of passengers, and avoiding traffic paralysis) of the IIoT-based VCTS [36]. This process can be quantified as follows:

$$\mathbb{E}[f(g)] > f_s \triangleq 1 - \frac{1}{\lambda_{\max}(A_e)}, \tag{24}$$

where $\lambda_{\max}(A_e)$ represents the maximum eigenvalue of the train status estimation matrix A_e.

Following DoS attacks, MA packet dropout is inevitable following DoS attacks. In the study, the MA packet dropout is used to improve the estimate error. Notably, $\phi(t_k) = (1 - \vartheta(t_k))\delta_i(t_k)$ is defined as the estimate error at $\vartheta(t_k) = 1$, implying that DoS attacks cause MA packet dropout at time t_k. It is aimed at minimizing the estimate error with the MA packet dropout. Therefore, the estimate error is highly corrected with the performance indicated by the train-ground communication system, and the cost function in formula (13a)–(13c) can be written as follows:

$$\min \lim_{\Delta T \longrightarrow \infty} \sup \frac{1}{\Delta T} \mathbb{E}\left[\sum_{t_k=T_1}^{T_2} \phi\left[z\left(t_k\right), g\left(t_k\right), f\left(t_k\right)\right]\right], \quad (25)$$

where $z(t_k)$ represents the probability of the MA packet dropout with DoS attacks at time t_k.

The estimate error of the train status information during transmission failure caused by DoS attacks in the IIoT-based VCTS is expressed as follows:

$$\begin{aligned}\phi\left[z\left(t_k\right), g\left(t_k\right), f\left(t_k\right)\right] \triangleq \left[\left[1-f\left(t_k\right)\right] z\left(t_k\right)^T z\left(t_k\right)\right. \\ -\lambda_e \cdot \rho\left(g\left(t_k\right), f\left(t_k\right)\right),\end{aligned} \quad (26)$$

where λ_e represents a weight constant.

For convenience, we write formula (26) as follows:

$$\phi(z, g, f) = (1-f) z^T z - \lambda_e \cdot \rho(g, f), \quad (27)$$

where z, f, and g indicate abbreviated forms of $z(t_k)$, $f(t_k)$, and $g(t_k)$, respectively.

Moreover, the optimal attack strategy from an attacker's standpoint is to achieve a trade-off between the rate of packet dropout and the cost of attack energy. Therefore, the attack strategy can be given by

$$\begin{aligned}\rho\left(t_k\right) &= \rho\left(g\left(t_k\right), f\left(t_k\right)\right) \\ &= \inf\left\{\rho\left(t_k\right) \mid f\left(t_k\right) \leq \overline{f}, \rho \in \left[0, \rho_{\max}\right]\right\},\end{aligned} \quad (28)$$

where it is assumed that $\rho(g(t_k), f(t_k))$ is continuous with respect to $f(t_k)$ and $g(t_k)$ [40] and $\overline{f}$ indicates the successful minimum packet transmission rate of the system testing specification, while ρ represents the abbreviated form of $\rho(t_k)$.

The status set $\mathcal{O}(z(t_k), g(t_k)) \in \mathcal{O}(z, g)$ is required to quantify the relationship between the attacks power and the packet transmission from the attacker's standpoint. Therefore, the optimal attack strategy is described as follows:

$$\mathcal{O}(z, g) = \begin{cases} f_{\min}\left(t_k\right), & \mathbb{E}\left(z^T z\right) > 1-\overline{f} \text{ or } g \leq \overline{g}, \\ \left[f_{\min}\left(t_k\right), f_{\max}\left(t_k\right)\right], & \text{otherwise}, \end{cases} \quad (29)$$

where $f_{\min}(t_k) = f(g, \rho_{\max})$, while $f_{\max}(t_k) = f(g, 0)$, whereas $\overline{g}$ indicates the maximum signal attenuation of the IIoT-VCTS system testing specification when the train convoy is decoupled.

Packet dropout occurs without DoS attacks when the channel qualities of the T2T and T2G communication systems are lower compared to the minimal performance of the VCTS system. This implies that there is a lack of attacks power in the communication channel.

The cost function is based on the optimal attack strategy; therefore, it is important to evaluate the relationship between energy limits and packet loss. A hypothesize can be formulated based on Theorem 3.5 [41] that a unique function $\Pi(z, g, f)$ exists to satisfy the following equation:

$$\begin{aligned}f^*(z, g, f) = \min_{f\left(t_k\right) \in \mathcal{O}(z, g)} \sup \phi(z, g, f) - \Pi^* \\ + \mathbb{E}\left\{\Pi\left(z^+, g^+\right) \mid z, g, f\right\},\end{aligned} \quad (30)$$

where $f^*(z, g, f)$ represents the optimal packet reception rate from which the effects of attack strategy have been estimated and z^+ indicates the estimated error in the next step, while g^+ indicates the signal attenuation in the next step. Π^* is expressed as follows:

$$\Pi^* = \mathbb{E}\{\Pi(z, g)\}. \quad (31)$$

The cost function can be converted by formula (30). Therefore, a suboptimal attack strategy can be obtained because the optimal cost function has been transformed to generate an effective solution. However, this expression cannot be used to estimate the status information of the front train in this step; therefore, the cost function is transformed. In the next section, the expression of function $\Pi(z, g, f)$ is described, while the minimum error estimation covariance matrix is generated.

The status transition probability can therefore be defined as $\Pr(z^+, g^+ | z, g, f)$ with $\vartheta(t_k) = 1$. $\mathscr{B}(f)$ represents the distribution of DoS attacks, subject to Bernoulli distribution [42]. Then, the attack strategy is considered as a Markov stochastic process as follows [43]:

$$\Pr\left(z^+, g^+ | z, g, f\right) = f \mathcal{N}_{0,W}\left(z^+\right) + (1-f) \mathcal{N}_{Az,W}\left(z^+\right), \quad (32)$$

where $\mathcal{N}_{0,W}$ and $\mathcal{N}_{Az,W}(z^+)$ indicate the transition status, $\mathcal{N}_{0,W}$ indicates the absence of DoS attacks at time t_k, and $\mathcal{N}_{Az,W}(z^+)$ indicates the occurrence of the MA packet dropout caused by the DoS attacks. $\Pi^*(z, g, f)$ can be expressed as follows according to formula (32):

$$\begin{aligned}\mathbb{E}\left\{\Pi\left(z^+, g^+\right) | z, g, f\right\} = f \mathbb{E}_{g^+, w_e}\left[\Pi\left(g^+, w_e\right)\right] \\ + (1-f) \mathbb{E}_{g^+, w_e}\left[\Pi\left(g^+, Az + w_e\right)\right].\end{aligned} \quad (33)$$

The following final cost function is ultimately expressed as follows:

$$f^*(z, g, f) = \min_{f \cdot \left(t_k\right) \in \mathcal{O}(z, g)} \sup\left\{-\lambda_e \rho(g, z) + (1-f) \cdot \Lambda(z, g)\right\}, \quad (34)$$

with

$$\Lambda(z, g) = \mathbb{E}_{g, w_e}\left[\mathbf{\Pi}\left(g^+, Az + w_e\right) - \mathbf{\Pi}(z, g)\right] + z^T z. \quad (35)$$

Therefore, the attack strategy is expressed as follows:

$$\rho^*(z, g, f) = \underset{\rho\left(t_k\right) \in \left[0, \rho_{\max}\right]}{\arg\min} \left\{-\lambda_e \cdot \rho(z, g) + (1-f) \cdot \Lambda(z, g)\right\}. \quad (36)$$

The error estimation covariance matrix depends on the attack strategy as well as the cost function. These parameters are solved in the next subsection.

4.3. Solving the Optimal Attack Strategy with Energy Limits. The minimum error estimation covariance matrix in the suboptimal attack strategy case is described in this section. The suboptimal solution method is expressed as a π function [44]. The DoS attack power strategy is expressed according

to the suboptimal attack strategy from lemmas in references [45, 46].

$$\rho_\pi^*(z,g,f)=\begin{cases}0, & g\le\overline{g}\text{ or }\Lambda(z,g),\\ >\lambda_e\left(\dfrac{\rho^s(t_k)-g(t_k)}{\widehat{s}_i(t_k)}-\varpi-1\right), & \\ \dfrac{\rho^s(t_k)-g(t_k)}{\widehat{s}_i(t_k)}-\varpi-1 & \text{otherwise.}\end{cases}\tag{37}$$

Due to the impact of the DoS attacks on the IIoT-based VCTS, the attacker seeks to interrupt the rapidly growing passengers flow at the station. MA dropout without jamming occurs when the quality of wireless communication channels is lower than the signal attenuation limit. However, it is a challenge for the attacker to cause MA dropout when the T2T and T2G communication environments are effective. In the situation of the quality of the wireless communication channel is the highest, the attack power is zero under the optimal attack strategy (as shown in equation (37)). During the duration of $[T_1, T_2]$, the attacker is required to establish attacks power by the SINR of the route map. The attacker selects a transmission power to make a packet successful rate that is less than the communication limit for the safety of the IIoT-based VCTS system when the train convoy is decoupled caused by DoS attacks. In an actual underground system, the SINR of the entire rail route can be measured by the attacker, posing a serious threat to the system.

The $\Lambda(z,g)$ can be obtained from equation (35); however, it is unsolved. A unique expression of $\Lambda(z,g)$ is described in the current section. Equation (38) can be derived from equation (11) as follows:

$$\begin{aligned}\phi_\pi(t_k)=\min\lim_{\Delta T\longrightarrow\infty}\sup\frac{1}{\Delta T}\mathbb{E}\left[\sum_{t_k=T_1}^{T_2}(1-\overline{f})z^Tz\right]\\ -\lambda_e\mathbb{E}_g\left[\rho\left(g(t_k),f(t_k)\right)\right],\end{aligned}\tag{38}$$

where $\overline{f}$ larger than f_s, as presented in equation (24).

A hypothesized that the condition formula (24) satisfies is defined to simplify $\phi_\pi(t_k)$ according to [40] as follows:

$$\phi_\pi(t_k)=\mathbf{Tr}\left(\chi_i^-(t_k)\right)-\lambda_e\mathbb{E}_g\left[\rho\left(g(t_k),f(t_k)\right)\right],\tag{39}$$

where $\mathbf{Tr}(\bullet)$ represents the trace function.

The above analyses indicate that the predicting error covariance matrix, which is iterated at each communication cycle, can be described by formula (41). Notably, the status estimation approach can be obtained from formula (13a)–(13c) and formulae (40)–(42). Therefore, Theorem 1 can be expressed as follows:

$$\rho_\pi^*(z,g,f)=\begin{cases}0, & g\le\overline{g}\text{ or }\dfrac{1}{1-\overline{f}}z^TM_ez>\lambda_e\left(\dfrac{\rho^s(t_k)-g(t_k)}{\widehat{s}_i(t_k)}-\varpi-1\right),\\ \dfrac{\rho^s(t_k)-g(t_k)}{\widehat{s}_i(t_k)}-\varpi-1, & \text{otherwise.}\end{cases}\tag{40}$$

Theorem 1. The optimal attack strategy with energy limits from the attacker's standpoint can be expressed as formula (40).

The expected corresponding cost function of the IIoT-based VCTS can be calculated as follows:

$$\Pi_\pi(z,g)(t_k)=\frac{1-f}{1-\overline{f}}z^TM_ez-\lambda_e\mathbb{E}_g\left[\rho\left(g(t_k),f(t_k)\right)\right],\tag{41}$$

where $\chi_i^-(t_k)$ and M_e follow the Lyapunov equation as shown in the following formulae -(42)-(44) [47]:

$$\chi_i^-(t_k)=(1-\overline{f})\left[A_e\chi_i^-(t_{k-1})A_e^T+Q\right],\tag{42}$$

$$M_e=(1-\overline{f})\left[A_eM_eA_e^T+Q\right].\tag{43}$$

Thus, $\Lambda_\pi(z,g)$ can be expressed as follows:

$$\Lambda_\pi\left(z(t_k),g(t_k)\right)=\frac{1}{1-\overline{f}}z^TM_ez.\tag{44}$$

Energy constraints are the most important characteristic of DoS attacks; therefore, they are discussed together on the part of the attacker. Under subway environments and tunnels do not provide a charging point for the attacker. Therefore, when the attacker intends to interfere with the T2T and T2G communication subsystems, energy constraints affect DoS attacks. The novel status estimation approach is based on the optimal attack strategy on the part of energy constraints of attacks. Studies would be further conducted to investigate other features of DoS attacks.

Contrary to the traditional method, in the enhanced Kalman filtering approach, the error estimation covariance matrix is calculated based on the optimal attack strategy. First, we have designed a cost function based on attack energy limits of DoS attacks, on the part of the attacker. Then, the optimization model, which consists of energy limits, signal attenuation, as well as the probability of successful rate of the packet transmission in the control system, was proposed. Next, we transformed the cost function into the minimum error estimation covariance matrix, while we defined a Markov stochastic process to match the failed

transmission or not. For achieving the combination of the packet dropout rate, the cyber performance, and the attack energy, which is a physical index, we also have transformed those performance into SER and SINR, which can be measured directly. Last, the minimum error estimation covariance matrix in the optimal attack strategy case is obtained on Theorem 1, which was calculated by a combination of the optimization model and feature performance of the IIoT-based VCTS system.

The evaluation criteria for mapping the train status estimation approach in the physical layer of the IIoT-based VCTS system against DoS attacks are described in the next section.

5. Evaluation Criterion of the Effects of DoS Attacks on the IIoT-Based VCTS System

We defined the security criterion to evaluate the performance of the train status estimation approach. The evaluation criterion can be divided into the physical dynamics of train operation and the passenger's satisfaction, which is inversely proportional to the traveling time of passengers. In addition, sensitivity indices including the train dynamic schedule, the train operational delay covariance, and the waiting time of the passenger are introduced to evaluate the effects on the passenger's traveling time [48, 49].

5.1. Train Dynamic Schedule. Train operation should be compliant with the train dynamic schedule and MA of ZC to ensure urban transportation safety and punctuality of urban transportation [48]. Profiles of trains' operation, where they are over space and time from one station to the destination station, are presented in Figure 5. The train dynamic schedule should be sensitive to the train operation. The dotted line in Figure 5 indicates an increased traveling time of train i from the station B to the station D, which is caused by frequent emergency brakes [49]. Notably, this spacing deviation indicates the difference from the train $i+1$ operation schedule, under normal conditions. This operation deviation is proportional to delays, which are caused by frequent emergency brakes and speed limits under DoS attacks.

5.2. Train Operational Delay Covariance. Performance indices including the delay covariance and the average waiting time of passengers are defined to investigate passenger satisfaction, which indicates passenger flow and traveling time [48]. Headway is defined as the time difference between the departure time of train i and the departure time of the train $i+1$ at station k. The delay represents the error between the headway under normal conditions and the actual headway under DoS attacks. In this study, σ_h is defined as the weight sum of train operational delay covariance as described below. This index indicates variations in the passenger travel time, which is accumulated by the train operational delay and the waiting time of the passenger under DoS attacks [49].

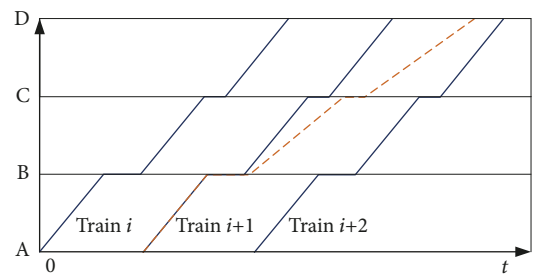

Figure 5: Train dynamic schedule.

Figure 6: Communication test link between trains and ground terminal.

Table 1: Parameters used in the simulations.

Parameters	Value
Tracing acceleration	1 m/s^2
The resistance acceleration	0.02 m/s^2
Emergency brake deceleration	1.2 m/s^2
Service brake deceleration	1 m/s^2
The length of the train	118 m
The mass of the train	1 ton
The headway between trains	120 s
The speed limits	22.2 m/s
The number of stations	14
The number of trains	12
The communication period	200 m/s
The measurement noise error	6
The passenger's leaving rate	1.7person/sec
n_c	3

$$\sigma_h = \frac{1}{n-1}\sum_{i=2}^{n}\sum_{k=1}^{m}\left(he_{i,k} - he\right)^2 \cdot ws_k, \tag{45}$$

where $he_{i,k}$ and he represent the actual operational headways of train i at station k, under DoS attacks and the operational headway with the original schedule, respectively; ws_k indicates the weight constant to map the passenger flow at station k; and m indicates the station number of the whole rail line, while n indicates the number of trains on the railway. Passenger flow for the underground railway varies among different stations. The term ws_k represents the weight

Table 2: Route parameters referenced to the Yizhuang railway.

Station name	Distance between adjacent stations (m)	Average passenger's arriving rate (person per sec)
Songjiazhuang	2,631	77/600
Xiaocun	1,275	271/600
Xiaohongmen	2,366	74/600
Jiugong	1,982	189/600
Yizhuangqiao	993	16/600
Yizhuang culture park	1,538	31/600
Wanyuanjie	1,280	192/600
Rongjingdongjie	1,354	132/600
Rongchangdongjie	2,338	16/600
Tongjinanlu	2,265	66/600
Jinghailu	2,086	46/600
Ciqunan	1,286	60/600
Ciqu	1,334	50/600
Yizhuangqiao (open soon)	—	0

value at station k to indicate the actual passenger flow in each station. For example, station B is an exchange station in the route map, where passenger waiting time at station B is longer compared to that at other stations (as shown in Figure 5).

The weight value for each station is defined according to the average passenger arrival rate for evaluation of the actual passenger flow at station k as follows:

$$ws_k = \frac{lr_k}{\sum_{k=1}^{m} lr_k}, \tag{46}$$

where lr_k indicates the arriving number per second of passengers at station k. The $\sigma_{h\_\text{average}}^{k}$ denotes the average train operational delay covariance, which represents the average train operational delay at station k.

$$\sigma_{h\_\text{average}}^{k} = \frac{1}{n-1}\sum_{i=2}^{n}\left(he_{i,k} - he\right)^2. \tag{47}$$

6. Simulation Results and Discussion

In this section, the performance effectiveness of the train status estimation approach under DoS attacks is evaluated. The simulation consists of three parts. Firstly, the simulation environments and main parameters are presented. Secondly, the energy distribution of the optimal DoS attack strategy is visualized using the MATLAB 2016 tool. Last, the performance improvement of the IIoT-based VCTS under the status estimation approach is analyzed.

6.1. Simulation Environment and Parameters. The simulation environment and parameters are referenced to the Beijing Yizhuang urban railway route, which is located in the southeast of Beijing, covering a total length of 23.3 km with 14 stations as shown in Figure 6. The simulation route comprises wayside techniques and wayside APs deployed along the track stretching over a 200 m distance. The wireless communication system is LTE-M. All relevant parameters for simulating are outlined in Table 1.

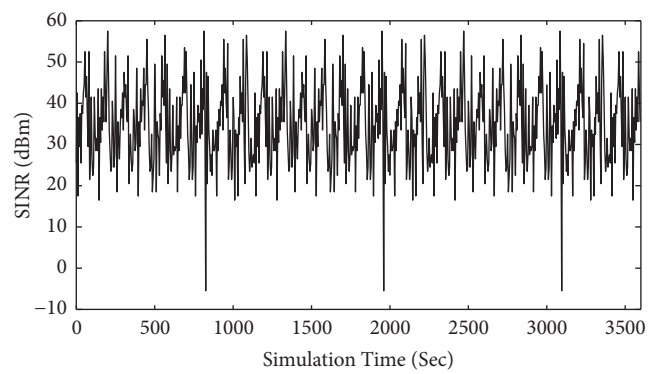

Figure 7: Measurement values of SINR with 1 hour in the Yizhuang line.

The performance of the train estimation approach was analyzed using the number of stranded passengers. In this case, the average departure time is set at 1.7 seconds per person, while the passenger flow varies across stations. Other parameters of the Yizhuang subway line are presented in Table 2.

6.2. The Optimal Attack Strategy. The propagation of signals in the IIoT-based VCTS system is similar to the propagation of electromagnetic waves in the waveguide [50]. This implies that the DoS attacks are proportional to the distance between the attacker and the victim ZC. For accurate simulation, it is assumed that the location of the attacker is adjacent to the position of victim ZC, and the attacker is alone.

Signal attenuation in the IIoT-based VCTS system is attributed to the accumulation of the fast fading model and the shadow fading model in the tunnel [50, 51]. Moreover, the optimal attack strategy is based on the SINR in the normal underground environment. Therefore, to simulate a practical realistic electromagnetic environment, values of $\hat{s}_i(t_k)$ are measured at the Yizhuang subway line as shown in Figure 7. MATLAB simulation is performed using the train status estimation approach simulation software. In this study, the sum of attack limits on part of the attacker was

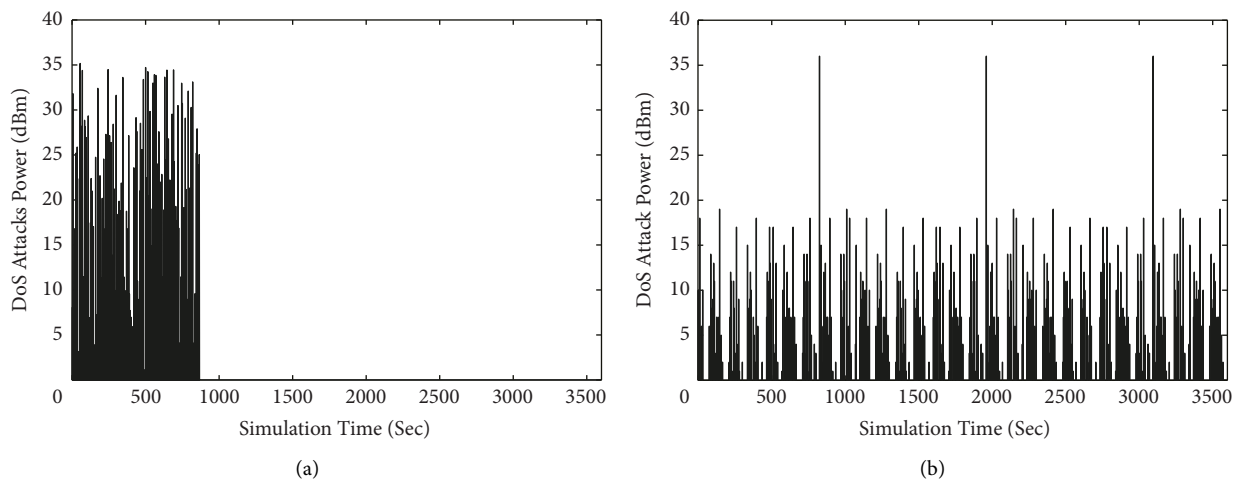

Figure 8: The energy distribution comparison of specific attacks strategies: (a) the random attacks strategy and (b) the optimal attacks strategy.

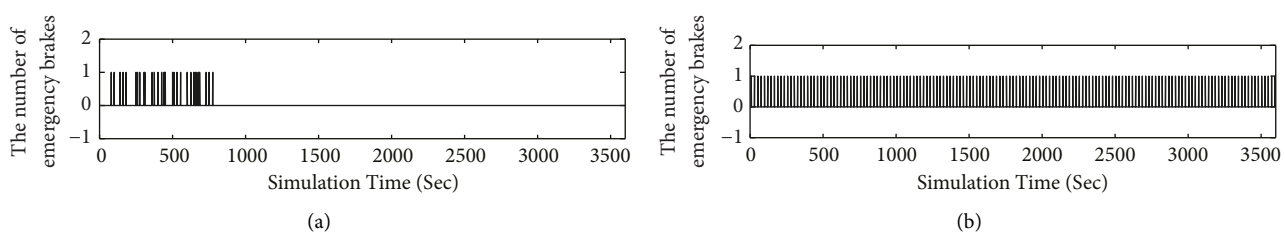

Figure 9: Comparison of the number of emergency brakes caused by specific attacks strategies: (a) the random attacks strategy and (b) the optional attacks strategy.

assumed to be 3000 W. Other parameters used in simulation include, $\rho^s(t_k) = 30\,\text{mW}$, $\varpi = 3\,\text{dBm}$, $\imath = 2\,\text{dBm}$, $\overline{f} = 0.95$, and $\overline{g} = 12\,\text{dBm}$.

The performance of optimal attack strategy and random attacks in determining the impacts of DoS attacks on the IIoT-based VCTS system is presented in Figure 8. The number of attacks power at 867 s is decreased (as shown in Figure 8(a)). Contrary to the energy distribution, the optimal attack strategy performs better with regard to the duration of DoS attacks, compared to the random attack strategy, which is characterized by one energy constraint. The number of emergency brakes, which is a measure of the effects of DoS attacks on the IIoT-based VCTS system, is associated with delays in the dynamic operation of the train. The number of emergency brakes is evaluated to determine the advantage of the optimal attack strategy as shown in Figure 9. The number of emergency brakes in the optimal attack strategy (i.e., 309) is higher compared to that of the random attack strategy (i.e., 32). This can be attributed to the higher energy consumed in the random attack strategy.

6.3. Result of the Physical Layer of the Train Status Estimation Approach. The effects of DoS attacks on the IIoT-based VCTS system were quantified using appropriate evaluation criteria, including speed/distance trajectories of the train, train dynamic schedule, train operational delay covariance, and average waiting time of the passenger, to effectively evaluate the status estimation approach. In addition, the performance of the status estimation approach was compared with that of the conventional methods to assess the effectiveness of the proposed approach. Representative methods used for comparison include the intrusion detection (IDS) method [52], which is widely used to identify DoS attacks and to evaluate the status estimation approach. Several studies have used the estimation approach based on game-theoretic (SEBG) [27]. The SEBG and IDS approaches are compared to the status estimation approach in the subsequent subsection.

6.3.1. Speed/Distance Trajectories of Trains. The x-axis in Figure 10 shows station positions, while the y-axis shows train velocities. The line chart presented in Figure 10(b) shows the trajectories of 12 trains, which are decoupled from three train convoys caused by DoS attacks. The operation speed of members of the train convoy during the control area of ZC represents the region between the third station and the tenth station, which is defined by the multiple ZC control areas in Section 2. However, the speed limits, which are due to emergency braking, disappeared under the estimation approach as shown in Figure 10(c). Train trajectories, which are not limited by speed limits, can reduce the delay in train operation under DoS attacks. The simulation results indicate the validity and stability of the defense strategy. Similarly, SEBG and IDS approaches can limit the

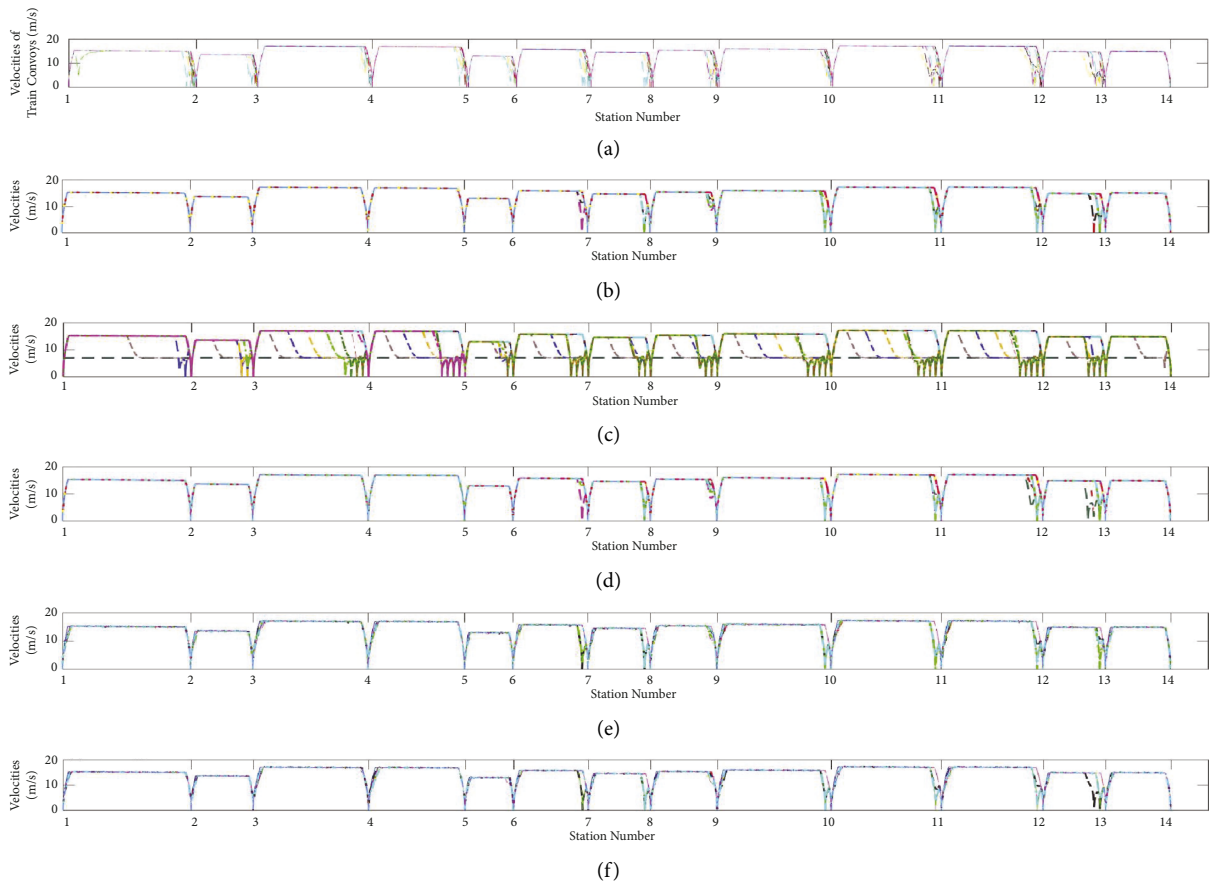

Figure 10: Train's trajectories in specific scenarios: (a) virtual coupling in IIoT-based VCTS, (b) decoupling trains in IIoT-based VCTS, (c) impacts of DoS attacks on the IIoT-based VCTS, (d) train status estimation approach, (e) SEBG approach, and (f) IDS approach.

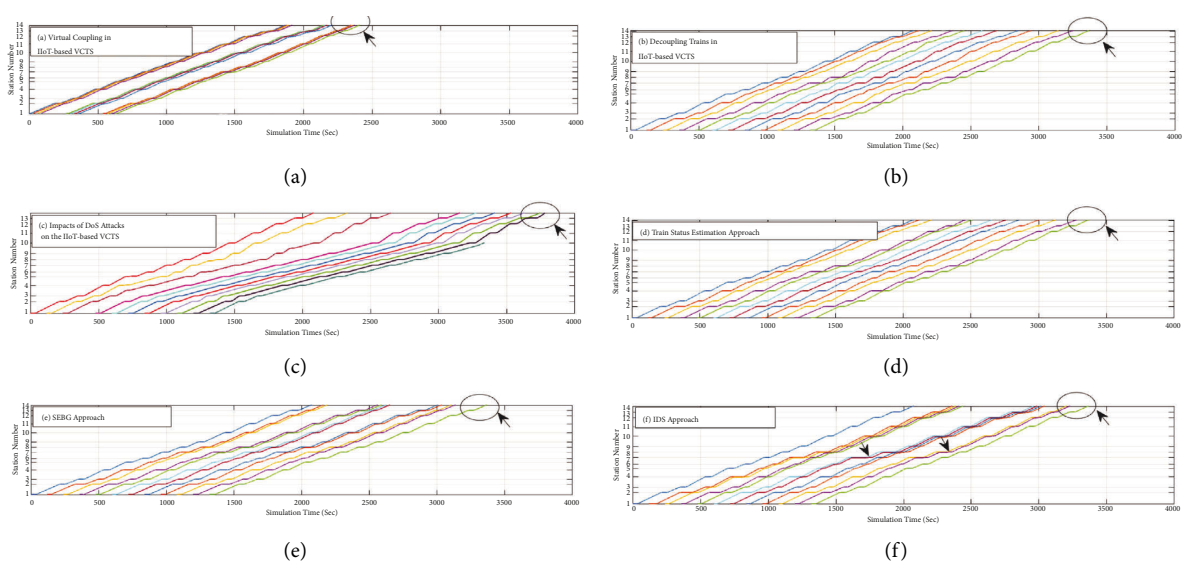

Figure 11: Dynamic schedule in specific scenarios: (a) virtual coupling in IIoT-based VCTS, (b) decoupling trains in IIoT-based VCTS, (c) impacts of DoS attacks on the IIoT-based VCTS, (d) train status estimation approach, (e) SEBG approach, and (f) IDS approach.

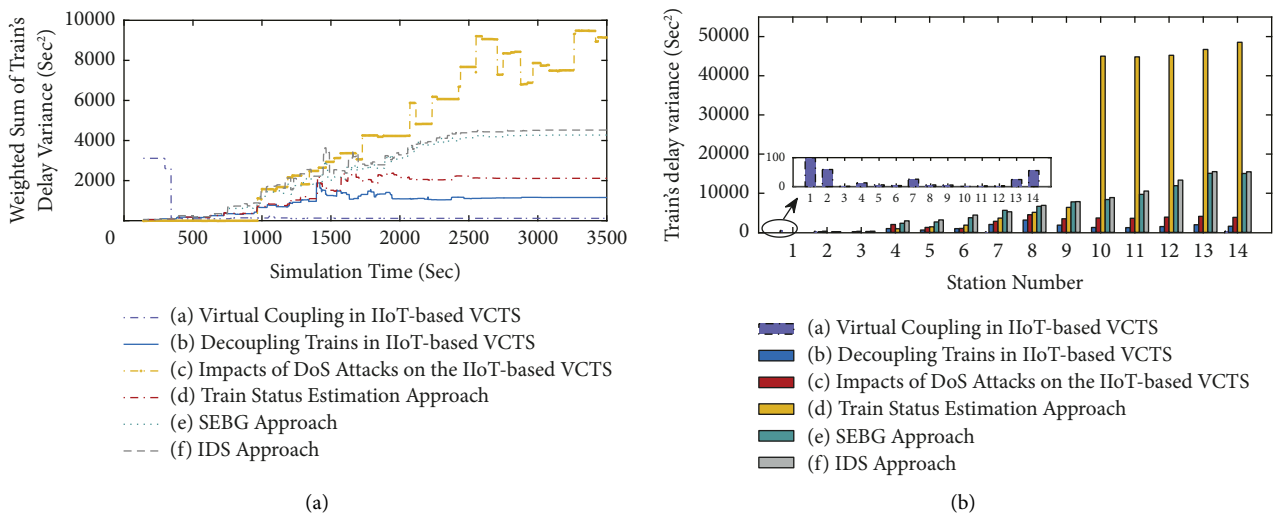

Figure 12: Train operational delay variance in specific scenarios.

number of emergency brakes due to DoS attacks. When compared to the effects of the train status estimation approach, the effects of these conventional methods on speed are significant. An increase in effects is attributed to the processing time of SEBG and IDS approaches.

6.3.2. Train Dynamic Schedule. Dynamic schedules of the Yizhuang subway line in specific scenarios with the 12 trains are shown in Figure 11. The simulation represents the 1 hour schedule of the train. The train dynamic schedule under DoS attacks with delays in train operations exceeding normal operations is presented in the chart in Figure 11. MAs overdue or dropout indicates the unreliable communication network under DoS attacks; thus, the subsequent train does not receive the MA in real time, resulting in the frequent emergency braking, occurring between the 1400 s and the 3000 s. Emergency braking causes delays in the arrival time for the train under DoS attacks as shown in Figure 11(c). The simulation result under the train status estimation approach is presented in Figure 11(d). The effects of DoS attacks on train operations are minimal better between the 2400 s to the 3000 s. These findings indicate that the dynamic schedule under the train status estimation approach resembles that of the original timetable and that this approach significantly minimizes delays in train operations (as shown in Figures 11(c) and 11(d)). On the contrary, delays in train operations of SEBG and IDS approaches are superior, compared to the status estimation approach. These results show different degrees of delay within one hour (as shown in Figures 11(e) and 11(f)), mainly with the IDS approach, which were attributed to the ineffective detection time. However, the arrival time of the last train for SEBG and IDS approaches is not significantly different when compared to the arrival time under normal conditions (as shown in Figure 11(e)). This can be attributed to manual interventions when train operation delays are extended beyond the specified limit in the subway. Notably, a small margin between adjacent dynamic operational curves improves the serious safety risk (as shown in Figure 11(f)).

6.3.3. Train Operational Delay Covariance. Variations in the weighed sum of train operational delay variance in the Yizhuang line are presented in Figure 12(a). The findings showed a general upward trend in the weight sum of train operational delay variance. When the train convoy is decoupled, the delay fluctuated from 1,400 to 2,000. However, when the wireless communication is under DoS attacks, the weight sum of train operational delay variance shows an upward trend reaching a maximum value sevenfold higher compared to the maximum value under a normal scenario. The red line in Figure 12(a) indicates that the train status estimation approach in comparison with other specific scenarios is superior. In particular, for the status estimation approach, the weight sum of train operational delay variance shows an upward trend, reaching a peak value below 2,300. This indicates that the status estimation approach is effective and consistent with the normal environment. Analysis shows a steady increase in delay variance of the train under SEBG and IDS approaches, with the maximum values less than 4,500 and 4,100, respectively.

The train operational delay covariance for each station, which is increased for each station, in the Yizhuang line under DoS attacks is shown in Figure 12(b). Notably, between the tenth and the fourteenth stations, operational delay covariance of the train under DoS attacks is higher compared to those under the trains decoupling and under the state estimation approach. These findings indicate that the delay time is proportional to the distance covered under DoS attacks. Findings for the estimation approach under DoS attacks are shown in Figure 12(b). These results indicate that compared to conventional methods, the estimation approach significantly reduces the operational delay time of the train. A steady rise in the number of delays from 0 to 15,000 is observed in the SEBG approach (as shown in

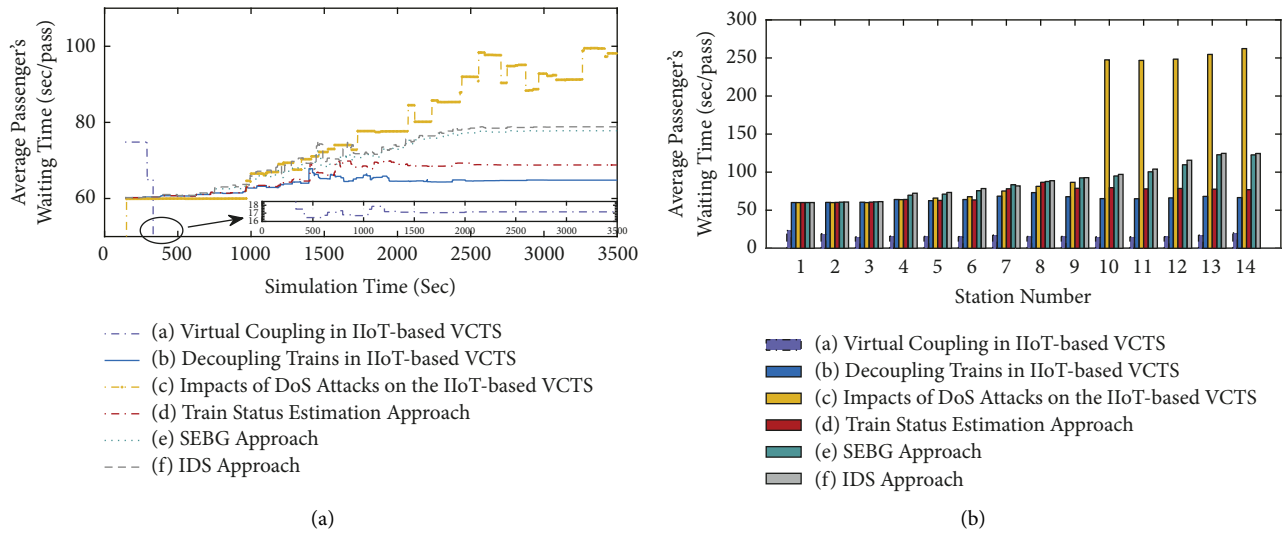

Figure 13: Average waiting time of the passenger in specific scenarios.

Figure 12(b)). Notably, when compared to the SEBG approach, the increase in the delay variance of the train in the IDS approach is not significantly different.

6.3.4. Average Waiting Time of the Passenger. Variations in the average waiting time of passengers, which is the average value of the fourteen stations, are presented in Figure 13(a). The average waiting time of the passenger rapidly increases when the DoS attacks are jammed in wireless communication. For the train status estimation approach, SEBG approach, and IDS approach, the average waiting times of passengers increased (as shown in the line chart Figure 13(a)). However, analyses revealed a steady trend in both the normal environment and under the train status estimation approach scenario, with a waiting duration of less than seventies seconds, in the entire process of train operation. These findings indicate that the train status estimation strategy is effective.

Findings for variations in the average waiting time of passengers at each station are presented in Figure 13(b). Significant increases in the average waiting time of passengers are observed under the DoS attacks, SEBG approach, and IDS approach. Notably, the train status estimation approach effectively decreases schedule delay.

7. Conclusion and Future Work

In this study, a novel train status estimation approach was established for the protection safety and punctuality in the IIoT-based VCTS system under DoS attacks. DoS attacks can affect T2T communications; as a result, it causes train convoy decoupling and enormous packet dropout. For mitigating and estimating the effects of DoS attacks on the IIoT-based VCTS system, we consider that the attack strategy of a rational attacker is optimal with the energy limited, which will most cause the system state offset, and explore a trade-off between the best gain of the attack strategy and the performance of the system, such as the real-time capability, train operational delays, the train dynamic schedule, trajectories of trains, and the average waiting time of passengers. In the study, six specific scenarios were defined to investigate the impacts of DoS attacks on the IIoT-based VCTS system. Final findings show that the train status estimation approach can mitigate the effects of DoS attacks on the punctuality of train dynamic schedule and effectively enhance the train operation safety prominently under DoS attacks. Moreover, further studies would be performed to evaluate other features of DoS attacks.

Data Availability

The data are available upon request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by grants from the China Railway Corporation Foundation under grant number K2021X001, the Natural Science Foundation of China under Grants (61973026), the Beijing Municipal Education Commission Funding (I20H100010, I19H100010), the Beijing Natural Science Foundation (L201002), in part by the Beijing Jiaotong University Project under Grant (RCS2021ZZ005), and Fundamental Research Funds for the Central Universities (2021CZ107).

References

[1] R. Parise, H. Dittus, J. Winter, and A. Lehner, "Reasoning functional requirements for virtually coupled train sets: Communication," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 12–17, 2019.

[2] H. Wang, Q. Zhao, S. Lin et al., "A reinforcement learning empowered cooperative control approach for iiot-based virtually coupled train sets," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4935–4945, 2021.

[3] S. Ma, B. Bu, and H. Wang, "A virtual coupling approach based on event-triggering control for cbtc systems under jamming attacks," in *Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 1–6, IEEE, Victoria, BC, Canada, 18 November 2020 - 16 December 2020.

[4] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[5] F. Flammini, S. Marrone, R. Nardone, A. Petrillo, S. Santini, and V. Vittorini, "Towards railway virtual coupling,"vol. 7, pp. 1–6, in *Proceedings of the IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles International Transportation Electrification Conference (ESARS-ITEC)*, vol. 7, IEEE, Nottingham, UK, Nov 2018.

[6] E. Quaglietta, M. Wang, and R. M Goverde, "A multi-state train-following model for the analysis of virtual coupling railway operations," *Journal of Rail Transport Planning & Management*, vol. 15, Article ID 100195, 2020.

[7] J. Aoun, E. Quaglietta, R. M. Goverde et al., "A hybrid delphi-ahp multi-criteria analysis of moving block and virtual coupling railway signalling," *Transportation Research Part C: Emerging Technologies*, vol. 129, pp. 1–22, 2021.

[8] D. Wu, J. Liu, H. Wang, and T. Tang, "A cpn-based approach for studying impacts of communication delays on safety and availability of safety-critical distributed networked control systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3033–3042, 2022.

[9] Y. Li, L. Zhu, H. Wang, F. R. Yu, and S. Liu, "A cross-layer defense scheme for edge intelligence-enabled cbtc systems against mitm attacks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2286–2298, 2021.

[10] C. Di Meo, M. Di Vaio, F. Flammini, R. Nardone, S. Santini, and V. Vittorini, "Ertms/etcs virtual coupling: proof of concept and numerical analysis," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2019.

[11] J. Felez, Y. Kim, and F. Borrelli, "A model predictive control approach for virtual coupling in railways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2728–2739, July 2019.

[12] W. P. M. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nešić, "Networked control systems with communication constraints: tradeoffs between transmission intervals, delays and performance," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1781–1796, Aug 2010.

[13] K. Wang, P. Xu, C.-M. Chen, S. Kumari, M. Shojafar, and M. Alazab, "Neural architecture search for robust networks in 6g-enabled massive iot domain," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5332–5339, 2021.

[14] A. Razaque, F. Amsaad, M. Abdulgader et al., "A mobility-aware human-centric cyber-physical system for efficient and secure smart healthcare," *IEEE Internet of Things Journal*, vol. 99, p. 1, 2022.

[15] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks," *Alexandria Engineering Journal*, vol. 54, no. 4, pp. 1115–1126, 2015.

[16] J. Pan, Q. Peng, S. Zhan, and J Bai, "Multiscenario-based train headway analysis under virtual coupling system," *Journal of Advanced Transportation*, vol. 10, pp. 1–20, 2021.

[17] A. D. Gupta, *Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security*, IGI Global, USA, 2016.

[18] F. Taylor, *Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives*, CRC Press, 2018.

[19] C. Chen, C. Wang, T. Qiu, M. Atiquzzaman, and D. O. Wu, "Caching in vehicular named data networking: architecture, schemes and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2378–2407, 2020.

[20] C. Chen, Y. Zhang, Z. Wang, S. Wan, and Q. Pei, "Distributed computation offloading method based on deep reinforcement learning in icv," *Applied Soft Computing*, vol. 103, Article ID 107108, 2021.

[21] L. Zhu, H. Liang, H. Wang, B. Ning, and T. Tang, "Joint security and train control design in blockchain-empowered cbtc system," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8119–8129, 2022.

[22] A. Almomani, M. Alauthman, F. Albalas, O. Dorgham, and A. Obeidat, "An online intrusion detection system to cloud computing based on neucube algorithms," *International Journal of Cloud Applications and Computing*, vol. 8, no. 2, pp. 96–112, 2018.

[23] A. Al Dmour, M. Almiani, T. Aidja, and A. Razaque, "Context-aware latency reduction protocol for secure encryption and decryption," *International Journal of High Performance Computing and Networking*, vol. 12, no. 3, p. 251, 2018.

[24] Y. Wu, Z. Wei, J. Weng, and R. H. Deng, "Position manipulation attacks to balise-based train automatic stop control," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5287–5301, 2018.

[25] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 1–2700, 2022.

[26] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Optimal linear cyber-attack on remote state estimation," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 4–13, 2017.

[27] Y. Li, D. E. Quevedo, S. Dey, and L. Shi, "Sinr-based dos attack on remote state estimation: a game-theoretic approach," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 632–642, 2017.

[28] H. Zhang and W. X. Zheng, "Denial-of-service power dispatch against linear quadratic control via a fading channel," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 3032–3039, 2018.

[29] Q. Geng, L. Zhao, L. Li, and F. Liu, "A dynamic controller design for trajectory tracking control of wheeled mobile robot under stochastic denial of service attacks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. Early Access, p. 1, 2022.

[30] L. Zhu, Y. Li, F. R. Yu, B. Ning, T. Tang, and X. Wang, "Cross-layer defense methods for jamming-resistant cbtc systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7266–7278, 2021.

[31] M. Zhou, C. Liu, A. Abiri Jahromi, D. Kundur, J. Wu, and C. Long, "Revealing vulnerability of n-1 secure power systems to coordinated cyber-physical attacks," *IEEE Transactions on Power Systems*, vol. Early Access, p. 1, 2022.

[32] S. Stickel, M. Schenker, H. Dittus et al., "Technical feasibility analysis and introduction strategy of the virtually coupled train set concept," *Scientific Reports*, vol. 12, no. 4248, pp. 1–13, 2022.

[33] X. Wang, L. Liu, T. Tang, and W. Sun, "Enhancing communication-based train control systems through train-to-train communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1544–1561, 2019.

[34] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in wsns," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, Fourth 2009.

[35] Z. A. Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 1–10, 2018.

[36] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (sdn) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2016.

[37] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Inferring distributed reflection denial of service attacks from darknet," *Computer Communications*, vol. 62, no. 1/2, pp. 59–71, 2015.

[38] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*, University of North Carolina, 1995.

[39] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[40] L. Shi, P. Cheng, and J. Chen, "Sensor data scheduling for optimal state estimation with communication energy constraint," *Automatica*, vol. 47, no. 8, pp. 1693–1698, 2011.

[41] O. Vega Amaya, "The average cost optimality equation: a fixed point approach," *Boletín De La Sociedad Matemática Mexicana Tercera Serie*, vol. 9, no. 1, pp. 85–195, 2003.

[42] C. De Persis and P. Tesi, "Networked control of nonlinear systems under Denial-of-Service," *Systems & Control Letters*, vol. 96, pp. 124–131, 2016.

[43] K. Kinjo, E. Uchibe, and K. Doya, "Evaluation of linearly solvable Markov decision process with dynamic model learning in a mobile robot navigation task," *Frontiers in Neurorobotics*, vol. 7, p. 7, 2013.

[44] H. Zhang, P. Cheng, L. Shi, and J. Chen, "Optimal dos attack scheduling in wireless networked control system," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 843–852, 2016.

[45] D. E. Quevedo and A. Ahlen, "A predictive power control scheme for energy efficient state estimation via wireless sensor networks," in *Proceedings of the 2008 47th IEEE Conference on Decision and Control*, pp. 1103–1108, IEEE, Cancun, Mexico, 09-11 December 2008.

[46] D. E. Quevedo, A. AhleN, and J. Ostergaard, "Energy efficient state estimation with wireless sensors through the use of predictive power control and coding," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4811–4823, 2010.

[47] T. Stykel and V. Simoncini, "Krylov subspace methods for projected Lyapunov equations," *Applied Numerical Mathematics*, vol. 62, no. 1, pp. 35–50, 2012.

[48] D. Canca, E. Barrena, E. Algaba, and A. Zarzo, "Design and analysis of demand-adapted railway timetables," *Journal of Advanced Transportation*, vol. 48, no. 2, pp. 119–137, 2014.

[49] E. Barrena, D. Canca, L. C. Coelho, and G. Laporte, "Single-line rail rapid transit timetabling under dynamic passenger demand," *Transportation Research Part B*, vol. 70, no. C, pp. 134–150, 2014.

[50] H. Wang, F. R. Yu, and H. Jiang, "Modeling of radio channels with leaky coaxial cable for lte-m based cbtc systems," *IEEE Communications Letters*, vol. 20, no. 5, pp. 1038–1041, 2016.

[51] H. W. Wang, B. Ning, H. L. Jiang, W. N. Liu, and D. Beijing, "Research on propagation characteristics of 2.4 GHz WLAN in tunnels for CBTC train ground communication systems," *Journal of the China Railway Society*, vol. 35, no. 10, pp. 52–58, 2013.

[52] J. Li, Z. Zhao, R. Li, and H. Zhang, "Ai-based two-stage intrusion detection for software defined iot networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2093–2102, 2019.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 8380239, 10 pages
https://doi.org/10.1155/2022/8380239

Research Article

A Homomorphic Signcryption-Based Privacy Preserving Federated Learning Framework for IoTs

Weidong Du,[1,2] **Min Li**,[1] **Yiliang Han**,[2] **Xu An Wang**,[2] **and Zhaoying Wei**[3]

[1]*Xi'an Hi-Tech Research Institute, Xi'an 710025, China*
[2]*College of Cryptography, Engineering University of PAP, Xi'an 710086, China*
[3]*College of Science, Xi'an Shiyou University, Xi'an 710065, China*

Correspondence should be addressed to Min Li; proflimin@163.com

Received 23 May 2022; Accepted 17 August 2022; Published 22 September 2022

Academic Editor: Chen Chen

Federated learning (FL) enables clients to train a machine learning model collaboratively by just aggregating their model parameters, which makes it very useful in empowering the IoTs with intelligence. To prevent privacy information leakage from parameters during aggregation, many FL frameworks use homomorphic encryption to protect client's parameters. However, a secure federated learning framework should not only protect privacy of the parameters but also guarantee integrity of the aggregated results. In this paper, we propose an efficient homomorphic signcryption framework that can encrypt and sign the parameters in one go. According to the additive homomorphic property of our framework, it allows aggregating the signcryptions of parameters securely. Thus, our framework can both verify the integrity of the aggregated results and protect the privacy of the parameters. Moreover, we employ the blinding technique to resist collusion attacks between internal curious clients and the server and leverage the Chinese Remainder Theorem to improve efficiency. Finally, we simulate our framework in FedML. Extensive experimental results on four benchmark datasets demonstrate that our framework can protect privacy without compromising model performance, and our framework is more efficient than similar frameworks.

1. Introduction

Traditional machine learning (ML) uses huge amounts of data collected from various sources to train models. However, data sharing from different devices or organizations may disclose privacy information about the owners. To solve the dilemma between protecting data privacy and leveraging the AI benefits to these data sensitive domains, federated learning (FL) [1, 2] is proposed to train ML models without sharing data directly. Because FL can protect the privacy while utilizing data, it has great application prospect in many scenarios [3–6], especially in IoTs [7, 8].

FL enables devices to collaboratively build a global ML model by only aggregating their model parameters with their data kept at their local storage. However, it still has the problem of privacy leakage. Existing researches [9–11] revealed that the exposed gradients still retain sensitive information about the training data. To avoid exposing gradients of a client during the training process, many privacy preserving frameworks based on cryptographic techniques have been proposed. Because homomorphic encryption (HE) allows aggregating the gradients through encryption envelope, privacy preserving FL frameworks based on HE have aroused many researchers' interests [9, 12–14].

However, because HE consists of complex cryptographic operations, these frameworks always bring heavy overhead. On the other hand, all clients in these frameworks share the same key pairs. If the server colludes with any client to get the secret key, the privacy protections will be invalid. Apart from efficiency and collusion attack issues, most privacy preserving FL frameworks ignore verifying the integrity of the aggregated results. However, a malicious or compromised server may forge the aggregated results to seduce clients to expose more sensitive information. Therefore, a privacy preserving FL framework should consider efficiency, privacy, and integrity.

Table 1: Comparison of PPFLS.

PPFL	Updates privacy	Model privacy	Collusion resistant	Verifiability	Model supported
Shokri & Shmatikov	No	No	No	No	Linear & deep model
Geyer et al.	No	No	No	No	Linear & deep model
Bonawitz et al.	Yes	No	Yes	No	Linear & deep model
Phong et al.	Yes	Yes	No	No	Linear & deep model
Batchcrypt	Yes	Yes	Yes	No	Linear & deep model
Ma et al.	Yes	Yes	No	Yes	Linear & deep model
Zheng et al.	Yes	No	Yes	Yes	Linear model
Xu et al.	Yes	No	Yes	Yes	Linear & deep model
CRT-Paillier	Yes	Yes	No	Yes	Linear & deep model
VFL	Yes	Yes	Yes	Yes	Linear & deep model
Our framework	Yes	Yes	Yes	Yes	Linear & deep model

We note that, though our framework has the same property as VFL, our framework is more efficient both in computation and in communication, which will be proved in Section 5.

This paper aims to address these problems by designing an efficient homomorphic signcryption based privacy preserving framework for FL in IoTs. We compare our framework with others in Table 1. Our contributions are summarized as follows:

(i) We design a homomorphic signcryption mechanism to encrypt and sign the clients' gradients at the same time. According to its additive homomorphic property, our mechanism allows the server to aggregate the gradients and signatures securely and allows the clients to verify the integrity (correctness) of the aggregated results.

(ii) We employ the blinding technique to resist collusion attacks. Even if $n-2$ clients try to collude with the server, they cannot steal privacy information of other clients' parameters.

(iii) We combine the clipping and quantizing technique in Batchcrypt [13] and the Chinese Remainder Theorem to reduce the number of complex cryptographic operations and ciphertexts. Thus, our framework is more efficient in computation and communication than similar frameworks.

(iv) We present a comprehensive analysis for the security of our framework, which proves the privacy security of the gradients and the model as well as the integrity of the aggregated results. Finally, we simulate our framework in FedML on four benchmark datasets (and corresponding models): Federated MNIST, CINIC-10, CIFAR-10, and CIFAR-100. The experimental results demonstrate that our framework has high computation and communication efficiency.

The rest of the paper is organized as follows: the preliminaries and related works are briefly introduced in Section 2. The detailed procedures of our framework are presented in Section 3. The security analysis and experimental evaluation are provided in Sections 4 and 5, respectively. Finally, we conclude our paper in Section 6.

2. Preliminaries and Related Work

In this section, we introduce the preliminaries and related works of privacy preserving FL.

2.1. Paillier One-Way Trapdoor Permutation. In this paper, we employ the Paillier one-way trapdoor permutation [15] to realize the homomorphic signcryption mechanism. The details of the Paillier one-way trapdoor permutation are described as follows:

(i) Key generation: select two large prime numbers p and q randomly with $\gcd(pq, (p-1)(q-1)) = 1$ and compute the modulus $N = pq$ and the least common multiple $\lambda = \mathrm{lcm}(p-1, q-1)$. Select a random group generator $g \in \mathbb{Z}^*_{N^2}$, where the order of g divides n. The key pairs is $((N, g), \lambda)$.

(ii) Encryption: for any message $m < N^2$, split m into m_1, m_2 with $m = m_1 + Nm_2$. The ciphertext of m is $c = g^{m_1} m_2^N \bmod N^2$.

(iii) Decryption: the decryption process is as follows:

$$\begin{aligned} m_1 &= \frac{L\left(c^\lambda \bmod N^2\right)}{L\left(g^\lambda \bmod N^2\right)} \bmod N, \\ c' &= cg^{-m_1} \bmod N, \\ m_2 &= c'^{\,N^{-1} \bmod \lambda} \bmod N, \\ m &= m_1 + Nm_2. \end{aligned} \tag{1}$$

The L function is defined as $L(u) = (u-1)/N$.

2.2. Batchcrypt. In Batchcrypt [13], the authors proposed a method to clip and quantize the gradients; it consists of two functions:

(i) $\alpha = dACIQ(s, V, v)$: C=compute the clipping threshold α according to the number s of gradients W, the maximum gradient V, and the minimum gradient v of each layer.

(ii) $\{\widetilde{w_i}\}$ = Quantize$(\{w_i\}, \{\alpha_i\}, n)$: quantize the gradient w_i into r bits according to the scaled quantization range $[-n\alpha, n\alpha]$ in case the sum of gradients from all clients overflows. Here, n denote the number of clients.

Batchcrypt [13] allows clipping and quantizing the gradients into fixed bit width integers, which can reduce the number of parameters in encryption and decryption.

2.3. Chinese Remainder Theorem. Suppose $m_0, m_2, \ldots, m_K$ are K positive pairwise coprime integers. Let $M = m_1 \cdot m_2 \cdots m_K$. Then there exists one unique integer satisfying the following group of congruences: $y \equiv a_1 \bmod m_1, y \equiv a_2 \bmod m_2, \ldots, y \equiv a_K \bmod m_K$. Similarly, we can unpack y to get $a_i \equiv y \bmod m_i (i = 1, 2, \ldots, K)$. To ease description, we denote the packing function as $y = CRT(\{a_1, a_2, \ldots, a_K\}, \{m_1, m_2, \ldots, m_K\})$ and the unpacking function as $\{a_1, a_2, \ldots, a_K\} = CRT\_\text{inverse}(y, \{m_1, m_2, \ldots, m_K\})$. For any two packed data y_i and y_j, the equation $y_i + y_j \equiv a_l^i + a_l^j \bmod m_l (l = 1, 2, \ldots, K)$ holds. Thus, we know that the CRT satisfies additive homomorphism.

2.4. Privacy Preserving Federated Learning. Many researches have been devoted to privacy preserving FL (PPFL), and they are mainly based on cryptographic methods. Shokri and Shmatikov [16] employed selective parameter update atop differential privacy to protect training record privacy. Geyer et al. [17] applied differential privacy directly to guarantee client level differential privacy. But these differential privacy based approaches reduced model accuracy significantly. Bonawitz et al. [18] leveraged secure aggregation to protect privacy in FL. Nevertheless, it exposes the trained model in the plaintext form. To address this problem, Phong et al. [9] proposed an additively homomorphic encryption based FL framework. However, it brought heavy computation and communication overhead because the encryption scheme involves complex cryptographic operations and large ciphertexts. To improve efficiency, Batchcrypt [13] tried to encode groups of quantized gradients large integers, but their method cannot defeat collusion attacks among internal curious clients and the server.

2.5. Verifiable Federated Learning. In FL, a malicious or compromised server may falsify aggregated results returned to the clients. Several methods have been proposed to solve this problem.

Ma et al. [19] used bilinear aggregate signature [20] for integrity verification. But the verification process involves all clients. For a large number of clients, this process is time-consuming. Xu et al. [21] employed a homomorphic hash function to verify integrity, but it cannot protect the privacy of the jointly trained model. Zheng et al. [22] designed Helen, a framework utilized to secure multiparty computing protocol [23], to verify the correctness of the aggregated results. Nevertheless, the framework does not support training complex models such as neural networks. Zhang et al. (referred to as CRT-Paillier in Section 5) [24] combined bilinear aggregate signature and Paillier encryption [15] to protect privacy and integrity of the aggregated results. But they cannot resist collusion attacks between the server and internal curious clients. Fu et al. proposed VFL [25] by employing Lagrange interpolation to realize secure aggregation and check the integrity of the aggregated results, but the parameter splitting process is costly, making it unsuitable for large deep learning models.

To summarize, among PPFL frameworks, many of them ignored verifying the correctness of the aggregated results. For verifiable frameworks, they either cannot apply to large nonlinear models, or cannot protect the privacy of the model, or cannot resist collusion attacks between the server and the clients. Our work is to address these problems.

3. Homomorphic Signcryption-Based Privacy Preserving Federated Learning Framework for IoT

3.1. Threat Model and Design Goals. In our framework, we assume security threats from three different perspectives:

(i) Internal curious clients may try to steal privacy information about other clients by inspecting the ciphertexts transmitted during the training process. Here we should note that, because the ultimate goal of the clients in is to train a good ML model, the curious clients may collude with the server to steal private information about other clients, but they will not collude with the sever to tamper the aggregated results.

(ii) The server may try to steal the jointly trained model because of its great economic value. But the model should be the common property of the clients.

(iii) The server may return falsified aggregated results to clients driven by some unexpected motivations.

Consequently, to enable a privacy preserving FL framework under the aforementioned threat model, the design goals are summarized as follows:

(i) Correctness: when the server and clients operate strictly according to the protocol, the aggregated results should be correct. This ensures that the clients achieve their goal and the final model has good performance.

(ii) Data privacy: data privacy means the privacy of a client's training data and gradients. The server and curious clients cannot gather any private information about the data and gradients from messages they receive.

(iii) Model privacy: model privacy refers to the privacy of the jointly trained model. The server or any other party not in the framework cannot steal the model by inspecting the immediate data transmitted during the learning process.

(iv) Verifiability: all clients can verify the integrity of the aggregation of parameters so that clients can detect the malicious behavior if the server returns tampered aggregated results.

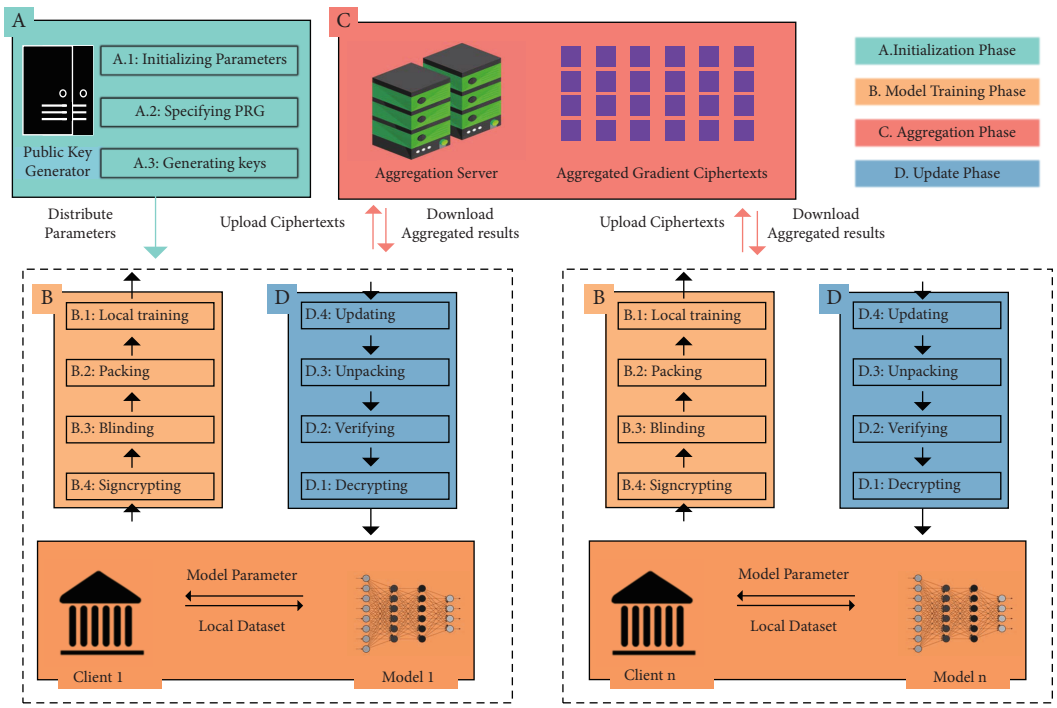

Figure 1: Overview of our framework.

3.2. Overview. As illustrated in Figure 1, our framework consists of 4 phases: initialization, model training, aggregation, and update. We summarize the 4 phases as follows:

(i) Initialization phase: the Public Key Generator (PKG) determines the model parameters and generates Paillier keypairs, Pseudo-Random Generator (*PRG*), random seed S_0, and K pairwise coprime large numbers $\{m_l\}_{l=1,2,\ldots,K}$ for all clients.

(ii) Model training phase: all clients train the ML model locally clip and quantize the gradients into r bits long integers, pack them using CRT, mask them with blinding factors, and then signcrypt them.

(iii) Aggregation phase: the server aggregates the ciphertexts from all clients and distributes the result to them.

(iv) Update phase: each client firstly decrypts the aggregated ciphertexts and then verifies the integrity of the aggregated gradients. If the result is correct, it unpacks them into plain aggregated gradients and updates their local model accordingly. Otherwise, it would terminate the learning process.

3.3. Detailed Construction

3.3.1. Initialization. For simplicity, we assume there are n clients, and each client is uniquely indexed by a number i. The set of client index is represented by index $= \{1, 2, \ldots, n\}$. In our framework, the PKG needs to initialize the model parameters and generate keys and PRG as follows:

(1) Initializing model parameters: the PKG specifies the architecture of the model, randomly initializes the model parameters as $M_0 = \{M_i\}_{i=1,2,\ldots,T}$ (T denotes the number of layers of the learning model), and specifies an appropriate learning rate η.

(2) Specifying packing parameters: the PKG generates K positive integers $\{m_i\}_{i=1,2,\ldots,K}$ that are pairwise coprime $\gcd(m_i, m_j) = 1\,(\forall i \neq j)$. We define $M = \prod_{i=1}^{k} m_i$. These integers are used to pack gradients.

(3) Specifying blinding parameters: the PKG randomly chooses n integer sets $\{s_1^j\}_{j=1,2,\ldots,|M_0|}$, $\{s_2^j\}_{j=1,2,\ldots,|M_0|}, \ldots, \{s_n^j\}_{j=1,2,\ldots,|M_0|} \in (\mathbb{Z}_N^*)^{|M_0|}$ such that $\sum_{i=1}^{n} s_i^j = 0$ for $j = 1, 2, \ldots, |M_0|$ ($|M_0|$ stands for the number of parameters of the model) and then sends $\{s_i^j\}_{j=1,2,\ldots,|M_0|}$ to client i.

(4) Specifying PRG: the PKG sends each client the same pseudorandom generator $\mathrm{PRG}(\cdot)$ and seed S_0. The pseudorandom generator is used to synchronize random numbers among clients for signcrypting the gradients in each training round.

(5) Generating signcryption keys: the PKG sends each client a Paillier key pair $((N, g_1), \lambda)$ for encryption and decryption and a secret value g_2^x to sign the gradients, where $N = pq, g_1 \in \mathbb{Z}_{N^2}^*$, $g_2 \in \mathbb{Z}_N^*$, and $x \in N$.

3.3.2. Model Training Phase. The model training phase of client $i\,(i \in \text{index})$ consists of the following four steps:

(1) Local training: in $R-th$ round of training, client i trains its local model on dataset D_i and computes the gradients G_i.

(2) Packing: client i sends the layer-wise maximum value V_i^j, minimum value v_i^j, and the size s_j of layer $j(j = 1, 2, \ldots, T)$ to the server. After receiving those layer-wise clipping parameters from all clients, the aggregator server calculates the clipping threshold $\alpha_j = dACIQ(s_j, \mathrm{Max}(V_i^j)_{i=1,2,\ldots,n}, \mathrm{in}(v_i^j)_{i=1,2,\ldots,n})$, where Max and Min compute the maximum and minimum of a set, respectively. After receiving the clipping thresholds, client i quantizes its gradients with $\widetilde{G}_i = \mathrm{Quantize}(G_i, \{\alpha_j\}_{j=1,2,\ldots,T}, n)$, client i partitions the quantized gradients into $L = \left\lceil \left| \widetilde{G}_i \right| / K \right\rceil$ groups $P_i = \{p_i^1 = \{p_i^{1l}\}_{l=1,2,\ldots,K}, p_i^2 = \{p_i^{2l}\}_{l=1,2,\ldots,K}, \ldots, p_i^L = \{p_i^{Ll}\}_{l=1,2,\ldots,K}\}$. If $|\widetilde{G}_i|$ is not divisible by K, $\widetilde{G}_i$ should be padded with 0 s. Then client i packs them into $W_i = \{w_i^1, w_i^2, \ldots, w_i^L\}$ with $w_i^j = CRT(p_i^j, \{m_l\}_{l=1,2,\ldots,K})$ and $w_i^j \in \mathbb{F}_M$. After packing, the number of cryptographic operations and ciphertexts is greatly reduced; thus the computation and communication efficiency is improved.

(3) Blinding: to resist collusion attacks between curious clients and the aggregation server, client i blinds w_i^j by adding its blinding factors to get $\widetilde{W}_i = \{\widetilde{w}_i^1, \widetilde{w}_i^2, \ldots, \widetilde{w}_i^l\} = \{w_i^1 + s_i^1, w_i^2 + s_i^2, \ldots, w_i^L + s_i^L\}$.

(4) Signcrypting: for each blinded gradient $\widetilde{w}_i^j \in \widetilde{W}_i$, client i computes its signature $\sigma_i^j = g_2^{x\widetilde{w}_i^j} \bmod N$ and uses the PRG to synchronize with other clients a random number $r_R = S_R = PRG(S_{R-1})$, where S_R will be used as seed for the PRG in the next round of training. Finally, clients i compute a signcryption of $\widetilde{w}_i^j$ as $\mathrm{cipher}_i^j = g_i^{\widetilde{w}_i^j} (\sigma_i^j r_R)^N \bmod N^2$.

We summarize the steps of model training phase in Algorithm 1.

3.3.3. Aggregation Phase.

(1) Aggregation: after receiving the ciphertexts from all clients, the server aggregates them according to the layer index $j(j = 1, 2, \ldots, L)$. Because Paillier trapdoor permutation is additively homomorphic, the sever computes

$$\begin{aligned}\mathrm{cipher}^j &= \prod_{i=1}^{n} \mathrm{cipher}_i^j,\\ &= \prod_{i=1}^{n} g_1^{w_i^j + s_i^j} \left(g_2^{x\left(w_i^j + s_i^j r_R\right)^N} \right),\\ &= g_1^{\sum_{i=1}^{n} w_i^j + \sum_{i=1}^{n} s_i^j} \left(g_2^{x \sum_{i=1}^{n} w_i^j + x \sum_{i=1}^{n} s_i^j} r_R^n \right)^N \bmod N^2.\end{aligned} \tag{2}$$

Because $\sum_{i=1}^{n} s_i^j = 0$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, |M_0|$, we have

$$\mathrm{cipher}^j = g_1^{\sum_{i=1}^{n} w_i^j} \left(g_2^{x \sum_{i=1}^{n} w_i^j} r_R^n \right)^N \bmod N^2. \tag{3}$$

3.3.4. Update Phase. After receiving the aggregated ciphertexts $\{\mathrm{cipher}^j\}_{j=1,2,\ldots,L}$, each client completes the update phase as follows.

(1) Decrypting: client i firstly decrypts the $j-th$ ciphertext:

$$\begin{aligned}W^j &= \frac{L\left(\left(\mathrm{cipher}^j\right) \bmod N^2\right)}{L\left(g_1 \lambda \bmod N^2\right)} \bmod N^2 = \sum_{i=1}^{n} w^j + \sum_{i=1}^{n} s^j\\ &= \sum_{i=1}^{n} w^j.\end{aligned} \tag{4}$$

Then it computes the aggregated signature as follows:

$$\begin{aligned}\widetilde{\mathrm{cipher}}^j &= \mathrm{cipher}^j g_i^{-W^i} \bmod N,\\ \sigma^j &= \left(\widetilde{cipher}^j\right)^{N^{-1} \bmod \lambda} r_R^{-nN} \bmod N\\ &= g_2^{x \sum_{i=1}^{n} w_i^{j \bmod \lambda}} \bmod N.\end{aligned} \tag{5}$$

(2) Verifying: for each client i, if the equation $g_2^{xW^j \bmod \lambda} \bmod N \equiv \sigma^j$ holds, the aggregation results are correct. Otherwise, it is supposed to be falsified, and the learning process terminates.

(3) Unpacking: if the termination condition is not satisfied, client i unpacks the plain aggregated gradients $G_R = \sum_{i=1}^{n} G_i = CRT\_\mathrm{inverse}(W, \{m_1, m_2, \cdots, m_K\})$, where $W = \{W^1, W^2, \ldots, W^L\} = \{\sum_{i=1}^{n} w_i^1, \sum_{i=1}^{n} w_i^2, \ldots, \sum_{i=1}^{n} w_i^L\}$.

(4) Updating: client i updates the model parameters $M_R = M_{R-1} - (\eta/n) G_R$ and prepares for the next round of aggregation.

We summarize the steps of update phase in Algorithm 2.

4. Security Analysis

Theorem 1. *The privacy of the clients' gradients is protected against internal curious clients.*

Proof. Suppose an internal curious client i intercepts a ciphertext from client l by intercepting the messages between client l and the server. Then client i has $\mathrm{cipher}_l^j = g_1^{\widetilde{w}_l^j} (\sigma_l^j r_R)^N \bmod N^2 (j = 1, 2, \ldots, L)$. Though it can decrypt them to get $\widetilde{w}_l^j = w_l^j + s_l^j$ and $g_2^{x(w_l^j + s_l^j)}$, it cannot infer any information about the gradients w_l^j because the blinding factor s_l^j is hidden from it. Therefore, our framework can protect the privacy of the clients' gradients against internal curious clients. □

Input: Training round R, model parameters M_{R-1}, dataset D_i,quantization bit width r, blinding factors $\left\{s_i^j\right\}_{j=1,2,\ldots,|M_0|}$, PRG, random seed S_{R-1}, Paillier key pairs $((N, g_1), \lambda)$, secret value g_2^x
Output: Signcryption of the masked gradients $Cipher_i^j$
functionMODELTRAINING
Compute gradients G_i based on M_{R-1} and D_i
Send layer-wise gradients Max-Min values and sizes $\left\{V_i^j, v_i^j, s_j\right\}_{j=1,2,\ldots,T}$ to the aggregation server
Clip G_i with corresponding threshold $\left\{n\alpha_j\right\}_{j=1,2,\ldots,T}$ (Advance Scaling) and quantize them into r bits
Batch the quantized gradients $\widetilde{G}_i$ layer by layer into $W_i = \{w_i^1, w_i^2, \ldots, w_i^L\} = \text{Encode}(\widetilde{G}_i)$
Blind W_i with $\left\{s_i^j\right\}_{j=1,2,\ldots,|M_0|}$ to compute $\widetilde{W}_i = \left\{\widetilde{w}_i^1, \widetilde{w}_i^2, \ldots, \widetilde{w}_i^L\right\} = \{w_i^1 + s_i^1, w_i^2 + s_i^2, \ldots, w_i^L + s_i^L\}$
Sign each blinded gradient $\widetilde{w}_i^j \in \widetilde{W}_i$ with g_2^x, use the PRG to generate the synchronizing random number $r_R = S_R = PRG(s_{R-1})$, compute the signcryption $\text{cipher}_i^j = g_i^{\widetilde{w}_i^j} (\sigma_i^j r_R)^N \bmod N^2$.
Send $\left\{cipher_i^j\right\}_{j=1,2,\ldots,T}$ to the aggregation server
end function

ALGORITHM 1: Model training phase of client i.

Input: Aggregated ciphertexts $\{cipher^j\}_{j=1,2,\ldots,L}$
Output: Updated Model M_R
functionUPDATE
Decrypt $\left\{\text{cipher}^j\right\}_{j=1,2,\ldots,L}$ to get $W = \{W^j\}_{j=1,2,\ldots,L} = \left\{\sum i = 1 n w_i^j + s_i^j\right\}_{j=1,2,\ldots,L}$, compute $\widetilde{\text{cipher}}^j = \text{cipher}^j g_i^{-W^i} \bmod N$ and $\sigma^j = (\widetilde{cipher}^j)^{N^{-1} \bmod \lambda} r_R^{-nN} \bmod N = g_2^{x \sum_{i=1} n b_i^j \bmod \lambda} \bmod N$
Verify the integrity of the aggregated gradients by checking if the equation $g_2^{xW^j \bmod \lambda} \bmod N \equiv \sigma^j$ holds. If not, terminate the learning process.
Decode the aggregated gradients to get $G_R = \sum_{i=1}^n G_i = \text{Decode}(W)$.
Update model with $M_R = M_{R-1} - (\eta/n) G_R$.
end function

ALGORITHM 2: Update phase of client i.

Theorem 2. *The privacy of the clients' gradients is protected against collusion attack between $n-2$ clients.*

Proof. Assume the set of collusion clients is $|Adv|$ with $|Adv| = 2$. For clients $i, l \notin Adv$, their ciphertexts $\text{cipher}_i^j = g_1^{\widetilde{w}_i^j} (\sigma_i^j r_R)^N \bmod N^2$ and $\text{cipher}_l^j = g_1^{\widetilde{w}_l^j} (\sigma_l^j r_R)^N \bmod N^2$ $(j = 1, 2, \ldots, L)$ contain blinding factors s_i^j and s_l^j that are unknown to clients in Adv and the server. For any aggregated ciphertexts include $cipher_i^j$ and $cipher_l^j$, the situation is similar. Thus, even when the server colludes with $n-2$ clients, they cannot infer any information about other clients' gradients. □

Theorem 3. *In our framework, each client can independently verify the correctness of the aggregated results, and the server cannot deceive the clients with tampered aggregated results.*

Proof. If the clients receive the correct aggregated results, obviously the equation $g_2^{xW^j} = \sigma^j$ $(j = 1, 2, \ldots, L)$ holds. Assume a client receives forged aggregated results, and without loss of generality, we assume the sever falsified the aggregated results to

$$\begin{aligned} \widehat{\text{cipher}} &= g_1^{\Delta_1} g_2^{\Delta_2} \prod_{i=1}^{n} \text{cipher}_i^j \\ &= g_1^{\sum_{i=1}^{n} (\widetilde{w}_i^j + \Delta_1)} \left(g_2^{x \sum_{i=1}^{n} (\widetilde{w}_i^j + \Delta_2)} r_R^n \right)^N \bmod N^2. \end{aligned} \tag{6}$$

If the malicious server tries to successfully fool the verification mechanism of our framework, it must make sure the following equation holds: $g_2^{\sum_{i=1}^{n} (x\widetilde{w}_i^j + x\Delta_1)} = g_2^{x \sum_{i=1}^{n} (\widetilde{w}_i^j + \Delta_2)}$; namely, the server should select Δ_1 and Δ_2 that satisfy $x\Delta_1 = \Delta_2$. However, it is impossible because g_2^x is confidential to the server. Therefore, we prove that the server cannot deceive the clients with tampered aggregated results. □

5. Performance Evaluation

We evaluate the performance of our framework in this section. Specifically, we first compare its accuracy against that of the original FL framework in four benchmark datasets: Federated MNIST, CINIC-10, CIFAR-10, and CIFAR-100. The models corresponding to the datasets are listed in Table 2. Then we compare its computation and communication cost with that of the other two verifiable PPFL frameworks: VFL [25] and CRT-Paillier [24].

Table 2: Datasets and corresponding models.

Datasets	Model structure	Model parameters (#)	Model size (MB)
CINIC-10	ResNet-56	591322	18.05
CIFAR-100	ResNet-56	591322	18.05
Federated EMNIST	CNN	1206590	36.82
CIFAR-10	ResNet-18+ group normalization	11232612	342.79

We should note that though ResNet-18 has less layers than ResNet-56; because of the group normalization operation, it has more parameters than ResNet-56.

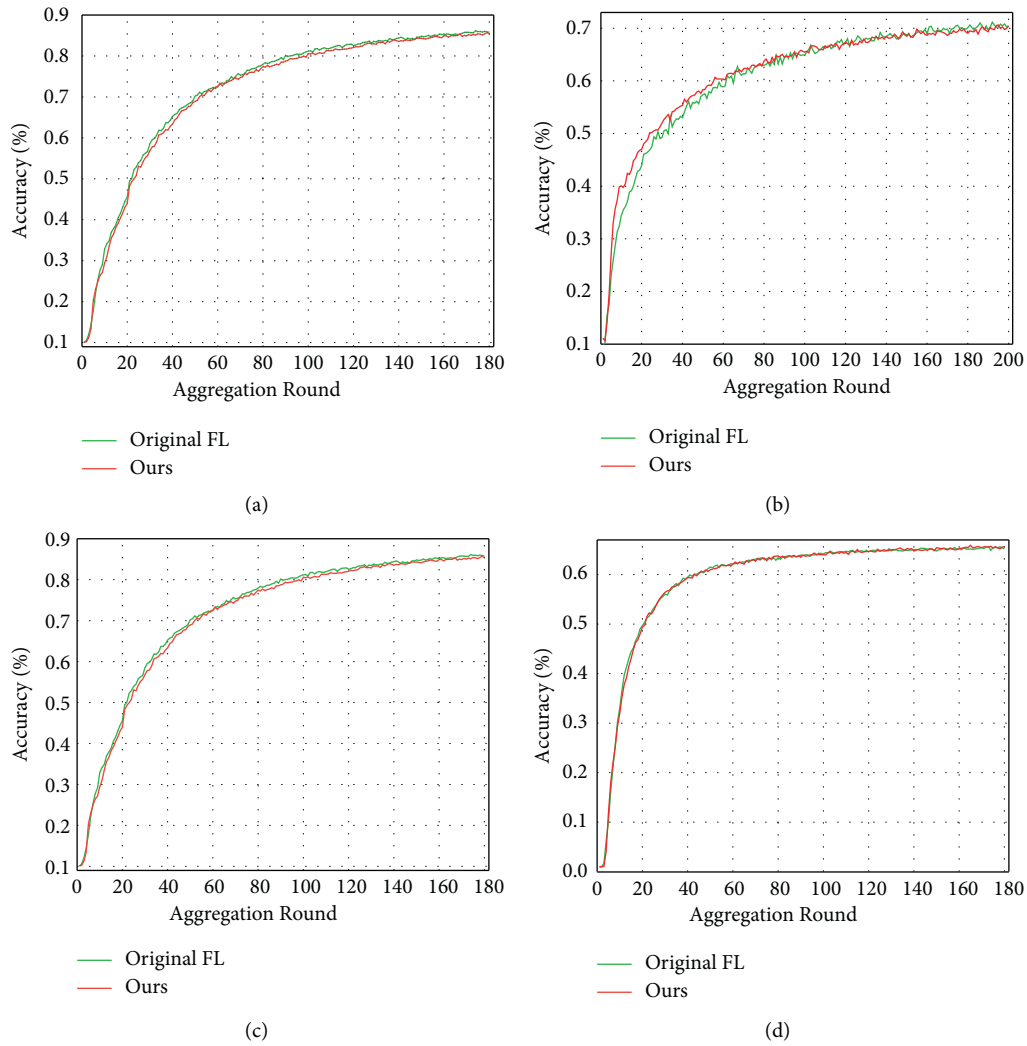

Figure 2: Comparison of performance of original FL and our framework. (a) Accuracy on FMNIST, (b) accuracy on CINIC-10, (c) accuracy on CIFAR-10, and (d) accuracy on CIFAR-100.

5.1. Experimental Setup. Our evaluation experiments are conducted on a Dell T7920 workstation with 1 Intel Xeon Silver 4210R CPU and 32 GB RAM. The OS is Ubuntu 18.04. We employ the FedML [26] in its standalone simulation computing paradigms to build the baseline framework. In both our and CRT-Paillier frameworks, the open-sourced python-Paillier [27] is adopted as the Paillier HE implementation. In our experiments, according to VFL [25] and CRT-Paillier [24], the gradients in their frameworks are of 32-bit length, while the gradients in our framework are clipped and quantized into 16-bit length with the dACIQ and Quantize function. The Paillier key size in both our framework and CRT-Paillier is set to 2048 bits, just as recommended in NIST [28].

5.2. Discussion of Results. Accuracy: we compared the performance of our framework with that of the original FL framework (Figure 2). For FMNIST, CINIC-10, CIFAR-10, and CIFAR-100, our framework can achieve an accuracy of 81.01%, 70.39%, 85.36%, and 65.60%, respectively, which is very close to that of the original FL, which has an accuracy of

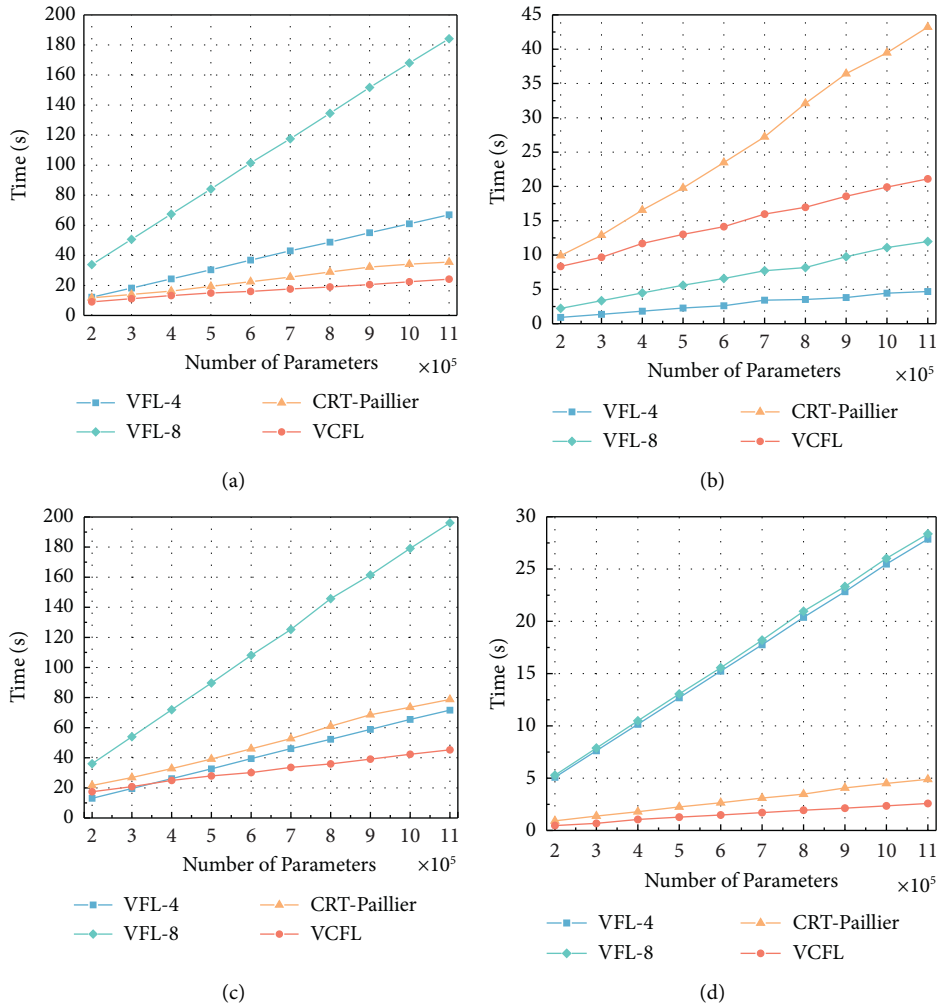

Figure 3: Comparison of time cost for VFL, CRT-Paillier, and our framework. (a) Encryption cost of a client, (b) decryption of a client, (c) total cost of a client, and (d) Aggregation cost of the server.

80.99%, 70.49%, 85.81%, and 65.23%, respectively. The results demonstrate that our framework can guarantee the privacy of the learned model without compromising its performance. In fact, our quantized training sometimes has better results. Prior quantization work has observed a similar phenomenon [29], where the stochasticity introduced by quantization can reduce overfitting, similar to the function of dropout layer [30].

Time cost, in Figure 3, we compared the time cost of our framework with that of VFL and CRT-Paillier. In VFL, the degree of the interpolation polynomials is set to $m = 4$ (VFL-4) and $m = 8$ (VFL-8), respectively. It is known that the security level and cost of VFL are higher with larger m. In CRT-Paillier, because the plaintext in Paillier encryption should be less than the modulus N, the maximum group size is 60 to avoid overflow error. While the gradient in our framework quantized into 16 bits, the maximum batch size is 120.

Figure 3(a) shows that the encryption cost per client of all frameworks increases linearly with the amount of parameters. The encryption costs of our framework are 9.06 s for 2×10^5 parameters and 24.17 s for 11×10^5 parameters, respectively. In addition, the encryption costs of CRT-Paillier are 11.73 s and 35.55 s, VFL-8 are 33.83 s and 184.15 s, and VFL-4 are 12.19 s and 66.99 s. Though both our framework and CRT-Paillier employ Paillier encryption to protect privacy, our framework needs fewer encryption operations and takes less time because its batch size is larger. For VFL, though the interpolation is very fast, the parameter splitting costs are expensive. Thus, our framework is more efficient than VFL-8 and VFL-4.

Figure 3(b) presents the decryption cost of all frameworks. The decryption costs of our framework are 8.35 s for 2×10^5 parameters and 21.10 s for 11×10^5 parameters. The encryption costs of CRT-Paillier are 9.91 s and 43.21 s, VFL-8 are 2.22 s and 11.87 s, and VFL-4 are 0.91 s and 4.68 s. Because our framework and CRT-Paillier utilize the Paillier scheme to protect privacy, which involves exponentiations and modular multiplications with large integers, the decryption costs of both our framework and CRT-Paillier are a little higher than VFL. But our framework takes less time than CRT-Paillier because of its larger batch size.

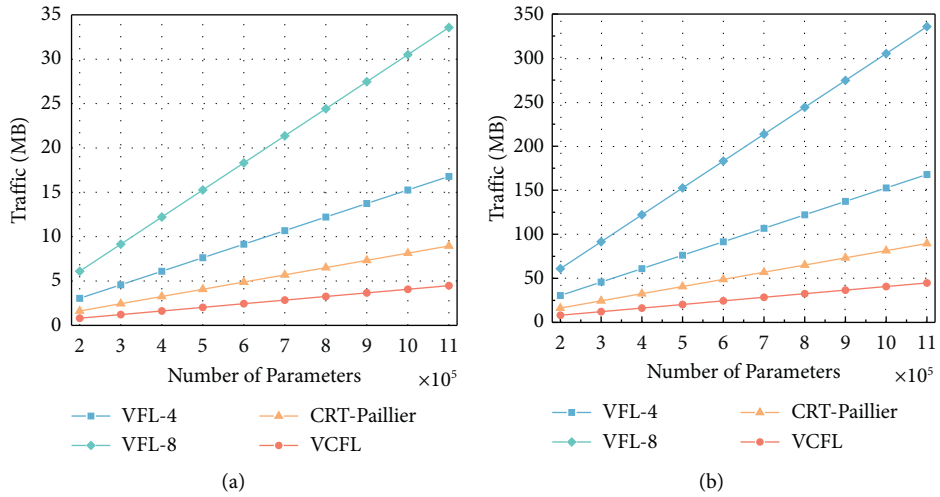

Figure 4: Comparison of communication cost of VFL, CRT-Paillier, and our framework. (a) Traffic of a client; (b) traffic of the server.

Figure 3(c) displays the total time cost of a client. From the figure we know that the total cost of a client of our framework is less than both VFL and CRT-Paillier, and the gap grows linearly with the number of parameters. We should note that, as giant companies or organizations prefer large deep learning models to boost their performance, our framework is more suitable for large AI models.

For the server, the overhead is caused by secure aggregation. Figure 3(d) shows the aggregation cost of the server. Because of larger batching size, our framework needs fewer sum operations on the ciphertexts. Thus, our framework is more efficient than CRT-Paillier and VFL for the server.

Communication cost: we compared the communication cost of our framework with that of VFL and CRT-Paillier in Figure 4. Since we simulate different FL frameworks in the standalone computing paradigms of FedML, we use the ciphertext size exchanged between the server and clients as the metric for communication cost. In VFL-4, each gradient is randomly split into 4 parameters. Though the authors keep the number of parameters the same as the original model by employing CRT to batch parameters, the size of the batched results grows accordingly. Thus, the ciphertext expansion rate for VFL-4 is 4. Similarly, the ciphertext expansion factor of VFL-8 is 8. In CRT-Paillier framework, every 60 gradients are grouped together and then are encrypted to get a 2048×2 bit-length ciphertext, the ciphertext expansion factor is approximately $2048 \times 260 \times 32 \approx 2.13$. Similarly, the ciphertext expansion factor of our framework is approximately $2048 \times 2120 \times 32 \approx 1.07$. Therefore, our framework is more communication efficient than VFL and CRT-Paillier.

6. Conclusion

In this paper, we have designed a preserving FL framework for IoT based on the homomorphic signcryption mechanism we designed. In our framework, each client can aggregate the gradients securely and verify the integrity of the aggregated results. Besides, our framework can also resist the collusion attacks between the server and at most $n - 2$ clients. Finally, experiments on four benchmark datasets show that our framework can protect the privacy and integrity of the learned model while guaranteeing its performance, and our framework is more efficient in computation and communication than existing similar frameworks. In future work, we will try to design more flexible privacy preserving framework that allows dynamic joining in and dropping out of clients.

Data Availability

MNIST dataset is available at https://yann.lecun.com/exdb/mnist/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Innovative Research Team in Engineering University of PAP (KYTD201805) and Natural Science Foundation of Shaanxi Province (2020JM-537).

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," 2017, https://arxiv.org/abs/1602.05629.

[2] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," 2017, https://arxiv.org/abs/1610.05492.

[3] V. Perifanis, G. Drosatos, G. Stamatelatos, and P. S. Efraimidis, "FedPOIRec: Privacy Preserving Federated POI Recommendation with Social Influence," 2021, https://arxiv.org/abs/2112.11134.

[4] T. Yang, G. Andrew, H. Eichner et al., "Applied Federated Learning: Improving Google Keyboard Query Suggestions," p. 02903, 2018, https://arxiv.org/abs/1812.02903.

[5] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156–165, 2019.

[6] G. Szegedi, P. Kiss, and T. Horváth, *Evolutionary Federated Learning on EEG-Data 8,* ITAT, 2019.

[7] C. Wang, C. Chen, Q. Pei, Z. Jiang, and S. Xu, "An information centric in-network caching scheme for 5g-enabled internet of connected vehicles," *IEEE Transactions on Mobile Computing*, vol. 1, p. 1, 2021.

[8] C. Chen, L. Liu, S. Wan, X. Hui, and Q. Pei, "Data dissemination for industry 4.0 applications in internet of vehicles based on short-term traffic prediction," *ACM Transactions on Internet Technology*, vol. 22, no. 1, pp. 1–18, 2022.

[9] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *Technical Reports Series*, vol. 715, 2017.

[10] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," 2017, https://arxiv.org/abs/1610.05820.

[11] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond Inferring Class Representatives: User-Level Privacy Leakage from Federated Learning," 2018, https://arxiv.org/abs/1812.00535.

[12] C. Liu, S. Chakraborty, and D. Verma, "Secure model fusion for distributed learning using partial homomorphic encryption," *Policy-Based Autonomic Data Governance*, vol. 11550, pp. 154–179, 2019.

[13] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning 15," in *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference*, pp. 493–506, U S A, July 2020.

[14] J. Ma, S.-A. Naas, S. Sigg, and X. Lyu, "Privacy-preserving Federated Learning Based on Multi-Key Homomorphic Encryption," 2021, https://arxiv.org/abs/2104.06824.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT '99*, J. Stern, Ed., vol. 1592, pp. 223–238, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[16] 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, ACM, Denver Colorado USA, October 2015.

[17] R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective," 2018, https://arxiv.org/abs/1712.07557.

[18] K. Bonawitz, V. Ivanov, B. Kreuter et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, ACM, Dallas Texas USA, October 2017.

[19] X. Ma, F. Zhang, X. Chen, and J. Shen, "Privacy preserving multi-party computation delegation for deep learning in cloud computing," *Information Sciences*, vol. 459, pp. 103–116, 2018.

[20] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Lecture Notes in Computer Science*, G. Goos, J. Hartmanis, J. van Leeuwen, and E. Biham, Eds., vol. 2656, pp. 416–432, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[21] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2020.

[22] W. Zheng, R. A. Popa, J. E. Gonzalez, and I. Stoica, "Helen: maliciously secure coopetitive learning for linear models," *2019 IEEE Symposium on Security and Privacy (SP)*, vol. 1, pp. 724–738, 2019.

[23] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Lecture Notes in Computer Science*, R. Safavi-Naini and R. Canetti, Eds., vol. 7417, pp. 643–662, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[24] X. Zhang, A. Fu, H. Wang, C. Zhou, and Z. Chen, "A privacy-preserving and verifiable federated learning scheme," in *Proceedings of the ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, Dublin, Ireland, June 2020.

[25] A. Fu, X. Zhang, N. Xiong, Y. Gao, and H. Wang, "VFL: A Verifiable Federated Learning with Privacy-Preserving for Big Data in Industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, 2022.

[26] C. He, S. Li, J. So et al., *FedML: A Research Library and Benchmark for Federated Machine Learning*, https://arxiv.org/abs/2007.13518, 2020.

[27] *DATA61, C. Python Paillier Library*, https://github.com/data61/python-paillier, 2013.

[28] E. Barker, L. Chen, A. Roginsky, A. Vassilev, R. Davis, and S. Simon, *Recommendation for pair-wise key establishment using integer factorization cryptography*, National Institute of Standards and Technology, Gaithersburg, MD, 2019.

[29] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients," 2016, https://arxiv.org/abs/1606.06160.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting 30," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 7507573, 12 pages
https://doi.org/10.1155/2022/7507573

Research Article

A V2P Collision Risk Warning Method based on LSTM in IOV

Ruoyu Pan,[1] **Lihua Jie**,[1] **Xinyue Zhang**,[1] **Shengli Pang**,[1] **Honggang Wang**,[1] **and Zhaoying Wei**[2]

[1]*Institute of Communication Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China*
[2]*Institute of College of Science, Xi'an Shiyou University, Xi'an, China*

Correspondence should be addressed to Honggang Wang; wanghonggang@xupt.edu.cn

Received 16 May 2022; Revised 13 June 2022; Accepted 25 June 2022; Published 31 July 2022

Academic Editor: Chen Chen

With the evolution of communication networks, the Internet of Vehicles (IOV) continues to accelerate the safe and rapid development of autonomous vehicles. Vehicle-to-Pedestrian (V2P) communication is a key technology in autonomous vehicles and a potential solution to realize collaborative intelligence between vehicles and pedestrians. However, the existing V2P communication early warning system does not consider the uncertainty of pedestrian trajectory, and the determination of the collision area is limited to a single point, resulting in an inaccurate system judgment and limited improvement of traffic efficiency. This paper designs a new autonomous-oriented V2P communication network architecture and completes a V2P communication early warning system based on Long Range (LoRa). A V2P anticollision model is established, and a new V2P collision risk early warning method is proposed. In this method, danger index is introduced into the early warning of collision between pedestrian and vehicle. The long short-term memory (LSTM) artificial neural network is used to predict the pedestrian's trajectory, so as to deduce the pedestrian-vehicle collision risk area when the pedestrian trajectory is uncertain. Meanwhile, the confidence probability is used to judge whether the pedestrian and vehicle are warned. The simulation shows that the V2P collision risk warning method proposed in this paper has good performance, which can accurately warn the pedestrian and vehicle under different vehicle speeds and Global Positioning System (GPS) positioning errors. At the same time, it reflects the characteristics of intelligence brought by using LSTM methods. Using the V2P communication early warning system based on LoRa to verify the experimental results show that when the GPS positioning accuracy is submeter level, the prediction accuracy is greater than 98%. The results of the proposed method show good performance and high detection rate.

1. Introduction

Automated system intelligent control in the automatic driving field is widely used. In recent years, autonomous driving technology has made considerable progress. Academia and industry are actively promoting it around the world, injecting new vitality into the research of autonomous vehicles technology [1, 2]. Self-driving automobiles rely on the collaboration of artificial intelligence, wireless communication, visual computing, radar, surveillance equipment, and global positioning systems, which can automatically and safely drive without manual intervention [3]. Relying on advanced vehicle intelligent control technology, self-driving automobiles not only liberate manpower but also reduce the human error of drivers and even reduce the incidence of traffic accidents to zero. In addition, self-driving automobiles can encourage people to carpool, greatly reduce the use of cars, increase the capacity of major roads, improve road capacity, and reduce air pollution caused by car exhaust. Therefore, the emergence of self-driving automobiles provides an ideal solution to problems such as traffic accidents, traffic congestion, energy consumption, and environmental pollution.

Vehicles and pedestrians are the main participants in traffic, so V2P is one of the core issues of autonomous driving [4–6]. It can even predict changes in traffic conditions. By exchanging the key information of vehicle and pedestrian directly, the efficiency of road traffic can be effectively improved and the intelligent road environment of

vehicle-road coordination can be realized. It is a key factor for the automobile intelligence.

Automobile intelligence requires vehicles to have the ability to communicate with road users. Automobile intelligence requires accurate acquisition and analysis of the following information to predict the future behavior of road users and generate corresponding responses, including user behavior information such as route, location, speed, and direction. This interaction is crucial between vehicles and pedestrians. The movement of vehicles, such as cars, trucks and buses, is governed by clear rules and environmental conditions. However, as the main participants of road traffic, pedestrians are not considered because of their complex, dynamic, and random behavior [4]. This paper considers the uncertain trajectory factors of the pedestrian and designs a V2P early warning system to effectively improve the accuracy of the warning system.

Pedestrians, cyclists, and motorized two-wheeler operators are called Vulnerable Road Users (VRUs) [7]. The protection of VRUs is a common topic in V2P [8]. Advanced Driver Assistance System (ADAS) uses sensor technology [9–11], the far-infrared method [12], computer vision [13], and a combination of methods [14] to detect pedestrian location information.

Compared with ADAS, the Internet of Vehicles (IOV) can detect the relative motion state of vehicles and pedestrians and perceive potentially dangerous situations more accurately in the case of None Light of Sight (NLOS), to achieve cooperative safety to ensure safer and more efficient transportation. Wireless communication technology is the cornerstone of the development of the IOV, which has been extensively studied by researchers. Different wireless communication technologies have different advantages and disadvantages. As shown in Table 1, LTE towers can cover 16 KM, but their energy consumption is very large and the Doppler effect caused by high carrier frequency is difficult to eliminate [11, 15]. Millimeter-wave (MmWAVE) has abundant bandwidth and short wavelength. At the same time, MmWAVE is an emerging technology that faces many challenges. For example, it is easy to be blocked by obstacles due to the short wavelength, so relayed forwarding is required, but the core problem of rapid selection of the optimal relay has not been solved [16, 17]. Wi-Fi has a high data transmission rate, which usually has a communication range of 100–150 meters. Driving at a speed of 50 km/h can meet the requirements. However, in the suburban areas, with the typical speed of 100 km/h, the requirements cannot be met because drivers have less reaction time to take action. Moreover, Wi-Fi wireless communication transmission cannot achieve low-latency bidirectional communication [18].

While LoRa strikes a balance between transmission range and data transfer rate. A LoRa base station can cover 15 km with data rates ranging from 300 bps to 37.5 kbps, depending on the spreading factor (SF) and channel bandwidth [19]. In addition, the LoRa base station can accommodate 1 million terminals, and simultaneously receive multiple transmissions using different spreading factors. When using a spreading factor of 12, the transmit power is 14 dB, and the transmission to the base station with a distance of 420 m is 96.7% [20]. Simultaneous reception of multiple spreading factors (between 7 and 12) creates a third possibility beyond time and frequency. In addition, the VRUs protection system of V2P communication requires low-power, low-cost, and small wireless communication equipment to ensure low-latency and high-reliability transmission of data under high bandwidth within the transmission range. Therefore, wearable devices with these characteristics of LoRa wireless communication technology can meet the requirements of VRUs protection system for V2P communication.

Regarding the establishment of anticollision models, many warning systems have been proposed in the existing literature. In [21], a V2X coordination system based on cellular and 802.11 p radio: SafeNav android application is proposed, the interface has set a predetermined collision area, if multiple traffic participants enter the collision area, visual and audible alerts are generated, and the color of the traffic participant changes to red and beep sound is emitted. In [5], a mobile application that supports pedestrians and vehicles was proposed by Hussein et al., the screen is active when the user interacts with the mobile phone to detect the position coordinates of the mobile phone. The system calculates the collision point and determines whether there is a collision. Increase the VRU's visual situational awareness of the location near the automatically and manually controlled vehicles in the form of user-friendly operation. All the above studies require manual intervention and are not highly automated, which may bring certain security risks [22, 23]. Liu et al. developed a warning system POFS, which uses IEEE 802.11 p and Wi-Fi as communication channels to warn pedestrians by predicting collisions and issuing alerts [24]. Ho et al. designed a WIFI-based anticollision system, which encapsulates longitude, latitude, direction, and speed in a service set identifier (SSID) for transmission, and judges whether there is a collision risk by parsing the SSID of vehicles and pedestrians [25]. The warning system based on dedicated short-range communication (DSRC) and Wi-Fi has relatively high accuracy, but the applicable distance is limited. Xiog et al. proposed a graded warning system, which divides the warning into two types: danger warning and emergency braking according to the position coordinates of the pedestrian crossing road [26]. Bastani Zadeh proposed a three-level collision warning system, in which the first stage is used to activate the system to deal with various dangerous situations, the second stage is to extract important features to evaluate the collision risk by using fuzzy reasoning engine, and the third stage is to divide the warning alarms into low, medium, and high risk [27]. The prediction of the future trajectory of pedestrians is necessary to improve the accuracy of the anticollision system because pedestrians, as major participants in road traffic, may change direction suddenly according to external factors such as objects and vehicles [28]. Ideally, a pedestrian's walking destination determines its trajectory; however, in the real world, its destination is not always known. Therefore, it is necessary to learn the behavior characteristics of pedestrians based on the past time trajectory sequence. However, the above methods do not

Table 1: Technical parameters of wireless communication.

	Maximum transmission distance	Spectrum	Energy consumption
LTE	16 km	1.25~20 MHz	High
MmWAVE	3.8 km	6~100 GHz	High
WiFi	150 m	2.4 GHz	Low
Zigbee	100 m	2.4 GHz	Low
Bluetooth	50 m	2.4 GHz	Low
LoRa	15 km	433/868/915 MHz	Low

consider the uncertainty of pedestrian trajectory, and the methods for judging collision are limited to whether there is a collision point, which should be an area, rather than a single point. In order to solve the uncertainty problem, introducing the neural network is a suitable choice [29–31]. For the description of the collision area, previous studies mostly used the maximum likelihood solution or the difference between the major and minor axes of the ellipse are used to describe the uncertainty in the position [32, 33].

This paper proposes the early warning system, which adopts LoRa for data transmission and GPS for data acquisition. The main contributions are in two aspects: 1. In the V2P communication problem, the model not only considers whether there is a collision risk from the perspective of strict physical calculation but also introduces a risk index to measure the degree of risk to reduce the judgment error. 2. The algorithm takes the uncertainty of the pedestrian trajectory into the model. In this process, the future uncertainty trajectory of the pedestrian is represented as a collision area given by a confidence ellipse, and the corresponding confidence probability is given.

The remainder of this paper is organized as follows: Section 2 introduces the V2P communication network architecture based on the LSTM and the test environment used in this paper. Section 3 introduces the probabilistic prediction method and the algorithm for judging whether the warning is given, which effectively solves the V2P anticollision problem while considering pedestrian trajectory prediction. In Section 4, the simulation analysis of the algorithm is given, and the real scene tests are carried out by using LoRa equipment to predict pedestrian behavior. Based on the pedestrian's future trajectory, analyze the collision area and collision probability of the pedestrian under different motion trajectories. Finally, the conclusion of this paper is given in Section 5.

2. Autonomous-Oriented V2P Communication Network Architecture

2.1. V2P Communication Network Architecture. This section proposes a three-layer LoRa multiparty collaborative network architecture, including a traffic cloud data center, regional platforms, and terminals, as shown in Figure 1. In addition to data transmission, the network also needs to assist in collaborative computing. The gateway, regional platform, and traffic cloud data center adopt 4G, 5G communication, which evolves to a higher level with the network updates.

The traffic cloud data center realizes the storage and analysis of massive terminals data. The regional platform (including the LoRa network server and edge computing) processes packets received by LoRa gateways, pedestrian terminals, and vehicle terminals in real time. The computational decision unit in edge computing is based on LSTM, which makes the system autonomously and predicts the collision risk more accurately. Because the LSTM enables the V2P system to better solve the problems caused by various pedestrians and the uncertainty of the environment. Most of the functions need to be implemented in the regional platform, checking whether the terminal is online, filtering redundant packets, etc. The terminal is responsible for the collection and transmission of location data. The vehicle terminal includes data collection, data processing, and safety decision-making. The pedestrian terminal and vehicle terminal can interact directly. The data from terminals also can be forwarded through the gateway. This paper does not discuss the vehicle terminal, focusing on the pedestrian terminal.

2.2. Network Server, Gateway, and V2P Pedestrian Terminals. The functions of the network server include communication control, protocol processing, device management, and network maintenance of the LoRa network. The network server helps users register, manage, and monitor LoRa devices, including terminals and gateways. It can also encrypt, decrypt and process application layer payloads. Meanwhile, it can support the application of various encryption or encoding methods to ensure data security and improve transmission efficiency.

The gateway implements the topological link of the LoRa network by receiving data from the terminal and forwarding it to the network server or forwarding the data message from the network server to the terminal.

Pedestrian terminals are used to collect data and send it to the network server through the gateway, or execute commands and tasks from the network server.

Pedestrian terminals and gateways are the foundation of the LoRa network. The gateway can interact with LoRa terminals, collect data such as signal-noise-ratio (SNR), received-signal-strength-indicator (RSSI), and transmit them to the base station for analysis. The terminals can implement different functions according to different application requirements. The gateway uploads all uplink packets to the network server. On the contrary, in the downlink, the LoRa gateway executes the transmission requests from the network server. The gateway usually relays

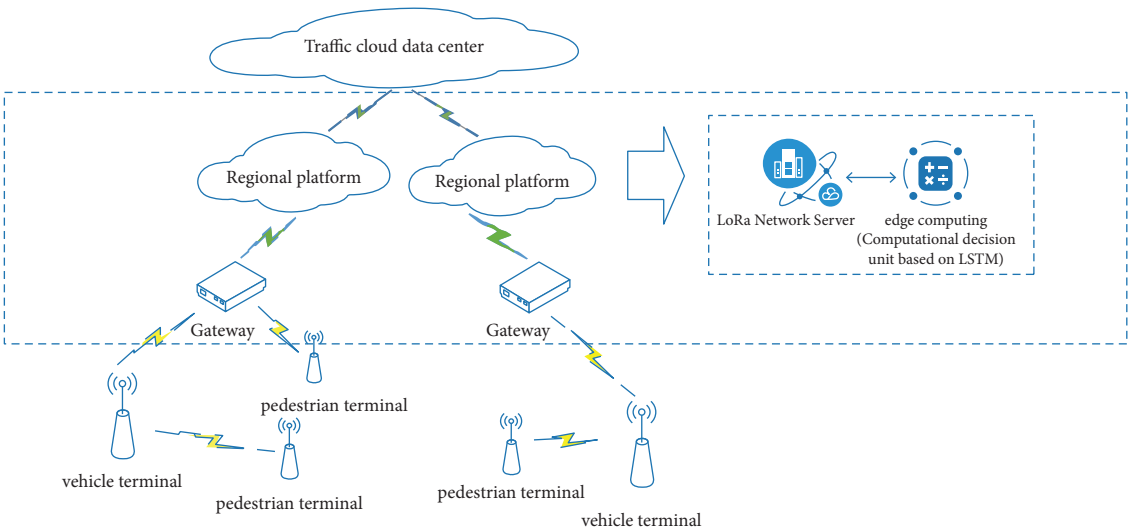

Figure 1: LoRa network architecture under the V2P scenario.

the received uplink/downlink packets without processing the payload data. Due to its low-power consumption and long-distance transmission characteristics, the LoRa network provides more reliable terminal location data for the judgment of the V2P communication warning system to improve the accuracy of the system.

2.3. Test Environment. The LoRa system used in the test consists of a gateway and a terminal. The gateway receives and processes messages sent by the terminal. Based on the LoRa network, a test terminal for pedestrians is specially developed. To obtain the position, speed, and direction of the pedestrian and the vehicle, the pedestrian and the vehicle are equipped with GPS to locate them and predict future trajectories based on the pedestrian's past data.

The pedestrian terminal of the test is composed of a LoRa module and GPS with submeter accuracy. The microcontroller unit (MCU) adopts an STM32 chip, the RF module adopts a LoRa RFM98 chip, and the GPS operating frequency is 1 Hz. The hardware frame and physical diagram of the pedestrian terminal are shown in Figure 2. The GPS module and the MCU communicate one way through Universal Synchronous/Asynchronous Receiver/Transmitter (USART). The LoRa module communicates two way with the MCU through the SPI (Serial Peripheral Interface) bus. The hardware frame and physical diagram of the gateway are shown in Figure 3, which consists of four groups of the same module. The gateway can support simultaneous communication with four different frequency bandwidths, in which the LoRa module conducts two-way communication with the MCU through the SPI bus.

The gateway is located on the top of the fifth floor of the building of Xi'an University of Posts and Telecommunications. The pedestrian wears the terminal and walks at a constant speed on the campus. The test path is divided into linear path and nonlinear path.

The paths of the pedestrian wearing the terminal are shown in Figure 4. The actual measurement scene is selected in Xi'an University of Posts and Telecommunications under the Gaode map. The gateway collects the data and sends it to the network server to map the data on the Gaode map. As shown in Figure 4, the blue is the pedestrian walking trajectory, the left is the linear trajectory, and the right is the nonlinear trajectory.

3. Probabilistic Prediction

3.1. Model of Pedestrian-Vehicle Anticollision. This paper analyzes and models the scene as shown in Figure 5, both the pedestrian (X_P, Y_P) and the vehicle (x_v, y_v) keep moving at a constant speed. The pedestrian is located at the driver's NLOS position and the trajectory is uncertain. The vehicle trajectory tends to be a straight line, and a collision occurs near $p(x, y)$.

Assuming that the pedestrian-vehicle path has a certain angle, the red area $p(x, y)$ in Figure 5 is the area where a collision may occur. The model building will be divided into two steps, minimum safety distance and judgment conditions.

The t_{tot} represents the reaction time for the driver to receive a collision warning and take action.

$$t_{tot} = t_{cesta} + t_{comm\ d} + t_{react} + t_{resp}, \tag{1}$$

where t_{cesta} is communication connection delay, t_{commd} is information transmission delay, t_{react} is driver reaction time, and t_{resp} is the time that driver takes action.

3.1.1. Minimum Safety Distance. The minimum safety distance of vehicle anticollision is

$$d = S_1 + S_2 = v_v \cdot t_{tot} + \frac{v_v^2}{2g\mu}, \tag{2}$$

where t_{tot} represents the reaction time for the driver to receive a collision warning and take action. The speed of the vehicle is v_v, s_1 is the distance passed by the driver during the

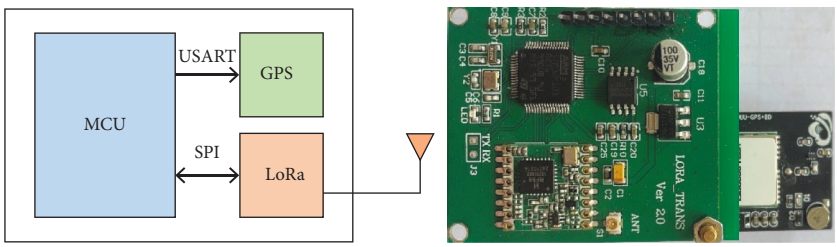

Figure 2: Pedestrian terminal (the left is the hardware frame, and the right is the physical diagram).

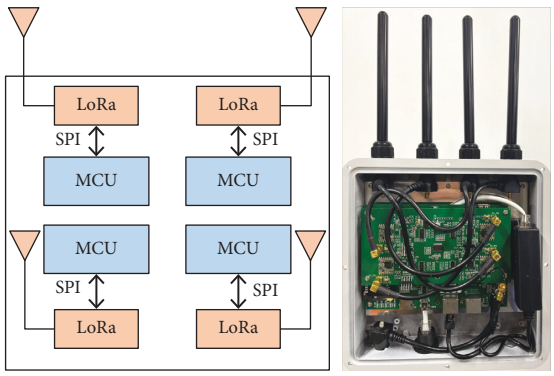

Figure 3: Gateway (the left is the hardware frame, and the right is the physical diagram).

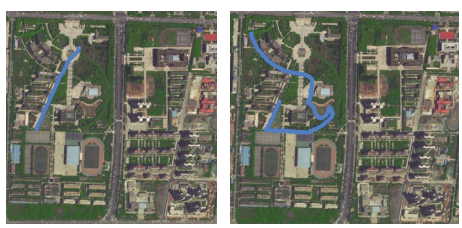

Figure 4: Pedestrian path map measured by the pedestrian wearing terminal (at Xi'an university of posts and telecommunications).

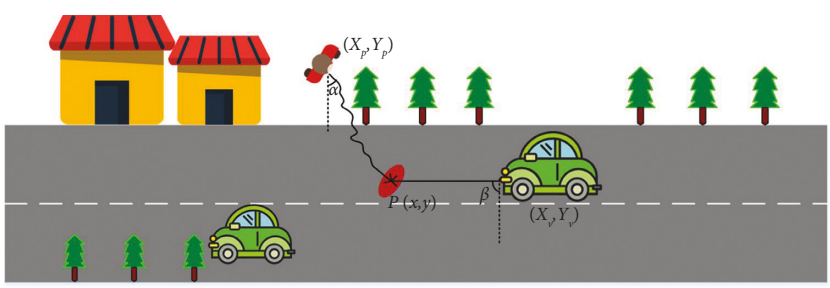

Figure 5: Pedestrian-vehicle collision scenario.

period when the driver reacts and takes action, s_2 is the braking distance of the vehicle, $g = 9.8\,\mathrm{m}/s^2$, and μ is the friction coefficient.

3.1.2. Judgment Conditions. Obtain two sets of position information (x_1, y_1), (x_2, y_2) for the pedestrian and (x_3, y_3), (x_4, y_4) for the vehicle and determine whether there is a collision point $p(x, y)$ through (3).

$$\begin{aligned} x &= ((x_3 - x_4) * ((y_1 - y_2) * x_1 - (x_1 - x_2) * y_1) \\ &\quad - \frac{(x_1 - x_2) * ((y_3 - y_4) * x_3 - (x_3 - x_4) * y_3)}{((y_1 - y_2) * (x_3 - x_4) - (y_3 - y_4) * (x_1 - x_2))}, \\ y &= ((y_3 - y_4) * ((y_1 - y_2) * x_1 - (x_1 - x_2) * y_1) \\ &\quad - \frac{(y_1 - y_2) * ((y_3 - y_4) * x_3 - (x_3 - x_4) * y_3)}{((y_1 - y_2) * (x_3 - x_4) - (y_3 - y_4) * (x_1 - x_2))}. \end{aligned} \tag{3}$$

The time TTC can be calculated through (4) and (5).

$$TTC = \left|TTC_V - TTC_P\right| < \delta, \tag{4}$$

$$\delta = \frac{(W_{car} + d)}{V_v}, \tag{5}$$

where TTC_P is the time that the pedestrian maintains the current speed to the collision point $p(x, y)$, TTC_v is the time that the car maintains the current speed to the collision point $p(x, y)$, W_{car} is the width of the vehicle, and the arrival time difference is δ. δ represents the preset threshold, which determines whether the road condition can be considered dangerous.

Because TTC (the time difference between the pedestrian and the vehicle arriving at the collision point) is not the only factor in determining the potentially dangerous road conditions, the S danger index is also taken into account. This danger index is also considered to decide whether to issue a warning. The danger index S is defined by

$$S(TTC, d) = \begin{cases} \dfrac{1}{2\pi\sigma^2}\exp\left(-\dfrac{1}{2}\dfrac{TTC_V^2}{\sigma^2 d}\right), \text{collision point } p \text{ exists}, \\ 0, \text{collision point } p \text{ not exists}, \end{cases} \tag{6}$$

where σ is the accuracy of GPS. According to the typical vehicle speed of 40 km/h, the threshold is divided into two types, 0.01 and 0.1. When the vehicle speed is greater than 40 km/h and S is smaller than 0.01, a warning is given, otherwise, a warning is given when S is smaller than 0.1.

3.2. Trajectory Prediction. Each pedestrian has different movement patterns: speeds, accelerations, and gaits. Therefore, it is necessary to build a model to understand and learn these specific characteristics of pedestrian motion, so as to predict the future trajectories of pedestrians. Neural network is considered to be an effective method to deal with real-time problems, which has been widely used in machine control [34,35] to improve the degree of machine automation [36]. When dealing with the long-term dependence of the input sequence, the traditional RNN network will produce the problem of gradient descent or gradient disappearance. In order to solve this problem, LSTM was proposed by Hochreiter and Schmidhuber [37], and then it was widely used to solve the problem of time series prediction.

LSTM has three gates (update i_t, forget f_t, and output y_t), block input, and memory cells c_t. The output of the block is repeatedly connected back to the input of the block and all gates. The data goes through the horizontal line of the highest point in Figure 6, which is called the cell state. The memory cell uses a function tanh whose value range is $[-1, 1]$ as the activation function. The forgetting gate controls whether the information in the memory cell of the previous time step is transmitted to the current time step, and the forgetting gate and the memory cell achieve better control of information flow.

In Figure 6, f_t, g_t, i_t, c_t, o_t are

$$\begin{aligned} f_t &= \sigma\left(\overline{f}_t\right) = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right), \\ g_t &= \tanh\left(\overline{g}_t\right) = \tanh\left(W_{xg}x_t + W_{hg}h_{t-1} + b_g\right), \\ i_t &= \sigma\left(\overline{i}_t\right) = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right), \\ c_t &= c_{t-1} \odot f_t + g_t \odot i_t, \\ o_t &= \sigma\left(\overline{o}_t\right) = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right). \end{aligned} \tag{7}$$

In this paper, the LSTM neural network is used to make cyclic multistep predictions for the future trajectory of the pedestrian. Among them, the activation function is relu, the loss function is MSE, the optimizer is Adam, the metric is accuracy, and the epoch is 100. GPS location information (latitude and longitude) of the pedestrian is used as the input of the LSTM. The cross-validation method is used for machine learning to improve the accuracy of machine learning and avoid overfitting and underfitting. The output is the location information, mean, and covariance of the pedestrian's future trajectory.

3.3. Collision Probability. The mean and covariance of the pedestrian's future trajectory can be obtained by LSTM for trajectory prediction, which can indicate the predicted area that the car needs to avoid, called the collision area R. For the ith pedestrian's movement, it is assumed that the probability of its future location at a certain position depends on the collision area R_i at time t, and R_i is based on the mean $\mu_i(t) = [\mu_x, \mu_y]^T$ and covariance $\sum_i(t) = \mathbb{R}^{2\times 2}$ at time t. For convenience, the variable t will be ignored in the following description of this chapter, so that we can assume that the future position of the ith pedestrian moving at time t can be expressed as $P_{i,f} \sim N(\mu_i, \sum_i)$.

Remark 1. Since the covariance matrix $\sum_i$ is a real symmetric, positive semi-definite matrix, and the eigenvalues are real numbers, there is an orthogonal matrix Q_i composed of the eigenvectors $\sum_i$. The eigenvalues of the covariance matrix $\sum_i$ can be decomposed into

$$\sum_i = Q_i \Lambda_i Q_i^T, \tag{8}$$

where $\Lambda_i = di\ ag(\lambda_j)$, $j = 1, 2$, λ_j sort in descending order, $\lambda_1 \geq \lambda_2$, j represents a dimension in the environment.

In this section, two key results (on the construction of the ellipse and the calculation of scaling factors) need to be explained, which are crucial parts of the model prediction.

Lemma 1. *An ellipse can be constructed from a transformation of the unit circle by first stretching it along each axis with the ratio* $\sqrt{\lambda_i}$, *then rotating the ellipse through* Q_i, *and finally translating the center* m_i *of the distribution to the origin according to the following inverse Mahalanobis transformation:*

$$R_i = Q_i \Lambda_i^{1/2} Q_i^T u + \mu_i, \tag{9}$$

where $u \sim N(0, I_2)$ *is the unit circle of a two-dimensional normal distribution. In this paper,* I_2 *denotes the identity matrix with the size of* 2×2.

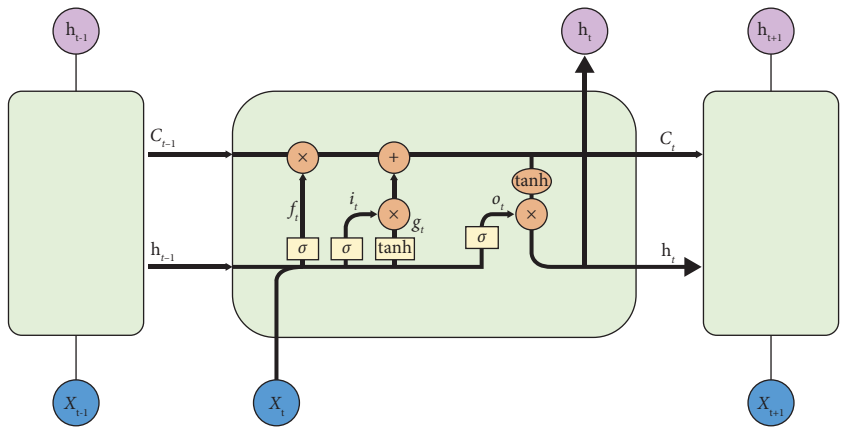

Figure 6: Repeating modules in LSTM (i_t is update gate, f_t is forget gate, and y_t is output gate).

Proof. Mapping a unit circle by the square root of the covariance matrix, $\sum_i^{1/2}$ determines an ellipse whose principle semiaxes depend on the eigenvalues of this matrix, and whose direction is related to the corresponding eigenvectors. To represent this ellipse graphically, the Mahalanobis transform eliminates the correlation between variables and normalizes each variable by its variance. Therefore, an ellipse can be constructed from the transformation of the unit circle, according to the inverse Mahalanobis transformation (9).

Next here, the scaling factor is calculated and the confidence probability that the collision area is an ellipse is derived. □

Lemma 2. *The scaling factorrcan be calculated approximately by*

$$F(r) = \mathrm{P}(r) - \Phi,$$

$$\widetilde{F}(r) = \widetilde{\mathrm{P}}(r) = \sqrt{\frac{2}{\pi}}\exp\left(-\frac{r^2}{2}\right) + \frac{\exp\left(-\left(r^2/2\right)\right)}{\sqrt{2}\Gamma(3/2)}\left(r^2 - 1\right). \tag{10}$$

Proof. Based on Lemma 2, the square of the Mahalanobis distance (scaling factor r) of the possible positions $P_{i,f}$ to its mean μ_i can be calculated by

$$r^2 = \left(P_{i,f} - m_i\right)^T \sum_i^{-1}\left(P_{i,f} - m_i\right). \tag{11}$$

Bring into the above equation through (8) and (9), the magnified ellipse with scaling factor r can be obtained, relying on the chi-square χ^2 distribution with the degree of freedom $\sigma = 2$, as shown in

$$\Pr\left(r^2 \le \chi^2_{\sigma=2,p}\right) = \Phi, \tag{12}$$

which can be expressed as

$$\Pr\left(\left(a_{i,f} - \mu_i\right)^T \sum_i^{-1}\left(a_{i,f} - \mu_i\right) \le r^2\right) = \Pr\left(u^T u \le r^2\right). \tag{13}$$

The confidence probability for an arbitrary ellipse with any factor r can be calculated, as shown in

$$\begin{aligned} P(r) &= \Pr\left(u^T u \le r^2\right) \\ &= \int\int 2\pi^{-(3/2)} \exp\left(\frac{u_1^2 + u_2^2}{2}\right) du_1 du_2 \\ &= erf\left(\frac{r}{\sqrt{2}}\right) - \left(\frac{r}{\sqrt{2}}\right)\frac{\exp\left(-\left(r^2/2\right)\right)}{\Gamma(3/2)}, \end{aligned} \tag{14}$$

where

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp\left(-t^2\right)\mathrm{d}t, \tag{15}$$

is the standard error function and Γ is the gamma function. The scaling factor $r = 2.7955$, 3.3682 when the corresponding confidence probabilities $\mathrm{P}(r) = 95\%$, 99%, respectively. □

3.4. Algorithm Description. The system obtains the location information of pedestrians and vehicles in real time, uses LSTM to predict the future trajectory of the pedestrian, and obtains the probability density function of the pedestrian trajectory, which obeys a multidimensional normal distribution function. It preliminarily infers the collision area between the pedestrian and the vehicle, and finally uses the confidence probability to determine whether to give a warning to the pedestrian and the vehicle.

In practice, in order to better describe the real data D_{real}, adding a normal distribution function to ideal data D to approximate real data D_{real}

$$D_{\text{real}} = D + N(\mu, \sigma). \tag{16}$$

The system algorithm description is as follows (Algorithm 1):

4. Result Analysis

4.1. Influencing Factors of System Collision Risk Warning Time. The influence of GPS positioning error, vehicle speed, and other uncertain factors on the anticollision

Input: Two sets of GPS data for pedestrian and vehicle
Output: A risk of collision occurs or not
(1) Acquire 2 sets of GPS data continuously (pedestrian and vehicle) (X_{p1}, Y_{p1}), $(X_{p2}, Y_{p2})(X_{v1}, Y_{v1})(X_{v2}, Y_{v2})$
if there is a collision point $p(p_x, p_y)$
if $TTC = |TTC_V - TTC_P| < \delta$ and $\delta = (W_{car} + d)/V_v$
if the confidence probability corresponding to the collision area generated by the predicted future trajectory $P(r) > 95\%$
Output: A risk of collision will occur.
(2) Calculate the direction angle of pedestrian and vehicle α, β
if $0 < |\alpha| < 90^o$ or $0 < |\beta| < 90^o$
$\max\{X_{p2}, X_{p2}, X_{v1}, X_{v2}\} > P_x$, stop getting data
else
$\min\{X_{p2}, X_{p2}, X_{v1}, X_{v2}\} < P_x$, stop getting data

Algorithm 1: Collision warning.

system model is studied through the simulation. The simulation scene is shown in Figure 5. The pedestrian and the vehicle start at the same time and keep running at a uniform speed. It is assumed that the collision occurs 200 seconds after the pedestrian and vehicle movement. In order to be closer to the actual situation, the error is added to the position data during simulation. This paper assumes that the error obeys the normal distribution. The pedestrian speed is 1 m/s, and the driving speed is variable. The reaction time t_{tot} for the driver receiving the collision warning and taking action is 0.83 s [38], the friction coefficient μ is 0.8, and the vehicle length is 5 m.

According to the idea of control variables, by setting different GPS positioning errors and driving speeds, the change of the warning time of pedestrian-vehicle collision risk is analyzed as follows.

4.1.1. The Relationship between Driving Speed and Collision Risk Assessment Time. Control the GPS positioning error $\sigma = 1$, i.e., the error is 1 m, and the driving speed is changed. The changing relationship of collision risk assessment time is compared when the driving speed is 30 km/h, 40 km/h, and 60 km/h. The difference of collision risk assessment time under the different vehicle speeds is shown in Figure 7. The horizontal axis is the time in seconds. The vertical axis is the time difference between the vehicle and the pedestrian arriving at the collision point, and the unit is second. The coordinates marked in Figure 7 are the warning time and *TTC*. The higher the speed, the earlier the collision warning. In Figure 7(a), when the driving speed is 30 km/h, the time difference between the pedestrian and the vehicle arriving at the collision point is 1.96 s, and the system sends out a danger warning in 198 s, i.e., 2 seconds in advance. In Figure 7(c), when the driving speed is 60 km/h, the time difference between the pedestrian and the vehicle arriving at the collision point is 2.19 s, and the system issues a danger warning in 195 s. When the positioning error is 1 meter, a warning is given at typical speeds within 2 to 5 seconds before the collision point arrives, and the results show that the algorithm is feasible.

4.1.2. The Effect of GPS Positioning Error on the Collision Risk Assessment Time. Change the GPS positioning accuracy when the vehicle speed is constant and compare the changing relationship of the collision risk early warning evaluation time when the positioning accuracy is 1 m and 10 m. Under different GPS positioning errors, the collision risk assessment time is different as shown in Figure 7. The positioning errors in Figures 7(a)–7(c) are 1 m, and the positioning errors in Figures 7(d)–7(f) are 10 m. The greater the GPS positioning error, the earlier the collision warning. In Figures 7(a), 7(d), when the vehicle speed is 30 km/h, the time difference between the pedestrian and the vehicle reaching the collision point is 1.96 s when the positioning accuracy is 1 m and 10 m, and the system issues a danger warning in 190 s. When the vehicle speed is 60 km/h, the time difference between the pedestrian and the vehicle reaching the collision point is 2.19 s, and the system sends a hazard warning in the 180 s. The results show that the positioning error does not affect the time difference of arrival, but only affects the warning time. Because the danger index should be considered in the warning, and the danger index is related to the positioning error.

4.2. Collision Area and Corresponding Confidence Probability. Based on the anticollision model, when v_V is 30 km/h, 40 km/h, or 60 km/h, the warning time is obtained when the GPS positioning error $\sigma = 1$ is used. 500×4 and 1250×2 sample data are used for straight line and nonstraight line, respectively. The sample data are predicted by LSTM. After the model training is completed, the prediction takes only 1~2 seconds, which can meet the application requirements. This is because only one set of data is predicted per round, so the prediction time is faster. The units in Tables 2 and 3 are meters. In this paper, 10.2.1 represents the use of ten data to predict the pedestrian position after two seconds, i.e., the first number 10 is the number of data used by the LSTM, the second number 2 is the position data predicted by the LSTM after 2 seconds, and the third number 1 represents the 1 position data predicted by the LSTM. If the pedestrian keeps moving in a straight line, using 6 seconds or 10 seconds to predict the position offset after multiple steps is 0 meters, but for nonlinear situations, it is better to use ten seconds to predict the position

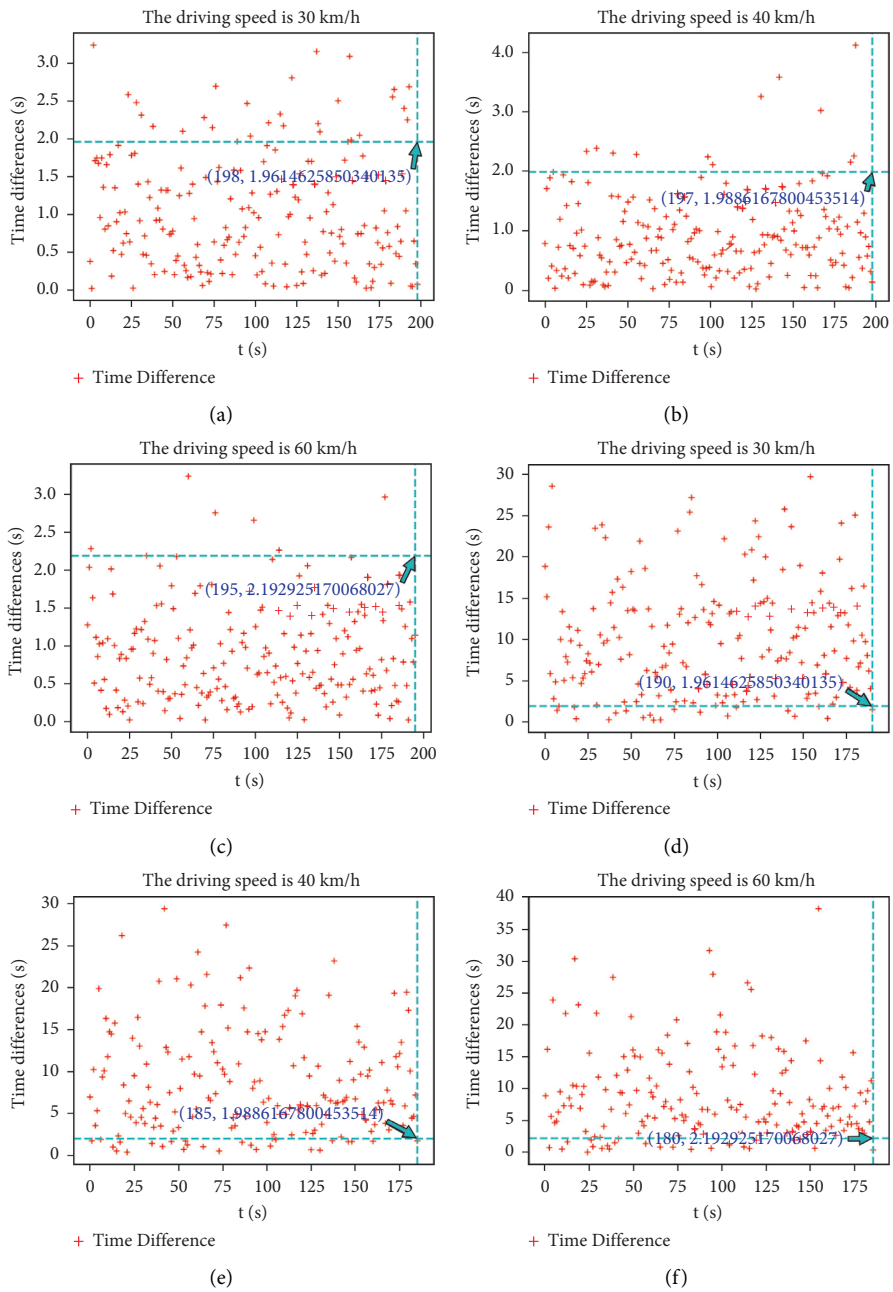

Figure 7: The influence of system positioning accuracy on the pedestrian collision risk detection (a, b, c positioning error is 1 m, d, e, f positioning error is 10 m).

Table 2: Pedestrian future trajectory prediction error.

	Linear (m)	Nonlinear (m)
6.5.1	0.0	11.3
6.2.1	0.0	7.6
10.5.1	0.0	1.5
10.2.1	0.0	0.6

Table 3: Collision area and corresponding confidence probability.

	Linear (m) (%)	Nonlinear (m) (%)
6.5.1	98.61	99.39
6.2.1	99.57	99.39
10.5.1	99.12	99.2
10.2.1	99.57	99.73

after multiple steps. This is because when the pedestrian trajectories are nonlinear, machine learning needs to learn more features, so more data are required. At the same time, the accuracy of the algorithm is also verified.

Here, a nonlinear prediction trajectory in 6.5.1 is used as an example to describe the prediction performance. The collision area of the uncertainty position prediction method is described in Figure 8. If the vehicle continues in its current

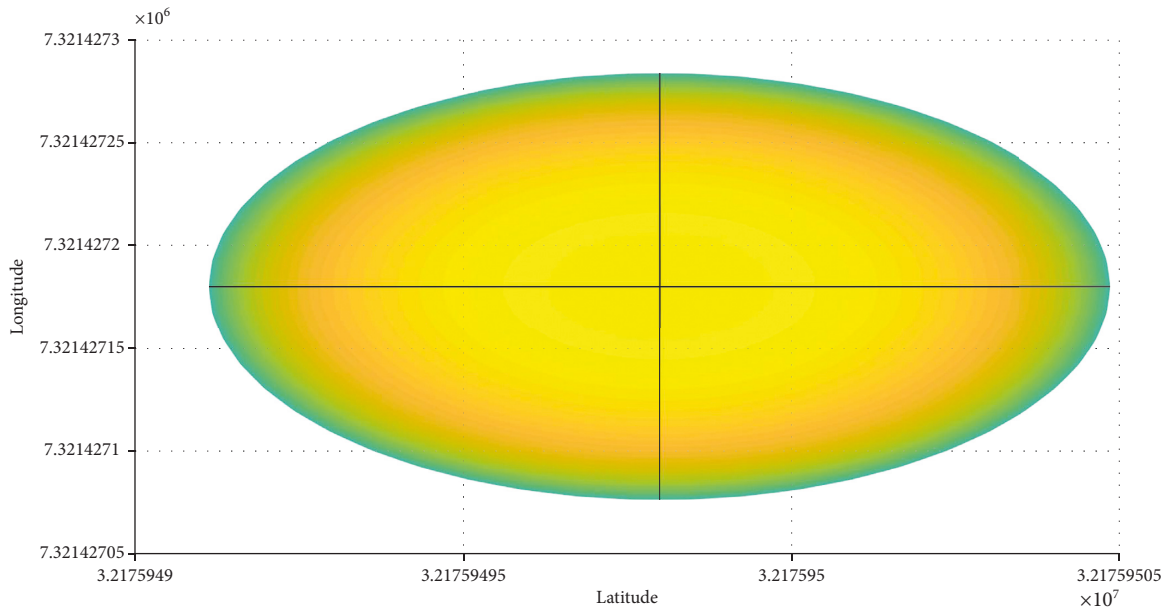

Figure 8: The collision area corresponding to a predicted trajectory (the horizontal and vertical coordinates are the latitude and longitude after mapping the Miller coordinate system to the plane coordinate system).

state, it will collide with the pedestrian in the collision zone with a probability of 99.39%. Among them, the center point of the ellipse is the mean value, and the colored area is the collision area. The closer to the center point, the smaller the covariance and the higher the probability of collision.

5. Conclusions

This work focuses on the problems caused by the uncertainty of pedestrian trajectory in the V2P scene, such as the determination of the collision area and the calculation of the collision probability. To do so, this paper proposed a new communication network architecture and V2P collision warning system. The system considers the GPS positioning accuracy, typical vehicle speed, the uncertainty of pedestrian behavior, and introduces a danger index to measure the degree of risk. The pedestrian behavior is predicted by using LSTM to set the collision area and gives the corresponding confidence probability. The simulation shows the influence of positioning accuracy and the vehicle speed on the warning time, which is left to the driver as the reaction time to take measures. The test gives the collision area and the corresponding confidence probability. Compared with the existing literature, the proposed method is to systematically realize the early warning system. The designed V2P early warning system is feasible for effectively improving road traffic safety and has certain reference value and practical significance for pedestrian safety protection research. The results have strong generality and reliability and can be applied to more common pedestrian and vehicle scenarios.

The proposed research method is applicable to the communication problem between a vehicle and a pedestrian. For future work, we are working on the problem of communication between multiple vehicles and multiple pedestrians. It is proposed to use filters to obtain the vehicle and the pedestrian with collision risk.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Key Industry Innovation Chain Project of Shaanxi Province (nos. 2021ZDLGY07-10 and 2021ZDLNY03-08), the Science and Technology Plan Project of Shaanxi Province (no. 2022GY-045), Natural Science Foundation of Shaanxi Province (no. 2020JM-537), the Key Research and Development Plan of Shaanxi Province (no. 2018ZDXM-GY-041), Scientific Research Program Funded by Shaanxi Provincial Education Department (Program no. 21JC030), the Science and Technology Plan Project of Xi'an (no. 2019GXYD17.3), and Graduate Innovation Fund of Xi'an University of Posts and Telecommunications (CXJJLZ202018).

References

[1] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "TrafficPredict: trajectory prediction for heterogeneous traffic-agents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Hawaii, U.S.A, February 2019.

[2] C. Chen, Y. R. Zhang, Z. Wang, S. Wan, and Q. Pei, "Distributed computation offloading method based on deep

reinforcement learning in ICV," *Applied Soft Computing*, vol. 103, Article ID 107108, 2021.

[3] P. A. Hancock, I. Nourbakhsh, and J. Stewart, "On the future of transportation in an era of automated and autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 16, pp. 7684–7691, 2019.

[4] A. Rasouli and J. K. Tsotsos, "Autonomous vehicles that interact with pedestrians: a survey of theory and practice," *Proceedings of IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 26–29, 2019.

[5] A. Hussein, F. García, J. M. Armingol, and O. M. Cristina, "P2V and V2P communication for Pedestrian warning on the basis of Autonomous Vehicles," in *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–4, LeblonRio de Janeiro, Brazil, November 2016.

[6] W. Cunningham, "Honda tech warns drivers of pedestrian presence," 2017, https://www.cnet.com/%20roadshow/news/honda-tech-warns-drivers-of-pedestrian-presence/.

[7] Who, "Global Status Report on Road Safety 2018: Summary," Technical. Report, World Health Organization (WHO), Geneva, 2018.

[8] R. Q. Malik, K. N. Ramli, Z. H. Kareem, M. I. Habelalmatee, A. H. Abbas, and A. Alamoody, "An overview on V2P communication system: architecture and application," in *Proceedings of the International Conference on Engineering Technology and its Applications (IICETA)*, pp. 6-7, Najaf, Iraq, September 2020.

[9] D. Steinhauser, P. Held, B. Thoresz, and T. Brandmeier, "Towards safe autonomous driving: challenges of pedestrian detection in rain with automotive radar," in *Proceedings of the 17th European Radar Conference (EuRAD)*, Utrecht, Netherlands, January 2021.

[10] B. Lv, R. Sun, H. Xu, and R. Yue, "Automatic vehicle-pedestrian conflict identification with trajectories of road users extracted from roadside lidar sensors using a rule-based method," *IEEE Access*, vol. 7, pp. 161594–161606, 2019.

[11] C. Y. Li, G. Salinas, P. H. Huang, G. H. Tu, G. H. Hsu, and T. Y. Hsieh, "V2PSense: enabling cellular-based V2P collision warning service through mobile sensing," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 20–24, Kansas City, MO, U.S.A, May 2018.

[12] D. Olmeda, C. Premebida, U. Nunes, J. M. Armingol, and A. D. L. Escalera, "Pedestrian detection in far infrared images," *Integrated Computer-Aided Engineering*, vol. 20, no. 4, pp. 347–360, 2013.

[13] A. Prioletti, A. Mogelmose, P. Grisleri, M. M. Trivedi, A. Broggi, and T. B. Moeslund, "Part-based pedestrian detection and feature-based tracking for driver assistance: real-time, robust algorithms, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1346–1359, 2013.

[14] F. García, J. García, A. Ponz, J. M. Armingol, and A. D. L. Escalera, "Context aided pedestrian detection for danger estimation based on laser scanner and computer vision," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6646–6661, 2014.

[15] S. H. Sun, J. L. Hu, Y. Peng, X. M. Pan, L. Zhao, and J. Y. Fang, "Support for vehicle-to-everything services based on LTE," *IEEE Wireless Communications*, vol. 23, no. 3, pp. 4–8, 2016.

[16] L. Kong, L. Ye, F. Wu, M. Tao, G. Chen, and A. V. Vasilakos, "Autonomous relay for millimeter-wave wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 9, pp. 2127–2136, 2017.

[17] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, "Overview of millimeter wave communications for fifth-generation (5G) wireless networks-with a focus on propagation models," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6213–6230, 2017.

[18] P. Merdrignac, O. Shagdar, and F. Nashashibi, "Fusion of perception and V2P communication systems for the safety of vulnerable road users," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1740–1751, 2017.

[19] 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.

[20] J. Petajajarvi, K. Mikhaylov, M. Hamalainen, and J. Iinatti, "Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring," in *Proceedings of the 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, Worcester, MA, U.S.A, March 2016.

[21] S. S. Magdum, A. Franklin, B. R. Tamma, and D. S. Pawar, "SafeNav: a cooperative V2X system using cellular and 802.11p based radios opportunistically for safe navigation," in *Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, September 2020.

[22] Y. Ren, Z. Zhao, Q. Yang, and K. S. Hong, "Adaptive neural-network boundary control for a flexible manipulator with input constraints and model uncertainties," *IEEE Transactions on Cybernetics*, vol. 51, no. 10, pp. 4796–4807, 2021.

[23] Z. J. Liu, J. Shi, X. N. Zhao, and H. X. Li, "Adaptive fuzzy event-triggered control of aerial refueling hose system with actuator failures," *IEEE Transactions on Fuzzy Systems*, 2021.

[24] Z. Liu, L. Pu, Z. Meng, X. Yang, K. Zhu, and L. Zhang, "POFS: a novel pedestrian-oriented forewarning system for vulnerable pedestrian safety," in *Proceedings of the 2015 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 19–23, Shenzhen, China, October 2015.

[25] P. Ho and J. Chen, "WiSafe: wi-fi pedestrian collision avoidance system," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4564–4578, 2017.

[26] G. Xiog, T. Yang, M. Li, Y. Zhang, W. Song, and J. Gong, "A novel V2X-based pedestrian collision avoidance system and the effects analysis of communication delay and packet loss on its application," in *Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, vol. 12, Madrid, Spain, September 2018.

[27] R. Bastani Zadeh, M. Ghatee, and H. R. Eftekhari, "Three-phases smartphone-based warning system to protect vulnerable road users under fuzzy conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2086–2098, 2018.

[28] B. I. Sighencea, R. I. Stanciu, and C. D. Căleanu, "A review of deep learning-based methods for pedestrian trajectory prediction," *Sensors*, vol. 21, no. 22, p. 7543, 2021.

[29] Y. M. Jiang, Y. N. Wang, and Z. Miao, J. Na, Z. Zhao, and C. Yang, Composite-learning-based adaptive neural control for dual-arm robots with relative motion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1010–1021, 2022.

[30] G. Q. Zhang, J Li, X. Jin, and C. Liu, "Robust adaptive neural control for wing-sail-assisted vehicle via the multiport event-triggered approach," *IEEE Transactions on Cybernetics*, pp. 1–13, 2021.

[31] C. Chen, J. G. Jiang, and Y. Zhou, N. Lv, X. Liang, and S. Wan, An edge intelligence empowered flooding process prediction using Internet of things in smart city," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 66–78, 2022.

[32] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 24–28, Vancouver, Canada, September 2017.

[33] R. Pepy and A. Lambert, "Safe path planning in an uncertain-configuration space using RRT," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 9, Beijing, China, October 2006.

[34] G. Q. Zhang, S. Liu, J. Q. Li, and X. Zhang, "LVS guidance principle and adaptive neural fault-tolerant formation control for underactuated vehicles with the event-triggered input," *Ocean Engineering*, vol. 229, Article ID 108927, 2021.

[35] Z. J. Zhao, Y. Ren, C. Mu, T. Zou, and K. S. Hong, "Adaptive neural-network-based fault-tolerant control for a flexible string with composite disturbance observer and input constraints," *IEEE Transactions on Cybernetics*, 2021.

[36] C. X. Liu, G. L. Wen, Z. Zhao, and R. Sedaghati, "Neural-network-based sliding-mode control of an uncertain robot using dynamic model approximated switching gain," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2339–2346, 2021.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] K. David and A. Flach, "CAR-2-X and pedestrian safety," *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 70–76, 2010.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5827056, 11 pages
https://doi.org/10.1155/2022/5827056

Research Article

A GRU-Based Lightweight System for CAN Intrusion Detection in Real Time

Haoyu Ma, Jianqiu Cao, Bo Mi, Darong Huang, Yang Liu, and Shaoqian Li

School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China

Correspondence should be addressed to Bo Mi; mi_bo@163.com

Received 6 April 2022; Accepted 3 June 2022; Published 27 June 2022

Academic Editor: Chen Chen

With the rapid development of vehicular networking and intelligence, more interfaces are adopted by cars to interact with the external world. Accordingly, this also brings enormous security risks, which are potentially catastrophic due to communication loopholes. Since the Controller Area Network (CAN) is critical to the transmission of commands among vehicular components, it has become a prime target for hacker research and attack. Considering that the CAN bus is commonly used and its protocol is always flawed, how to efficiently detect the intrusions against it has become an evitable problem. In this paper, we presented an intrusion detection system that can be rapidly deployed inside the vehicle. Aiming at achieving the goal of real-time detection, we devised a feature extraction algorithm with low complexity and thoroughly exploited its advantages via a GRU-based lightweight neural network. The experiment was physically conducted on in-vehicle embedded devices using publicly available datasets. Experiment results illustrated that our intrusion detection system could be rapidly deployed with high classification and real-time performance. Moreover, we also discussed how an intrusion detection system could work with OTA services to improve the intelligence of vehicular operating systems and prevent potential attacks.

1. Introduction

The deep integration of high-tech network technology and automobile technology has greatly promoted the rapid development of intelligent connected vehicle technology [1]. As an important network under the vehicle network system [2], the vehicle intranet is responsible for the information interaction between the vehicle and the outside world, and the vehicle and its drivers and passengers [3]. In the vehicle intranet, CAN (Controller Area Network), as an important underlying control network of the vehicle, is mainly used to transmit the state information and control information of the vehicle, thus ensuring the smooth and safe running of the vehicle. But its communication mode is broadcast communication, with almost no encryption means. Therefore, once hackers invade cars, CAN is almost completely exposed to the outside world, which seriously affects driving safety [4]. In recent years, due to concerns about vehicle safety, some research institutions have conducted network security studies on cars using the Internet of Vehicles; for example, in September 2016, Keen Lab cracked Tesla's in-car central control to remotely control the car over long distances (up to 12 miles away) [5]. This caused the model *S* to suddenly stop while it was moving. The Keen Lab attack used a wireless network to hack into the CAN, exposing vulnerabilities in some automakers' network domain isolation.

As the case of cracking Tesla shows, CAN is directly related to the safety of cars. The safety of cars, as necessary vehicles used by people for travel, is particularly important for the whole society, transportation system, every family, and personal safety. Intrusion detection system detects network attacks by monitoring network traffic [6]. In order to improve the overall security of the vehicle by solving the security problems in the vehicle network, this paper intends to use the intrusion detection system deployed in embedded devices to detect network threats and provide early warning functions. The system can provide threat intelligence for the vehicle manufacturers and assist them to repair the vulnerabilities in the software.

In this study, we present a GRU-based lightweight system for CAN intrusion detection. The major contributions of our study are as follows:

(i) The architecture and process of the vehicle internal network intrusion detection system are proposed, which can update the detection model quickly based on the existing OTA system, improve the accuracy of the model, and reduce the failure rate.

(ii) A feature extraction algorithm with low computational complexity is proposed based on the weak computing and storage capability of vehicle-mounted computing devices.

(iii) GRU units with fewer computational parameters than LSTM are used as important hidden layers in the neural network model, and a neural network model with simple structure, low depth, and a few hidden layers is designed

(iv) In the experiment, the Jetson Xavier NX embedded computing device already mounted on the vehicle is used as the intrusion detection device to test whether the intrusion detection system has real-time performance under the condition of low computing power. And we use the evaluation indicators commonly used in classification tasks to evaluate the model classification performance, and illustrate its advantages by comparing it with those in other studies.

The rest of this paper is organized as follows. Section 2 introduces and discusses the CAN intrusion detection methods used in the existing literature; we present their shortcomings and under-researched areas of current detection models. Section 3 details our proposed intrusion detection system and discusses the low latency benefits brought by the unified memory model in embedded devices. Section 5 describes the experimental setup and experimental results. Based on the experimental results in Section 5, in Section 6 we conclude our work.

2. Related Works

In the field of CAN bus intrusion detection, researchers have focused their research on how to improve the classification performance of detection models. The existing detection models are mainly classified into statistical-based models and machine learning-based models. However, CAN bus intrusion detection systems eventually need to be deployed inside the vehicle, and previous researchers have rarely conducted simulated experiments using actual in-vehicle computing environments.

In recent years, some researchers have proposed different algorithms and models for intrusion detection system in CAN. In 2008, Larson et al. [7] proposed a CAN intrusion detection method based on vehicle communication protocol by studying CAN protocol, which could monitor the protocol-violating messages and abnormal message sending behaviors in the network. Muter [8] proposed a network anomaly detection method based on the multi-detection theory. Eight sensors were used to monitor frame ID, data load, message frequency, and message order, and finally the detection results of sensors were integrated to identify abnormal attacks. In addition, Muter introduced the entropy theory into CAN anomaly detection and calculated ID information entropy to judge the anomaly [9]. Han et al. [10] proposed a detection scheme based on ID packet cycle and designed experiments for verification. This scheme considers only the perspective of packet ID entropy or period and cannot resist tampering attacks that modify data domains.

Some researchers choose data-driven research methods, which generate datasets by collecting massive normal CAN traffic and CAN traffic generated by intrusion behaviors. CAN traffic is classified based on the machine learning model trained by the above datasets. Cheng et al. proposed TCAN-ID [11] based on TCAN (Temporal Convolutional Attention Network). This method not only includes a new feature extraction algorithm, but also uses attention mechanism [12] in neural network to improve model performance. Taylor et al. [13] proposed an anomaly detection method based on LSTM [14]. The trained LSTM model can predict the message of the next state and compare the predicted value and the actual state value with the set threshold value to judge whether the anomaly is abnormal. The experimental results show that LSTM classified the sequential CAN traffic well, but the experimental platform used by the authors is a high computing performance server, so the intrusion detection system cannot guarantee the real-time performance when it is deployed in the actual vehicle-mounted computing equipment. Seo et al. [15] used generative adversarial network learning to detect unknown attacks using only normal data, and this IDS frame is called GIDS. The GAN characteristics of GIDS eliminate the need of intrusion detection system for a large number of abnormal data samples, but the GAN model has certain difficulties in training and deployment, and the detection model itself lacks the support of real data.

In addition to the above detection methods based on deep learning model, the study [16] proposed an intrusion detection algorithm based on SVM. However, experiments show that SVM algorithm performs poorly in multi-classification tasks and is suitable for binary classification detection system. For ensemble learning, AdaBoost, as a common ensemble learning algorithm, is often used to construct a strong classifier in intrusion detection tasks. Reference [17] proposed the idea of using decision tree as weak classifier and AdaBoost iterative integration for intrusion detection, which can reduce the overall computational complexity of the system.

Through the above related research, it can be seen that the vehicle internal network intrusion detection system needs to fit the actual computing power of the vehicle. In particular, for neural network-based intrusion detection models, it is necessary to test whether the detection model runs faster than the CAN traffic generation speed.

3. Proposed CAN Intrusion Detection System

3.1. System Architecture. We need to introduce CAN bus attack model first. In this paper, the attack is defined as

connecting to CAN bus through the vulnerability between the external network and the internal network of the vehicle and injecting attack messages through the external network to achieve the attack effect. As shown in Figure 1, attacks on CAN bus topology in this paper are divided into three categories: DoS Attack, Fuzzy Attack, and Spoofing Attack.

Among them, DoS (Denial of Service) means that the hacker uses 0000 to fill CAN ID and random data payload to encapsulate CAN message for injection. According to the CAN bus protocol, the smaller the CAN ID value is, the higher the message priority is, so the bus will be disturbed by the fake message injected by the attacker and cannot send and receive normal messages.

Fuzzy Attack is similar to DoS Attack, but the CAN ID and data payload of messages injected in Fuzzy Attack are completely random, making it more hidden than DoS and thus not easy to detect by conventional anomaly detection methods. However, this type of attack will also interfere with the onboard electronic equipment connected to the bus, which will also affect vehicle safety.

The Spoofing Attack is more targeted than the previous two attacks because it uses a specified CAN ID to populate the message. This attack can be performed on a specified in-vehicle device connected to the CAN bus and may cause the device to malfunction. Since vehicle manufacturers generally do not disclose the CAN ID and data load specification of the onboard device, this attack requires special tools (e.g., social engineering). If an attacker spoofs a specific in-vehicle device such as a wheel speed sensor, this can have a significant impact on vehicle driving safety.

In Figure 2, we show the difference between an x64 [18] architecture based server and an ARM [19] architecture based embedded device. With the improvement of the requirements for intelligent vehicles, some automobile companies have installed embedded devices with neural network reasoning ability in the interior of the newly produced vehicles, and these devices all use ARM architecture as the processor architecture. If we look at these two architectures from an evolutionary perspective, computational offloading may be the solution for the future [20].

In this paper, the neural network model will be used as the CAN bus traffic classifier. We assume that the central server stores a large number of CAN data frame time series samples and the server has high computing power to train the neural network model. The built-in GPU or the neural network module embedded in the SoC (System on a Chip) has the inference ability of the neural network model that meets the real-time detection.

The types of cybersecurity threats faced by in-vehicle networks are increasing, and intrusion detection systems deployed inside vehicles will not be able to cope with new types of attacks if they cannot be updated in a timely manner. Due to the development of wireless communication technology, many vehicles are equipped with OTA (Over-the-Air) technology [21] systems based on 4G/5G transmission. Furthermore, the OTA system can allow the vehicle to download software updates from a certified remote cloud server and can send back some of the original vehicle data to the vehicle manufacturer's backend server. The intrusion detection model proposed in this paper can be delivered to each vehicle through the OTA system, and model inference is performed based on the embedded device in the vehicle. On the other hand, the model could also be transmitted more quickly and securely through a potentially widespread vehicular ad hoc network [22].

When the CAN traffic generated by the suspected intrusion behavior is automatically identified by the detection system, the intrusion detection system will upload the original suspected malicious CAN traffic log to the OTA server, and then the OTA server will send the traffic log to the log analysis center for analysis to generate intrusion judgment and report and feed back the logs determined to be intrusion behaviors after analysis to the vehicle manufacturer's vulnerability crisis management department. The vulnerability threat treatment center will analyze the log content to find out the vulnerabilities in the vehicle software to be repaired, and push the repaired system to the vehicles affected by the vulnerability through the OTA server. All the above processes are based on the good communication efficiency between the vehicle and the remote server. If the communication conditions are poor, the intelligent caching strategy [23] can also be used for data exchange.

At the same time, the CAN bus data frame sample group after log analysis will also be uploaded to the model training server through the OTA server. In the model training server, new samples will be added to the training set of existing labels or as samples of a new attack type label. After the server uses the latest training set to train the model, the deep learning model will be sent to the OTA server, and the OTA server will update the intrusion detection system deployed on the embedded device in the vehicle. Then, the updated intrusion detection system will have better classification performance. The specific data transmission schematic diagram is shown in Figure 3.

Figure 4 shows the process of the intrusion detection system proposed in this paper, and the specific implementation of each process will be described in the following sections.

3.2. Data Structure of the CAN Frame. In the CAN2.0 B technical specification [24], CAN messages are divided into four types: data frames, remote frames, error frames, and overload frames. This article will focus on the data frames used in normal communication.

In Figure 5, the number of each region represents the number of the binary digits occupied by that region. The meaning of each field in a CAN frame is described below:

(a) SOF: It is the start flag bit of the frame. On the CAN bus, a dominant bit indicates the start of a packet.

(b) CAN ID: It is the unique identifier of the ECU to which the message is passed. A smaller value in this field indicates a higher priority for the message.

(c) DLC: It represents the length of the message frame data payload.

(d) Data payload: It represents the actual data passed by the data frame.

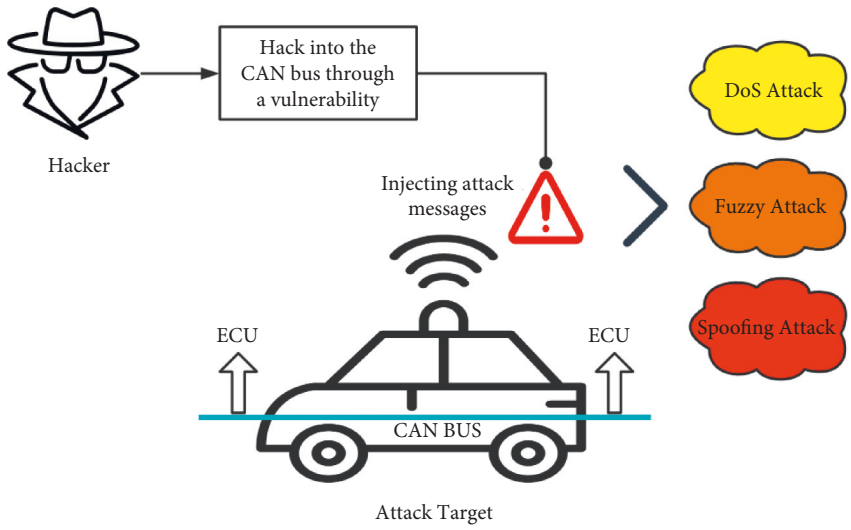

Figure 1: Schematic diagram of CAN bus attack.

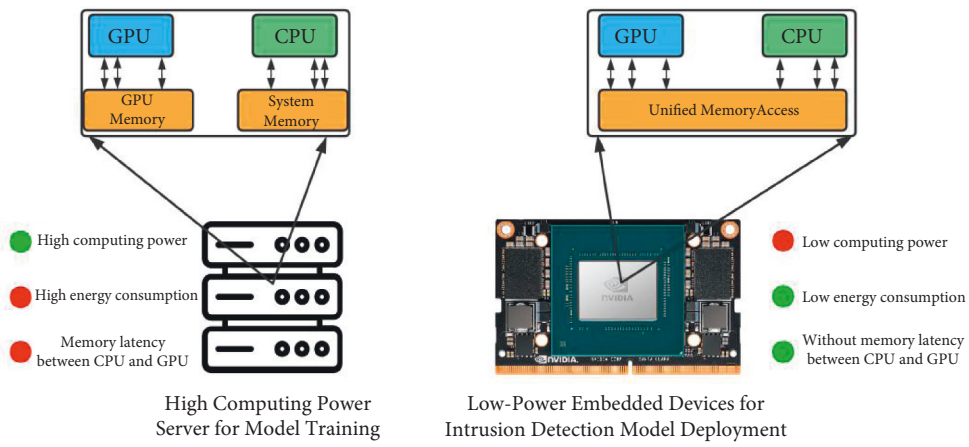

Figure 2: Intrusion detection system model training and deployment device.

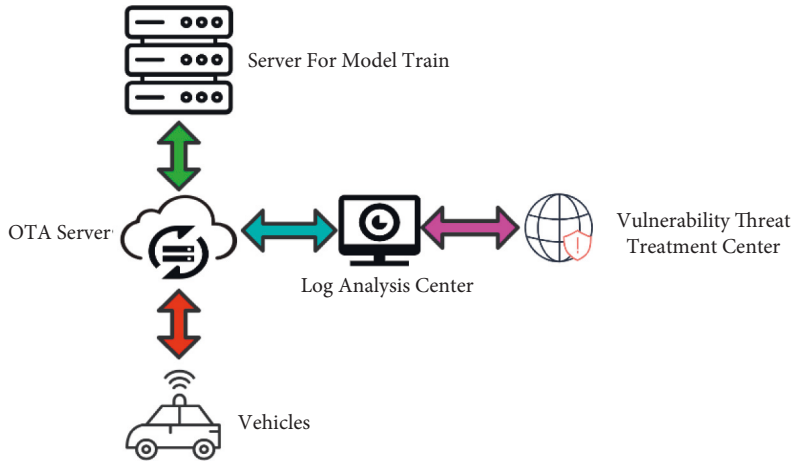

Figure 3: Scalable CAN bus intrusion detection system.

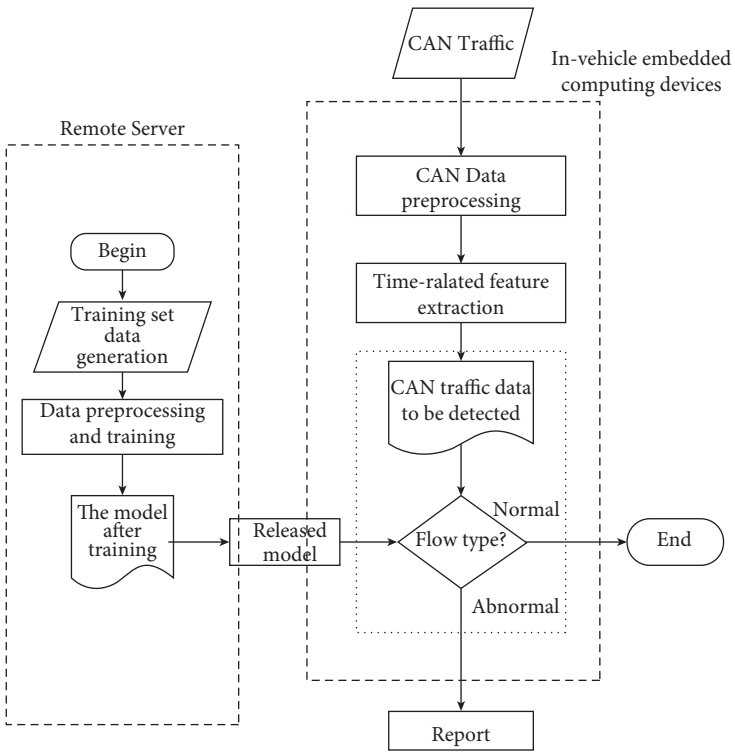

Figure 4: Intrusion detection flowchart.

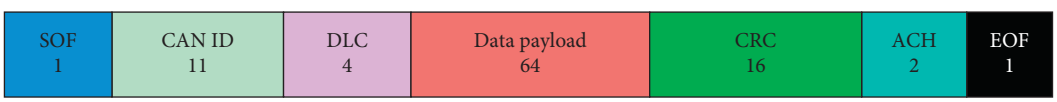

Figure 5: Structure of a CAN bus packet.

(e) CRC (Cyclic Redundancy Check): It ensures that the sender and receiver receive the correct data.

(f) ACK: It is the response field of the receiving end.

(g) EOF: It is the flag bit that indicates the end of a CAN bus frame.

In the structure of the CAN bus data packet, SOF and EOF are the flag bits of the start and end of the data frame, so they do not have the value of intrusion behavior characteristics or feature engineering. In the same way, CRC and ACK, as the fields for verifying the correctness of data transmission, can ensure that no errors will occur in the data transmission process. Since this paper only considers the intrusion detection and classification of correctly transmitted frames, this part of the data is not considered.

3.3. Data Preprocessing. In order to facilitate the subsequent feature extraction steps and deep learning model operations, we need to perform data preprocessing on the original CAN data frames. For ensuring the universality of our proposed intrusion detection algorithm, this paper uses public datasets to design data preprocessing and feature extraction for intrusion detection systems and divide the processed data into training sets and test sets. The open source in-vehicle CAN intrusion dataset used in this article is provided by a study in Korea [15]. The dataset includes the three attack types described in the previous sections of this article, and the samples in this dataset are raw data on the CAN data bus.

This public dataset includes the following attributes: TimeStamp, CAN ID, DLC, and Data Payload. The first row of the dataset is a Unix timestamp, which is the decimal number of seconds since January 1, 1970 (midnight in UTC/GMT). Since the time series features of CAN data frames are important features input to the intrusion detection model, we need to relativize the timestamps. In the process of dataset collection, all types of CAN data frames are collected continuously without interruption, so the timestamp of the first data frame can be used as the reference time, and the time difference of other data frames can be calculated by this reference. So, it is assumed that there are n samples in the dataset and the timestamp in each sample is represented as t; then, the timestamp of the $i-$ th sample is processed as follows:

$$t_i = t_i - t_0. \tag{1}$$

For CAN ID and Data Payload, the original dataset is stored in hexadecimal. In order to facilitate subsequent feature extraction and neural network training, we need to convert it to decimal. The conversion formula is as follows:

$$\text{CANID}(H) = \{H_3, H_2, H_1, H_0\}, \tag{2}$$

$$\begin{aligned}\text{CANID}(D) = {} & H_3 * 16^3 + H_2 * 16^2 \\ & + H_1 * 16^1 + H_0 * 16^0,\end{aligned} \tag{3}$$

$$\text{DataPayload}(H) = \{(H_1, H_0)_1, \ldots (H_1, H_0)_n\}, \tag{4}$$

$$\begin{aligned}\text{DataPayload}(D) = {} & \{(H_1 * 16^1, H_0 * 16^0)_1, \\ & \ldots, (H_1 * 16^1, H_0 * 16^0)_n\}.\end{aligned} \tag{5}$$

In formula (5), the value of n is determined by DLC, DLC ≤ 8.The original CAN data frame dataset collected in time series is shown in Figure 6.

Time Stamp	CAN ID	N	Data Payload
1478193190.056566,	0140,	8,	00, 00, 00, 00, 10, 29, 2a, 24
1478193190.056817	02c0,	8,	15, 00, 00, 00, 00, 00, 00 ,00
1478193190.057058,	0350,	8,	05, 20, 44, 68, 77, 00, 00, 7e
1478193190.057304,	0370,	8,	00, 20, 00, 00, 00, 00, 00 ,00
1478193190.057542,	043f,	8,	00, 00, 00, 00, 00 ,00, 00, 00

Figure 6: CAN data frame with time correlation.

3.4. Feature Extraction. As can be seen from Figure 4, the feature extraction algorithm runs on the embedded device in the vehicle, and the corresponding computing power is provided by the CPU based on the ARM architecture. Because the CPU computing power of the embedded device is weak, and the time consumed by the feature extraction algorithm will directly affect the response time of the intrusion detection system, this paper proposes a sliding window strategy based on a fixed number of messages and extracts features in the statistical domain of time series by extracting features from the CAN time series data frames passing through the window.

In the research [24], Bozdal et al. used the idea of wavelet transform to extract features from time series data frames in CAN bus, but the calculation of wavelet transform is more complicated, which may affect the real-time performance of the system. Therefore, the feature extraction method in this paper aims to reduce the computational complexity as the first purpose. First, feature extraction is performed on the preprocessed payload field. The decimal values of each byte in the field are added to obtain the sum feature of the payload field named payload_sum.If the size of the sliding window is defined as w, the feature extraction algorithm will extract the features of w packets at a time. The data payload contains n bytes in total, and each byte corresponds to a decimal number d. Therefore, the sum feature of packet m_i is calculated as follows:

$$\text{payload\_sum} = d_0 + d_1 + \cdots + d_n. \tag{6}$$

Since the messages contained in the sliding window are sorted by timestamp, in order to measure the time deviation of the samples to be detected in the window, this paper uses the variance formula to measure the deviation of the preprocessed timestamps. Defining the timestamp of a message as t, we calculate the variance as follows:

$$\text{time\_var} = \frac{(t_1 - \bar{t})^2 + (t_2 - \bar{t})^2 + \cdots + (t_w - \bar{t})^2}{w}, \tag{7}$$

$$\bar{t} = \frac{t_1 + t_2 + \ldots + t_w}{w}. \tag{8}$$

To measure the time interval between two messages with adjacent timestamps, we use the difference between two adjacent messages as the time difference feature of the message. For a message m in the window, it is necessary to calculate the timestamp difference between it and the previous message:

$$\text{time\_}d = t_m - t_{m-1}. \tag{9}$$

At the same time, in order to avoid the interference of aperiodic messages on the feature extraction algorithm, it is also necessary to calculate the median absolute error of the timestamp.

$$m(\text{time\_}d_i) = \frac{\text{time\_}d_1 + \text{time\_}d_2 + \cdots + \text{time\_}d_w}{w}, \tag{10}$$

$$\text{time\_}d\ \text{MEAD} = \text{median}(\text{time\_}d_i - m(\text{time\_}d_i)). \tag{11}$$

The intrusion detection system proposed in this paper will use the features calculated above and the CAN ID (D), Data Payload (D), and DLC of the dataset itself as input to the neural network for training and evaluation.

3.5. CAN Message Classification and Model Optimization. The detection model proposed in this paper will work on the in-vehicle embedded device, where the GPU provides the computing power for model inference and the unified memory provides the data cache space. Since the computing power and storage space provided by in-vehicle devices are limited, the design goal of our proposed neural network structure is to reduce the computational overhead as much as possible while ensuring the model detection performance.

GRU (Gate Recurrent Unit) [25] is a type of Recurrent Neural Network (RNN) [26]. Like LSTM (Long Short-Term Memory) [27], it is also proposed to solve problems such as long-term memory and gradients in backpropagation. GRU, a variant of LSTM, combines the forget gate and the input gate into a single update gate. It also mixes the cell state and the hidden state, plus some other detailed changes; the final model is simpler than the standard LSTM model. According to the experimental results of the literature [28–30], LSTM and GRU have little difference in performance on datasets with time series correlation, and GRU models even have higher performance evaluations in some datasets. From the earliest GRU literature [25], it is shown that the number of parameters calculated by GRU element is less than that by LSTM, which indicates that the time of model training and inference can be reduced. Moreover, the research [31–33] shows that GRU takes less time in training and inference than LSTM under several different types of datasets.

For the above reasons, we adopt GRU as the main hidden layer of the neural network model. The internal structure of GRU is shown in Figure 7(a). In Figure 7(a), h_{t-1} is the state

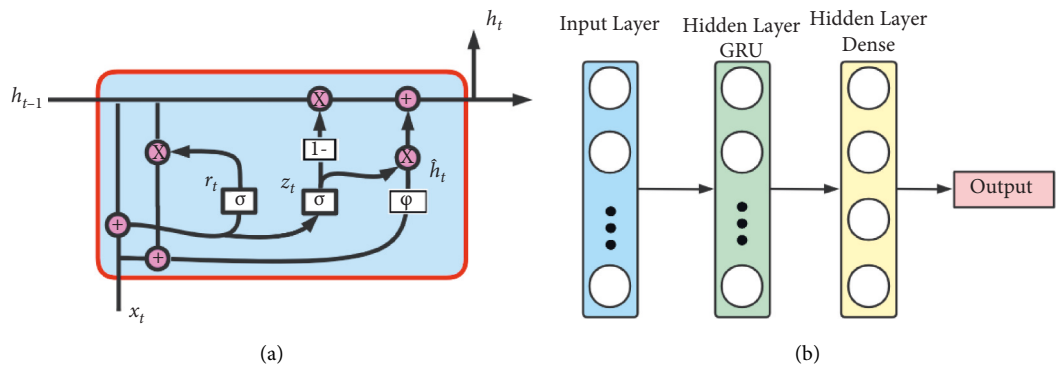

Figure 7: GRU cell structure and neural network structure.

of the previous moment relative to the current moment t. x_t and h_t are the input and output of the GRU module at the current moment, respectively. r_t and z_t are two key structures in the GRU module, namely, reset gate and update gate. Each gate is a simple neural network. And, in order to make the output of the gate fixed between 0 and 1, the activation function of the neural network is using the sigmoid function (as shown in (12). $\hat{h}_t$ is the output candidate value after reset gate processing. The structure of the GRU module is expressed by the formula as follows:

$$S(x) = \frac{1}{1+e^{-x}}, \tag{12}$$

$$r_t = \sigma\left(W_{rh}h_{t-1} + W_{rx}x_t\right), \tag{13}$$

$$z_t = \sigma\left(W_{zh}h_{t-1} + W_{zx}x_t\right), \tag{14}$$

$$\hat{h}_t = \tanh\left[W_{hh}\left(r_t \circ h_{t-1}\right) + W_{hx}x_t\right], \tag{15}$$

$$h_t = \left(1 - z_t\right) \circ \hat{h} + z_t \circ h_{t-1}. \tag{16}$$

Among them, W_{rh} and W_{rx} are the parameters in the reset gate; W_{zh} and W_{zx} are the parameters in the update gate; W_{hh} and W_{hx} are the parameters in the process of obtaining the output candidate value $\hat{h}_t$; and the operator "$\circ$" means to multiply the array elements in turn.

As shown in Figure 7(b), after the input 3-Dim data passes through the hidden layer composed of GRU units, it will be input to the fully connected layer for classification. Since the CAN intrusion detection task in this paper is a multi-classification task, the SoftMax function [34] will be used as the activation function in the last layer.

$$a_j^L = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}}. \tag{17}$$

We use the common cross-entropy loss function as the basis for network backpropagation; in information theory, cross-entropy is defined as (18). When the last layer in the network uses the SoftMax function as the activation function, the output of the last layer can be regarded as a distribution, and the cross-entropy loss function used by the neural network is shown in (19):

$$j = -\int p(x)\log g(x)\mathrm{d}x, \tag{18}$$

$$\text{loss} = -t_j \log\left(y_i\right). \tag{19}$$

Among them, j indicates that the sample belongs to the $j-$th class and y indicates the output value of the last layer.

4. Experiment

4.1. Experimental Purpose and Configuration

4.1.1. Experimental Purpose 1. First, it is necessary to verify the effectiveness of the CAN intrusion detection process proposed in this paper by evaluating the performance of the trained intrusion detection model on the test set. And, in order to fairly evaluate the performance differences between various machine learning models, this paper selects three representative machine learning models in the field of intrusion detection and compares them with the CAN message classification model proposed in this paper. The three machine learning models are AdaBoost [17], SVM (Support-Vector Machines) [16], and LSTM. Among them, AdaBoost is an ensemble learning algorithm. In this paper, a decision tree suitable for multi-classification tasks is used as the learner integrated in it. Unlike AdaBoost, the original SVM is not suitable for the multi-classification task in this paper, so we use linear classification SVM for comparative experiments. Different from AdaBoost and SVM, LSTM has strong similarity with the GRU model used in this paper, but the calculation is more complicated.

4.1.2. Experimental Purpose 2. To evaluate whether server-trained models can perform real-time inference on embedded devices, we use the Nvidia Jetson NX device used in the vehicle to test the actual deployment of the model, and the test content is the number of CAN messages detected every 100 ms. The test timing starts from the original test set data input, after data preprocessing and feature extraction; it is input to the neural network for prediction; and the timing ends after the classification result is output. In order to reflect the efficiency of the neural network model used in this article, we will use LSTM as the baseline to identify the

Table 1: Server configuration used for model training.

Category	Parameters
CPU	Intel Xeon 4210R
RAM	32 GB
GPU	Nvidia RTX 3090
Operation system	Ubuntu 18.04 LTS
CUDA version	11.1
Machine learning platform	TensorFlow 2.6 + Scikit-learn 0.23

Table 2: Embedded device configuration for model inference.

Category	Parameters
CPU	Nvidia Carmel ARM
RAM	8 GB unified memory
GPU	384-core Nvidia Volta GPU
Operation system	Ubuntu for Jetson
CUDA version	10.2
Machine learning platform	TensorFlow 2.4 + Scikit-learn 0.24

(a) (b)

Figure 8: Actual experimental device.

difference in detection speed between the two neural network models.

We use the server platform based on x64 architecture as the model training server in Figure 2. The hardware and software configuration of the platform are shown in Table 1.

To evaluate the inference performance on in-vehicle embedded devices, this paper uses an ARM-based Nvidia Jetson Xavier NX system as the experimental platform for the deployment of the intrusion detection system in Figure 2. The hardware and software configuration of the experimental platform are shown in Table 2.

Figure 8(a) shows the high computing power server used for intrusion detection model training, with average energy consumption of 800 watts. Figure 8(b) shows the experimental equipment deployed for the intrusion detection system. Under the radiator of the equipment is the Jetson Xavier NX embedded system, which has average power consumption of 10 watts.

4.2. Hyperparameter Settings. In order to ensure that subsequent researchers can reproduce the experimental results of this paper, we list the hyperparameters used for each machine learning training in the experiment. As shown in Tables 3 to 6, for a fair side-by-side comparison, we list the model hyperparameters used for the experiments.

4.3. Experimental Result. In the performance evaluation of several supervised learning models, in order to reduce the time consumption during training and ensure the fairness of the experiment, part of the original CAN bus dataset was extracted as training set and test set. The specific numbers are listed in Table 7. In addition, all models were trained and evaluated using the same dataset. In Table 8, we evaluate the effectiveness of the intrusion detection system against three attack behaviors using precision, recall, and $F1$-score. For each attack, precision is defined as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{20}$$

The recall rate represents how many of all attack messages were detected:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{21}$$

$F1$-score (equilibrium average) is a calculation result that comprehensively considers the precision and recall of the model, and the value is more inclined to the index with a smaller value. The larger the $F1$-score, the higher the quality of the model.

$$F1 - \text{score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{22}$$

Table 3: Parameters used in SVM.

Parameter name	Parameter
Penalty coefficient of error term	0.3
Kernel	SVC-linear
Probability	False
Max_iter	1000000
Random state	None

Table 4: Parameters used in AdaBoost.

Parameter name	Parameter
Base estimator	Decision tree
Number of base classifier cycles	50
Probability	False
Learning rate	1.0
Random state	None
Algorithm	SAMME.R

Table 5: Training parameters used in LSTM.

Parameter name	Parameter
Epoch	20
Num of LSTM units	40
Num of dense units	4
Batch size	64
Learning rate	0.01
Weight decay	$1e-8$
Optimizer	Adam [35]

Table 6: Training parameters used in GRU.

Parameter name	Parameter
Epoch	20
Num of LSTM units	40
Num of dense units	4
Batch size	64
Learning rate	0.01
Weight decay	$1e-8$
Optimizer	Adam [35]

Table 7: CAN intrusion dataset categories and corresponding sample numbers.

Dataset label	Num of original samples	Num of training set samples	Num of test set samples
Benign	14037000	200000	50000
DoS	587500	40000	10000
Fuzzy	491000	40000	10000
Spoofing	1134000	40000	10000

In Table 9, we present the experimental results for experimental purpose 2. A continuous period of timestamps is selected in the test set, and a total of 5000 consecutive CAN messages are available in this time block in the experiment. The real-time performance of the intrusion detection system is measured by calculating the time elapsed for message preprocessing and feature extraction as well as classification by the neural network model. After calculating the test set within forty consecutive time intervals, the CAN bus has about **200** messages or 200 CAN data frames every 100 ms. This paper uses the embedded devices in Table 2 for the experiments.

Table 8: Classification performance statistics.

DoS Attack	Precision	Recall	$F1$-Score
SVM [16]	0.8723	0.8744	0.8733
AdaBoost [17]	0.9898	0.9923	0.9910
LSTM [13]	0.9923	0.9901	0.9912
Ours	**0.9993**	**0.9991**	**0.9992**
Fuzzy Attack	Precision	Recall	$F1$-score
SVM [16]	0.8423	0.8312	0.8367
AdaBoost [17]	0.8995	0.9123	0.9059
LSTM [13]	**0.9976**	**0.9924**	**0.9950**
Ours	0.9932	0.9913	0.9922
Spoofing Attack	Precision	Recall	$F1$-score
SVM [16]	0.9312	0.9215	0.9263
AdaBoost [17]	0.9289	0.9216	0.9252
LSTM [13]	0.9932	0.9912	0.9922
Ours	**0.9995**	**0.9931**	**0.9963**

Table 9: Intrusion detection system real-time experimental results.

DL model	Num of detected messages in 100 ms	Time of detecting 5000 messages
LSTM	470	1120 ms
Ours	**650**	**890 ms**

5. Conclusion

In order to improve the security of CAN bus and avoid the attack against the important equipment on the vehicle due to the vulnerability, we study a lightweight intrusion detection system which can be deployed on the embedded equipment on the vehicle. In view of the problems of insufficient real-time performance and integration difficulties of existing intrusion detection systems, this study designs a lightweight real-time intrusion detection system suitable for vehicle-mounted CAN, which can carry out online intrusion detection of message data in the network in real time, identify intrusion messages, and secure vehicle-mounted CAN. It is proved that the intrusion detection system proposed by us has the possibility of practical deployment by conducting two contrast experiments with different purposes and setting uniform performance evaluation index.

The focus of this research is to ensure that the intrusion detection system has high sensitivity to the three attack methods in the CAN bus on the basis of simplifying the neural network structure and reducing the computational complexity. The purpose of this design is to successfully deploy the intrusion detection system into the embedded computing device on the vehicle, and we also describe the process of updating the intrusion detection model and analyzing the intrusion log.

In the future, we plan to study intrusion detection models that are simpler to compute and do not rely on GPUs, while completing the experiment on a platform with weaker computing power than Jetson. In CAN message feature extraction, we will consider using presentation learning to form end-to-end detection [36]. At the same time, considering that user privacy data on CAN bus may cause user privacy disclosure, we will consider using some privacy protection frameworks [37] to protect user privacy.

Data Availability

The CAN intrusion dataset used to support the findings of this study was supplied by Hacking and Countermeasure Research Lab under license and so cannot be made freely available. Requests for access to these data should be made to Huy Kang Kim, cenda@korea.ac.kr.

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 under Grant 61903053; the Science and Technology Research Program of Chongqing Municipal Education Commission under Grants KJZD-K201800701, KJCX2020033, and the Opening Project of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security under Grant AGK2020006.

References

[1] C. Chen, L. Liu, S. Wan, X. Hui, and Q. Pei, "Data dissemination for industry 4.0 applications in internet of vehicles based on short-term traffic prediction," *ACM Transactions on Internet Technology*, vol. 22, pp. 1–18, 2022.

[2] C. Chen, Y. Zeng, H. Li, Y. Liu, and S. Wan, "A multi-hop task offloading decision model in MEC-enabled internet of vehicles," *IEEE Internet of Things Journal*, p. 1, 2022.

[3] S. Wan, S. Ding, and C. Chen, "Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles," *Pattern Recognition*, vol. 121, Article ID 108146, 2022.

[4] C. J. Castillo, S. Zeadally and J. A. Guerrero-Ibanez, Internet of vehicles: architecture, protocols, and security," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701–3709, 2018.

[5] S. Nie, L. Liu, and Y. Du, "Free-fall: hacking tesla from wireless to can bus," *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017.

[6] R. A. Kemmerer and G. Vigna, "Intrusion detection: a brief history and overview," *Computer*, vol. 35, pp. supl27–supl30, 2002.

[7] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 220–225, IEEE, Eindhoven, Netherlands, June 2008.

[8] M. Müter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in *Proceedings of the 2010 Sixth International Conference on Information Assurance and Security*, pp. 92–98, IEEE, Atlanta, GA, USA, August 2010.

[9] M. Müter and A. Naim, "Entropy-based anomaly detection for in-vehicle networks," in *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1110–1115, IEEE, Baden-Baden, Germany, June 2011.

[10] M. L. Han, B. I. Kwak, and H. K. Kim, "Anomaly intrusion detection method for vehicular networks based on survival analysis," *Vehicular communications*, vol. 14, pp. 52–63, 2018.

[11] P. Cheng, K. Xu, S. Li, and Mu Han, "TCAN-IDS: intrusion detection system for internet of vehicle using temporal convolutional attention network," *Symmetry*, vol. 14, no. 2, p. 310, 2022.

[12] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: an interpretable predictive model for healthcare using reverse time attention mechanism," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[13] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 130–139, IEEE, Montreal, QC, Canada, 2016 October.

[14] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[15] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proceedings of the In 2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pp. 1–6, IEEE, Belfast, Ireland, Aug2018.

[16] D. Jing and H.-B. Chen, "SVM based network intrusion detection for the UNSW-NB15 dataset," in *Proceedings of the 2019 IEEE 13th International Conference on ASIC (ASICON)*, pp. 1–4, IEEE, Chongqing, China, Nov2019.

[17] W. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 2, pp. 577–583, 2008.

[18] M. Matsui, *How far can we go on the x64 processors."In International Workshop on Fast Software Encryption*, pp. 341–358, Springer, Berlin, Heidelberg, 2006.

[19] D.. Jaggar, "ARM architecture and systems," *IEEE micro*, vol. 17, no. 4, pp. 9–11, 1997.

[20] C. Chen, Y. Zhang, Z. Wang, S. Wan, and Q. Pei, "Distributed computation offloading method based on deep reinforcement learning in ICV," *Applied Soft Computing*, vol. 103, Article ID 107108, 2021.

[21] S. Halder, A. Ghosal, and M. Conti, "Secure over-the-air software updates in connected vehicles: a survey," *Computer Networks*, vol. 178, Article ID 107343, 2020.

[22] B. Liu, D. Jia, J. Wang, K. Lu, and L. Wu, "Cloud-assisted safety message dissemination in VANET–cellular heterogeneous wireless network," *Ieee Systems Journal*, vol. 11, no. 1, pp. 128–139, 2017.

[23] C. Chen, J. Jiang, R. Fu, L. Chen, C. Li, and S. Wan, "An intelligent caching strategy considering time-space characteristics in vehicular named data networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021.

[24] M. Bozdal, M. Samie, and I. K. Jennions, "WINDS: a wavelet-based intrusion detection system for Controller Area Network (CAN)," *IEEE Access*, vol. 9, no. 2021, pp. 58621–58633.

[25] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597–1600, IEEE, Boston, MA, USA, Aug2017.

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," *California Univ San Diego La Jolla Inst for Cognitive Science*, 1985.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] P. T. Yamak, Y. Li, and P. K. Gadosey, "A comparison between arima, lstm, and gru for time series forecasting," in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pp. 49–55, Sanya China, Dec 2019.

[29] A. Sethia and P. Raut, *Application of LSTM, GRU and ICA for stock price prediction." In Information and Communication Technology for Intelligent Systems,*pp. 479–487, Springer, Singapore, 2019.

[30] M. R. Raza, W. Hussain, and J. Maria Merigó, "Cloud sentiment accuracy comparison using RNN, LSTM and GRU," in *Proceedings of the In 2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1–5, IEEE, Elazig, Turkey, October 2021.

[31] S. Yang, X. Yu, and Y. Zhou, "Lstm and gru neural network performance comparison study: taking yelp review dataset as an example," in *Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, pp. 98–101, IEEE, Shanghai, China, June 2020.

[32] C.-Bi Lin, Z. Dong, W.-K. Kuan, and Y.-Fa Huang, "A framework for fall detection based on OpenPose skeleton and LSTM/GRU models," *Applied Sciences*, vol. 11, no. 1, p. 329, 2020.

[33] X. Jiang, J. Sun, C. Li, and H. Ding, "Video image defogging recognition based on recurrent neural network," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3281–3288, 2018.

[34] R. A. Dunne and N. A. Campbell, "On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function," *Proc. 8th Aust. Conf. on the Neural Networks*, vol. 181p. 185, Melbourne, 1997.

[35] D. P. Kingma and Ba. Jimmy, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[36] L. Wu, C. Quan, C. Li, Q. Wang, B. Zheng, and X. Luo, "A context-aware user-item representation learning for item recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 2, pp. 1–29, 2019.

[37] Z. Wang, X. Pang, Y. Chen et al., "Privacy-preserving crowdsourced statistical data publishing with an untrusted server," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1356–1367, 2019.