

Journal of Advanced Transportation

# Graph-Based Big Data Analysis and Mining in Transportation Systems

Lead Guest Editor: JingXin Dong

Guest Editors: Ming Xu, Ling Huang, and Geqi Qi





---

# **Graph-Based Big Data Analysis and Mining in Transportation Systems**

Journal of Advanced Transportation

---

## **Graph-Based Big Data Analysis and Mining in Transportation Systems**

Lead Guest Editor: JingXin Dong

Guest Editors: Ming Xu, Ling Huang, and Geqi Qi





---

Copyright © 2022 Hindawi Limited. All rights reserved.
























This is a special issue published in "Journal of Advanced Transportation." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Associate Editors

Juan C. Cano , Spain  
Steven I. Chien , USA  
Antonio Comi , Italy  
Zhi-Chun Li, China  
Jinjun Tang , China

## Academic Editors

Kun An, China  
Shriniwas Arkatkar, India  
José M. Armingol , Spain  
Socrates Basbas , Greece  
Francesco Bella , Italy  
Abdelaziz Bensrhair, France  
Hui Bi, China  
María Calderon, Spain  
Tiziana Campisi , Italy  
Giulio E. Cantarella , Italy  
Maria Castro , Spain  
Mei Chen , USA  
Maria Vittoria Corazza , Italy  
Andrea D'Ariano, Italy  
Stefano De Luca , Italy  
Rocío De Oña , Spain  
Luigi Dell'Olio , Spain  
Cédric Demonceaux , France  
Sunder Lall Dhingra, India  
Roberta Di Pace , Italy  
Dilum Dissanayake , United Kingdom  
Jing Dong , USA  
Yuchuan Du , China  
Juan-Antonio Escareno, France  
Domokos Esztergár-Kiss , Hungary  
Saber Fallah , United Kingdom  
Gianfranco Fancello , Italy  
Zhixiang Fang , China  
Francesco Galante , Italy  
Yuan Gao , China  
Laura Garach, Spain  
Indrajit Ghosh , India  
Rosa G. González-Ramírez, Chile  
Ren-Yong Guo , China

Yanyong Guo , China  
Jérôme Ha#rri, France  
Hocine Imine, France  
Umar Iqbal , Canada  
Rui Jiang , China  
Peter J. Jin, USA  
Sheng Jin , China  
Victor L. Knoop , The Netherlands  
Eduardo Lalla , The Netherlands  
Michela Le Pira , Italy  
Jaeyoung Lee , USA  
Seungjae Lee, Republic of Korea  
Ruimin Li , China  
Zhenning Li , China  
Christian Liebchen , Germany  
Tao Liu, China  
Chung-Cheng Lu , Taiwan  
Filomena Mauriello , Italy  
Luis Miranda-Moreno, Canada  
Rakesh Mishra, United Kingdom  
Tomio Miwa , Japan  
Andrea Monteriù , Italy  
Sara Moridpour , Australia  
Giuseppe Musolino , Italy  
Jose E. Naranjo , Spain  
Mehdi Nourinejad , Canada  
Eneko Osaba , Spain  
Dongjoo Park , Republic of Korea  
Luca Pugi , Italy  
Alessandro Severino , Italy  
Nirajan Shiwakoti , Australia  
Michele D. Simoni, Sweden  
Ziqi Song , USA  
Amanda Stathopoulos , USA  
Daxin Tian , China  
Alejandro Tirachini, Chile  
Long Truong , Australia  
Avinash Unnikrishnan , USA  
Pascal Vasseur , France  
Antonino Vitetta , Italy  
S. Travis Waller, Australia  
Bohui Wang, China  
Jianbin Xin , China




---

Hongtai Yang , China  
Vincent F. Yu , Taiwan  
Mustafa Zeybek, Turkey  
Jing Zhao, China  
Ming Zhong , China  
Yajie Zou , China

# Contents

## **Conflict Probability Prediction and Safety Assessment of Straight-Left Traffic Flow at Signalized Intersections**

Yingying Ma , Zihao Zhang , and Jiabin Wu 


Research Article (14 pages), Article ID 8233424, Volume 2022 (2022)

## **Spatiotemporal DeepWalk Gated Recurrent Neural Network: A Deep Learning Framework for Traffic Learning and Forecasting**

Jian Yang , Jinhong Li , Lu Wei , Lei Gao , and Fuqi Mao 


Research Article (11 pages), Article ID 4260244, Volume 2022 (2022)

## **Bi-GRCN: A Spatio-Temporal Traffic Flow Prediction Model Based on Graph Neural Network**

Wenhao Jiang , Yunpeng Xiao, Yanbing Liu, Qilie Liu, and Zheng Li 



Research Article (12 pages), Article ID 5221362, Volume 2022 (2022)

## **Research on Recommendation Algorithm of Joint Light Graph Convolution Network and DropEdge**

Haicheng Qu, Jiangtao Guo , and Yanji Jiang


Research Article (12 pages), Article ID 3843021, Volume 2022 (2022)

## **Online Traffic Accident Spatial-Temporal Post-Impact Prediction Model on Highways Based on Spiking Neural Networks**

Duwei Li , Jianping Wu , and Depin Peng






Research Article (20 pages), Article ID 9290921, Volume 2021 (2021)

## **MDGCN: Multiple Graph Convolutional Network Based on the Differential Calculation for Passenger Flow Forecasting in Urban Rail Transit**

Chenxi Wang, Huizhen Zhang , Shuilin Yao, Wenlong Yu, and Ming Ye

Research Article (10 pages), Article ID 2956151, Volume 2021 (2021)

## **Prediction of Road Network Traffic State Using the NARX Neural Network**

Ziwen Song , Feng Sun , Rongji Zhang , Yingcui Du , and Chenchen Li 

Research Article (17 pages), Article ID 2564211, Volume 2021 (2021)

## **Multichannel Speech Enhancement in Vehicle Environment Based on Interchannel Attention Mechanism**

Xueli Shen , Zhenxing Liang , Shiyin Li , and Yanji Jiang 



Research Article (9 pages), Article ID 9453911, Volume 2021 (2021)

## **Graphical Optimization Method for Symmetrical Bidirectional Corridor Progression**

Kai Lu , Shuyan Jiang , Yiming Zhao , Yongjie Lin , and Yinhai Wang 

Research Article (18 pages), Article ID 7649214, Volume 2021 (2021)

## **Mining Travel Time of Airport Ferry Network Based on Historical Trajectory Data**

Cong Ding , Jun Bi , Dongfan Xie, Xiaomei Zhao, and Yi Liu



Research Article (11 pages), Article ID 9231451, Volume 2021 (2021)

**An Influence Analytical Model of Dedicated Bus Lane on Network Traffic by Macroscopic Fundamental Diagram**

Yingying Ma , Yuanqi Xie , and Yongjie Lin 




Research Article (18 pages), Article ID 2617732, Volume 2021 (2021)

**Highway Traffic Speed Prediction in Rainy Environment Based on APSO-GRU**

Dongqing Han, Xin Yang, Guang Li, Shuangyin Wang, Zhen Wang , and Jiandong Zhao 

Research Article (11 pages), Article ID 4060740, Volume 2021 (2021)

**Correlation Analysis of External Environment Risk Factors for High-Speed Railway Derailment Based on Unstructured Data**

Haixing Wang , Yuanlanduo Tian , and Hong Yin 

Research Article (11 pages), Article ID 6980617, Volume 2021 (2021)



## Research Article

# Conflict Probability Prediction and Safety Assessment of Straight-Left Traffic Flow at Signalized Intersections

Yingying Ma , Zihao Zhang , and Jiabin Wu 

*Department of Transportation Engineering, South China University of Technology, Guangzhou 510641, China*

Correspondence should be addressed to Jiabin Wu; 13670000003@sina.cn

Received 30 September 2021; Revised 24 March 2022; Accepted 28 April 2022; Published 25 June 2022

Academic Editor: Geqi Qi

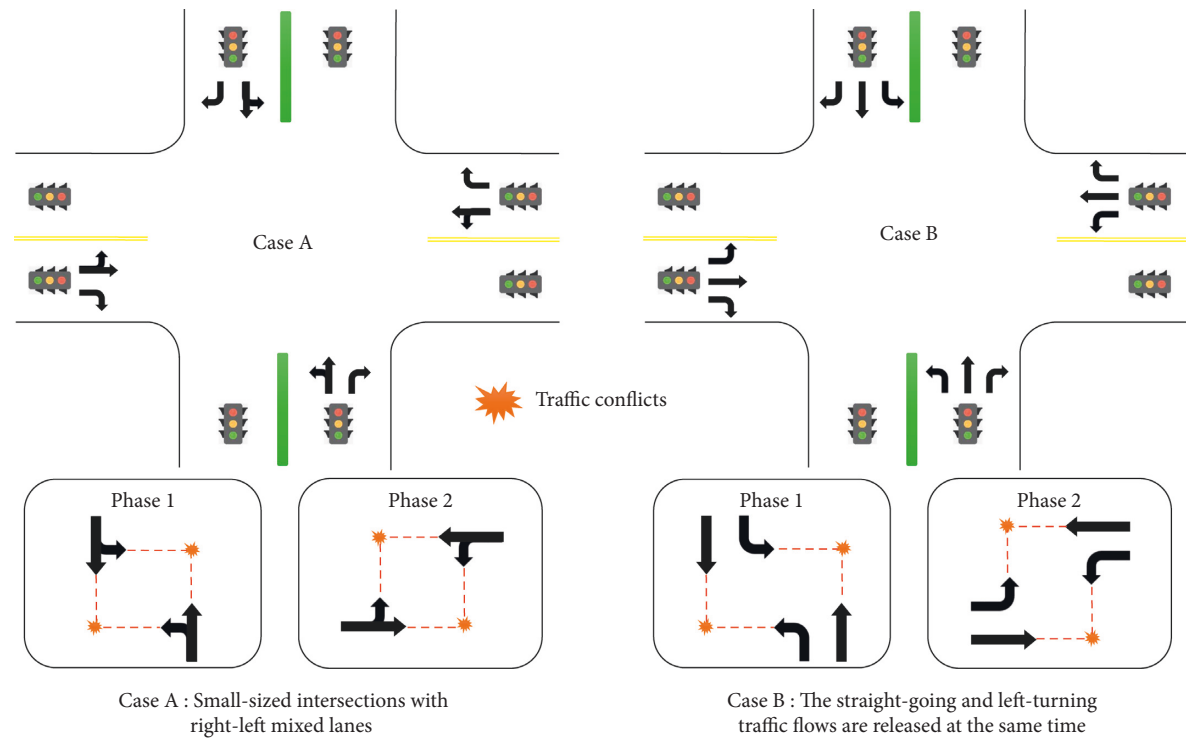
Copyright © 2022 Yingying Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The safety of signalized intersections is of great concern. To allow for an effective evaluation measure on the safety level of intersections, traffic conflict analysis methods are commonly used. However, the existing literature has mainly focused on the statistical prediction of conflicts by using surrogate measurements, among which the spatial-temporal characteristics of the potential conflicts have been less addressed. In addition, most of the relevant studies rely on precise trajectory data, and the results could be limited to engineering applications when real-time/comprehensive trajectory data are not available. To address these issues, this study proposes a SICP (signalized intersection conflict probability) model to predict a straight-left traffic flow conflict with a spatial-temporal distribution in the heat map, which could effectively evaluate the traffic safety of the existing or prebuilt signalized intersections on urban roads. Firstly, the impact of vehicle movement characteristics on traffic conflict at signalized intersections was considered by incorporating the vehicle movement trajectory. Secondly, the signal phase was categorized in several stages (each phase contains switching and nonswitching stages); then, a vehicle-vehicle conflicts probability prediction model was established by integrating both horizontal and vertical arrival probability. Finally, to validate the performance of the proposed model, the measured data were collected from the intersection of Wushan road and Yuehan road in Tianhe District, Guangzhou, China. SSAM ( Surrogate Safety Assessment Model ) traffic conflict simulation was used to analyze the traffic conflict in the actual data and compared to the SICP model. A case study was conducted to reveal the evolution mechanism of the conflict risk coefficient at the signalized intersection and to estimate the safety status under the various security optimization strategies. The experimental results verified the effectiveness of the SICP model, indicating that the proposed model is effective in evaluating the safety level of existing or prebuilt signalized intersections.

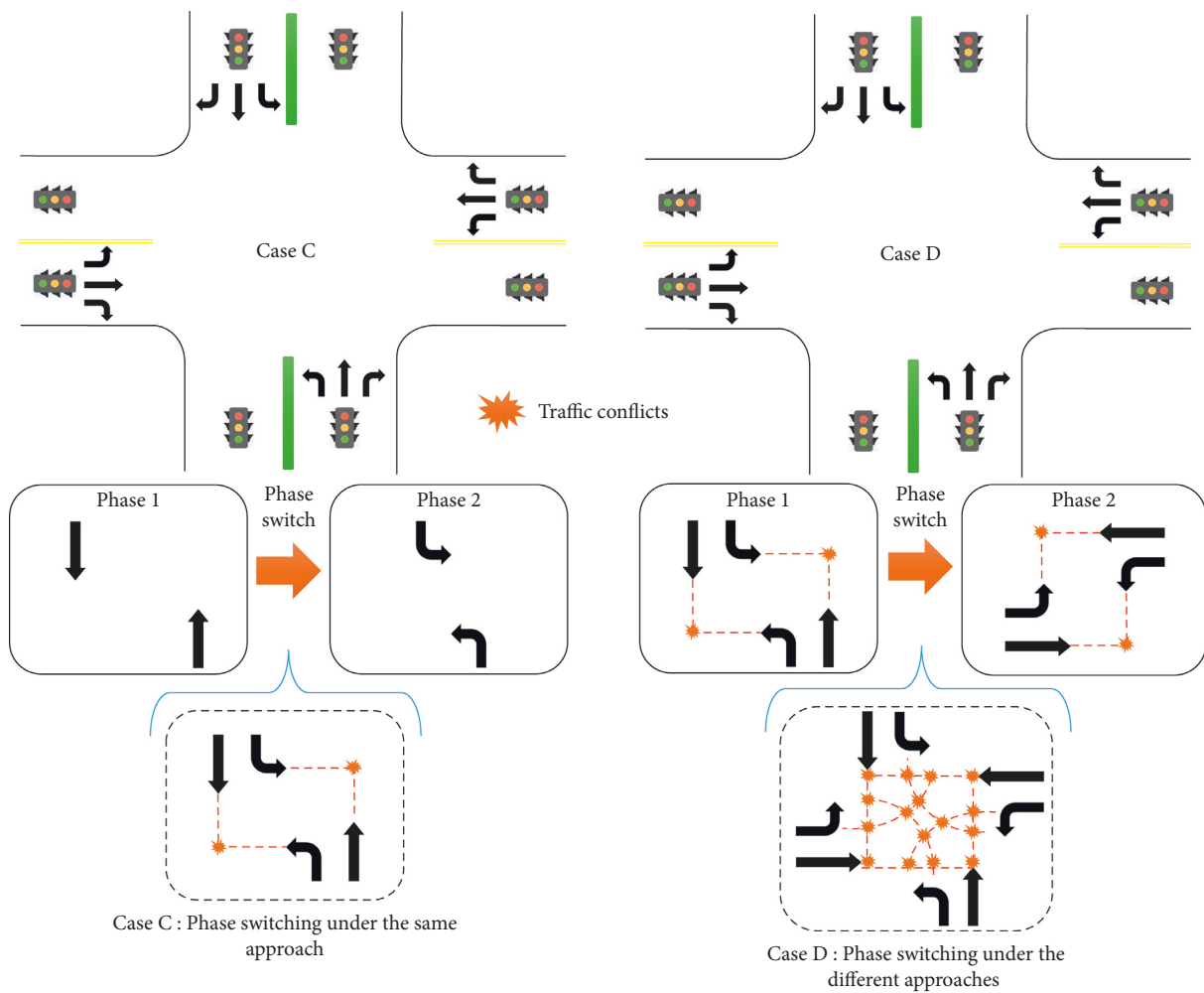
## 1. Introduction

As the bottleneck of the urban road network, signalized intersections play an important role in the operation of urban transportation, as various road users exist, such as motor vehicles, nonmotor vehicles, pedestrians, and other road users. However, signalized intersections are nodes with frequent traffic conflicts and congestion, as well as the common place of traffic accidents. As a key component to realize the distribution of the right way at intersections, signal control facilities are responsible for maintaining the normal operation of the traffic order. Although the existing urban traffic managers separate the right-going and left-turning traffic flows at signalized intersections in time or

space, there are still some small-sized intersections with right-left mixed lanes, or the straight-going and left-turning traffic flows are released at the same time due to unreasonable signal timing. At this time, there is the right-of-way competition between the straight-going traffic flow and the left-turning traffic flow, and this is an important cause of the frequent blocking and traffic conflicts at intersections, as shown in Figure 1(a). In addition, all-red intervals of some signalized intersections are not set or are too short to clear the vehicles of the last phase, which leads to serious conflicts during the phase switch, as shown in Figure 1(b). Therefore, the frequency and severity of traffic conflicts at such signalized intersections are higher than those in other areas. Therefore, traffic conflicts and collisions are more likely to



(a)



(b)

FIGURE 1: The traffic conflicts of some signalized intersections in one signal cycle. (a) Two common cases of traffic conflicts at signalized intersections during a phase. (b) Two common cases of traffic conflicts at signalized intersections during the phase switch.

occur at signalized intersections with the simultaneous release of straight and left traffic flow, and the safety assessment of this issue deserves more attention.

The safety evaluation methods of signalized intersections are mainly divided into two categories: one is the direct evaluation method based on traffic accident statistics, and the other is the indirect evaluation method based on traffic conflict analysis. The direct evaluation methods mainly include empirical modelling [1], gray evaluation [2], regression modelling [3], and other methods. The expected number of traffic accidents is taken as the evaluation index, which strongly relies on the historical accident data. By analyzing the historical data of road traffic accidents in 20 European countries, Al-Ghamdi [4] studied the correlation between the death toll of road traffic accidents and the population and the number of motor vehicles, performed regression analysis, and proposed the Smeed model. Guo et al. [5] collected historical traffic accident data by mobile apps, established an improved k-means algorithm, evaluated road traffic safety, and identified traffic accident black spots. Chen et al. [6] collected traffic and accident data from 246 nonsignalized intersections and proposed a generalized negative binomial regression model based on zero censorings to comprehensively evaluate the safety of intersections. Although the application of this method is direct and simple, the drawback is also acknowledged that it requires a significant amount of high-quality accident statistical data in most cases. Therefore, the application of the direct evaluation method in traffic safety analysis is somewhat limited due to the difficulties of obtaining traffic accident data, its unstable quality, and long collection period.

In contrast, indirect evaluation methods based on traffic conflict analysis are more widely used in traffic safety analysis. These studies use the frequency and severity of traffic conflicts to indirectly reflect the level of traffic safety, which mainly focuses on the effectiveness of traffic conflicts [7–19], traffic conflict evaluation indicators [20, 21], and data methods [22, 23]. To facilitate the extension and application of the definition of traffic conflict in practice, scholars have proposed a variety of traffic conflict indicators to quantify the proximity and interaction between two or more road users in time or space. The most commonly used conflict indicators include the time to collision (TTC) [24, 25], the postencroachment time (PET) [26–28], and the stopping distance index (SDI) [29–31]. For example, Hayward [25] proposed TTC as a measurement index of traffic conflict, which is defined as the time to collision when two vehicles maintain their existing speed and direction unchanged, to evaluate the traffic safety level of intersections. Mohamed et al. [32] collected TTC and the traffic parameters of each signal cycle at signal intersections and proposed a safety performance function based on traffic conflict by using a generalized linear model, which evaluated the traffic safety status of signalized intersections. Guo [33] collected traffic conflict data of vehicles at signalized intersections and proposed safety evaluation methods of conflicts between motor vehicles and motor vehicles, motor vehicles and nonmotor vehicles, and motor vehicles and pedestrians at signalized intersections based on the Bayesian method. Ma

et al. [34] built a traffic conflict identification model based on the PET algorithm in ramp merging areas, considering vehicle movement information and the influence of vehicle size on traffic conflict and provided a method for determining the severity of traffic conflict. Lareshyn et al. [35] proposed a traffic conflict measurement method based on extended Delta-V to evaluate the safety status of various intersections, from the perspective of traffic accident severity. Oh et al. [30] proposed stopping sight distances based on SDI to estimate whether a given car-following condition is safe. Hyden [36] proposed time to accident (TA) as an indicator to identify road traffic conflicts and then evaluated road safety. This kind of method is widely used, of which the model is stable and flexible, and the evaluation index is diversified, but it does not consider the actual size of the vehicle and is unable to evaluate the future situation of intersections. In addition, most of the previous studies deeply rely on real-time vehicle trajectory data, which are hard to access in engineering applications. Thus, these methods can only evaluate the safety of the existing signalized intersections and fail to assess the safety level of the design scheme or optimization scheme of the signalized intersections. Furthermore, few studies have focused on the spatial-temporal characteristics of the potential conflicts at signalized intersections, and instead they tend to pay more attention to the recognition accuracy of traffic conflicts. Finally, according to our latest knowledge, there is no literature concerning traffic conflict when two phases switch, which may be the most serious period of traffic conflict.

To address the above issues, this paper proposes a new method of conflict probability prediction for signalized intersections, considering the micro operation characteristics of vehicles based on their actual size. The best feature of the proposed model is that it has low traffic data requirements and only needs traffic flow and signal timing parameters instead of real-time vehicle trajectories or an enormous body of historical data. In other words, the model can not only evaluate the existing signalized intersections, but also analyze the traffic conflicts of prebuilt signalized intersections as long as the design scheme provides the related data. The secondary superiority is to consider the spatial-temporal characteristics of the potential conflicts to evaluate the comprehensive safety level of signalized intersections. On the one hand, the spatial distribution of traffic conflicts is deduced by merging areas of different vehicle running trajectories. Besides, the signal timing is divided into several stages based on the changes in the spatial distribution of the traffic conflicts. Another crucial creativity of this study is that the proposed model is especially concerned with traffic conflicts during phase switching, which is a very important period but is usually ignored in other researches. The simulation experiment and case analysis results are conducted to verify the effectiveness of the SICP model. These findings can provide theoretical and technical support for the channelization design and signal timing optimization of signalized intersections for traffic engineers.

The remainder of this paper is arranged as follows. Section 2 describes how to construct the cell system of a signalized intersection. Section 3 explains the modelling

process of the SICP model and the calculation of the safety assessment index. Section 4 gives some comparative experiments and results based on the actual data. Three typical case studies and discussions are illustrated in Section 5. Eventually, this paper ends with some conclusions and future work in Section 6.

## 2. Analysis on Micro Operation Characteristics of Vehicles at Signalized Intersections

**2.1. Cell Division and Its Approximation.** To effectively evaluate the internal traffic safety status of a signalized intersection, a two-dimensional Cartesian plane coordinate system is constructed by taking the intersection of the east-west direction and the north-south direction of the intersection as the coordinate origin. The east approach road is the positive direction of the  $X$ -axis, while the north approach road is the positive direction of the  $Y$ -axis. It is assumed that the width of the approach and exit lanes in all directions of the intersection are symmetrical. Let  $M$  and  $N$  denote the length and width of the intersection, respectively.  $M - 1$  and  $N - 1$  points are evenly inserted on the  $X$ -axis and  $Y$ -axis, respectively, and the intersection conflict area is evenly divided into  $M \times N$  cells, as shown in Figure 2.

$$\begin{aligned} -\frac{M}{2} &= x_0 < \dots < x_j < \dots < x_n = \frac{M}{2}, \\ -\frac{N}{2} &= y_0 < \dots < y_i < \dots < y_m = \frac{N}{2}, \end{aligned} \quad (1)$$

where  $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, n$ .  $m$  and  $n$  are positive integers, and its value must meet that the length and width of each cell are less than the vehicle width to reflect the arrival of vehicles. At this time, vehicles occupying any part of a cell are regarded as vehicles arriving at the cell.

In the coordinate system, the cell in row  $i$  and column  $j$  is denoted as  $\text{Rec}(i, j)$ . The probability of the vehicle occupying the cell can be approximately regarded as the probability of the vehicle appearing in the centroid of the cell.  $(x_{ij}, y_{ij})$  is used to represent the centroid coordinates of the cell  $\text{Rec}(i, j)$ , as shown in

$$x_{ij} = \frac{x_i + x_{i+1}}{2}, y_{ij} = \frac{y_j + y_{j+1}}{2}. \quad (2)$$

**2.2. Analysis of the Vehicle Trajectories.** Taking the signalized intersection with six two-way lanes where straight and left-turn vehicles are released simultaneously as an example, this paper describes the vehicle trajectories under different signal phases by corresponding mathematical expressions, as shown in Figure 3. Additionally, the trajectory equation used in this paper can also be applied to straight and left turn mixed traffic and the signalized intersections of other geometric conditions.

According to the lane function setting of the intersection, the vehicle trajectory equation of each approach is established. Taking the signalized intersection in Figure 3 as

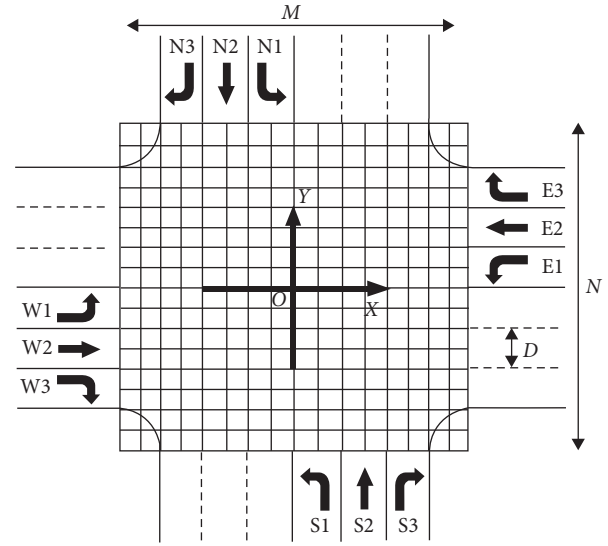


FIGURE 2: Coordinate system and cell division of conflict area in a signalized intersection.

an example, let a ( $a = 3$ ) denote the number of lanes at each approach road. The lanes are numbered from inside to outside in the order of small to large; that is, the lane number  $h = 1, \dots, a$ . Then, the vehicle trajectory equations of going straight and turning left and right at each approach road are as follows:

$$\frac{A}{2} + (h - 1)D \leq y_{ij} \leq \frac{A}{2} + hD, \quad (3)$$

$$(x_{ij} - x_{dw})^2 + (y_{ij} - y_{dw})^2 = R_w^2,$$

where  $R_w$  denotes the left turning or right turning radius of the vehicle in each approach, and  $R_w \in [R_{dw} - 0.5D, R_{dw} + 0.5D]$ .  $R_{dw}$  denotes the designed radius of left-turning or right-turning of the vehicle in each approach.  $(x_{dw}, y_{dw})$  denotes the center coordinates of the turning track circle at each approach.  $d$  represents the direction of the approach, and  $d = \{d_1, d_2, d_3, d_4\}$  denote the directions east, west, south, and north, respectively.  $w = \{l, r\}$  indicate the left and right turn directions, respectively.  $D$  denotes the width of the single lane, and  $A$  denotes the width of the median.

## 3. Signalized Intersection Conflict Probability Model

**3.1. Signal Phase Decomposition considering the Temporal and Spatial Evolution of Traffic Conflict.** In this paper, the SICP model divides signal phases at intersections based on the evolution of conflict mechanisms. Some small intersections may not set the all-red time or the all-red time is too short. When the signal phase switches periodically, the vehicles in the previous phase often fail to leave the intersection in time, which is likely to cause traffic conflicts with the vehicles starting in the next phase. During this period, the traffic conflict situation is relatively complex, and the composition of participants in the conflict changes. Thus, this process is

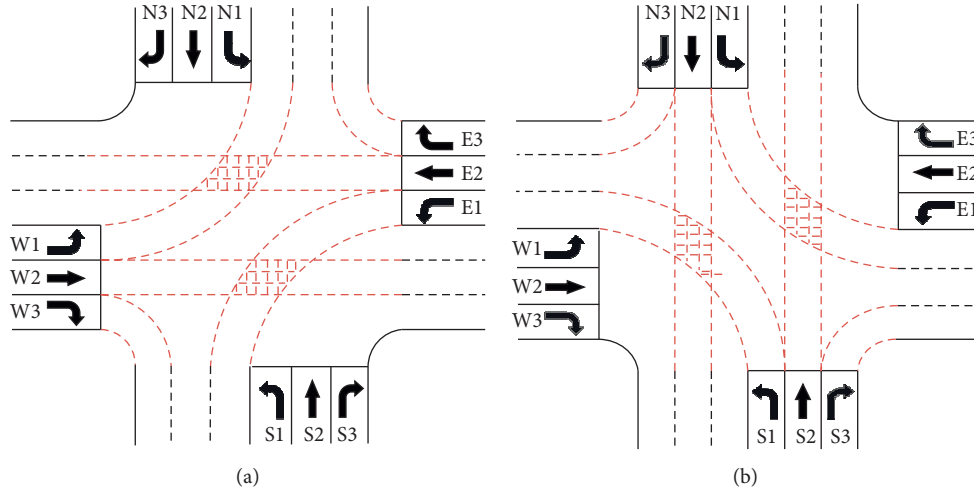


FIGURE 3: Vehicle trajectories at a signalized intersection. (a) Phase 1: vehicle trajectories of east-west approach. (b) Phase 2: vehicle trajectories of north-south approach.

defined as a phase switching stage, and other processes of the same phase are called the nonphase switching stage. Their critical time is the moment when the vehicles at the end of the previous phase leave the intersection.

To describe the generality of the phase division, the phase number in the signal timing scheme is set as  $Z$ . Each signal cycle can be divided into  $2Z$  phase stages based on the evolution of the conflict probability. That is, one phase includes the phase switching stage and nonphase switching stage, as shown in Figure 4. Suppose  $T_i$  denotes the duration of the  $i$ th phase,  $L_j$  denotes the distance of each traffic flow in the previous phase from the stop line to leave the intersection, and  $v_j$  denotes the average driving speed of each traffic flow in the previous phase. The durations of the phase switching stage and nonphase switching stage in the  $i$ th phase are

$$T_{i1} = \max \left\{ \frac{L_1}{v_1}, \dots, \frac{L_j}{v_j} \right\}, \quad (4)$$

$$T_{i2} = T_i - T_{i1},$$

where  $T_{i1}$  and  $T_{i2}$  are the duration of the phase switching phase and the nonphase switching phase in phase number  $i$ , respectively. For example,  $T_{11}$  and  $T_{21}$  denote the duration of the phase switching stage, and  $T_{12}$  and  $T_{22}$  denote the duration of the nonphase switching stage.  $j$  is the number of traffic flow at a signalized intersection,  $i = 1, \dots, Z$ .

**3.2. Conflict Probability Models considering Micro Operation Characteristics of Vehicles.** Considering the rear-end collision and the front or side collision from different angles, a conflict probability model of the vehicle for any cell at signalized intersections is established. Let  $k$  denote the vehicle trajectory, and there are at most three vehicle trajectories passing through a cell simultaneously ( $k \leq 3$ ) [28]. Then, the probability of traffic conflict in cell  $\text{Rec}(i, j)$  is

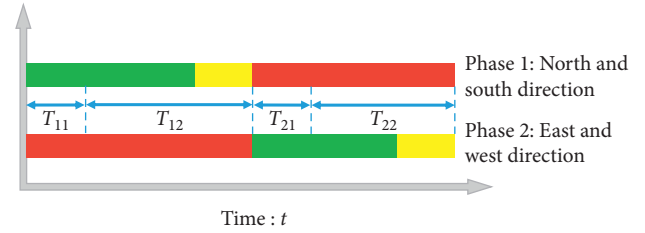


FIGURE 4: Diagram of phase decomposition of the signalized intersection (green, yellow, and red indicate the duration of the green, yellow, and red lights respectively).

$$P_{ij} = \sum_{k=1}^3 (P_{Aij}^k)^2 + P_{Aij}^1 \times P_{Aij}^2 + P_{Aij}^1 \times P_{Aij}^3 + P_{Aij}^2 \times P_{Aij}^3 + P_{Aij}^1 \times P_{Aij}^2 \times P_{Aij}^3, \quad (5)$$

$$P_{Aij}^k = P_{Dij}^k \times P_{Lij}^k, \quad k \leq 3,$$

where  $P_{ij}$  denotes the conflict probability of cell  $\text{Rec}(i, j)$ , and  $P_{Aij}^k$  denotes the probability that the vehicle reaches cell  $\text{Rec}(i, j)$  along with vehicle trajectory  $k$ . If there is no traffic flow arriving at cell  $\text{Rec}(i, j)$  along the vehicle trajectory, the arrival probability of the vehicle in cell  $\text{Rec}(i, j)$  is 0.  $P_{Lij}^k$  denotes the longitudinal arrival probability of the vehicle in cell  $\text{Rec}(i, j)$  along vehicle trajectory  $k$ ;  $P_{Dij}^k$  denotes the probability that the vehicle appears in cell  $\text{Rec}(i, j)$  along with vehicle trajectory  $k$  due to lateral offset.

**3.2.1. The Longitudinal Arrival Probability of Vehicle considering the Interference of the Signal Light.** It is assumed that most of the vehicles travel along the path set by the motion trajectory equation and ignore the heterogeneity of individual vehicles. The probability of vehicles appearing in the cell outside the coverage area of the vehicle trajectory equation can be approximately zero. For the cell covered by the vehicle trajectory equation, the longitudinal arrival probability of the vehicle in the cell is discussed in two cases

considering the signal light. (1) If the signal light is green or yellow, the vehicles are allowed to enter the intersection, and the longitudinal arrival probability of that in the cell is not zero. (2) If the signal light is red, the vehicles are not allowed to enter the intersection, and the longitudinal arrival probability of that appearing in the cell is zero. Therefore, this paper mainly focuses on the longitudinal arrival probability of the vehicle in the cell in the first case.

Due to the limitation of signal control facilities, the way that the vehicles pass the intersection is mainly divided into two cases: one is to pass after waiting in line, and the other is to pass freely (without stopping to pass). When the signal light turns green from red, the vehicles begin to pass through the intersection successively from stop status, and their headways obey the lognormal distribution model [37]. If the green time is greater than the queue dissipation time, the headway of the vehicles arriving at the intersection in the remaining green time obeys the Poisson distribution model. In a statistical sense, the arrival probability of traffic flow during the green light can be regarded as the weighted result of the arrival probability of the above two types of traffic flows. The proportion of the two types of traffic flow can be determined by the ratio of dissipation time of the queue length to the effective green time. Hence, the longitudinal arrival probability of the vehicle appearing in cell  $\text{Rec}(i, j)$  along with trajectory  $k$  in conflict time  $t$  is

$$P_{Lij}^k = \begin{cases} \left(1 - \frac{g_d}{g_e}\right) \times (1 - e^{-q_k t}) + \frac{g_d}{g_e} \times \frac{e^{-(\ln t - \mu_1)^2 / 2\sigma_1^2}}{t\sigma_1 \sqrt{2\pi}}, & g_e > g_d, \\ \frac{1}{t\sigma_1 \sqrt{2\pi}} e^{-(\ln t - \mu_1)^2 / 2\sigma_1^2}, & g_e \leq g_d, \end{cases} \quad (6)$$

$$g_d = \frac{q_k C}{S},$$

$$t = \min \left\{ \frac{L+B}{V_1}, \frac{L+B}{V_2} \right\},$$

where  $g_d$  is the dissipation time of the queue length,  $g_e$  is the effective green time,  $q_k$  is the traffic flow along with the vehicle trajectory  $k$ ,  $C$  is the signal period,  $S$  is the saturation flow rate, and  $t$  is the conflict time. If the time difference between the two cars appearing in the same cell successively is less than  $t$ , it would be seen as a traffic conflict.  $\mu_1$  and  $\sigma_1^2$  are the moment estimations of expectation and the variance of time headway, respectively.  $L$  is the length of the vehicle,  $B$  is the width of the vehicle, and  $B = f_1 B_1 + f_2 B_2 + f_3 B_3$ .  $f_1, f_2$  and  $f_3$  are the vehicle proportions of small, medium, and large vehicles, respectively.  $B_1, B_2$ , and  $B_3$  are the width of a small vehicle, medium vehicle, and large vehicle, respectively.  $V_1$  and  $V_2$  are the designed speed of the road where traffic flow may cause traffic conflicts. All vehicles are assumed to be small cars during the example verification.

**3.2.2. The Lateral Deviation Probability in Vehicle Trajectory.** Because the driver can not completely avoid operation error, there is a certain deviation (the distance between the

centerline of the vehicle and the centerline of the lane) in lane-keeping, and the deviation is subject to a normal distribution [34, 38, 39]. Thus, the probability of straight and turning vehicles appearing in cell  $\text{Rec}(i, j)$  due to lateral deviation can be obtained from

$$P_{Dij} = \begin{cases} \int_{y_{ij}+B/2-3D/2-A/2}^{D-B/2} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-(y-\mu_2)^2/2\sigma_2^2} dy, & y_{ij} \in Z_1, \\ 1, & y_{ij} \in Z_2, \\ \int_{y_{ij}+B/2-3D/2-A/2}^{D-B/2} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-(y-\mu_2)^2/2\sigma_2^2} dy, & y_{ij} \in Z_3, \end{cases} \quad (7)$$

$$P_{Dij} = \begin{cases} \int_{|R_w-R_{dw}-B/2|}^{D-B/2} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-(R_w-\mu_2)^2/2\sigma_2^2} dR_w, & R_w \in K_1, \\ 1, & R_w \in K_2, \\ \int_{|R_w-R_{dw}+B/2|}^{D-B/2} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-(R_w-\mu_2)^2/2\sigma_2^2} dR_w, & R_w \in K_3, \end{cases} \quad (8)$$

where  $P_{Dij}$  is the probability that the vehicle appears in cell  $\text{Rec}(i, j)$  due to lateral offset;  $\mu_2$  and  $\sigma_2^2$  are the mean and variance of the lateral deviation of straight or turning vehicles, respectively. The data can be obtained by fitting the lateral deviation position with the MATLAB toolbox.

### 3.3. Safety Assessment of Signalized Intersections





**3.3.1. Risk Level Classification.** The conflict probability output by the SICP model refers to the probability that two or more vehicles appear in a cell at the same time during the conflict time. Therefore, it can be converted into the ratio between the frequency of traffic conflicts and the traffic volume in the cell within a period (TC/MPCU). Pei et al. [40] conducted statistics on TC/MPCU values of 295 intersections in 33 cities in China and obtained four value ranges of risk levels. This paper adopts green, yellow, orange, and red to represent the change in risk level from low to high based on the statistical results. Detailed values are shown in Table 1.

**3.3.2. Risk Coefficient Calculation Based on the Duration of Different Phases.** To describe the overall safety of the intersection, the risk coefficient  $\delta$  is introduced as the comprehensive measurement, and its calculation formula is as follows:

$$\delta = \sum_{i=1}^Z (\alpha_{i1} \delta_{i1} + \alpha_{i2} \delta_{i2}), \quad (9)$$

$$\alpha_{ik} = \frac{T_{ik}}{C}, k = 1, 2,$$

TABLE 1: Risk level classification.

Risk level	1	2	3	4
Conflict probability	0~0.001	0.001~0.025	0.025~0.036	≥0.036
Color				

where  $\delta$  is the risk coefficient of the intersection,  $\alpha_{i1}$  and  $\alpha_{i2}$  are the proportion of the duration of phase switching stage and nonphase switching stage of phase  $i$  in a signal period, respectively, and  $\sum \alpha_{ik} = 1$ .  $\delta_{i1}$  and  $\delta_{i2}$  are the risk coefficients of phase switching stage and nonphase switching stage in phase  $i$ , respectively, which can be obtained as follows:

$$\delta_{ik} = \sum_{j=1}^4 \beta_j s_{ij}, k = 1, 2, \quad (10)$$

$$s_{ij} = \frac{n_{ij}}{m \times n}$$

where  $\beta_j$  is the weight of risk level  $j$ , which is used for risk calculation and analysis. The high-risk level is accompanied by large weight  $\beta_j$ .  $s_{ij}$  represents the ratio of the number of cells with risk level  $j$  to the total number of cells in phase  $i$ , and  $n_{ij}$  represents the number of cells with risk level  $j$  in phase  $i$ .

## 4. Model Verification

**4.1. Data Collection.** A SONY 4K HD camera was used to collect the traffic data of the intersection of Wushan Road and Yuehan Road in Guangzhou, China, from 16:00 to 18:00 on August 12, 2021. The collected parameters mainly include the geometric parameters of the intersection, the average speed of the traffic flow, traffic volume, steering ratio, large vehicle ratio, lateral deviation, time headway, and signal timing scheme. Among them, the geometric data of intersection is obtained by means of an artificial survey with a tape measure. The traffic speed is calculated by counting the time that the vehicles pass through the preset speed measuring interval after the green light (ignoring the first 5 vehicles), and the average value is taken. The calibration of all parameters is shown in Table 2. The signal timing scheme, traffic volume, steering ratio, large vehicle ratio, lateral deviation, and time headway are obtained through manual statistics. The signal timing scheme and traffic volume are shown in Table 3.

The data from the above table is input into the SICP model, and the heat maps of the predicted conflict probability distribution of four-phase stages are obtained, as shown in Figure 5. Figure 5(a) shows the heat map of traffic conflict probability in phase 1. At this time, part of the traffic flow on the north entrance of the previous phase does not completely leave the intersection, and the traffic flow on the east and west entrance is released at the same time. Figure 5(b) shows the heat map of traffic conflict probability in phase 2. The east-west traffic flow is normal, and the north traffic flow has completely left the intersection. Figures 5(c) and 5(d) show the heat map of traffic conflict probability in phase 3 and 4. Due to the setting of long full red time, the

traffic flow at the east-west entrance has completely left the intersection, and only the traffic flow at the north entrance can pass normally.

**4.2. Comparison of Conflict Distribution Based on SSAM.** The VISSIM simulation model is established according to the geometric parameters, traffic flow parameters, and signal timing scheme of the intersection. Using the track file data output by VISSIM, traffic conflicts are analyzed in the surrogate safety assessment model (SSAM), as shown in Figure 6.

It can be found that the SICP model predicts the traffic conflict distribution of the intersection, and the result is consistent with SSAM. There are subtle differences between the two models, because the traffic conflicts of SSAM include rear-end conflicts, which are not within the scope of the SICP model. To verify the accuracy of the SICP model, sixteen conflict cells are selected on the heat maps of SICP and SSAM. The number of traffic conflicts is calculated by the SICP model and SSAM within an hour, as shown in Table 4. However, the conflict probability in this paper is defined as the probability that two or more vehicles arrive at the same cell during conflict time  $t$ . Thus, the number of traffic conflicts of a cell in an hour is that its conflict probability is multiplied by the number of conflict times.

From Table 4, it is found that the number of traffic conflicts between SSAM and SICP is similar in terms of the same conflict positions, and a little error is inevitable. The main reason for the difference is the lack of conflict sample size, and the prediction time is only one hour. If there are enough sample data, the SICP model will be better. However, the conflict number and the evolution trend of traffic conflicts given by the SICP model are close to those of SSAM, which indicates the effectiveness of the SICP model in terms of the prediction of traffic conflicts.

## 5. Case Study

To analyze the impact of typical optimization strategies on the safety level of intersections, this study adopts the SICP model to evaluate the risk status of intersections under three strategies, including nonsignalized strategies, signal timing optimization strategies, and traffic flow control strategies. A case study is conducted to provide feasible strategies for the safety optimization of nonsignalized or signalized intersections. This study takes the T-shaped intersection of Wushan road and Yuehan road in Guangzhou, China, as a case study. In this case,  $\beta_j$  is set as 1, 3, 5, and 7 according to the risk level based on previous research experience [28, 31].

**5.1. Safety Analysis of Nonsignalized Strategy.** The SICP model is used to analyze the safety status of the case

TABLE 2: Parameter calibration.

Parameter	Value	Explanation
$D$ (m)	3.75	Lane width
$M$ (m)	21.5	The length of intersection
$N$ (m)	22.5	The width of intersection
$A$ (m)	0.5	The width of the median
$B_1$ (m)	1.78	The width of a small car
$L$ (m)	3.5	The length of vehicle
$m$	40	Cell rows
$n$	40	Cell columns
$V_j$ (km/h)	12.8	Average speed
$S$ (pcu/h)	1800	Saturated flow rate
$\mu_1$	3.25	Moment estimations of expectation of time headway
$\sigma_1^2$	10.32	Moment estimations of variance of time headway
$\mu_2$	0.294	The mean of the lateral deviation of straight or turning vehicles
$\sigma_2^2$	0.0098	The variance of the lateral deviation of straight or turning vehicles

TABLE 3: Traffic flow and signal timing parameters at the intersection of Wushan road and Yuehan road.

Approach direction	East		West		North	
Symbol	E	W	W	E	N	N
Lane function	Straight and right	Straight	Straight and left	Straight	Left	Right
Flow (pcu/h)	358	484	324	416	43	108
Large car ratio (%)	20.16	7.02	5.99	27.44	0	0
Turn ratio (%)	8.7	—	18.66	—	—	—
Phase	East-west				North	
Green time(s)	49				22	
Yellow time(s)	3				3	
Full red time(s)	25				0	
Cycle(s)	102					

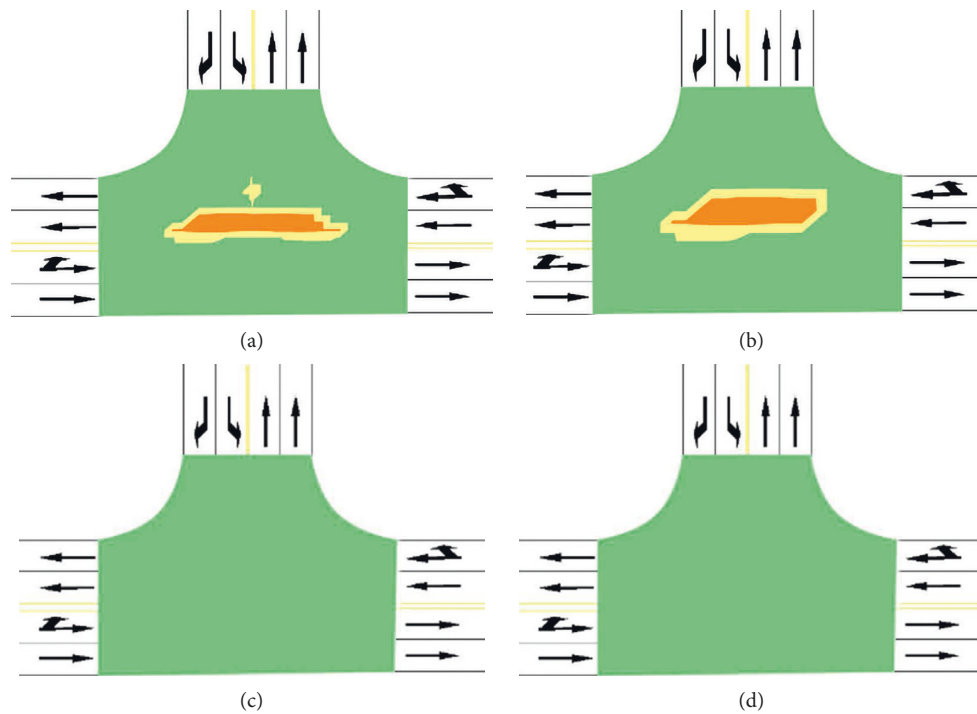


FIGURE 5: The heat map of the predicted conflict probability in different phases. (a) Phase stage 1. (b) Phase stage 2. (c) Phase stage 3. (d) Phase stage 4.



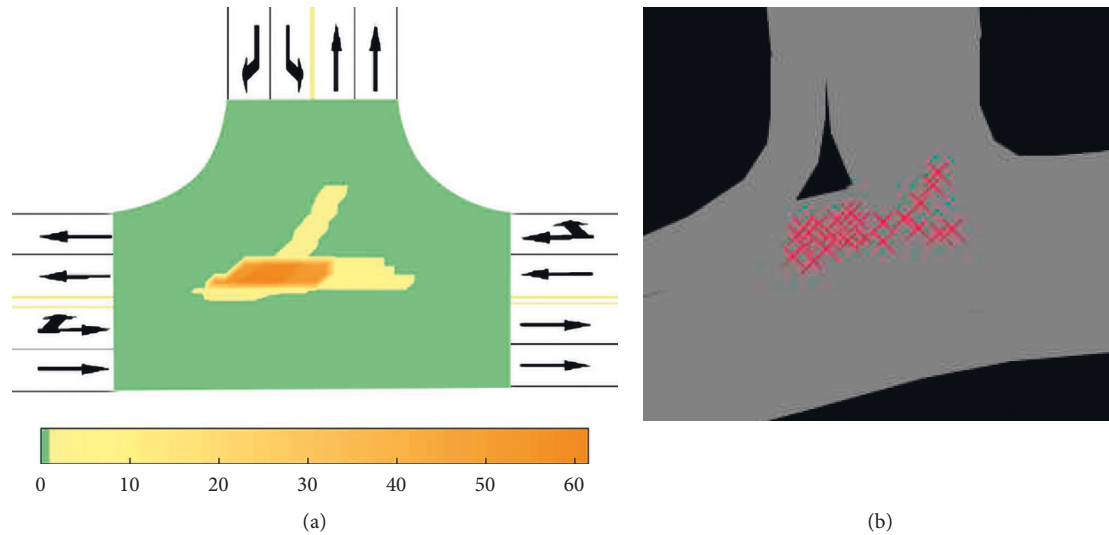


FIGURE 6: Comparisons between SSAM and SICIP model. (a) The distribution of traffic conflict predicted by SICIP. (b) The distribution of traffic conflict predicted by SSAM.

TABLE 4: Comparison of conflict number between SSAM and SICIP model.

Cell coordinate	SSAM	SICIP
(-2, 2.5)	60	59
(-2.5, 2)	69	62
(-1.5, 1.5)	74	62
(3.5, -4.5)	0	0
(0, 5)	4	2
(5, 2)	1	1
(-3.5, 1.5)	66	59
(4.9, -2.5)	0	0
(-5, -2)	0	0
(-6, 1.5)	51	52
(-2, 2)	64	62
(3, 5)	2	1
(5, 1.8)	7	8
(4.7, -3)	0	0
(3, 3)	8	7
(2, 1.5)	6	8

intersection under the nonsignalized strategy and the existing signalized strategy and compare their difference in terms of the traffic conflict distribution. Due to the significant differences of vehicle arrival characteristics between the nonsignalized strategy and the signalized strategy, the Poisson distribution model is introduced to calculate the longitudinal arrival probability of vehicles in the SICIP model [28, 31]. The heat maps of traffic conflict probability and conflict distribution under nonsignalized strategy at intersections are shown in Figure 7.

In Figure 7, the safety status of the signalized intersection changes with the signal phase, and the evolution of conflict probability and distribution area present with obvious periodicity, while the safety status of the nonsignalized intersection remains unchanged. In Figure 7(a), the traffic flow of each approach to the nonsignalized intersection can pass at any time. Thus, the traffic conflict probability of a nonsignalized intersection is static, and its conflict distribution is more intensive. Compared with phase 1 of the signalized

strategy, the conflict distribution of intersection under the nonsignalized strategy is similar, but the value of the conflict probability is smaller. The conflict area of the intersection under the nonsignalized strategy is larger than that in phase 2 of the signalized strategy, but its conflict probability is smaller. The conflict area and conflict probability of intersection under the nonsignalized strategy are both larger than those in phases 3 and 4 of the signalized strategy. In Figure 7(b), the conflict distribution of nonsignalized intersection is partially different from that of the original signalized intersection, such that conflicts are more frequent and the severe conflict area is wider. In terms of the distribution and severity of traffic conflict, the safety level of an intersection under the signalized strategy is higher than that under the nonsignalized strategy.

### 5.2. Safety Analysis of Signal Timing Optimization Strategy.

Based on the actual lane flow and the design conditions of the intersection, the mature Webster timing method is used to optimize the signal timing of the existing intersections. The signal timing parameters are shown in Table 5. The SICIP model is taken to evaluate the safety status of the intersection after signal timing optimization. The heat map of conflict probability of intersection in different phases under the signal timing optimization strategy is given in Figure 8.

From Table 5, the signal timing scheme of the Webster method is quite different from the original signal timing scheme, which is mainly reflected in the all-red intervals and green split. The all-red intervals refer to the time when the traffic lights approaches of the intersection are red. The green split refers to the ratio of the green light time of each phase to the signal cycle. In terms of all-red intervals, there are two changes between two continuous phases. After the traffic flow in the east-west direction is released, the all-red intervals are shortened to improve traffic efficiency. Furthermore, all-red intervals of 3 seconds are newly added to clear the vehicles after the traffic flow in the north direction is released. In terms of phase time, the green split of each

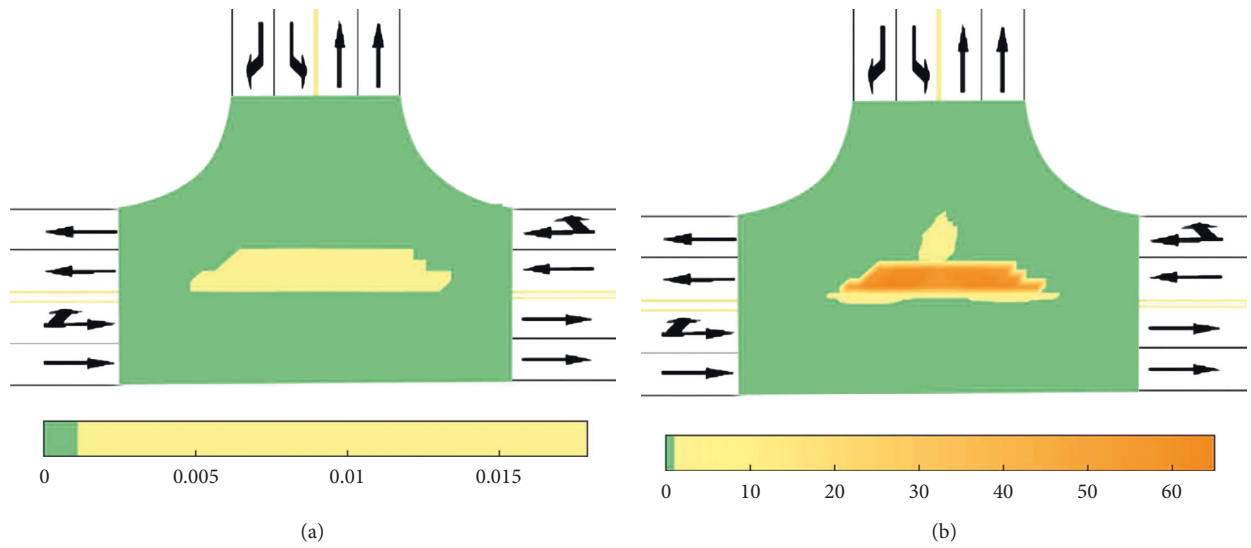


FIGURE 7: The traffic conflict probability and conflict distribution of intersection under nonsignalized strategy. (a) The heat map of traffic conflict probability. (b) Traffic conflict distribution in an hour.

TABLE 5: The signal timing parameters based on the Webster method.

Approach direction	Lane function	Phase	Green time (s)	Yellow time (s)	Full red time (s)	Cycle (s)
East	Straight and right	East-west	48	3	3	80
	Straight					
West	Straight and left	North	20	3	3	
	Straight					
North	Left	North	20	3	3	
	Right					

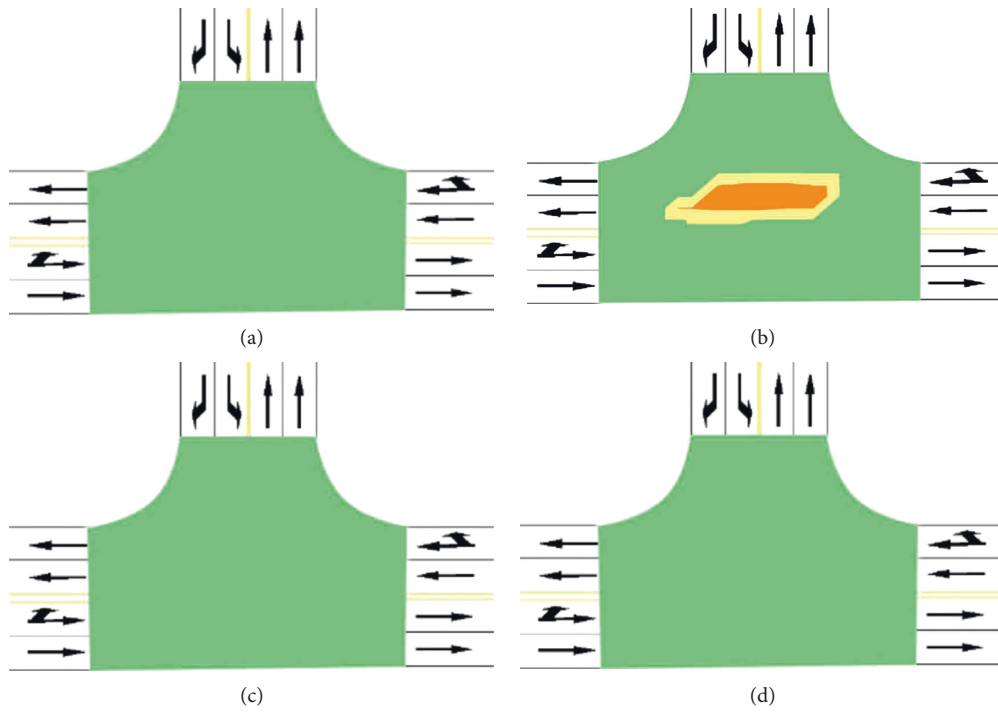


FIGURE 8: The heat map of conflict probability of intersection in different phases under signal timing optimization strategy. (a) Phase 1. (b) Phase 2. (c) Phase 3. (d) Phase 4.

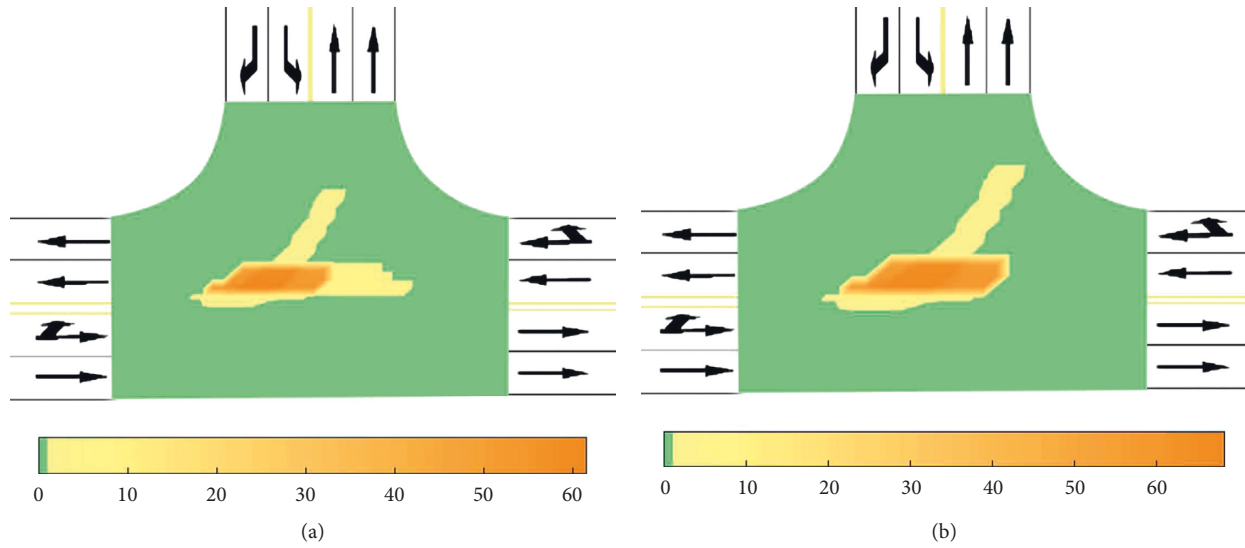


FIGURE 9: Conflict distribution of intersections before and after signal timing optimization. (a) Conflict distribution of intersection under original signal timing scheme. (b) Conflict distribution of intersection under the optimized signal timing scheme.

TABLE 6: Parameters and safety assessment of flow control measures (pcu/h).

/ Scheme	East approach		West approach		North approach		/ $\delta$
	Straight and right	Straight	Straight and left	Straight	Left	Right	
0	358	484	324	416	43	108	1.0465
1	251	339	324	416	43	108	1.0251
2	358	484	227	292	43	108	1.0451
3	358	484	324	416	31	76	1.0458

phase is positively correlated with the traffic flow. The larger traffic flow of the approach tends to go along with the larger green split of the associated phase.

From Figure 8, all red intervals of 3 seconds are added before phase 1, which avoids the traffic flow of the north approach leaving the intersection in time. The traffic conflicts between traffic flows of the north approach and the east-west approach in the original signal timing scheme are effectively eliminated. Thus, the conflict probability of phase 1 in the new signal timing scheme turns to zero. In other words, the conflict probability of the intersection is reduced after adopting the signal timing optimization scheme derived from the Webster method. The heat maps of the conflict distribution at intersections before and after signal timing optimization are shown in Figure 9.

In Figure 9, the conflict areas of the intersection under the optimized signal timing scheme significantly decline, and the conflict times slightly increase. The results show that the all-red intervals between two continuous phases can effectively cut down traffic conflict areas and improve the local safety of the intersection.

**5.3. Safety Analysis of Traffic Flow Control Strategy.** To analyze the internal mechanism between the traffic flow of the approach and the safety level of the signalized intersection, this paper adopts flow control measures for each approach.

Flow control is a commonly used management and control measure in intersections and can be implemented by restricting and guiding the traffic flow at upstream intersections. Under the condition that the design speed of each approach remains unchanged, the traffic flow of different approaches is adjusted and outputs the associated heat map of conflict distribution. This case has designed 3 kinds of traffic flow control schemes, as shown in Table 6. Scheme 0 is the existing scenario. From scheme 1 to scheme 3, the traffic flow of the East, West, and North approaches is reduced by 30%.

The safety assessment results of each flow control scheme are obtained by implementing the SICP model, and the related heat maps of the conflict times are shown in Figure 10.

From Table 6 and Figure 10, the traffic flow control strategy helps to reduce the conflict area with risk levels 3 and 4. Compared with the existing scheme, the serious conflict areas in each scheme have been lessened to varying degrees. The greater the reduction in traffic flow, the better the optimization effect on the safety level of the entire intersection, such as scheme 1 and scheme 2. To further clarify the inherent relationship between the safety status of the intersection and the flow in each approach, the traffic flow of each approach increases from 0 to 3000pcu/h, and the numerical change in the risk coefficient at the intersection is shown in Figure 11.

From Figure 11, the risk coefficient of intersections presents a stepped growth trend with the increase in the

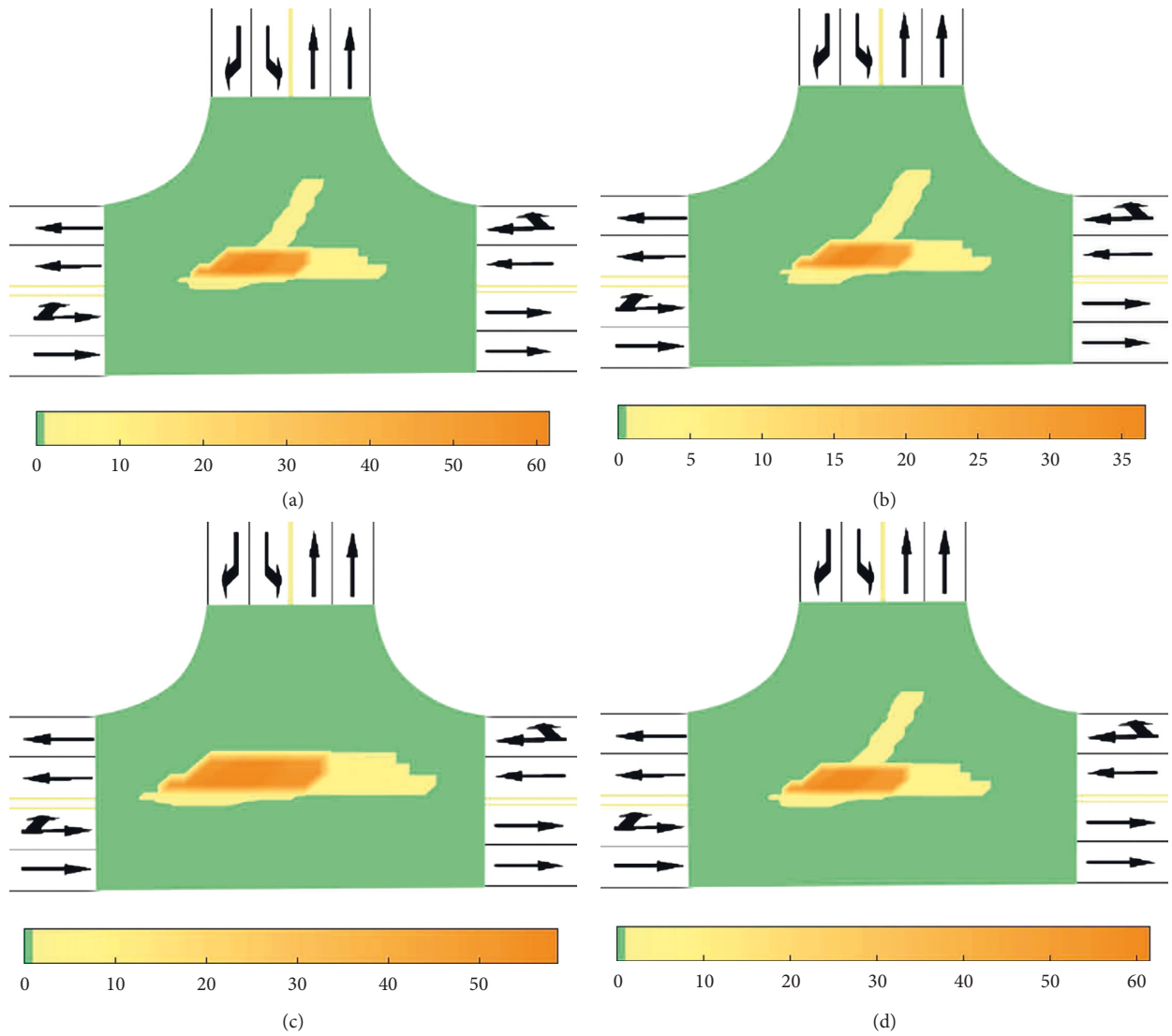


FIGURE 10: The heat maps of conflict times of intersection under different flow control schemes. (a) Existing scenario. (b) Scenario 1. (c) Scenario 2. (d) Scenario 3.

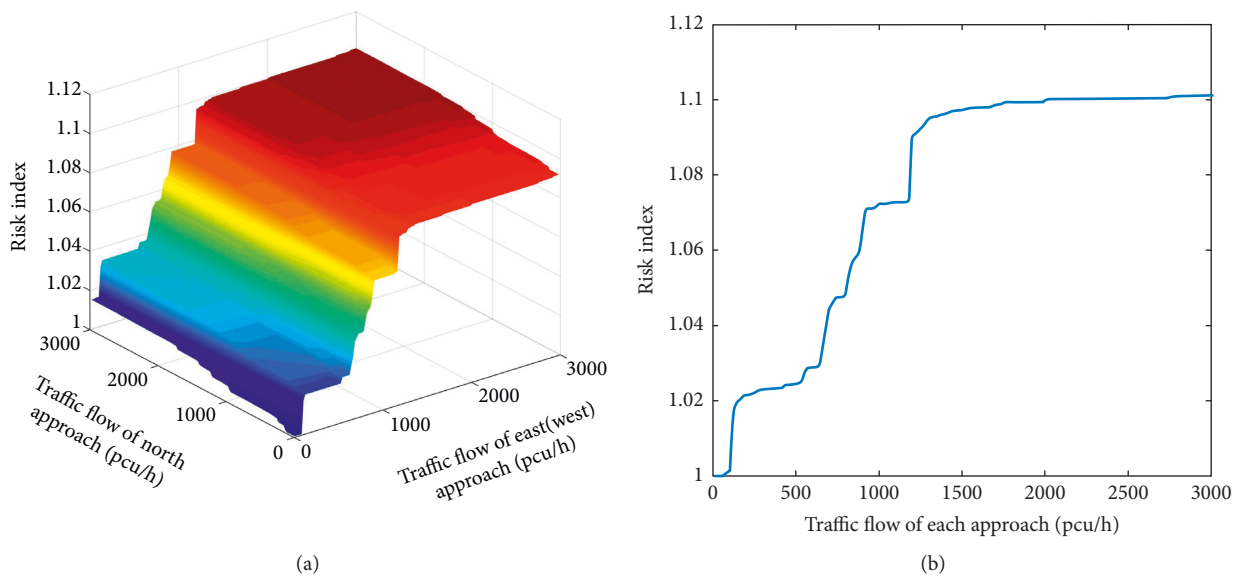


FIGURE 11: The relationship between risk index and traffic flow. (a) The relationship between risk index and the traffic flow of different approaches. (b) The relationship between risk factor and traffic flow of each approach.

traffic flow of each approach. In Figure 11(a), the risk coefficient is less affected by the changes in the traffic flow of the north approach and has a greater correlation with the traffic flow of the east-west approach. This means that the risk coefficient of the intersection is more influenced by the traffic flow of straight-left mixed lanes. In Figure 11(b), when the traffic flow of each approach increases from 0 pcu/h to 180 pcu/h, 520 pcu/h to 920 pcu/h, and 1180 pcu/h to 1300 pcu/h, the risk coefficient of the intersection rises rapidly. When the approach flow increased from 180 pcu/h to 520 pcu/h, 920 pcu/h to 1180 pcu/h, and 1300 pcu/h to 3000 pcu/h, the growth rate of the risk coefficient tended to be flat. Therefore, when a signalized intersection is allowed to release straight and left-turn vehicles at the same time and the traffic flow in that approach is greater than 1300 pcu/h, the safety level of the intersection is less affected by an increase in traffic flow.

## 6. Conclusions

To address the problem of potential traffic conflict prediction at signalized intersections, this study proposes a safety assessment method that considers the impact of signal timing and does not rely on real-time and historical data. The micro motion characteristics of vehicles are introduced to construct a vehicle conflict probability prediction model. By collecting the traffic data of actual signalized intersections, the effectiveness of the model is verified. Furthermore, the safety state of intersections under the nonsignalized strategy, signal timing optimization, and flow control strategy is exploited to provide feasible safety improvement suggestions for traffic engineers. The research findings can be summarized as follows:

- (i) The conflict probability of a signalized intersection tends to vary periodically with the change in phase, which cannot be described by the static safety assessment method.
- (ii) Under the same conditions, the overall safety level of signalized intersections is higher than that of nonsignalized intersections, with fewer conflicts per unit time and smaller conflict areas.
- (iii) The optimization of the signal timing scheme based on the Webster method is helpful to improve the safety of signalized intersections and effectively reduce the probability and number of traffic conflicts.
- (iv) The risk coefficient of a signalized intersection is significantly affected by the traffic flow of straight and left mixed lanes and goes up with the increasing traffic flow. When the approach flow of the intersection is greater than a critical value, the growth rate of the risk coefficient tends to be stable.

Although some encouraging progress has been made, developing an appropriate safety assessment of signalized intersections with the high-precision prediction of traffic conflict is less successful and still requires further research. Due to the heterogeneity of intersection geometric

conditions and drivers' driving habits, it is difficult to use a unified vehicle trajectory equation to summarize the vehicle trajectory at each approach. Thus, future research work will focus on improving the vehicle trajectory modeling of each approach. Additionally, the prediction effect of the SICIP model is easily affected by parameter calibration and deserves further development. More traffic data of typical intersections will be introduced to calibrate the model in the future, to improve the prediction accuracy of the model.

## Data Availability

All data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

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

## Acknowledgments

The authors acknowledge Dianchen Zhu from the Hong Kong Polytechnic University for his valuable comments on this research. This paper was supported by the General Project of the National Natural Science Foundation of China (grant no. 52072129) and the Natural Science Foundation of Guangdong Province (grant no. 2018A030313250).

## References

- [1] M. Ma, X. Yan, C. Wu, and H. Yin, "The effects of principal influential factors on traffic accident frequency at controlled intersections," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 40, no. 2, pp. 417–422, 2010.
- [2] M. He, X. Guo, Y. Chen, and L. Guo, "Application of improved gray clustering method in urban traffic safety assessment [J]," *Journal of Transport Information and Safety*, vol. 28, no. 1, pp. 104–107, 2010.
- [3] A. C. Lorion and B. Persaud, "Investigation of surrogate measures for safety assessment of urban two-way stop controlled intersections," *Canadian Journal of Civil Engineering*, vol. 42, no. 12, pp. 987–992, 2015.
- [4] A. S. Al-Ghamdi, "Analysis of traffic accidents at urban intersections in Riyadh," *Accident Analysis & Prevention*, vol. 35, no. 5, pp. 717–724, 2003.
- [5] L. Guo, J. Zhou, S. Dong, and S. C. Zhang, "Analysis of urban road traffic accidents based on improved k-means algorithm," *China Journal of Highway and Transport*, vol. 31, no. 4, pp. 270–279, 2018.
- [6] Y. Chen, H. Yuan, Z. Huang, and L. Wang, "Modeling intersection traffic crashes using a zero-truncated negative binomial model," *China Journal of Highway and Transport*, vol. 33, no. 4, pp. 146–154, 2020.
- [7] U. Engel, "To what extent do conflict studies replace accident analysis: validation of conflict studies, an international review [C]. Organisme national," *de securite routiere Pmceedings of "Evaluation 85" Paris*, vol. 85, pp. 324–343, 1985.
- [8] J. Glennon, "Critique of the traffic-conflict technique," *Transportation Research Record*, vol. 630, pp. 32–38, 1977.

- [9] D. Migletz, W. Glauz, and K. Bauer, *Relationships between Traffic Conflicts and Accidents Volume I-Executive Summary*, Federal Highway Administration, USA, 1985.
- [10] M. J. Williams, "Validity of the traffic conflicts technique," *Accident Analysis & Prevention*, vol. 13, no. 2, pp. 133–145, 1981.
- [11] B. Zimolong and H. Erke, *Experimental Validation of Traffic Conflict Technique*, Technical University of Braunschweig, West Germany, 1977.
- [12] C. Zegeer and R. Deen, "Traffic conflicts as a diagnostic tool in highway safety [J]. Transportation Research Record," *Journal of the Transportation Research Board*, vol. 667, no. 667, pp. 48–55, 1978.
- [13] B. Zimolong, H. Erke, and H. Gstalter, *Traffic Conflicts at Urban Junctions: Reliability and Validity Studies*, University of Technology, Beijing, China, 1983.
- [14] Y. Guo and M. T. M. M. S. Essa, "A comparison between simulated and field-measured conflicts for safety assessment of signalized intersections in Australia," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 96–110, 2019.
- [15] Y. Guo, T. Sayed, and M. Essa, "Real-time conflict-based Bayesian Tobit models for safety evaluation of signalized intersections," *Accident Analysis & Prevention*, vol. 144, Article ID 105660, 2020.
- [16] Y. Guo, T. Sayed, and L. Zheng, "A hierarchical bayesian peak over threshold approach for conflict-based before-after safety evaluation of leading pedestrian intervals," *Accident Analysis & Prevention*, vol. 147, Article ID 105772, 2020.
- [17] K. Gao, F. Han, P. Dong, N. Xiong, and R. Du, "Connected vehicle as a mobile sensor for real time queue length at signalized intersections," *Sensors*, vol. 19, no. 9, p. 2059, 2019.
- [18] L. Hu, J. Ou, J. Huang, and F. Y. B. H. L. Wang, "Safety evaluation of pedestrian-vehicle interaction at signalized intersections in changsha, China," *Journal of Transportation Safety & Security*, pp. 1–26, 2021.
- [19] L. Hu, X. Wu, J. Huang, and Y. W. Peng, "Investigation of clusters and injuries in pedestrian crashes using GIS in Changsha, China," *Safety Science*, vol. 127, Article ID 104710, 2020.
- [20] V. Guttinger, "Conflict observation in theory and practice," in *Proceedings of the International Calibration Study of Traffic Conflict Techniques*, Berlin, Germany, 1984.
- [21] C. Hyden and L. Linderholm, "The Swedish traffic-conflicts technique," in *Proceedings of the International Calibration Study of Traffic Conflict Techniques*, Springer Berlin, Heidelberg, Germany, January 1984.
- [22] S. Kamijo and Y. K. M. Matsushita, "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 108–118, 2000.
- [23] T. Sayed and S. Zein, "Traffic conflict standards for intersections," *Transportation Planning and Technology*, vol. 22, no. 4, pp. 309–323, 1999.
- [24] D. Gettman and L. Head, "Surrogate safety measures from traffic simulation models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1840, no. 1, pp. 104–115, 2003.
- [25] J. C. Hayward, "Near-miss determination through use of a scale of danger," *Highway Research Record*, vol. HRR384, 1972.
- [26] B. Allen, B. Shin, and P. Cooper, "Analysis of traffic conflict and collisions [J]. Transportation Research Record," *Journal of the Transportation Research Board*, vol. 667, pp. 67–74, 1978.
- [27] P. Cooper, "Experience with traffic conflicts in Canada with emphasis on "post encroachment time" techniques," in *Proceedings of the International calibration study of traffic conflict techniques*, Springer Berlin, Heidelberg, Germany, May 1984.
- [28] Y. Ma, X. Qin, O. Grembek, and Z. Chen, "Developing a safety heatmap of uncontrolled intersections using both conflict probability and severity," *Accident Analysis & Prevention*, vol. 113, pp. 303–316, 2018.
- [29] H. Park, C. Oh, J. Moon, and S. Kim, "Development of a lane change risk index using vehicle trajectory data," *Accident Analysis & Prevention*, vol. 110, pp. 1–8, 2018.
- [30] C. Oh, S. Park, and S. G. Ritchie, "A method for identifying rear-end collision risks using inductive loop detectors," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 295–301, 2006.
- [31] H. Wen, J. Wu, W. Qi, L. Wu, and K. Zhang, "CP-CS fusion model for on-ramp merging area in highway [J]," *Journal of South China University of Technology*, vol. 48, no. 2, pp. 50–57, 2020.
- [32] E. Mohamed and S. Tarek, "Traffic conflict models to evaluate the safety of signalized intersections at the cycle level," *Transportation Research Part C*, vol. 89, pp. 289–302, 2018.
- [33] yanyong Guo, *Signalized Intersection Safety Evaluation Techniques Based on Traffic Conflict Theory*, Southeast University, Bangladesh, 2016.
- [34] Y. Ma, S. Qi, H. Wu, and L. Fan, "Traffic conflict identification model based on post encroachment time algorithm in ramp merging area," *Journal of Transportation Systems Engineering and Information Technology*, vol. 18, no. 2, pp. 142–148, 2018.
- [35] A. Laureshyn, T. De Ceunynck, and C. Karlsson, "In search of the severity dimension of traffic events: extended Delta-V as a traffic conflict indicator," *Accident Analysis & Prevention*, vol. 98, pp. 46–56, 2017.
- [36] C. Hyden, "The development of a method for traffic safety evaluation," *The Swedish Traffic Conflicts Technique*, Bulletin Lund Institute of Technology, Sweden, 1987.
- [37] X. Jin, Y. Zhang, F. Wang, and L. D. Y. Z. Li, "Departure headways at signalized intersections: a log-normal distribution model approach," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 3, pp. 318–327, 2009.
- [38] J. Wu, H. Wen, and W. Qi, "A new method of temporal and spatial risk estimation for lane change considering conventional recognition defects," *Accident Analysis & Prevention*, vol. 148, 2020.
- [39] Y. Wang, S. Yang, and B. Pan, "Research of vehicle running track in highway straight section [J]," *Journal of Highway and Transportation Research and Development*, vol. 33, no. 2, pp. 111–119, 2016.
- [40] Y. Pei, *Road Traffic Safety*, p. 253, China Communications Press, Beijing, China, 2007.

## Research Article

# Spatiotemporal DeepWalk Gated Recurrent Neural Network: A Deep Learning Framework for Traffic Learning and Forecasting

Jian Yang <sup>1,2</sup>, Jinhong Li <sup>1,2</sup>, Lu Wei <sup>1</sup>, Lei Gao <sup>1,2</sup> and Fuqi Mao <sup>2</sup>

<sup>1</sup>Beijing Key Lab of Urban Road Traffic Intelligent Technology, North China University of Technology, Beijing 100144, China

<sup>2</sup>School of Computer Science and Technology, North China University of Technology, Beijing 100144, China

Correspondence should be addressed to Jian Yang; [yanj200045@163.com](mailto:yanj200045@163.com)

Received 12 August 2021; Revised 7 March 2022; Accepted 11 March 2022; Published 18 April 2022

Academic Editor: Ming Xu

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

As a typical spatiotemporal problem, there are three main challenges in traffic forecasting. First, the road network is a nonregular topology, and it is difficult to extract complex spatial dependence accurately. Second, there are short- and long-term dependencies between traffic dates. Third, there are many other factors besides the influence of spatiotemporal dependence, such as semantic characteristics. To address these issues, we propose a spatiotemporal DeepWalk gated recurrent unit model (ST-DWGRU), a deep learning framework that fuses spatial, temporal, and semantic features for traffic speed forecasting. In the framework, the spatial dependency between nodes of an entire road network is extracted by graph convolutional network (GCN), whereas the temporal dependency between speeds is captured by a gated recurrent unit network (GRU). DeepWalk is used to extract semantic information from road networks. Three publicly available datasets with different time granularities of 15, 30, and 60 min are used to validate the short- and long-time prediction effect of this model. The results show that the ST-DWGRU model significantly outperforms the state-of-the-art baselines.

## 1. Introduction

With the advancement of society and economy, traffic congestion is an inevitable common problem in big cities, which affects the development of cities and people's travel safety. Traffic speed prediction is a basic and important function of the intelligent transportation system (ITS). Accurate and real-time traffic speed prediction can effectively save travel costs and provide a reliable traffic management support for traffic management and route recommendation. However, traffic forecasting has many challenges because of traffic changes in different locations throughout the different timestamps and weather and accidents, etc., and factors will also have an impact on traffic. Therefore, there are complex spatiotemporal relationships in traffic data.

Several temporal- and spatial-dependent prediction methods are reported in the existing literature. The analysis methods are represented by the time-series analysis model "auto-regressive integrated moving average" (ARIMA) [1, 2]

and the Kalman filtering model [3, 4]. However, these models cannot effectively respond to the nonlinear and uncertain characteristics of the traffic data. Meanwhile, with the development of deep learning, numerous intelligent traffic prediction models have emerged [5, 6]. But, these models ignore the spatial characteristics in the traffic network. Several studies [7, 8] have used convolutional neural networks (CNN) to learn the spatial feature of road networks according to historical traffic maps. But, CNN is suitable for extracting Euclidean spatial features while the traffic road network involves a complex non-Euclidean spatial network. To solve this problem, graph convolution-based methods have been extensively developed recently [9, 10]. In summary, most of the existing literature considers only the traffic temporal and spatial correlations for traffic prediction, without considering the semantic information of the traffic road network; for example, similar functional areas have similar traffic patterns.

To better capture the complex spatiotemporal dependencies and semantic correlation hidden in traffic data, we

further propose a spatiotemporal DeepWalk gated recurrent unit model (ST-DWGRU), which can simultaneously learn the temporal, spatial, and semantic correlations of the road network. When validated using three real traffic datasets, the proposed model outperforms the state-of-the-art traffic forecasting baselines.

The main contributions of our work are as follows:

- (1) The ST-DWGRU model learns the spatial correlation between traffic speeds by GCN. The connectivity between traffic roads is represented by the adjacency matrix. Spatial correlation in traffic speed can be learned effectively by GCN
- (2) GRU was introduced in order to learn the temporal correlation between traffic speed data. Traffic speed itself is time-series, and historical traffic speed data and future data have time-series relationship, and GRU is used to obtain the hidden time-series relationship between traffic speed
- (3) Since the traffic patterns are different between different locations, DeepWalk is a position-aware graph embedding algorithm. DeepWalk can effectively obtain semantic information and enhance the accuracy of traffic speed prediction
- (4) Spatial features, temporal features, and semantic features are fused to improve the predictive power of the model. Traffic speed data are affected by other factors besides spatial and temporal correlation, such as semantic correlation, and the ST-DWGRU model that incorporates semantic correlation can effectively improve prediction accuracy.

## 2. Literature Review

Traffic forecasting problem is to predict traffic indicators such as traffic volume, speed, and travel time at a certain location at a certain time. Forecasting methods can generally be divided into two types of approaches: model-based and data-driven forecasting methods. The model-based forecasting methods rely on queueing theory [11] and traffic velocity models [12]. These models require some assumptions and prior knowledge. However, in reality, there are many factors that can have an impact on traffic such as weather and unforeseen events. Therefore, it is difficult for the model-based methods to accurately predict.

The data-driven methods are more flexible because they discover patterns from historical data and automatically deduce intrinsic connections between data without requiring many assumptions. These methods are divided into statistical prediction and machine learning methods. The main statistical prediction methods include ARIMA, linear regression models, the Kalman filtering model, and exponential smoothing (ES). For example, Ahmed and Cook [13] first introduced ARIMA to the traffic-forecasting problem. Hamed and others [14] used a simple ARIMA model to predict traffic volumes on urban arterials with good results. Ding and others [15] proposed a spatiotemporal STARIMA model for predicting traffic volumes. Moreover, there are

various variants of the ARIMA model for predicting traffic, such as Kohonen map ARIMA [16] and seasonal ARIMA [17]. Sun et al. [18] used a local linear predictor to address the issue of interval forecasting. Guo et al. [19] used an adaptive Kalman filtering model to predict traffic flow. Hinsbergen et al. [20] used an extended Kalman filtering model to estimate traffic state. Williams et al. [21] used seasonal ARIMA and winter's ES model to perform traffic flow prediction.

Statistical prediction methods do not respond well to traffic uncertainty. Compared with the statistical methods, the machine learning methods are more flexible. The machine learning methods are mainly used to learn traffic patterns in the road network using large amounts of historical data, and the main methods include the K-nearest neighbor algorithm model (KNN) [22], SVM model [23–25], and neural network (NN) model [26]. KNN predicts traffic speed by the distance between features; however, it has high time complexity and space complexity. SVM uses kernel functions for traffic prediction, but it is relatively difficult to find the right kernel function.

Recently, deep learning methods have evolved rapidly. Huang et al. [27] used a deep belief network (DBN) to learn features, a multitask regression layer for traffic flow prediction. Tan et al. [28] used DBN based on restricted Boltzmann machines for traffic flow prediction. However, none of these methods considered long-time dependencies in traffic; therefore, to solve this problem, an RNN and its variants LSTM and GRU were used to learn temporal features in traffic. Tian and others [29] used LSTM NN to predict short-term traffic flow; Fu and others [5] used LSTM and GRU NN to predict traffic flow. All these methods consider only the temporal dependency but ignore the spatial dependency. Wu et al. [30] proposed a novel deep architecture that combined CNN and LSTM to predict traffic flow (CLTFP). A 1-dimension CNN is used to capture spatial features of traffic flow, and two LSTMs are used to mine short-term variability and traffic flow periodicities. Cao et al. [31] proposed an interactive temporal recurrent convolution network for traffic prediction, where the CNN part learns network traffic as images to capture network-wide services' correlations, and the GRU part learns temporal features to help the interactive network traffic prediction. Although the above methods have made considerable progress in spatiotemporal learning, the CNNs are more effective in Euclidean space and cannot obtain spatial features more accurately for complex road network structures. Recently, with the development of graph CNN, it can acquire the spatial features of complex road networks more effectively. Yu et al. [32] proposed an ST-GCNN model for traffic prediction and achieved good results; Cui et al. [33] proposed a TGC-LSTM model to learn the interactions between roadways in the traffic network and proposed a network-wide traffic state; Zhao et al. [10] combined the GCN and GRU models for traffic prediction; Bogaerts et al. [34] combined the GCN and LSTM for long- and short-term traffic prediction using trajectory (GPS) data to achieve good results. However, the abovementioned methods only considered spatial and temporal dependencies, ignoring other



factors that affect traffic, such as emergencies. Wu et al. [35] analyzed the impact of weather and accidents on traffic; Zhang et al. [36] proposed a spatiotemporal residual network (ST-ResNet) model to predict the traffic flow while integrating weather factors; Yao et al. [8] proposed a deep multiview spatial-temporal network (DMVST-Net) model to learn temporal, spatial, and semantic features simultaneously; Song et al. [37] propose a match-then-predict method which integrates contextual matching and time-series prediction based on group method of data handling (GMDH) algorithm; Qu et al. [38] used deep neural networks to predict daily traffic flows while considering historical traffic flow data and contextual factor data. Ma et al. [39] propose a novel deep-learning-based method for daily traffic flow forecasting by taking contextual factors and traffic flow patterns into account. Ma et al. [40] use contextual factors to select historical days with the similar pattern to the target day as the training data for prediction algorithm. However, these methods do not consider the non-Euclidean topological relationship of the road network through clustering or CNN and cannot fully explore the spatial correlation and temporal correlation.

Based on the abovementioned background, this study proposes a new NN model that learns spatial, temporal, and semantic information simultaneously. Moreover, it makes accurate short- and long-term traffic predictions based on urban traffic road network information.

### 3. Methodology

**3.1. Problem Definition.** This study aims to predict future traffic speed after a time step  $T$  based on historical traffic speed information. To predict the traffic speed more accurately, this study not only considers the traffic-timing information and the spatial structure characteristics of the road network but also adds the semantic information of the road network, where similar functional areas have similar traffic patterns, as shown in the objective function.

$$y_{t+T} = f(x_{t-h,\dots,t}; m_{t-h,\dots,t}; A), \quad (1)$$

where  $T$  denotes the predicted time step,  $x_t$  denotes the historical traffic speed data,  $m_{t-h,\dots,t}$  denotes the semantic information of the road network, and  $A$  denotes the road network structure information. The objective of the model is to derive the function  $f$  from the complex temporal, spatial, and semantic information. As shown in Figure 1, the inputs to the model are the historical traffic speed data  $x_1, \dots, x_t$ , the road network structure matrix, and the final traffic semantic matrix.

**3.2. Traffic Speed Prediction Based on ST-DWGRU.** In this section, we elaborate on the architecture of the ensemble deep learning framework (ST-DWGRU). As shown in Figure 2, ST-DWGRU is composed of several ST-DWGRU layers, each of which contains three components: spatial feature extraction component, temporal feature extraction component, and semantic feature extraction component.

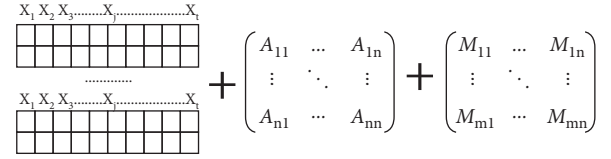


FIGURE 1: Model input.

The final prediction component outputs the prediction results.

In addition, each layer of ST-DWGRU learns spatiotemporal feature and semantic feature, which is different from the previous prediction model. Since most graph neural networks compute node embedding by aggregating information from each node's  $q$ -hop neighborhood and are thus structure-aware, GCNs cannot fully learn the location information of nodes [41]. The role of different location intersections in the road network is different, so we need the position information embedded in the nodes to represent the semantic information of different intersections. The proposed ST-DWGRU learns spatiotemporal feature and semantic feature, which is more suited to the actual situation. The details of each component are described as follows.

**3.3. Spatial Feature Extraction Component.** A CNN extracts spatial effects by summing the weights of surrounding pixels, which is particularly effective for Euclidean space. However, a CNN cannot be directly used for extracting the spatial features of road network structure because the number of neighboring intersections or road segments around the intersection or road segment in the actual road network is not fixed. Defferrard et al. [42] defined the convolution operation for graph structures, based on spectral theory. The urban road network is considered an undirected graph  $G = (V, E)$ , where  $V$  is the set of vertices in the graph and  $E$  is the set of the graph edges. Using the road network adjacency matrix as input, the graph convolution operation provides the road network structure features, and a two-layer GCN can be represented as [43].

$$\begin{aligned} Z &= f(X, A) \\ &= \text{soft max}(\hat{A}\text{ReLU}(\hat{A}XW^{(0)})W^{(1)}), \end{aligned} \quad (2)$$

where  $\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ ,  $\tilde{A} = A + I_N$  is the adjacency matrix of the road network with its connectivity,  $A$  is the adjacency matrix of the road network,  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ ,  $W^{(0)}$  is the weight parameter of the first layer, and  $W^{(1)}$  denotes the weight parameter of the second layer.

**3.4. Temporal Feature Extraction Component.** GRU is an LSTM network variant with only two gates: an updated gate and resets gate, as compared with the input gate, output gate, and forget gate of LSTM. It has only two gates: update gate and reset gate. Let the input sequence be  $X = (x_1, x_2, \dots, x_t)$  and then input to GRU to learn temporal features by GCN in

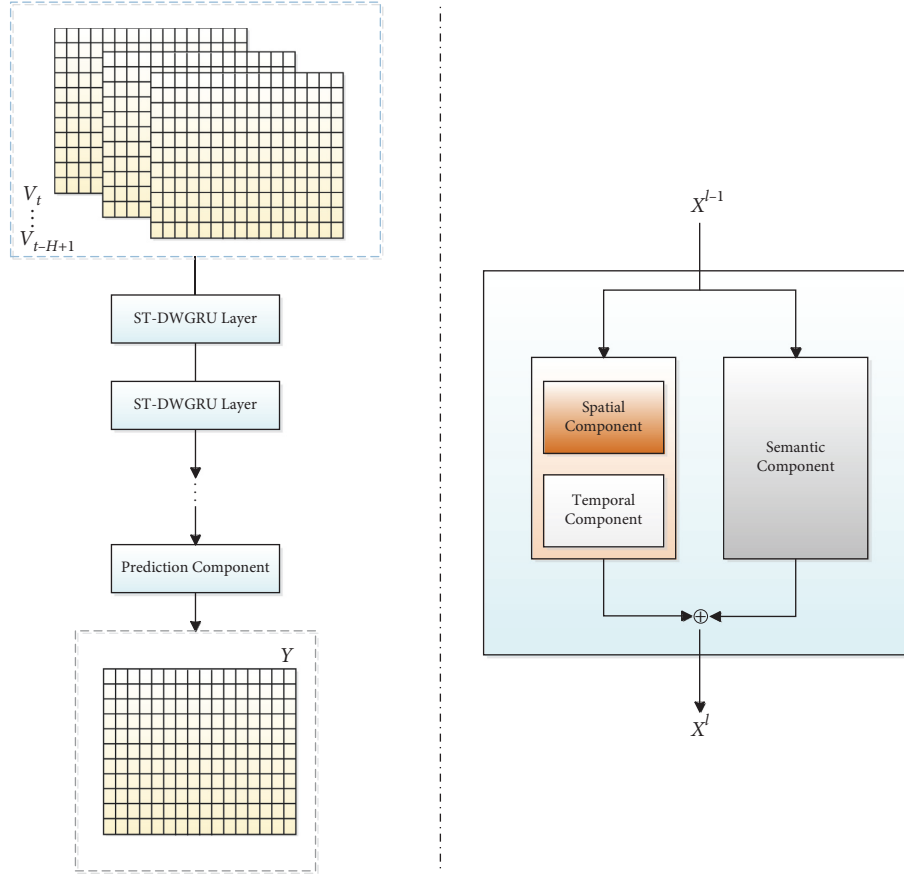


FIGURE 2: The left image is the framework of ST-DWGRU, where  $V_t$  is the traffic speed data and  $Y$  is the prediction. The right image is an ST-DWGRU layer.

advance. The specific calculation formula is shown as follows.

$$\begin{aligned}
 s_t &= GCN(x_t), \\
 r_t &= \sigma(W_r \cdot [h_{t-1}, s_t \cdot x_t] + b_r), \\
 z_t &= \sigma(W_z \cdot [h_{t-1}, s_t \cdot x_t] + b_z), \\
 \tilde{h}_t &= \tanh(W_h \cdot [r_t \odot h_{t-1}, s_t \cdot x_t] + b_h), \\
 h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t,
 \end{aligned} \tag{3}$$

where  $s_t$  is the GCN output at time  $t$ ,  $h_t$  is the hidden state at time  $t$ ,  $x_t$  is the current input,  $h_{t-1}$  is the hidden state at the previous time,  $r_t$  is the reset gate,  $z_t$  is the update gate,  $\tilde{h}_t$  is the current memory content, and  $h_t$  is the current hidden state. Where  $W$  and  $b$  denote the weights and biases of the network, respectively,  $\sigma$  denotes the sigmoid activation function, and  $\tanh$  is the tanh activation function. Therefore, the final GRU output has temporal and spatial features.

**3.5. Semantic Feature Extraction Component.** Generally, similar functional attribute areas exhibit similar traffic patterns. For example, traffic speed near city parks will be lower on weekends, while industrial parks will have smaller

traffic speed during morning and evening peak traffic periods. Therefore, the traffic pattern correlation for similar areas is relatively high, and this study uses the graph embedding technique to learn the representation of similar functions.

DeepWalk [44] is a position-aware node embedding representation learning algorithm used widely recently [41]. The algorithm is divided into two main steps: random walk and update procedure. A path with the root node is  $s_i$  denoted by  $W_{s_i}$ , and the nodes in the path are labeled as  $\{W_{s_i}^1, W_{s_i}^2, W_{s_i}^3, \dots, W_{s_i}^k\}$ , where  $W_{s_i}^k$  denotes the  $k$ th intersection or section in the path. All random walks are the same length in a truncated random walk, and the random walk sequence matrix of the entire path is obtained after traversing all intersections or road sections. The corresponding vector representation is  $s_i^j$  obtained from the Skip-gram algorithm, and the optimization objective is

$$\underset{\Phi}{\text{minimize}} \quad -\log \Pr\left(\frac{\{v_{i-w}, \dots, v_{i+w}\}}{v_i} \mid \Phi(v_i)\right). \tag{4}$$

This study obtains the final semantic information representation through a fully connected layer after obtaining the vector representation.

$$\hat{s} = f(W_i \cdot s_i + b_i), \quad (5)$$

where  $W_i$  and  $b_i$  are the learnable weights and biases.

**3.6. Prediction Component.** This study predicts traffic speed change information in the future based on the historical traffic speed information. Because the output of GRU has spatial and temporal features, the outputs of the three components are stitched together, and the concatenate operation is expressed as follows:

$$o_t^i = h_t^i \oplus \hat{s}_i. \quad (6)$$

The concatenated result is then passed through a fully connected network to obtain the predicted output of the model  $y_{t+T}^i$ , where  $T$  is the predicted time step, of the following form:

$$y_{t+T}^i = \text{sigmoid}(W_s o_t^i + b_s), \quad (7)$$

where  $W_s$  and  $b_s$  are the learnable weights and biases.

## 4. Experiments

**4.1. Dataset Description.** In the experiments of this paper, three real-world traffic datasets are used for the experiments. They are PeMSD4, PeMSD8 [45], and PeMS-BAY [46]. The three datasets are collected by California Performance of Transportation (PeMS). Details of the dataset are shown in Table 1. The sensor network of PeMS is shown in Figure 3.

PeMSD4 [45]: PeMSD4 has 307 detectors distributed across 29 roadways spanning January to February 2018. The dataset was collected from Caltrans Performance Measurement System (PeMS) and traffic data from the San Francisco Bay Area.

PeMSD8 [45]: PeMSD8 has 170 detectors distributed on 8 roads, between July and August 2016 in San Bernardino.

PeMS-BAY [46]: It is collected from Caltrans PeMS too, which has 325 sensors in the Bay Area, spanning from Jan 1, 2017, to May 31, 2017.

**4.2. Experimental Settings.** A PC (CPU: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz, Memory: 64 GB, GPU: NVIDIA TITAN XP) was used as the experimental platform, and TensorFlow1.14 was used to build the model.

In this study, the data are divided into two parts, respectively, 80% of the data as the training set, and the remaining as the test set. Predictions are performed for 15, 30, and 60 min, respectively.

The model's hyperparameters include learning rate, training rounds, number of hidden units, window size, and length of prediction time, where the learning rate is 0.001, the number of training epoch is 800, the number of hidden units is 64, the window size is 10, and the prediction time is 15, 30, and 60 min, respectively.

**4.3. Evaluation.** The root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage errors (MAPE) are used to evaluate the model. RMSE, MAE, and MAPE are defined as

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{m} \sum_{i=1}^m (Y_i - \hat{Y}_i)^2}, \\ \text{MAE} &= \frac{1}{m} \sum_{i=1}^m |Y_i - \hat{Y}_i|, \\ \text{MAPE} &= \frac{1}{m} \sum_{i=1}^m \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|, \end{aligned} \quad (8)$$

where  $m$  denotes the number of road sections and  $Y_t, \hat{Y}_t$  denote the true and predicted values of speed, respectively.

**4.4. Baselines.** The baseline methods for comparison with the ST-DWGRU are as follows:

- (i) HA : historical average
- (ii) ARIMA [17]: auto-regressive integrated moving average, which is often used in time series prediction
- (iii) STGCN [32]: spatiotemporal graph convolutional networks (STGCN), which combines graph convolution with gated CNNs
- (iv) DCRNN [47]: diffusion convolutional recurrent neural network (DCRNN), which combines bidirectional random walks with gated recurrent units
- (v) GWN [48]: graph neural network Graph WaveNet, which consists of stacked spatial-temporal layers and an output layer
- (vi) ASTGCN [49]: attention-based spatial-temporal graph convolutional network (ASTGCN), which mainly consists of three independent components to, respectively, model recent, daily-periodic, and weekly periodic dependencies
- (vii) LSGCN [50]: long short-term graph convolutional networks, which integrate both cosAtt and graph convolution networks (GCN) to handle spatial dependency and use gated linear units convolution (GLU) to capture complex temporal feature
- (viii) USTGCN [51]: unified spatiotemporal graph convolution network, which performs both spatial and temporal aggregation through spectral graph convolution on a spatiotemporal graph

**4.5. Experiment Results.** As shown in Table 2, the ST-DWGRU model performs the best in both short- and long-term predictions on all evaluation metrics. In particular, all metrics are optimal on the PeMSD4 dataset, while MAE is the second best on the PeMS-BAY dataset, the GWN model is the best, RMSE is second only to the USTGCN model on the PeMSD8 dataset, and the other metrics are still the best

TABLE 1: The details for PeMSD4, PeMSD8, and PEMS-BAY.

	PeMSD4	PeMSD8	PeMS-BAY
#Nodes	307	170	325
#Edges	340	295	2369
#Time steps	16992	17856	52116
Time span	2018/1–2018/2	2016/7–2016/8	2017/1–2017/6
Time interval	5 mins		
Daily range	0:00–24:00		

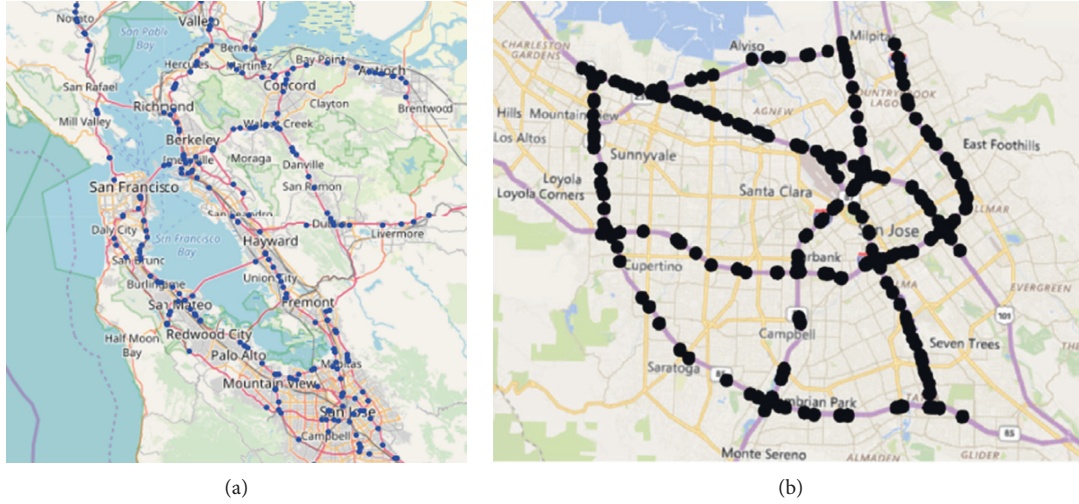


FIGURE 3: Sensor distribution of (a) PeMSD4 and (b) PeMS-BAY.

TABLE 2: Performance comparison of ST-DWGRU and other approaches on the datasets PeMSD4, PeMSD8, and PeMS-BAY.

Model	PeMSD4 (15/30/60 min)		
	MAE	RMSE	MAPE (%)
HA	2.54	4.96	5.56
ARIMA (2003)	2.51/2.75/3.21	5.72/6.34/7.36	5.32/5.69/6.56
DCRNN (2018)	1.35/1.77/2.26	2.94/4.06/5.28	2.68/3.71/5.10
STGCN (2018)	1.47/1.93/2.55	3.01/4.21/5.65	2.92/3.98/5.39
ASTGCN (2019)	2.12/2.42/2.73	3.96/4.59/5.21	4.16/4.80/5.46
GWN (2019)	<u>1.30/1.70/2.03</u>	<u>2.68/3.82/4.65</u>	<u>2.67/3.73/4.60</u>
LSGCN (2020)	1.45/1.82/2.22	2.93/3.92/4.83	2.90/3.84/4.85
USTGCN (2021)	1.40/1.64/2.03	2.69/3.19/4.25	2.81/3.23/4.32
ST-DWGRU (ours)	<b>1.20/1.48/1.90</b>	<b>2.40/3.12/4.01</b>	<b>2.21/2.75/3.53</b>
Model	PeMSD8 (15/30/60 min)		
	MAE	RMSE	MAPE (%)
HA	1.98	4.11	3.94
ARIMA (2003)	1.90/2.12/2.79	4.87/5.24/6.22	5.11/5.21/5.62
DCRNN (2018)	1.17/1.49/1.87	2.59/3.56/4.50	2.32/3.21/4.28
STGCN (2018)	1.19/1.59/2.25	2.62/3.61/4.68	2.34/3.24/4.54
ASTGCN (2019)	1.49/1.67/1.89	3.18/3.69/4.13	3.16/3.59/4.22
LSGCN (2020)	1.16/1.46/1.81	2.45/3.28/4.11	2.24/3.02/3.89
USTGCN (2021)	<u>1.14/1.25/1.70</u>	2.15/2.58/3.27	<u>2.07/2.35/3.22</u>
ST-DWGRU (ours)	<b>1.005/1.25/1.57</b>	2.08/2.70/3.49	<b>1.81/2.24/2.78</b>
Model	PeMS-BAY (15/30/60 min)		
	MAE	RMSE	MAPE (%)
HA	2.88	5.59	6.84
ARIMA (2003)	1.62/2.33/3.38	3.30/4.76/6.50	3.5/5.4/8.3
DCRNN (2018)	1.38/1.74/2.07	2.95/3.97/4.74	2.9/3.9/4.9
STGCN (2018)	1.46/2.00/2.67	3.01/4.31/5.73	2.9/4.1/5.4
GWN (2019)	<b>1.30/2.63/1.95</b>	<u>2.74/3.70/4.52</u>	<u>2.7/3.7/4.6</u>
ST-DWGRU (ours)	1.33/1.68/2.08	<b>2.52/3.28/4.17</b>	<b>2.59/3.32/4.14</b>

Bold is the best; underline is the second best.

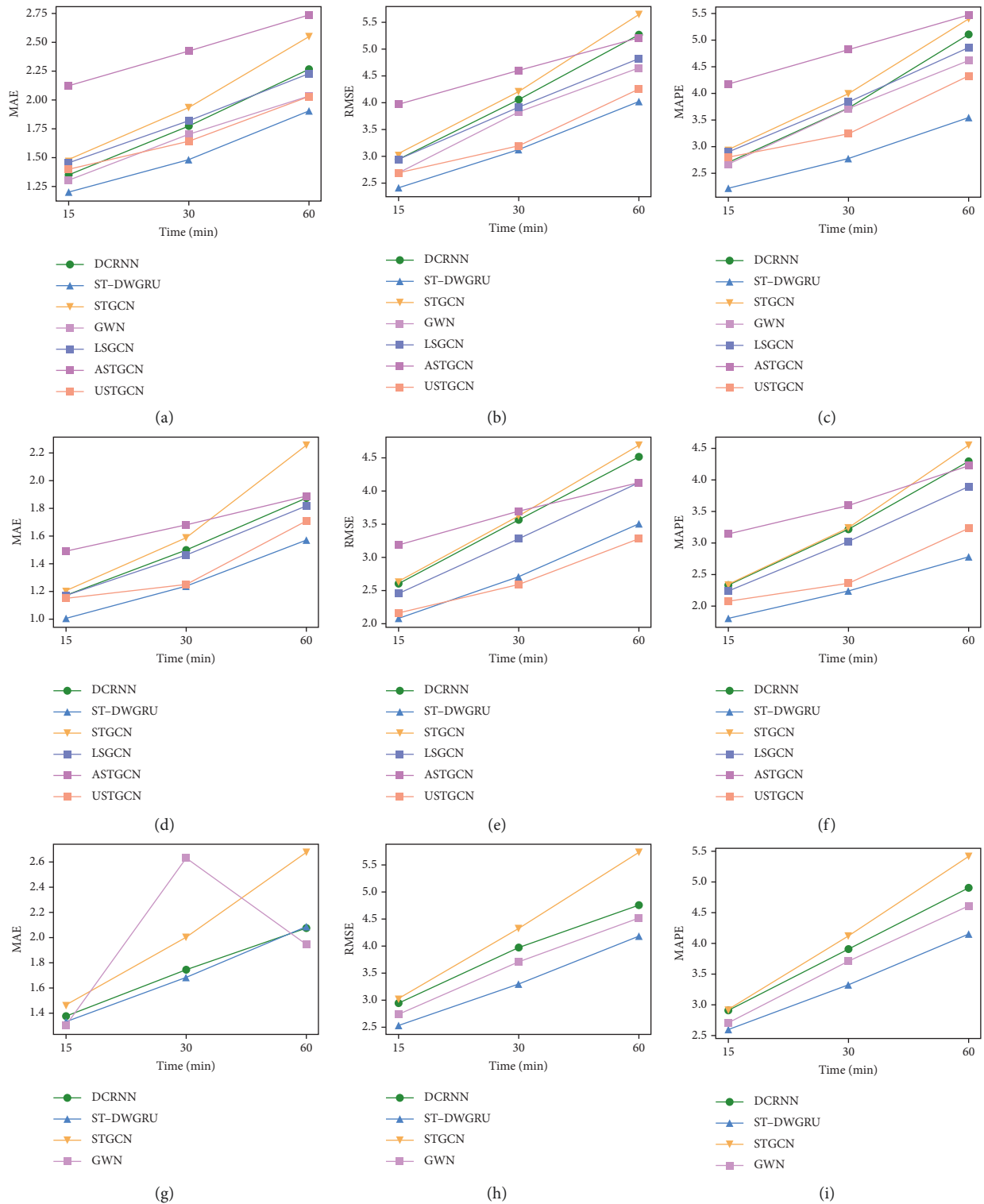


FIGURE 4: Trends in MAE and RMSE on PeMSD4, PeMSD8, and PeMS-BAY. (a) MAE of PeMSD4. (b) RMSE of PeMSD4. (c) MAPE of PeMSD4. (d) MAE of PeMSD8. (e) RMSE of PeMSD8. (f) MAPE of PeMSD8. (g) MAE of PeMS-BAY. (h) RMSE of PeMS-BAY. (i) MAPE of PeMS-BAY

on the ST-DWGRU model. From the results, it is shown that the ST-DWGRU model can capture spatiotemporal features and semantic features effectively for traffic prediction.

In contrast, traditional statistical methods such as HA and ARIMA perform the worst in short- and long-term prediction on three datasets because they cannot effectively capture complex spatiotemporal and semantic features.

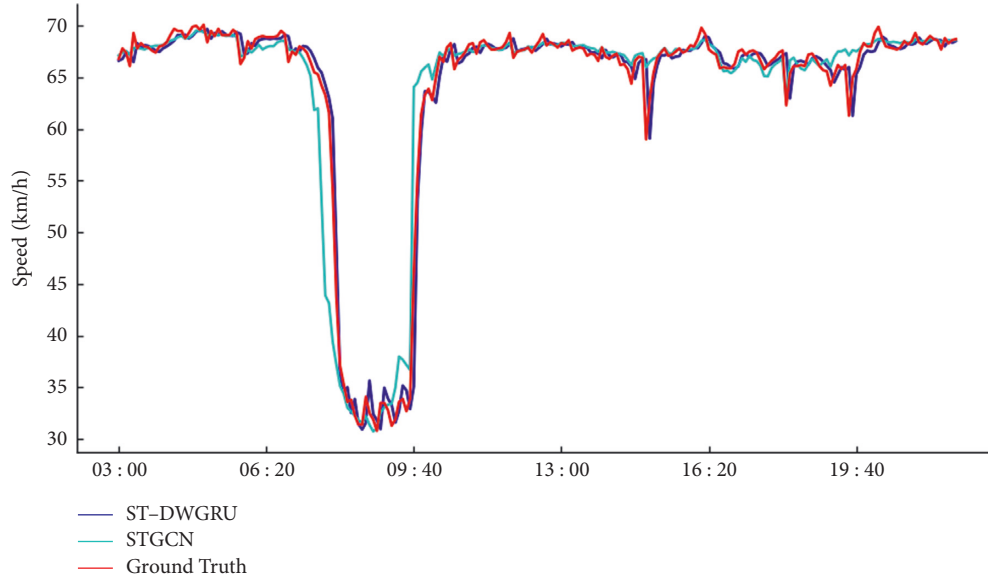


FIGURE 5: Speed prediction in extreme moments of the dataset PeMSD4.

The GWN model uses stacked spatiotemporal layers to handle spatial dependencies at different temporal levels, while the DCRNN uses diffusion convolution can effectively obtain complex spatial dependencies. Besides, it uses bidirectional graph random walk in combination with a sequence-to-sequence learning framework and scheduled sampling to obtain long-term temporal dependency. Because of the cumulative error of STGCN, STGCN does not perform as well as GWN and DCRNN.

The USTGCN performs both spatial and temporal aggregation through spectral graph convolution on a spatiotemporal graph. In addition to the relationship of spatiotemporal, USTGCN considers the important historical and current-day pattern. So, it performs better than GWN and LSGCN in the short- and long-term prediction of PeMSD4 and PeMSD8.

The ST-DWGRU model has the best overall performance on three datasets in addition to the second best MAE on the PeMS-BAY and RMSE on the PeMSD8. It is implied that in traffic prediction, in addition to spatiotemporal correlations, semantic correlation also has an important impact.

As can be seen from Figure 4, the errors of all models grow in the 15-, 30-, and 60-minute predictions on the three datasets, except for the GWN model in which the 60-minute MAE is smaller than the 30-minute on the PeMS-BAY dataset. However, the ST-DWGRU model has more flat growth. This indicates that the ST-DWGRU model is more robust on the short- and long-term forecasts.

Especially in the extreme moments of the morning and evening peaks, as the ST-DWGRU model can effectively obtain the spatiotemporal features and semantic features, the ST-DWGRU model can well predict the changes of traffic in extreme moments. The ST-DWGRU model can accurately predict extreme situations, as shown in Figure 5.

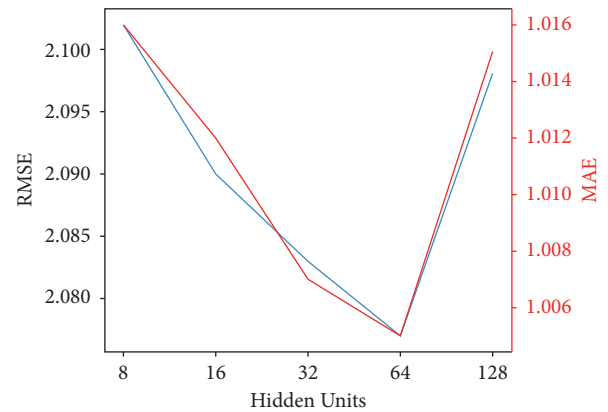


FIGURE 6: RMSE and MAE change with the number of hidden of the PeMSD8.

TABLE 3: Impact of different number of DeepWalk window size on PeMSD8 dataset.

Prediction (min)	Metrics	$W=5$	$W=10$	$W=15$	$W=20$
15	MAE	1.009	<b>1.005</b>	1.010	1.009
	RMSE	2.094	<b>2.077</b>	2.092	2.089
	MAPE	1.815	<b>1.805</b>	1.815	1.812
30	MAE	1.252	<b>1.245</b>	1.253	1.253
	RMSE	2.705	<b>2.699</b>	2.708	2.702
	MAPE	2.250	<b>2.239</b>	2.250	2.242
60	MAE	1.614	<b>1.569</b>	1.626	1.582
	RMSE	<b>3.456</b>	3.485	3.470	3.458
	MAPE	2.859	<b>2.783</b>	2.870	2.799

Bold shows the best.

4.6. *Parameter Sensitivity Analysis.* As shown in Figure 6, to determine the number of hidden units, we experimentally validate the hidden units as 8, 16, 32, 64, and 128, respectively. The results are shown in Figure 6, where the X-axis

represents the number of hidden units and the Y-axis represents the values of RMSE and MAE. As shown in the figure, when the number of hidden units is 64, the values of RMSE and MAE are minimum.

To determine how different DeepWalk window size in short- and long-term prediction, we have listed the performance of ST-DWGRU with different DeepWalk window size ( $W$ ) in Table 3. In Table 3, we can observe that in short- and long-term prediction, the performance of ST-DWGRU with  $W = 10$  is the best. With the increase of DeepWalk window size, it is not that the ST-DWGRU model is getting better. When  $W = 10$ , the ST-DWGRU model is more accurate in obtaining semantic information.

## 5. Conclusion

This paper proposes an urban road network traffic speed prediction model, which explores the potential spatiotemporal relationships and semantic information in the traffic speed data. The model is validated by applying three public datasets. Experimental results verify that compared with traditional HA and ARIMA prediction methods, the ST-DWGRU model has better prediction performance. Compared with the state-of-the-art traffic prediction methods DCRNN, STGCN, ASTGCN, GWN, LSGCN, and USTGCN, the ST-DWGRU model is also performed better. Due to the complexity of traffic, future research will focus on combining the attention mechanism, considering the influence of key road sections or intersections on traffic speed, and further exploring the effective acquisition of spatiotemporal relationships and the interpretability of the model in complex networks.

## Data Availability

Previously reported traffic data used to support this study are available. These prior studies (and datasets) are cited at relevant places within the text as references [45, 46].

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was financially supported by the project supported by the “National Natural Science Foundation of China” (61977001).

## References

- [1] L. Bai, “Urban rail transit normal and abnormal short-term passenger flow forecasting method,” *Journal of Transportation Systems Engineering and Information Technology*, vol. 17, no. 1, pp. 127–135, 2017.
- [2] X. Yang, Y. Zou, J. Tang, J. Liang, and M. Ijaz, “Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and machine learning models,” *Journal of Advanced Transportation*, vol. 2020, Article ID 9628957, 2020.
- [3] V. Marzano, A. Papola, F. Simonelli, and M. Papageorgiou, “A kalman filter for quasi-dynamic o-d flow estimation/updating,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3604–3612, 2018.
- [4] B. Li, W. Cheng, and L. Li, “Real-Time Prediction of lanebased queue lengths for signalized intersections,” *Journal of Advanced Transportation*, vol. 2018, Article ID 5020518, 2018.
- [5] F. Rui, Z. Zuo, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in *Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, Wuhan, China, November 2016.
- [6] Z. Cui, R. Ke, Z. Pu, and Y. Wang, “Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values,” *Transportation Research Part C: Emerging Technologies*, vol. 118, 2020.
- [7] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks,” *Sensors*, vol. 17, no. 7, p. 1501, 2017.
- [8] H. Yao, F. Wu, Ke. Jintao et al., “Deep multi-view spatial-temporal network for taxi demand prediction,” in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 2588–2595, New Orleans, LO, USA, February 2018.
- [9] Xi-qun. Chen, L.-xiao. Zhou, and Z. Cao, “Short-term network-wide traffic prediction based on graph convolutional network,” *Journal of Transportation Systems Engineering and Information Technology*, vol. 20, no. 4, pp. 49–55, 2020.
- [10] L. Zhao, Y. Song, C. Zhang et al., “T-GCN: a temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [11] X.-y. Xu, J. Liu, H.-y. Li, and J.-Q. Hu, “Analysis of subway station capacity with the use of queueing theory,” *Transportation Research Part C: Emerging Technologies*, vol. 38, no. 1, pp. 28–43, 2014.
- [12] W. Qi, L. I. Li, H. U. Jianming, and B. Zou, “Traffic velocity distributions for different spacings,” *Journal of Tsinghua University*, vol. 51, no. 3, pp. 309–312, 2011.
- [13] M. S. Ahmed and A. R. Cook, “Analysis of freeway traffic time-series data by using box-jenkins techniques,” *Transportation Research Record*, vol. 722, pp. 1–9, 1979.
- [14] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, “Short-term prediction of traffic volume in urban arterials,” *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.
- [15] Q. Y. Ding, X. F. Wang, X. Y. Zhang, and Z. Q. Sun, “Forecasting traffic volume with space-time ARIMA model,” *Advanced Materials Research*, vol. 156–157, pp. 979–983, 2010.
- [16] M. Van Der Voort, M. Dougherty, and S. Watson, “Combining kohonen maps with arima time series models to forecast traffic flow,” *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.
- [17] 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.
- [18] H. Sun, C. Zhang, and B. Ran, “Interval prediction for traffic time series using local linear predictor,” in *Proceedings of the International IEEE Conference on Intelligent Transportation Systems*, pp. 410–415, Washington, WA, USA, Nov. 2004.
- [19] J. Guo, W. Huang, and B. M. Williams, “Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

- [20] C. P. I. J. V. Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. V. Lint, and H. J. V. Zuylen, "Localized extended kalman filter for scalable real-time traffic state estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 385–394, 2012.
- [21] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1644, no. 1, pp. 132–141, 1998.
- [22] X. L. Zhang, H. E. Guo-Guang, and L. U. Hua-Pu, "Short-term traffic flow forecasting based on k-nearest neighbors non-parametric regression," *Journal of Systems Engineering*, vol. 24, no. 2, pp. 178–183, 2009.
- [23] J. Wang and Q. Shi, "Short-term traffic speed forecasting hybrid model based on Chaos-Wavelet Analysis-Support Vector Machine theory," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 219–232, 2013.
- [24] Y. Cong, J. Wang, and X. Li, "Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm," *Procedia Engineering*, vol. 137, pp. 59–68, 2016.
- [25] Y. Gu, D. Wei, and M. Zhao, "A new intelligent model for short time traffic flow prediction via EMD and PSO-SVM," *Lect. Notes Electr. Eng.*, vol. 113, pp. 59–66, 2012.
- [26] S. H. Huang and B. Ran, "An application of neural network on traffic speed prediction under adverse weather condition," in *Proceedings of the Transportation Research Board Annual Meeting*, no. 12–16, Washington, DC, USA, January 2003.
- [27] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.
- [28] H. Tan, X. Xuan, Y. Wu, Z. Zhong, and B. Ran, "A comparison of traffic flow prediction methods based on DBN," in *Proceedings of the Cota International Conference of Transportation Professionals*, pp. 273–283, Shanghai, China, July 2016.
- [29] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pp. 153–158, Chengdu, China, December 2015.
- [30] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," 2016, <https://arxiv.org/abs/1612.01022>.
- [31] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, "Interactive temporal recurrent convolution network for traffic prediction in data centers," *IEEE Access*, vol. 99, p. 1, 2017.
- [32] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional neural network: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
- [33] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.
- [34] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph CNN-LSTM neural network for short and longterm traffic forecasting based on trajectory data," *Transp. Res. C, Emerg. Technol.*, vol. 112, pp. 62–77, 2020.
- [35] F. Wu, H. Wang, and Z. Li, "Interpreting traffic dynamics using ubiquitous urban data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, vol. 69, ACM, 2016.
- [36] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1655–1661, San Francisco, CA, USA, 2017.
- [37] X. Song, W. Li, D. Ma, D. Wang, L. Qu, and Y. Wang, "A match-thenpredict method for daily traffic flow forecasting based on group method of data handling," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 11, pp. 982–998, 2018.
- [38] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Systems with Applications*, vol. 121, pp. 304–312, 2019.
- [39] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2627–2636, 2021.
- [40] M. Dongfang, X. B. Song, and J. C. Zhu, "Input data selection for daily traffic flow forecasting through contextual mining and intra-day pattern recognition," *Expert Systems with Applications*, vol. 176, 2021.
- [41] J. You, R. Ying, and J. Leskovec, "Position-aware graph neural networks," *ICML'19*, vol. 97, pp. 7134–7143, 2019.
- [42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 3844–3852, 2016.
- [43] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR) 2017*, Toulon, France, April 2017.
- [44] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, ACM, New York, NY, USA, August 2014.
- [45] R. Huang, C. Huang, Y. Liu, G. Dai, Kong, and Weiyang, "LSGCN: long short-term traffic prediction with graph convolutional networks," in *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*, pp. 2355–2361, 2020.
- [46] Q. Zhang, J. Chang, G. Meng, S. Xiang, and C. Pan, "Spatio-temporal graph structure learning for traffic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 1177–1185, New York, New York, USA, February 2020.
- [47] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proceedings of the ICLR*, Vancouver Convention Center, Vancouver, BC, Canada, April 2018.
- [48] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the IJCAI*, pp. 1907–1913, Macao, China, August 2019.
- [49] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatio-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI*, Honolulu, HI, USA, February 2019.



- [50] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "Lsgcn: long shortterm traffic prediction with graph convolutional networks," in *Proceedings of the IJCAI*, January 2021.
- [51] A. Roy, K. K. Roy, A. A. Ali, M. A. Amin, and A. K. M. M. Rahman, "Unified spatio-temporal modeling for traffic forecasting using graph neural network," in *Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Shenzhen, China, July 2021.

## Research Article

# Bi-GRCN: A Spatio-Temporal Traffic Flow Prediction Model Based on Graph Neural Network

Wenhao Jiang <sup>1,2</sup>, Yunpeng Xiao,<sup>3</sup> Yanbing Liu,<sup>1</sup> Qilie Liu,<sup>3</sup> and Zheng Li <sup>3</sup>

<sup>1</sup>School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

<sup>2</sup>College of Intelligent Information Engineering, Chongqing Aerospace Polytechnic, Chongqing 400021, China

<sup>3</sup>School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Zheng Li; [lizheng@cqupt.edu.cn](mailto:lizheng@cqupt.edu.cn)

Received 11 August 2021; Revised 2 January 2022; Accepted 4 January 2022; Published 1 February 2022

Academic Editor: JingXin Dong

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

Because traffic flow data has complex spatial dependence and temporal correlation, it is a challenging problem for researchers in the field of Intelligent Transportation to accurately predict traffic flow by analyzing spatio-temporal traffic data. Based on the idea of spatio-temporal data fusion, fully considering the correlation of traffic flow data in the time dimension and the dependence of spatial structure, this paper proposes a new spatio-temporal traffic flow prediction model based on Graph Neural Network (GNN), which is called Bidirectional-Graph Recurrent Convolutional Network (Bi-GRCN). First, aiming at the spatial dependence between traffic flow data and traffic roads, Graph Convolution Network (GCN) which can directly analyze complex non-Euclidean space data is selected for spatial dependence modeling, to extract the spatial dependence characteristics. Second, considering the temporal dependence of traffic flow data on historical data and future data in its time-series period, Bidirectional-Gate Recurrent Unit (Bi-GRU) is used to process historical data and future data at the same time, to learn the temporal correlation characteristics of data in the bidirectional time dimension from the input data. Finally, the full connection layer is used to fuse the extracted spatial features and the learned temporal features to optimize the prediction results so that the Bi-GRCN model can better extract the spatial dependence and temporal correlation of traffic flow data. The experimental results show that the model can not only effectively predict the short-term traffic flow but also get a good prediction effect in the medium- and long-term traffic flow prediction.

## 1. Introduction

Traffic flow prediction is to predict the future traffic flow of the road according to the historical traffic flow data. It is an important part of the Intelligent Transportation System and also provides a scientific suggestion for traffic planning and control [1, 2]. According to the predicted traffic flow conditions, the Transport Department can deploy and guide vehicles driving in advance to reduce traffic congestion, and the transport agency can select appropriate transport routes to improve travel efficiency [3]. However, due to the complex spatial and temporal characteristics of traffic flow, real-time and accurate traffic flow prediction is a huge challenge. Traffic flow shows correlation and dependence in time and space. Therefore, comprehensively considering the

time and space characteristics of traffic flow is the key to realize traffic flow prediction.

The time characteristic of traffic flow refers to the periodicity [4] and trend of changes [5] in traffic conditions over time. The traffic flow data shows periodic changes over time. For example, the traffic flow in the morning and evening peak periods on weekdays is significantly higher than that at other times and the traffic flow in the early morning is very small. The traffic flow data has a certain trend change with time, and the traffic flow at the historical time will have an impact on the traffic flow at the future time, especially on the flow at the adjacent time. For example, the traffic flow of a road bayonet from 8:00 to 8:15 A.M. will affect the traffic flow of the bayonet from 8:15 to 8:30 A.M.

The spatial correlation [6] of traffic flow means that the traffic condition of any road in the traffic road will be affected by the other roads, and has a spatial correlation with its adjacent or connected roads. The spatial dependence [7] of traffic flow is that the traffic conditions of the upstream roads will be transmitted to the downstream roads, and the traffic conditions of the downstream roads will also have a corresponding retrospective effect on the upstream roads, that is, from a spatial point of view, the geographically adjacent areas show strong spatial dependence. For example, if there are novice drivers on the upstream road driving slowly, the road congestion will directly lead to traffic congestion on the downstream road, and if there is slow traffic on the downstream road, the speed of the upstream road will also be affected accordingly.

The traditional traffic flow prediction method [8, 9] is to predict the future traffic flow by considering the time correlation of traffic flow data and learning the data characteristics of historical traffic flow, such as Kalman filtering model (KFM) [10, 11], Autoregressive Integrated Moving Average (ARIMA) model [12, 13], k-nearest neighbor model [14, 15], Bayesian model [16, 17], and so on. These methods consider the dynamic changes of traffic conditions with time but ignore the influence of space, so they can not accurately predict traffic conditions. To better describe the relationship between traffic flow and spatial characteristics, Neural Network is introduced to model the spatial characteristics of traffic flow data. However, traditional Neural Networks are usually used for the analysis of neatly arranged Euclidean data, such as text, images, and audio, and are not suitable for irregular traffic roads with complex topology. Therefore, the traditional Neural Network cannot deeply explore the spatial characteristics of traffic flow.

To better learn the complex spatial dependence and temporal correlation of traffic flow data and predict traffic flow more accurately, this paper proposes a spatio-temporal traffic flow prediction model based on a new Graph Neural Network (GNN), which is called Bidirectional-Graph Recurrent Convolutional Network (Bi-GRCN). The main contributions of this paper are as follows:

- (1) Aiming at the spatial dependence of traffic flow data, the Graph Convolution Network (GCN) is introduced and improved, and a new spatio-temporal traffic flow prediction model is proposed based on GNN. The spatial relationship between traffic flow and traffic road is studied, and the adjacency matrix without weight is constructed to represent the connection relationship of traffic road. Through the learning of GCN, the spatial dependence in traffic flow data is better captured, and a new traffic flow prediction model is constructed.
- (2) A traffic flow prediction model which could extract time features is constructed based on Bidirectional-Gate Recurrent Unit (Bi-GRU). Bi-GRU uses bidirectional layer-by-layer training and has good performance in feature extraction. Considering that traffic flow is time series data and has time-series correlation characteristics, Bi-GRU is used to

capture the time correlation characteristics hidden in the data time-series, and to learn the correlation relationship among traffic flow data, historical data, and future data, so that the predicted value can be obtained.

- (3) The idea of integrating spatio-temporal data is adopted to improve the prediction ability of the model. Traffic flow is the data that integrates spatial and temporal information. The temporal correlation between the traffic flow data which is divided by time slices and the hidden spatial dependence in each time slice is learned, and the temporal and spatial characteristics are fused through the full connection layer to improve the prediction accuracy of the model.

The rest of the paper is organized as follows: Section 2 shows the related research of traffic flow prediction. Section 3 introduces the definition and method of traffic flow prediction in detail. Section 4 explains the Bi-GRCN model for traffic flow prediction. Section 5 evaluates the prediction performance of the Bi-GRCN model through real-world traffic data sets, including model parameters, results analysis, and model interpretation. Section 6 is the conclusion of the paper.

## 2. Related Work

The existing traffic flow prediction models are divided into traditional traffic flow prediction models and traffic flow prediction models based on Machine Learning. The commonly used traditional flow prediction models include the Historical Average Model (HAM) [18], Kalman Filtering Model (KFM) [10, 11], and Autoregressive Integrated Moving Average Model (ARIMA) [12, 13]. HAM takes the average data of historical traffic flow as the result and the calculation is simple and efficient. KFM is a linear regression analysis model and it has the advantages of high precision and flexible selection of predictors. ARIMA forecasts the traffic flow by analyzing the relationship between historical and current traffic flow data and has strong interpretability.

The commonly used classical Machine Learning methods for traffic flow prediction include K-Nearest Neighbor (KNN) [14, 15], Support Vector Machine (SVM) [19, 20], and Decision Tree (DT) [21]. KNN is to find the flow data of K historical periods closest to the traffic flow in the predicted period; however, it has high computational complexity. SVM uses the trained SVM model of traffic flow prediction to forecast the traffic flow; however, the prediction ability of the model depends on the kernel function. DT realizes the Classification Forecasting of traffic flow through continuous feature selection, and it has the advantages of high calculating speed and high prediction accuracy, but it is easy to overfit.

Deep Learning models considering the temporal correlation of data include Recurrent Neural Networks (RNN) [22, 23], Long Short-Term Memory (LSTM) [24], and Gated Recurrent Unit (GRU) [25], while comprehensively considering spatial dependence and temporal correlation

include Convolutional Neural Network(CNN) [26, 27], Deep Belief Network(DBN) [28], and Stacked Autoencoder (SAE) [29]. RNN can effectively use the self-circulation mechanism, and they can learn long-term temporal correlation of traffic flow data well. LSTM transmits the time data through the gate unit, uses the memory unit to continuously store the updated data, and obtains the short-term and long-term temporal correlation of the traffic flow data. GRU has a simpler structure and fewer parameters compared with LSTM. Therefore, GRU is better than LSTM in training speed and operational efficiency. CNN is a classical feed-forward propagation Deep Learning model, which can capture the spatial dependence and temporal correlation of data at the same time. DBN consists of multiple Restricted Boltzmann Machine (RBM) [30] and can learn the traffic flow under the influence of spatial dependence between roads. SAE consists of multiple self-encoders and can learn multi-level features, so it can effectively mine the spatial dependence and temporal correlation in traffic flow data.

In recent years, Graph Neural Network (GNN) [31, 32] has become the most discussed topic in deep learning research, showing state-of-the-art performance in various traffic applications [33], such as traffic congestion, traffic safety, travel demand, automatic driving, and traffic monitoring. Because of GNN's ability to capture spatial dependency, which is represented using non-Euclidean graph structures, it is ideally suited to solve traffic prediction problems; for example, the Diffusion Convolutional Recurrent Neural Network (DCRNN) [34], Graph Attention Network (GAT) [35], and Graph WaveNet [36] models.

Binary Graph Convolutional Network (Bi-GCN) [37] binarizes both the network parameters and input node features, and Bi-Directional Graph Convolutional Networks (Bi-GCN) [38] explore both characteristics by operating on both top-down and bottom-up, and Graph Convolution [39] introduced into the segmentation task and proposes an improved Laplacian. The historical days [40] are selected and added for daily traffic flow forecasting through contextual mining. Incorporating contextual factors and traffic flow patterns [41], and a deep-learning-based method for daily traffic flow forecasting could be introduced. A deep neural network [42] based on historical traffic flow data and contextual factor data is proposed.

The GNN-based method utilizes various graph formulations, so it has been extended to other transportation modes. Based on this background, this paper proposes a new Deep Learning model on GNN [43], which can capture complex spatio-temporal characteristics from traffic flow data to further improve the accuracy of prediction.

### 3. Problem Definition

*3.1. Related Definition.* Traffic information is spatio-temporal data that has both spatial dependence and temporal correlation. Therefore, traffic conditions are not only affected by historical traffic conditions, but also by the upstream and downstream relationships in the road. The purpose of traffic flow prediction is to predict future traffic conditions based on historical information. Usually, traffic

conditions are mainly described by variables such as traffic flow, vehicle speed, and road occupancy. In this study, the traffic conditions are measured by vehicle speed. Taking account of the spatial and temporal characteristics of the vehicle speed, the vehicle speed is transformed into a spatio-temporal matrix containing time series data of historical traffic conditions and spatial characteristic data of road connections, to predict the vehicle speed for a while in the future.

*Definition 1. Topology Graphic  $G$  of Traffic Networks.*  $G = (V, E)$  is an unweighted matrix that represents the spatial dependence between traffic roads.  $V = \{v_1, v_2, \dots, v_N\}$ ,  $V_i \in V$ . In this situation,  $V$  is the collection of traffic roads,  $V_i$  presents one link in the road network, and  $N$  is the number of roads in the traffic networks.  $E$  is the set of all the edges in the road graphic reflecting the connection between roads. Meanwhile, the adjacency matrix  $A$  shown in equation (1) stores the connection information of roads in graphic  $G$ .

$$A_{ij} = \begin{cases} 1, & e_{ij} \in E \\ 0, & e_{ij} \notin E \end{cases} \quad (1)$$

The matrix  $A$  contains elements of 0 and 1 in equation (1), where  $e_{ij}$  represents the edge from  $v_i$  to  $v_j$  in the graphic  $G$ . If  $A_{ij} = 1$ , there is a link between  $v_i$  to  $v_j$ , and if  $A_{ij} = 0$ , there exist no links.

The graph structure is transformed into an unweighted adjacency matrix  $A$ , as shown in Figure 1.

*Definition 2. Temporal Feature Matrix  $X^{N \times P}$ .* The traffic information on the road networks suggests the temporal attribute features of roads, which is expressed as  $X^{N \times P}$ .  $N$  represents the number of roads and  $P$  represents the number of temporal attribute features of roads.  $x_i^t = (1/m) \sum_{i=1}^m v_i^t$ , where  $m$  represents the number of vehicles in  $t$  minutes,  $v_i^t$  is the average speed of the  $i$ -th vehicle, and  $x_i^t$  represents the average speed of vehicles on the  $i$ -th road section in  $t$  minutes.

*Definition 3. Spatio-Temporal Graphic  $G^t$  of Traffic Information.* The spatio-temporal situations of traffic information containing both the spatial characteristic information of traffic networks and the time-series characteristic data information of traffic conditions, which is expressed as  $G^t = (V, E, X^t)$ . In this way,  $G^t$  represents the traffic conditions attributed to the vehicle speed dynamically changing with time.  $V$  presents the collection of traffic roads,  $E$  presents the set of all the edges in roads networks, and  $X^t$  presents the time-series characteristic matrix of the vehicle speed at time  $t$ .

To solve traffic flow prediction problems, we could regard it as the mapping function  $f$  on the premise of acquiring temporal feature matrix  $X$  and road network topology  $G$ , and then calculate the traffic flow at the next  $T$  moment, as shown in

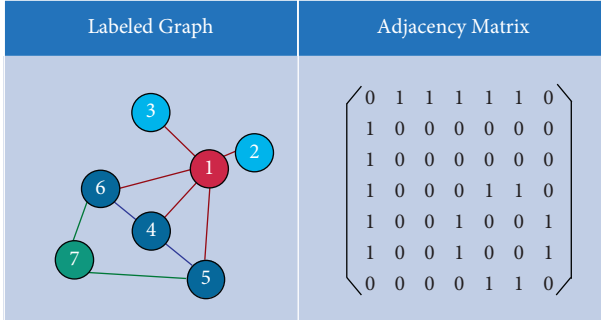


FIGURE 1: Transformation of the graph structure into an adjacency matrix without weights.

$$[X_{t+1}, \dots, X_{t+T}] = f(G; (X_{t-n}, \dots, X_t)), \quad (2)$$

where  $n$  is the length of the historical time series and  $T$  is the length of the time series to be predicted.

**3.2. Overview.** We proposed a model Bi-GRCN for traffic flow prediction, which is composed of both GCN and Bi-GRU. At first, input the data with spatial characteristics at historical moments into the GCN, and then obtain the spatial characteristics by using GCN to capture the topological structure of the traffic roads. Second, input the time series data with spatial characteristics into the Bi-GRU, and obtain the bidirectional time characteristics through the forward and backward information transmission between the gate units. Finally, the traffic flow prediction results will be obtained through the fusion of spatio-temporal data on the fully connected layer. The framework of the Bi-GRCN is shown in Figure 2.

## 4. The Proposed Method

The key problem to be solved in traffic flow prediction is to obtain the complex spatial dependence and temporal correlation of traffic flow data.

**4.1. Spatial Dependency Modeling.** The traffic flow in the real world is changing with the transformation of traffic road topology. The commonly used CNN Modeling method can obtain the spatial characteristics of data, but it can only act on regular Euclidean space data, and cannot capture the spatial dependence of complex traffic roads. GCN can widely process non-Euclidean space data and has been successfully applied to image classification, document analysis, and other fields. Considering the spatial dependence of traffic flow data on road topology, this paper uses GCN to process traffic flow data to better capture the spatial characteristics of the data. The structure of GCN is shown in Figure 3.

The GCN constructs a filter in the Fourier domain, and then acts on the nodes of the graph to capture the spatial characteristics between nodes. The GCN model is established by stacking multiple convolution layers. The calculation process of GCN is described in

$$\begin{aligned} H^{(l+1)} &= \sigma\left(\tilde{D}^{-(1/2)} \tilde{A} \tilde{D}^{-(1/2)} H^{(l)} W^{(l)}\right) \\ \tilde{A} &= A + I_N \\ \tilde{D} &= \sum_j \tilde{A}_{ij}, \end{aligned} \quad (3)$$

where  $\tilde{A}$  is the matrix with additional self-connections,  $I_N$  is the identity matrix,  $\tilde{D}$  is the degree matrix,  $H^{(l)}$  is the output of  $l$  layer,  $H^{(l+1)}$  is the output of  $l+1$  layer,  $W^{(l)}$  is the weight matrix, and  $\sigma$  is the sigmoid activation function.

In this model, the two-layer GCN is used to obtain the spatial characteristics of traffic flow data, as described in

$$\begin{aligned} f(X, A) &= \sigma(\hat{A} \text{ReLU}(\hat{A} X W_0) W_1), \\ \hat{A} &= \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}, \end{aligned} \quad (4)$$

$$W_0 \in R^{P \times H} \quad W_1 \in R^{H \times T}.$$

$\hat{A}$  is a symmetric normalized Laplacian, which is obtained by symmetrically normalizing the adjacency matrix  $A$ .  $W_0$  represents the weight matrix from the input layer to the hidden layer, and  $W_1$  represents the weight matrix from the hidden layer to the output layer.  $P$  is the length of the characteristic matrix  $X$ ,  $H$  is the number of hidden cells, and  $T$  is the prediction length. ReLU is a commonly used activation function in neural networks.

GCN learns the spatial characteristics of traffic flow data by setting the corresponding adjacency matrix for the traffic road code and the connection between traffic roads through a road is abstracted as a node. The spatial dependence characteristics of traffic roads are shown in Figure 4.

**4.2. Temporal Correlation Modeling.** The traffic flow in the real world fluctuates with the change of time. At present, the most commonly used neural network model for processing time series is the RNN model, but RNN has the defects of gradient explosion, gradient disappearance, and unable to save data for a long time. LSTM is a variant of RNN, which effectively solves the defects of RNN. LSTM is composed of input gate, forget gate, and output gate. The input gate and the forget gate are used to retain and forget the input information, and the output gate is used to export the current state. However, LSTM has some defects, such as complex model structure and long training time. GRU model replaces the input gate and forget gate with an update gate based on the LSTM model, which reduces the complexity of the model, reduces the training time, and improves the training efficiency.

As shown in Figure 5,  $x_t$  represents the traffic flow information at time  $t$ .  $z_t$  is the update gate used to retain the status information of the previous time to the current status.  $r_t$  is the reset gate for ignoring the state information of the previous time.  $\hat{h}_t$  is the memory for storing the information at time  $t$ .  $h_{t-1}$  represents the hidden state at time  $t-1$ ,  $h_t$  represents the output state at time  $t$ . GRU model obtains the state at time  $t$  through the hidden state at time  $t-1$  and the

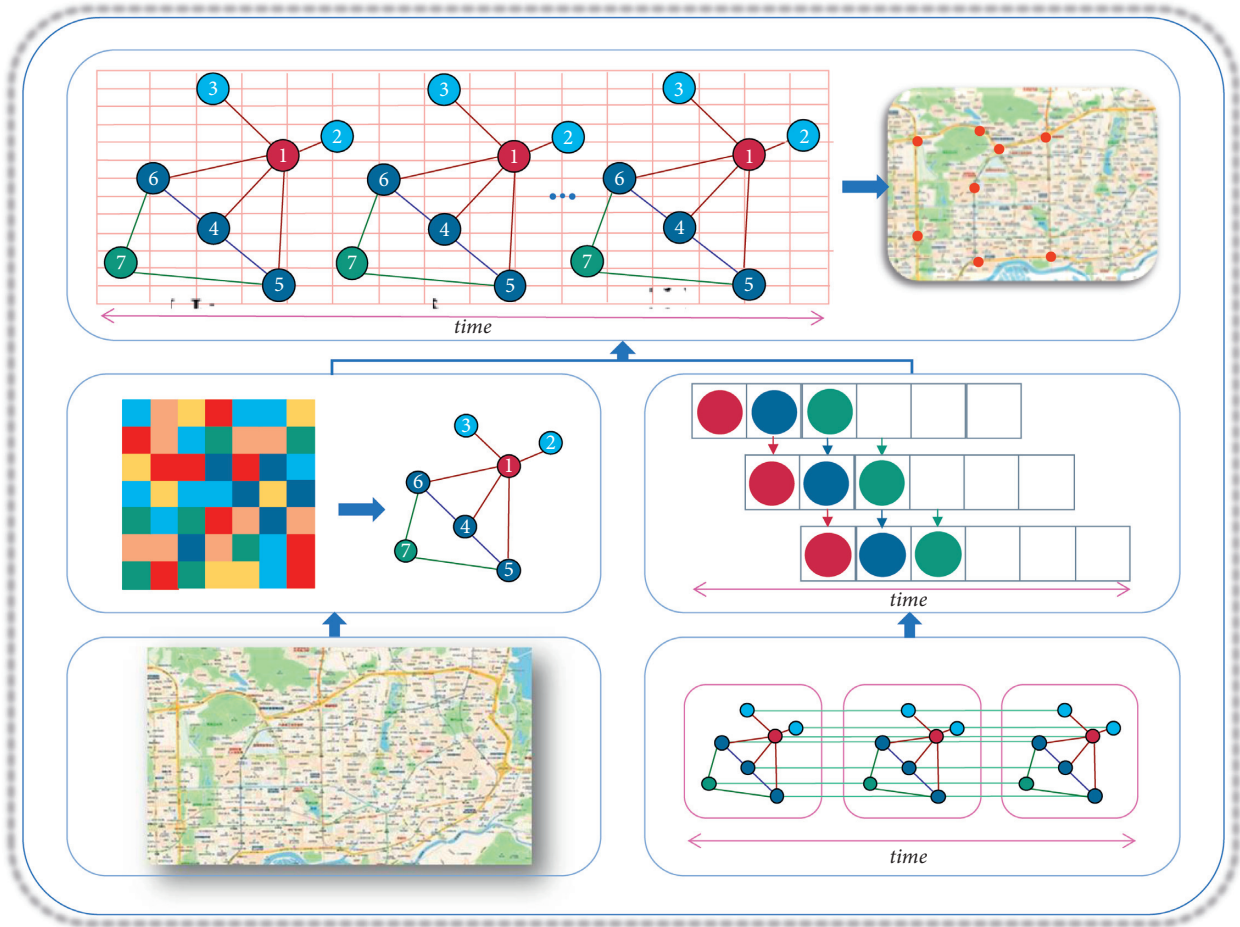


FIGURE 2: Framework of Bi-GRCN.

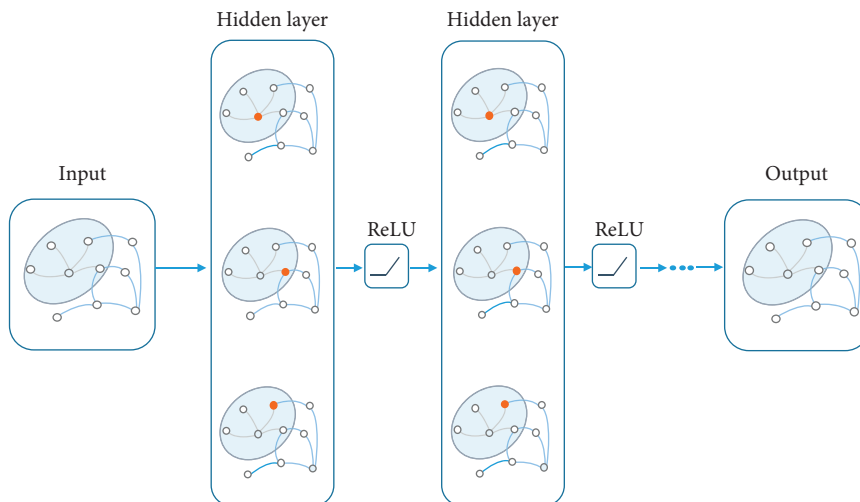


FIGURE 3: Structure of GCN.

current traffic flow data as inputs. The GRU model can not only capture the traffic flow information at the current time, but also retain the traffic flow information at the historical time, so it can learn temporal correlation. The structure of GRU is shown in Figure 5.

Considering traffic flow data has a bidirectional temporal correlation on historical data and future data. Bi-GRU is used to learn historical data and future data at the same time, to fully extract the temporal correlation. The structure of Bi-GRU is shown in Figure 6.

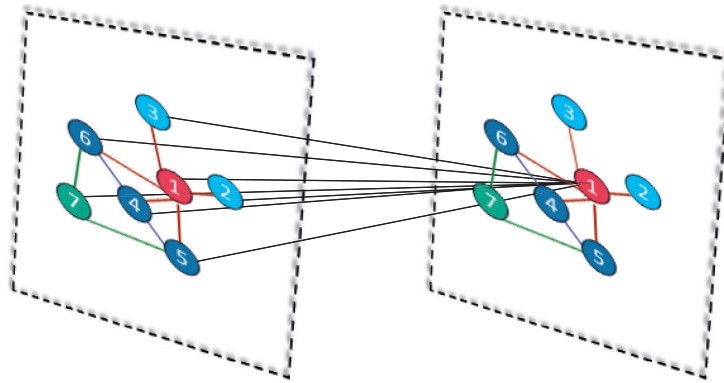


FIGURE 4: Spatial dependence characteristics of traffic roads.

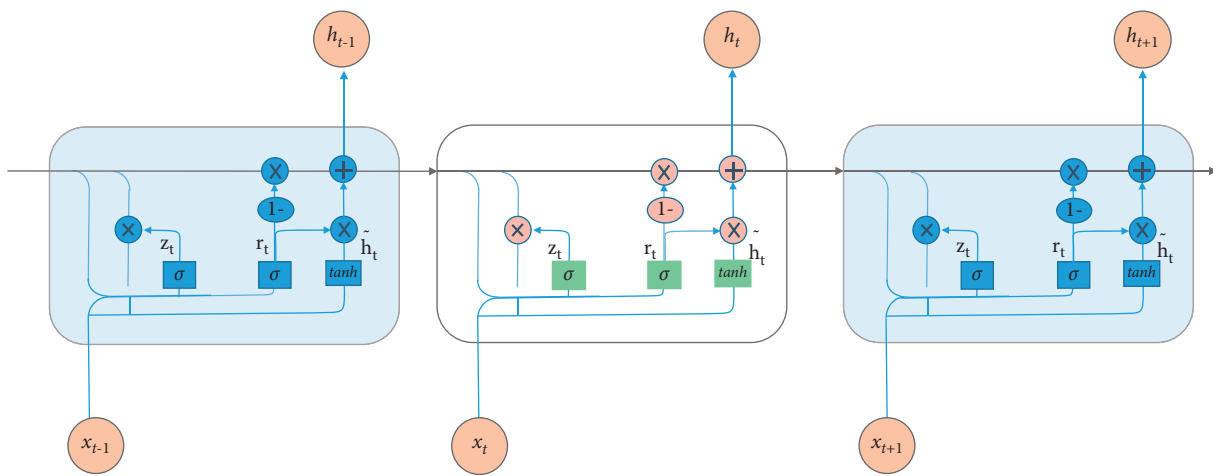


FIGURE 5: Structure of GRU.

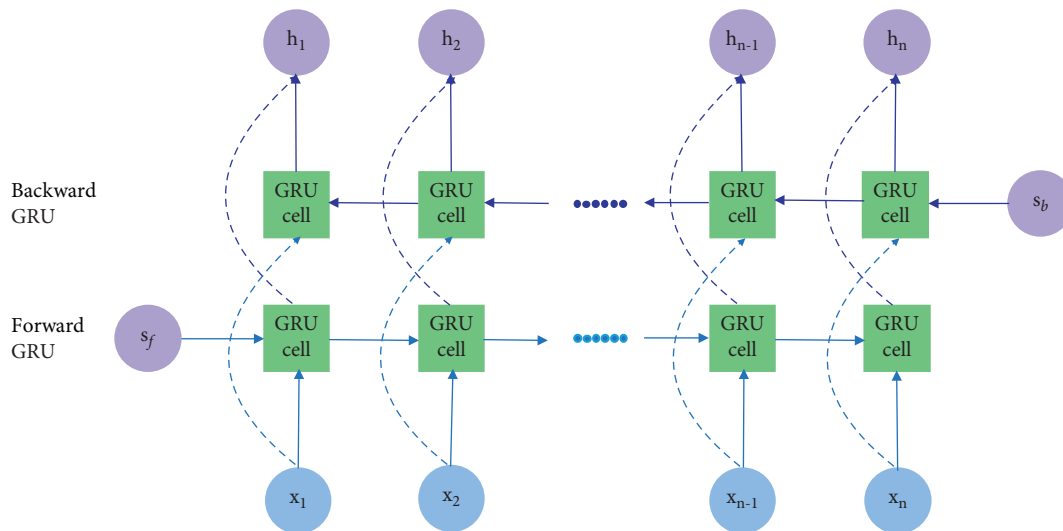


FIGURE 6: Structure of Bi-GRU.

4.3. *Spatio-Temporal Correlation Modeling.* To capture the spatial and temporal characteristics from traffic flow data at the same time, a new Graph Neural Network Model

based on GCN and Bi-GRU, which is called Bi-GRCN, is proposed in this paper. The calculation process is described in

$$z_{tf} = \sigma(W_z[f(A, X_t), h_{t-1}] + b_z), \quad (5)$$

$$r_{tf} = \sigma(W_r[f(A, X_t), h_{t-1}] + b_r), \quad (6)$$

$$\tilde{h}_{tf} = \tanh(W_h[f(A, X_t), (r_{tf} * h_{t-1})] + b_h), \quad (7)$$

$$h_{tf} = z_{tf} * h_{t-1} + (1 - z_{tf}) * \tilde{h}_{tf}, \quad (8)$$

$$z_{tb} = \sigma(W_z[f(A, X_t), h_{t+1}] + b_z), \quad (9)$$

$$r_{tb} = \sigma(W_r[f(A, X_t), h_{t+1}] + b_r), \quad (10)$$

$$\tilde{h}_{tb} = \tanh(W_h[f(A, X_t), (r_{tb} * h_{t+1})] + b_h), \quad (11)$$

$$h_{tb} = z_{tb} * h_{t+1} + (1 - z_{tb}) * \tilde{h}_{tb}, \quad (12)$$

$$h_t = 50\% * h_{tf} + 50\% * h_{tb}. \quad (13)$$

$h_{t-1}$  represents the output at time  $t-1$ ,  $h_t$  represents the output at time  $t$ , and  $h_{t+1}$  represents the output at time  $t+1$ .  $z_{tf}$  is the update gate of the forward GRU at time  $t$ , and  $z_{tb}$  is the update gate of backward GRU at time  $t$ .  $r_{tf}$  is the reset gate of the forward GRU at time  $t$ ,  $r_{tb}$  is the reset gate of backward GRU at time  $t$ .  $\tilde{h}_{tf}$  is the memory for storing the forward information at time  $t$ .  $\tilde{h}_{tb}$  is the memory for storing the backward information at time  $t$ .

Bi-GRCN obtains the topology of traffic road through GCN and the dynamic change of traffic flow with time through Bi-GRU. It then processes the complex spatial and temporal characteristics of traffic flow through the fully connected layer, and realizes traffic flow prediction finally.

**4.4. Loss Function.**  $Y_t$  represents the actual traffic speed and  $\hat{Y}_t$  represents the predicted traffic speed in Bi-GRCN. The goal of the model training is to minimize the error between the actual traffic speed and the predicted traffic speed. The loss function of the Bi-GRCN is shown in

$$loss = \|Y_t - \hat{Y}_t\|_2 + \lambda L_{reg}. \quad (14)$$

$\lambda$  is a hyperparameter.  $L_{reg}$  is introduced to avoid overfitting, and  $L_{reg}$  is the regularization term of  $L2$ .

## 5. Experiments and Analysis

### 5.1. Experimental Settings

**5.1.1. Experimental Data.** The experimental data set is the trajectory of taxis in Shenzhen from January 1 to January 31, 2015. The research area is 96 main roads in Luohu District. The experimental data are composed of an adjacency matrix that represents spatial dependence and a characteristic matrix that represents temporal correlation. The adjacency matrix has 96 rows and 96 columns to describe the spatial dependence between 96 roads. The

row number and column number have corresponded to the road number in the adjacency matrix. The values in the adjacency matrix represent the connection relationship between roads, 0 represents that the two roads are not connected, and 1 represents that the two roads are connected. The characteristic matrix describes the vehicle speed on the road, which changes with time. Each column represents a road, and each row represents the speed of a road in different periods. The vehicle speed on the road is calculated every 15 minutes, so the characteristic matrix has 2976 rows in total. Use 70% of the data as the training set and 30% of the data as the test set to predict the vehicle speed in the next 15 minutes, 30 minutes, 45 minutes, and 60 minutes.

**5.1.2. Baseline Methods.** To evaluate the performance of the proposed method, this paper uses the following baseline methods in comparison with Bi-GRCN:

HA [18]: the average value of historical traffic flow data is used as the predicted value of traffic flow.

ARIMA [12, 13]: traffic flow data are treated as random time series. The non-stationary data are transformed into stationary series data through multiple differential calculations, and then the traffic prediction value is obtained by using Autoregressive Moving Average (ARMA) [44].

SVR [45]: Support Vector Regression (SVR) uses regression analysis to solve the problem of traffic flow prediction based on the principle of SVM [19, 20]. The traffic parameters such as vehicle speed inputs the trained SVR and outputs the traffic flow prediction results in the corresponding period. The kernel function that has been selected is the key to using SVR. The kernel function used is a linear kernel in this paper.

GCN [31, 32]: GCN is a GNN [43] that uses the convolution operation. The traffic flow with spatial relationship inputs the trained GCN and outputs the traffic flow prediction results in the corresponding period.

GRU [25]: GRU uses a gate unit to select information and forget data at the same time, and the model has high training efficiency. The traffic flow with time attribute characteristics inputs the trained GRU and outputs the traffic flow prediction results in the corresponding period.

**5.1.3. Evaluation Methods.** Four metrics are used to evaluate the performance of Bi-GRCN, as shown in equations (15) to (18).

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N (y_i^j - \hat{y}_i^j)^2}. \quad (15)$$



Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N |y_i^j - \hat{y}_i^j|. \quad (16)$$

$$\text{Accuracy} = 1 - \frac{\|Y - \hat{Y}\|_F}{\|Y\|_F}. \quad (17)$$

$$\text{var} = 1 - \frac{\text{Var}\{Y - \hat{Y}\}}{\text{Var}\{Y\}}, \quad (18)$$

$y_i^j$  represents the real speed of the  $i$ -th road at time  $j$ , and  $\hat{y}_i^j$  represents the predicted speed of the  $i$ -th road at time  $j$ .  $N$  is the number of roads and  $M$  is the number of time samples.  $Y$  is the set of  $y_i^j$ , which represents the real speed of the road at different times.  $\hat{Y}$  is the set of  $\hat{y}_i^j$ , which represents the predicted speed of the road at different times. RMSE and MAE are used to evaluate the prediction error. The smaller the values of RMSE and MAE, the better the prediction effect. Accuracy and var are used to evaluate the prediction effect. The greater the values of Accuracy and var, the better the prediction effect.

## 5.2. Prediction of Performance Analysis

**5.2.1. Hyperparameters.** The setting of hyperparameters determines the prediction effect of Bi-GRCN. In the experiment, the hyperparameters of the Bi-GRCN mainly include batch size, training epoch, learning rate, and the number of hidden units. Comparing the prediction effect of batch size set to 32 or 64, the batch size is set to 32 in the experiment. Comparing the prediction effect of the training epoch set to 3000 or 5000, the training epoch is set to 3000 in the experiment. We manually set the learning rate to 0.001. The number of hidden units is the most important parameter of the Deep Learning Model. Different numbers of hidden units have a great impact on the prediction results. To choose the best value, we experiment with different hidden units. We choose the number of hidden units from [16, 32, 64, 80, 96, 100, 128] and analyze the change of prediction precision.

As shown in Figure 7, the horizontal axis represents the number of hidden units, and the vertical axis represents the values of RMSE and MAE. Figure 7 shows the results of RMSE and MAE for different hidden units. It can be seen that the prediction error is the smallest when the number is 128. As shown in Figure 8, the horizontal axis represents the number of hidden units, and the vertical axis represents the values of accuracy and var. Figure 8 shows the results of accuracy and var for different hidden units. It can be seen that the prediction precision is the maximum when the number is 128. Based on the four evaluation metrics in Figures 7 and 8, the prediction result is the best when the number of hidden units is set to 128. Therefore, we set the number of hidden units to 128 in the experiment.

**5.2.2. Comparative of Experiments Using Different Models.** We set the batch size to 32, training epoch to 3000, learning rate to 0.001, and the number of hidden units to 128 in the

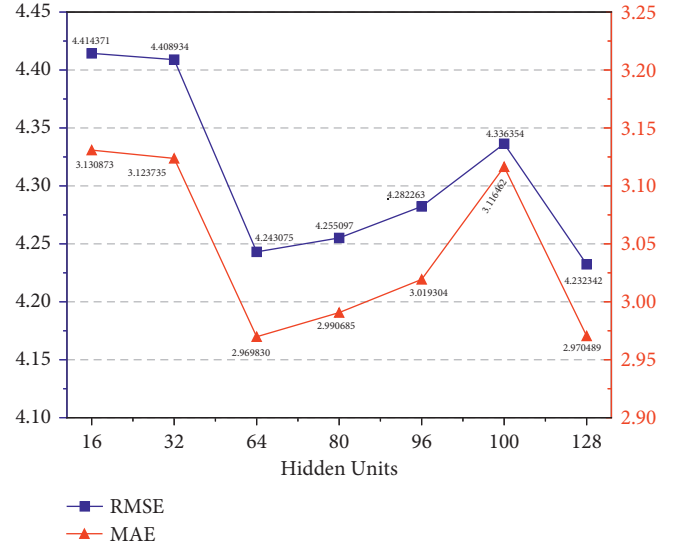


FIGURE 7: RMSE and MAE change with the number of hidden units.

Bi-GRCN model. 70% of the overall data set is used as the training dataset, and the remaining data is used as the testing dataset. The Bi-GRCN model is trained using the Adam optimizer. The prediction performance of the model is learned at four different time intervals of 15 minutes, 30 minutes, 45 minutes, and 60 minutes on the dataset. The prediction results of the Bi-GRCN model and other baseline methods are shown in Table 1.

**5.2.3. Analysis of Experimental Results.** Spatio-temporal prediction capability. To verify whether the Bi-GRCN model could capture spatial and temporal features from the dataset, we compare the Bi-GRCN with the GCN and the GRU. Compared with the GRU, which considers only temporal features, for 15 minutes, 30 minutes, 45 minutes, and 60 minutes traffic forecasting, the accuracy of the Bi-GRCN is increased by approximately 2.59%, 1.16%, 0.25%, and 0.85%, indicating that the Bi-GRCN can capture temporal correlation well. Compared with the GCN, which considers only spatial features, for 15 minutes, 30 minutes, 45 minutes, and 60 minutes traffic forecasting, the accuracy of the Bi-GRCN is increased by approximately 11.77%, 14.11%, 14.72%, and 16.02%, indicating that the Bi-GRCN can capture spatial dependence well. The accuracy comparison between GRU and Bi-GRCN is shown in Figure 9. The accuracy comparison between GCN and Bi-GRCN is shown in Figure 10.

**Model prediction ability.** According to the analysis of the data in Table 1, Bi-GRCN has better prediction performance than other baseline models. Compared with the GRU, GCN, HA, ARIMA, and SVR for 15 minutes, the RMSE of the Bi-GRCN is decreased by approximately 5.29%, 18.9%, 3.63%, 35%, and 7.57%, indicating that the Bi-GRCN can capture spatial dependence and temporal correlation well. The main reason for the worse prediction of ARIMA is that it is difficult to deal with long series of non-stationary data, and GCN ignores the temporal correlation of traffic flow data

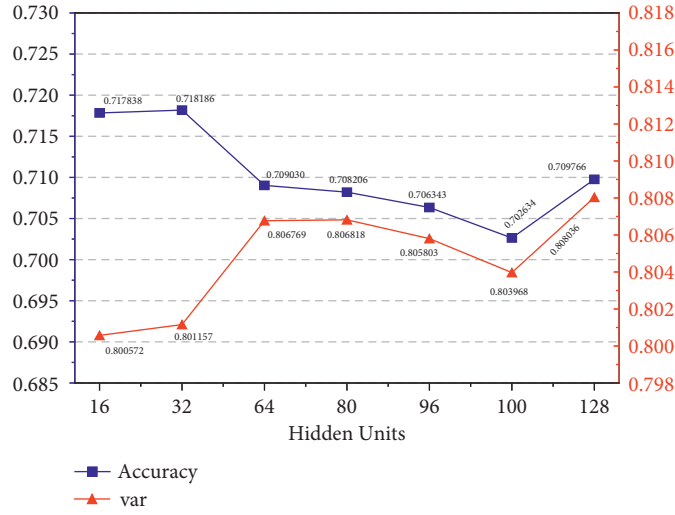


FIGURE 8: Accuracy and var change with the number of hidden units.

TABLE 1: Prediction results of the Bi-GRCN model and other baseline methods.

T (min)	Metric	Models					
		HA	ARIMA	SVR	GCN	GRU	Bi-GRCN
15	RMSE	4.389446	6.998807	4.228722	5.609166	4.802862	4.548942
	MAE	3.006430	4.991264	2.882457	4.410619	3.464756	3.239479
	Accuracy	0.699633	0.455428	0.710631	0.616168	0.671343	0.688718
	var	0.788113	-0.000145	0.803415	0.654015	0.746349	0.772483
30	RMSE	4.389446	6.998712	4.252060	5.635812	4.487037	4.369150
	MAE	3.006430	4.990788	2.935113	4.453700	3.218210	3.101176
	Accuracy	0.699633	0.455422	0.709021	0.614327	0.692941	0.701008
	var	0.788113	-0.000347	0.802071	0.650719	0.778851	0.790520
45	RMSE	4.389446	6.997727	4.280222	5.664346	4.372606	4.347006
	MAE	3.006430	4.990163	2.978076	4.462996	3.116036	3.077341
	Accuracy	0.699633	0.455437	0.707092	0.612372	0.700769	0.702521
	var	0.788113	-0.000512	0.800080	0.647190	0.789771	0.792385
60	RMSE	4.389446	6.989135	4.307416	5.677034	4.333217	4.245528
	MAE	3.006430	4.986023	3.011925	4.489186	3.086672	3.000904
	Accuracy	0.699633	0.455653	0.705231	0.611504	0.703465	0.709466
	var	0.788113	0.000986	0.798030	0.645671	0.793562	0.801980

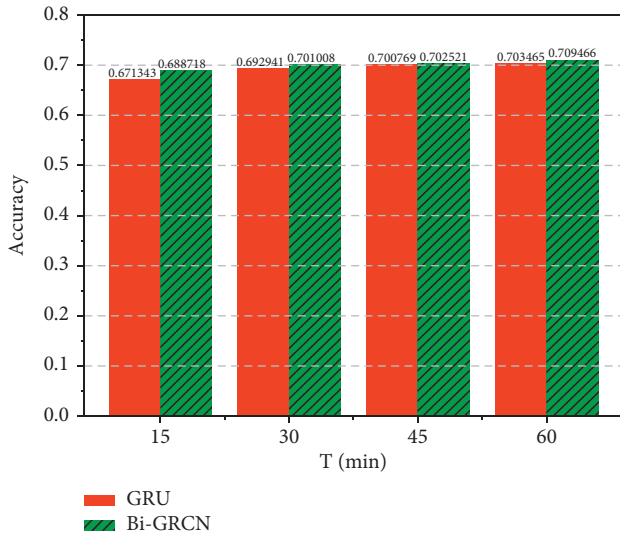


FIGURE 9: Accuracy comparison between GRU and Bi-GRCN.

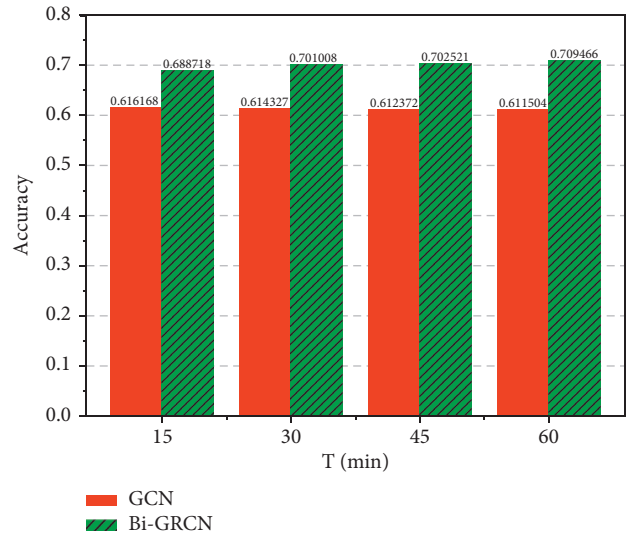


FIGURE 10: Accuracy comparison between GCN and Bi-GRCN.

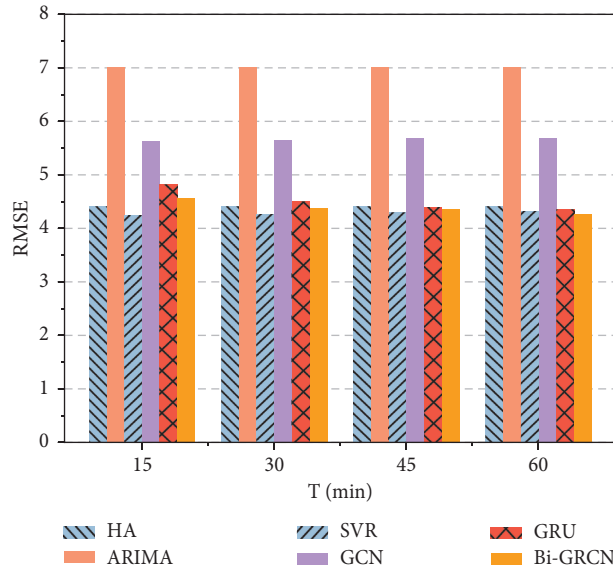


FIGURE 11: RMSE of various models.

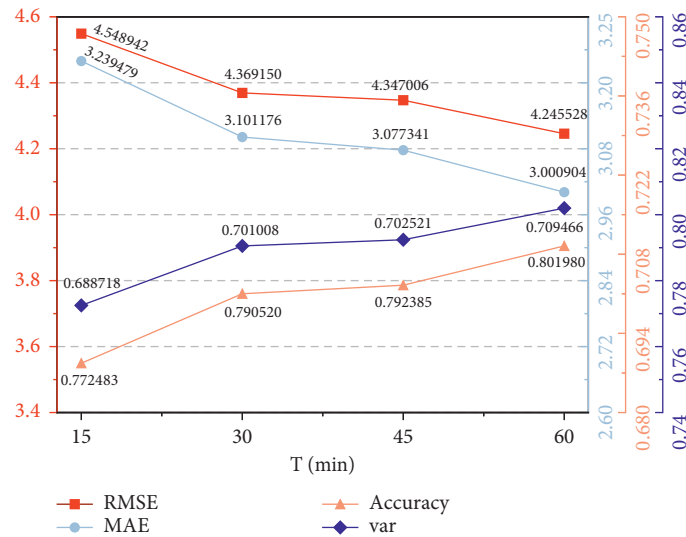


FIGURE 12: Prediction error and prediction accuracy of Bi-GRCN.

which is only considered the spatial dependence. RMSE of various models is shown in Figure 11.

Long-term prediction ability. As shown in Figure 12, the horizontal axis represents different times, and the vertical axis represents four evaluation metrics. RMSE and MAE represent the prediction error of Bi-GRCN. Accuracy and var represent the prediction accuracy of Bi-GRCN. The prediction results show that the prediction error and prediction accuracy of Bi-GRCN change little with time, indicating that Bi-GRCN has certain stability. No matter how the time changes, the model can obtain the best prediction results. Therefore, Bi-GRCN can be used not only for short-term traffic flow prediction, but also for medium-term and long-term traffic flow prediction.

## 6. Conclusion

We propose a new traffic flow prediction model Bi-GRCN based on GNN, which combines GCN and Bi-GRU. The traffic flow graph network is modeling, the road is represented by the nodes, the connection relationship between roads is represented by the edges, and the traffic flow information on the road is represented by the attributes of the nodes. We use real traffic data in the experiment, and compare Bi-GRCN with other Neural Network models and traditional traffic prediction methods. The experimental results show that compared with GCN and GRU, Bi-GRCN has higher accuracy and better traffic prediction performance. Compared with the traditional traffic prediction methods HA, ARIMA and SVR, Bi-GRCN is also more

effective. As weather, weekdays, holidays, traffic accidents, and other factors will also affect the prediction results, we will consider these factors in future research.

## Data Availability

The terms of use of the data used in this study do not allow the authors to distribute or publish the data directly. However, these data can be obtained directly from the following webpage: <https://opendata.sz.gov.cn/>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Grant no. 61772098), Chongqing Science and Technology Innovation Leading Talent Support Program (Grant no. CSTCCXLJRC201908), Basic and Advanced Research Projects of CSTC (Grant no. cstc2019jcyj-zdxmX0008), and the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant nos. KJQN201903005, KJZD-K201900605, and KJZD-K202003001).

## References

- [1] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.
- [2] C. Zheng, X. Fan, C. Wen, L. Chen, C. Wang, and J. Li, "DeepSTD: mining spatio-temporal disturbances of multiple context factors for citywide traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3744–3755, 2020.
- [3] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019.
- [4] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 166–180, 2018.
- [5] N. Polson, G. Polson, and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.
- [6] A. Ermagun and D. Levinson, "Spatiotemporal traffic forecasting: review and proposed directions," *Transport Reviews*, vol. 38, no. 6, pp. 786–814, 2018.
- [7] L. Kang, G. Hu, H. Huang, W. Lu, and L. Liu, "Urban traffic travel time short-term prediction model based on spatio-temporal feature extraction," *Journal of Advanced Transportation*, vol. 2020, Article ID 3247847, 2020.
- [8] Y. Jia, J. Wu, and M. Xu, "Traffic flow prediction with rainfall impact using a deep learning method," *Journal of Advanced Transportation*, vol. 2017, Article ID 6575947, 2017.
- [9] J. Wang, J. Lv, C. Wang, and Z. Zhang, "Dynamic route choice prediction model based on connected vehicle guidance characteristics," *Journal of Advanced Transportation*, vol. 2017, Article ID 6905431, 8 pages, 2017.
- [10] V. Marzano, A. Papola, F. Simonelli, and M. Papageorgiou, "A kalman filter for quasi-dynamic o-d flow estimation/updating," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3604–3612, 2018.
- [11] B. Li, W. Cheng, and L. Li, "Real-Time Prediction of lane-based queue lengths for signalized intersections," *Journal of Advanced Transportation*, vol. 2018, Article ID 5020518, 2018.
- [12] R. Salles, K. Belloze, F. G. Porto, P. H. Gonzalez, and E. Ogasawara, "Nonstationary time series transformation methods: an experimental review," *Knowledge-Based Systems*, vol. 164, pp. 274–291, 2019.
- [13] X. Yang, Y. Zou, J. Tang, J. Liang, and M. Ijaz, "Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and machine learning models," *Journal of Advanced Transportation*, vol. 2020, Article ID 9628957, 2020.
- [14] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.
- [15] B. Sun, W. Cheng, P. Goswami, and G. Bai, "Short-term traffic forecasting using self-adjusting k-nearest neighbours," *IET Intelligent Transport Systems*, vol. 12, no. 1, pp. 41–48, 2018.
- [16] Y. Li, X. Jiang, H. Zhu et al., "Multiple measures-based chaotic time series for traffic flow prediction based on Bayesian theory," *Nonlinear Dynamics*, vol. 85, no. 1, pp. 179–194, 2016.
- [17] Z. Zhu, B. Peng, C. Xiong, and L. Zhang, "Short-term traffic flow prediction with linear conditional Gaussian Bayesian network," *Journal of Advanced Transportation*, vol. 50, no. 6, pp. 1111–1123, 2016.
- [18] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.
- [19] X. Wang, N. Zhang, Y. Zhang, and Z. Shi, "Forecasting of short-term metro ridership with support vector machine online model," *Journal of Advanced Transportation*, vol. 2018, Article ID 3189238, 13 pages, 2018.
- [20] N. Karballaezadeh, S. D. Mohammadzadeh, S. Shamshirband, P. Hajikhodaverdikhan, A. Mosavi, and K.-W. Chau, "Prediction of remaining service life of pavement using an optimized support vector machine (case study of Semnan-Firuzkuh road)," *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 188–198, 2019.
- [21] H. Crosby, S. A. Jarvis, and P. Davis, "Spatially-Intensive decision tree prediction of traffic flow across the entire UK road network," in *Proceedings of the 2016 IEEE/ACM 20th international symposium on distributed simulation and real time applications*, pp. 116–119, IEEE, London, UK, September 2016.
- [22] C. D. Dorine, G. Wang, and J. Kim, "Forecasting pedestrian movements using recurrent neural networks an application of crowd monitoring data," *Sensors*, vol. 19, pp. 1–19, 2019.
- [23] A. Nejadettehad, H. Mahini, and B. Bahrak, "Short-term demand forecasting for online car-hailing services using recurrent neural networks," *Applied Artificial Intelligence*, vol. 34, no. 9, pp. 674–689, 2020.

- [24] B. Yang, S. Sun, J. Li, X. Lin, and Y. Tian, "Traffic flow prediction using LSTM with feature enhancement," *Neurocomputing*, vol. 332, pp. 320–327, 2019.
- [25] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: a GRU-based deep learning approach," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 578–585, 2018.
- [26] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning Traffic as Images A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction," *sensors*, vol. 17, pp. 1–16, 2017.
- [27] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.
- [28] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep Belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.
- [29] X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, "Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3235–3243, 2018.
- [30] N. Zhang, S. Ding, J. Zhang, and Y. Xue, "An overview on restricted Boltzmann machines," *Neurocomputing*, vol. 275, pp. 1186–1199, 2018.
- [31] X. Yin, G. Wu, J. wei, Y. shen, H. qi, and B. yin, "A comprehensive survey on traffic prediction," 2020.
- [32] M. Luca, G. Barlacchi, B. Lepri, and L. Pappalardo, "Deep learning for human mobility: a survey on data and models," 2020.
- [33] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.
- [34] L. Chen, K. Han, Q. Yin, and Z. Cao, "GDCRN: global diffusion convolutional residual network for traffic flow prediction," in *Proceedings of the International Conference on Knowledge Science, Engineering and Management*, Springer, New York City, NY, USA, August 2020.
- [35] Y. Xie, Y. Zhang, M. Gong, Z. Tang, and C. Han, "MGAT: multi-view graph attention networks," *Neural Networks*, vol. 132, pp. 180–189, 2020.
- [36] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for Deep Spatial-Temporal Graph Modeling," in *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*, pp. 1907–1913, Vienna, Austria, August 2019.
- [37] J. Wang, Y. Wang, Z. Yang, L. Yang, and Y. Guo, "Bi-GCN: binary graph convolutional network," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1561–1570, June 2021.
- [38] T. Bian, Xi Xiao, T. Xu et al., "Rumor detection on social media with Bi-directional graph convolutional networks," January 2020, <https://arxiv.org/abs/2001.06362>.
- [39] Y. Meng, H. Zhang, D. Gao et al., "BI-GCN: boundary-aware input-dependent graph convolution network for biomedical image segmentation," *cs.CV*<https://arxiv.org/abs/2110.14775>, October 2021.
- [40] D. Ma, X. Ben Song, J. Zhu, and W. Ma, "Input data selection for daily traffic flow forecasting through contextual mining and intra-day pattern recognition," *Expert Systems With Applications*, vol. 176, pp. 1–12, Article ID 114902, 2021.
- [41] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2627–2636, 2021.
- [42] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Systems with Applications*, vol. 121, pp. 304–312, 2019.
- [43] Z. Lu, W. Lv, Y. Cao, Z. Xie, H. Peng, and B. Du, "LSTM variants meet graph neural networks for road speed prediction," *Neurocomputing*, vol. 400, pp. 34–45, 2020.
- [44] Y.-J. Wu, F. Chen, C.-T. Lu, and S. Yang, "Urban traffic flow prediction using a spatio-temporal random effects model," *Journal of Intelligent Transportation Systems*, vol. 20, no. 3, pp. 282–293, 2016.
- [45] Y. Xu, H. Chen, Q.-J. Kong, X. Zhai, and Y. Liu, "Urban traffic flow prediction: a spatio-temporal variable selection-based approach," *Journal of Advanced Transportation*, vol. 50, no. 4, pp. 489–506, 2016.

## Research Article

# Research on Recommendation Algorithm of Joint Light Graph Convolution Network and DropEdge

Haicheng Qu, Jiangtao Guo , and Yanji Jiang

College of Software, Liaoning Technical University, Liaoning, Huludao 125105, China

Correspondence should be addressed to Jiangtao Guo; 471921121@stu.lntu.edu.cn

Received 16 October 2021; Accepted 22 December 2021; Published 27 January 2022

Academic Editor: JingXin Dong

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

Overfitting in a deep neural network leads to low recommendation precision and high loss. To mitigate these issues in a deep neural network-based recommendation algorithm, we propose a recommendation algorithm, LG-DropEdge, joint light graph convolutional network, and the DropEdge. First, to reduce the cost of data storage and calculation, we initialize user and item embedding in the embedding layer of the algorithm. Then, to obtain high-order interaction relationships to optimize the embedding representation, we enrich the embedding by injecting high-order connectivity relationships in the convolutional layer. In the training phase, DropEdge is used to randomly discard connected relationships (interaction edges) to prevent overfitting. Finally, to reasonably aggregate the embedding results learned on all layers, the weighted average is expressed as the final embedding, so that users can make preferences in the item. We conduct experiments on three public datasets, using two performance indicators; namely, recall and NDCG, are used for evaluation. For the Gowalla dataset, compared with the optimal baseline method, recall@20 and ndcg@20 increased by 2.53% and 2.39%, respectively. For the Yelp2018 dataset, recall@20 and ndcg@20 increased by 6.17% and 5.58%, respectively. For the Amazon-book dataset, recall@20 and ndcg@20 increased by 4.82% and 4.67%, respectively. The results show that LG-DropEdge can not only reduce the degree of neural network overfitting but also improve the recommended results' precision.

## 1. Introduction

Personalized recommendation is a common recommendation method that has been widely used in social media, advertising, e-commerce [1], and other online services. It effectively alleviates the difficulties involved in a user obtaining personalized content due to the explosive growth of information. Its goal is to estimate the likelihood of users to adopt a product based on historical interaction behaviors, such as purchases and clicks. Thus, many studies have focused on recommendation algorithms that are in line with user needs and are practical.

Collaborative filtering (CF) is a method used to build a large-scale recommendation system. Its advantages are strong interpretability and high maturity. The general idea of the algorithm is to predict the items that users may be interested in by analyzing the interaction between users and items. At a high level, the similarity measure between users is

based on user rating history, so that ratings from like-minded users can be used to predict the ratings of interested users; it can also be based on the ratings of users who were interested in the past. To realize this idea, a common approach is to reconstruct historical interactions by parameterizing users and items and then predict user preferences based on parameters [2]. Traditional CF recommendation algorithms can be grouped into two categories: neighborhood-based CF methods [3, 4] and matrix factorization (MF) [5–7] methods. MF have been studied more extensively. It is usually assumed that the scoring matrix can be approximated by two lower triangular matrices. On the basis of matrix factorization, probabilistic MF (PMF) optimizes the maximum likelihood by minimizing the mean square error between the observed items and the reconstructed level [5]. Biased MF improves PMF by merging user and item specific deviations and global deviations [6]. In practical applications, the CF method has some problems, such as a

learning process not being displayed, the high-level interaction information between the user and the item not being considered, and the implicit relationship between the user and the item being ignored.

In recent years, the deep-learning algorithm represented by Convolutional Neural Networks (CNNs) has made great progress in many aspects [8, 9], but its design mostly uses regular Euclidean data (which can be expressed in the form of a sequence or a two-dimensional grid), such as image, voice, and natural language. CNNs are suitable for Euclidean data [10] but have limitations in networks with non-Euclidean structures. This is an issue because not everything can be represented as a sequence or two-dimensional grid, such as social networks or chemical molecules. As these data can be regarded as special cases of graph-structured data, researchers naturally think of generalizing CNN to graphs.

## 2. Related Research

*2.1. Graph Convolutional Network.* A graph convolutional network (GCN) [11] can overcome the problems of CNNs only being applicable to Euclidean data and can capture the characteristics of a network structure. Development has been rapid and research in this direction is generally divided into two categories: methods based on spectral decomposition and methods based on spatial structure. Spectral decomposition methods mainly deal with the spectral domain of the graph. A spectral network [12] defines the convolution operation in the Fourier domain by calculating the feature decomposition of the Laplacian matrix of the graph, but this convolution operation will cause the convolution kernel not to meet the locality. Henaff et al. proposed introducing a parameter term with smooth coefficients to solve the locality problem of the convolution kernel [13]. Defferrard et al. proposed ChebNet, using K-order convolution to define the graph; the convolution can avoid the calculation of redundant Laplacian matrix eigenvectors [14]. Kipf and Welling proposed reducing the convolution operation to first order [15], which greatly reduces the calculation of graph convolution. As a simplification of spectral decomposition methods, the GCN was formally proposed. To learn the implicit relationship between different nodes, Li et al. proposed a residual Laplacian that an adaptive graph convolution network should be added to the original graph [16]. Methods based on spatial structure mainly deal with graphs of different structures and directly define the convolution operation on the graph. GraphSAGE (Graph Sample and AggreGatE) is a method to generate the embedding vector of the target vertex by learning a function that aggregates the representation of neighbor nodes and calculates the node representation inductively [17]. Unlike earlier methods, the graph attention network (GAT) innovatively uses a self-attention mechanism to provide different weights for the heterogeneity of different nodes [18]. Based on GAT, the heterogeneous graph attention network (HAT) refined two attention mechanisms, namely, node-level attention and semantic-level attention [19].

Due to the powerful expressive ability of graphs, graph-based recommendation algorithms have become one of the most popular research methods. The goal is to reorganize the interactive data into a user-item bipartite graph and use the high-level connectivity between users and items to enrich its representation. PinSage [20] uses local convolution to mark the nodes of the graph-structured data and uses multiple convolution modules to aggregate the local neighborhood features of the nodes to generate node embeddings. Graph Convolutional Matrix Completion (GC-MC) [21] applies a GCN to the user-item graph but only uses the signal of the first-order neighbor. Through random browsing on the graph, Hop-Rec [22] combines matrix decomposition and a graph structure from the neighborhood of each user to obtain high-level information from items. Neural Graph Collaborative Filtering (NGCF) proposed by Wang et al. [23] encodes collaborative signals hidden in user-item interactions and spreads and embeds them in bipartite graphs to achieve high-level neighborhood aggregation. However, its direct inheritance of GCN makes the design quite substantial with high algorithm complexity and difficult algorithm training. Based on the improvement of the traditional GCN, a series of new algorithms have been proposed. Simple Graph Convolution (SGC) [24] eliminates the GCN layer and employs a nonlinear relation between time and a linear algorithm to reduce the complexity of the algorithm.

*2.2. LightGCN Algorithm.* The previous works that combine GCNs with recommendation mostly inherit GCN and increase its generalization ability. Based on an in-depth analysis of GCN, from the perspective of simplifying GCN design to make it more concise and more suitable for recommendation algorithms, a new algorithm was proposed: light graph convolutional network (LightGCN) [25]. It uses only the GCN as the foundation for neighborhood aggregation. It has been verified via thorough ablation experiments on the special transformation and nonlinear activation of NGCF inherited from the GCN. The study concluded that the two operations inherited from GCN, nonlinear activation and feature transformation, do not make a positive contribution to NGCF. Furthermore, removing them can significantly improve recommendation precision, showing that certain operations in the GCN bring no benefit to the recommendation task and reduce the effectiveness of the algorithm. Therefore, in LightGCN, each user (or item) is first associated with ID embedding, which is then propagated on the user-item interaction graph to enrich its representation. Finally, the embeddings and weights learned in different layers are combined for recommendation prediction. This not only makes the entire algorithm concise in structure, but also gives a great performance improvement compared to other recommendation methods.

*2.3. DropEdge.* DropEdge [26] involves randomly removing a certain number of edges from the input graph in each training phase to address overfitting and oversmoothing. Overfitting occurs due to the use of a parameter algorithm to fit the distribution of limited training data. The learned

algorithm fits the training data well but is not suitable for the test data. Overfitting weakens the generalization ability of small dataset. Oversmoothing isolates the output representation from the input features as the network depth increases, thus hindering algorithm training. DropEdge can be regarded as a data enhancement technique that increases the randomness and diversity of input data, thereby better preventing overfitting. It can also be regarded as reducing message passing. Losing some edges makes the node connections sparser, which avoids oversmoothing to a certain extent when the GCN is deep.

In addition, DropEdge is different from Dropout [27] and DropNode [28]. Dropout disrupts the feature matrix by randomly setting the feature dimension to zero, but because it does not change the adjacency matrix, it does not have an obvious impact on the overfitting problem. DropNode samples the subgraph and is used for small batch processing. Its principle is that discarding certain nodes can be understood as a special kind of edge discarding. DropNode is for nodes, and edge discarding is achieved indirectly by discarding nodes, whereas DropEdge is edge-oriented and can retain the characteristics of all nodes. That is, with DropEdge, the node only loses the interaction with a certain node, but this does not affect the interaction between the node and other nodes, providing greater flexibility and wider applicability.

Therefore, we propose a joint LightGCN and DropEdge recommendation algorithm named LG-DropEdge. This algorithm is based on the concept of the LightGCN algorithm and integrates the DropEdge which slightly improves the algorithm prediction.

The main contributions of this paper are as follows: (1) we proposed new hybrid recommendation algorithm (2) adding DropEdge to the GCN to enrich input and reduce message passing and (3) changing the final representation of LightGCN from the original average of each layer to a weighted average. Experiments on multiple public datasets verified its advantages and performance.

### 3. Methods

This section describes the proposed LG-DropEdge. The algorithm can be divided into two parts: a light graph convolution algorithm, which is the basic core part of the algorithm and DropEdge to mitigate overfitting and oversmoothing problems caused by deep networks.

**3.1. LG-DropEdge Algorithm.** The recommendation algorithm based on the GCN uses the topological structure of the graph to spread and aggregate the information of neighboring nodes and learn the embedding of nodes. The algorithm structure is shown in Figure 1. It is usually divided into three layers: an embedding layer, a convolution layer, and a prediction layer. LightGCN is based on NGCF [23]. Ablation experiments show that the two operations inherited from GCN feature transformation and nonlinear activation do not bring any benefits but negatively impact algorithm training by increasing difficulty. Removing them can significantly improve precision. This reflects that adding

useless operations to the target task in GCN does not bring any benefits and reduces effectiveness.

**3.1.1. Embedding Layer.** In the LG-DropEdge, the main tasks of the embedding layer are to express the entities (users and items) and relationships in the user-item interaction diagram as low-dimensional vectors and to retain all the information of the interaction diagram, which can reduce data storage and calculation costs and filter out some noise data. Following the mainstream recommendation algorithm [23], the IDs of users and items are mapped to vectors using one-hot encoding, denoted by  $v_u \in R^h$  and  $v_i \in R^h$ , where  $h$  is the embedding size. Here,  $v_u^{(0)}$  and  $v_i^{(0)}$  are the initial vectors of user embedding and item embedding, respectively. The number of users  $u$  is  $N$  and the number of items  $i$  is  $M$ .

**3.1.2. Convolutional Layer.** After obtaining the embedding representation of the user (item) node, based on the graph neural network message passing method rule [23], the collaborative signal is obtained on the interactive graph structure and the embeddings of the user and item are optimized. This mainly involves the construction of signals and the aggregation (update) of node embedding. The advantage lies in the embedded representation that can be displayed to associate users and items with high-level collaboration information. This section shows the embedding learning process of the first-order signal and the higher-order signal extended from the first-order signal.

(1) *First-Order Signal.* The consumer interaction between the user and the item can be used as a characteristic of the user and can be regarded as a collaborative signal of two users.

Structure of the signal: for users and items  $(u, i)$  that have an interactive relationship, the resulting signal is defined as

$$S_{u \leftarrow i} = f(v_u, v_i, d_{ui}), \quad (1)$$

where  $S_{u \leftarrow i}$  is the propagated signal,  $v_u$  and  $v_i$  are embedded inputs, and  $d_{ui}$  is the attenuation coefficient of each propagation on the control  $(u, i)$ . Finally,  $f(\cdot)$  is the signal encoding function, expressed as

$$S_{u \leftarrow i} = \frac{1}{\sqrt{|N_u||N_i|}} v_i, \quad (2)$$

where  $d_{ui}$  is set to the graph Laplacian norm  $1/\sqrt{|N_u||N_i|}$  and  $N_u$  and  $N_i$  denote the first-order neighbor sets of the user  $u$  and the item  $i$ , respectively.

Aggregation of node embedding: this is used to enrich the form of  $u$  embedding by summarizing the signals of neighbors near the user node  $u$ . The aggregate function is defined as

$$v_u^{(1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u||N_i|}} S_{u \leftarrow i}, \quad (3)$$

where  $v_u^{(1)}$  represents the embedded representation of the user  $u$  obtained after the first embedding and propagation.



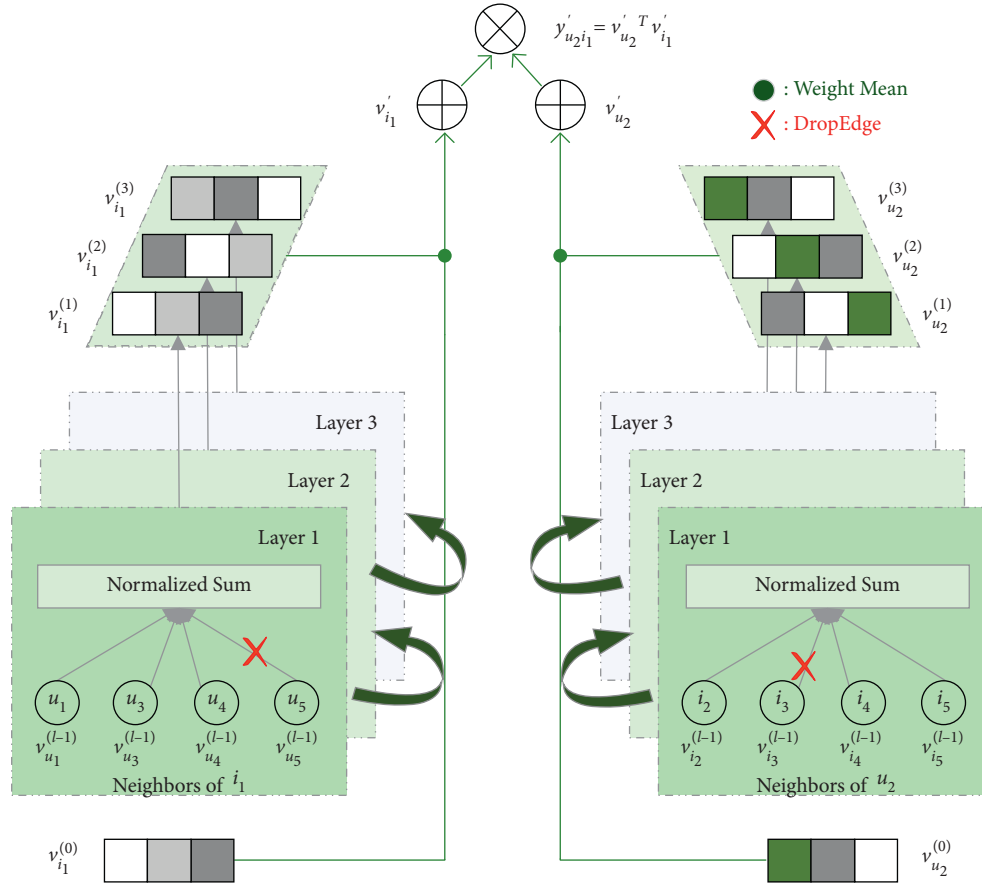


FIGURE 1: LG-DropEdge algorithm structure.

The self-connection  $S$  of  $u$  is not considered here. Similarly, the representation  $v_u^{u \leftarrow u}$  of the item obtained through embedding and propagation can be obtained.

(2) *High-Order Signals*. Through the improvement of first-order signals, multiple layers of embedding can be stacked to represent high-order collaborative signals. This high-level signal enables better interpretability in estimating the correlation between users and items.

Signal structure: enrich the signal by stacking multiple layers of embedding, defined as follows:

$$S_{u \leftarrow i}^{(l)} = \frac{1}{\sqrt{|N_u| |N_i|}} v_i^{(l-1)}. \quad (4)$$

Node embedding aggregation: aggregate multilayer signals, which can receive signals propagated from layer neighbors, are defined as follows:

$$v_u^{(l)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_u| |N_i|}} S_{u \leftarrow i}^{(l-1)}, \quad (5)$$

where  $S_{u \leftarrow i}^{(l-1)}$  is the representation generated from the previous signal transfer step, which is used to store collaborative signals from  $(l-1)$  layer neighbours. Similarly, the  $l$  layer representation of the item  $i$  can be obtained.

3.1.3. *Prediction Layer*. After spreading the  $l$  layer, the  $l$  representation of the user  $u$ , namely,  $\{v_u^{(1)}, v_u^{(2)}, \dots, v_u^{(l)}\}$ , can be obtained. The embeddings obtained at each layer are further combined to form the final representation of the user (item):

$$v'_u = \frac{\sum_{l=0}^L \alpha_l v_u^{(l)}}{\sum_{l=0}^L \alpha_l} \quad (6)$$

$$v'_i = \frac{\sum_{l=0}^L \alpha_l v_i^{(l)}}{\sum_{l=0}^L \alpha_l},$$

where  $\alpha_l > 0$  represents the importance of layer embedding in forming the final embedding, which is equivalent to an attention mechanism. This is because, when the number of layers increases, the weight is on a downward trend. The final representation is changed from the original average of each layer to a weighted average. The deeper the layer, the greater the weight, which emphasizes the importance of the deep signal. To prevent the algorithm becoming too complex,  $\alpha_l$  is set to  $(1/k+1)$ .

Finally, the inner product is used to estimate the user's preference for the item:

$$y'_{ui} = v_u'^T v'_i. \quad (7)$$

It can be used as the ranking score generated by the recommendation.

**3.2. DropEdge Module.** The main task of DropEdge [26] is to randomly lose a certain number of edges from the input graph at each training phase, which can be understood as data enrichment. This can increase the randomness and diversity of the input data, which can mitigate overfitting. DropEdge’s approach can also be understood as the simplification of data transmission. Losing some edges makes node connections sparser, which can effectively avoid oversmoothing.

To apply the DropEdge, (5) is transformed into a matrix form, as follows:

$$V^{(l)} = LV^{(l-1)}, \quad (8)$$

where  $V^{(l)} \in R^{(N+M) \times d}$  is the representation of users and items obtained after embedding in the propagation layer and  $d$  is the embedding size. The initial signal  $V^{(0)}$  is set to  $V$ ; that is,  $v_u^{(0)} = v_u$  and  $v_i^{(0)} = v_i$ . Finally,  $L$  is the Laplacian matrix (symmetrical normalization) of the user-item interaction graph, which is defined as

$$L = D^{-(1/2)}AD^{-(1/2)} \quad (9)$$

$$A = \begin{bmatrix} 0 & R \\ R^T & 0 \end{bmatrix},$$

where  $A$  is the adjacency matrix,  $D \in R^{(M+N) \times (M+N)}$  is the pair angle matrix,  $D_{ii} = |N_i|$  is the number of nonzero entries in the  $i$ -th row vector of the adjacency matrix  $A$ , and  $R \in R^{M \times N}$  is the user-item interaction matrix. Finally, the final embedding matrix used for algorithm prediction is

$$V = \alpha_0 V^{(0)} + \alpha_1 V^{(1)} + \dots + \alpha_L V^{(L)} \quad (10)$$

$$= \alpha_0 V^{(0)} + \alpha_1 LV^{(0)} + \dots + \alpha_L L^{(L)} V^{(0)}.$$

In each training phase, DropEdge randomly selects a certain percentage of edges of the input graph. In other words, it randomly sets  $Vp$  nonzero elements in the adjacency matrix  $A$  to zero, where  $V$  is the total number of edges, and  $p$  is the loss rate. The obtained adjacency matrix  $A_{\text{drop}}$  is expressed as

$$A_{\text{drop}} = A - A', \quad (11)$$

where  $A'$  is expanded from a random subset of the original edge size of  $Vp$ . Then, the matrix is normalized and expressed as  $\widehat{A}_{\text{drop}}$ , which is used to replace  $A$  in the convolution operation in (9).

**3.3. Algorithm Training.** Reference algorithm [2], using paired Bayesian Personalized Ranking (BPR) loss [29], considers the relative order between observed and unobserved user-item interactions and encourages the prediction of observed items to be higher than that of unobserved items; the interactions with corresponding observed items are more reflective of user preferences.

The loss function of the algorithm is

$$\text{Loss} = - \sum_{(u,i,j) \in O} \ln \sigma(y_{ui}' - y_{uj}') + \lambda \|V^{(0)}\|_2^2, \quad (12)$$

where  $O = \{(u, i, j) | (u, i) \in R^+, (u, j) \in R^-\}$  is the paired training data,  $R^+$  is the observed interaction,  $R^-$  is the unobserved interaction,  $\sigma(\cdot)$  is a sigmoid function,  $\lambda$  controls the intensity of  $L_2$  regularization, and the training parameter is only the embedding of the 0th layer; that is,  $\Theta = \{V^{(0)}\}$ . A small batch of the Adam [30] optimizer is used to predict the algorithm and update the algorithm parameters.

## 4. Experiments

To verify the recommended performance of the proposed LG-DropEdge, this study used the PyTorch deep-learning framework. The operating system used in the experiment was Windows 10, the graphics card was a Nvidia Titan V, and the CPU was an i7-8700K. The Python version was 3.6. Experiments and analysis were conducted using the Pycharm2020 development tool and PyTorch deep-learning framework.

To compare the performance of the proposed algorithm with other algorithms and verify the effectiveness of the algorithm’s own modules, we designed three sets of experiments:

- (Q1) Compared with the most advanced LightGCN, how is the performance of LG-DropEdge compared?
- (Q2) How do the settings of different improved modules affect the performance of LG-DropEdge?
- (Q3) How do the hyperparameter settings (such as edge loss rate) impact the effectiveness of LG-DropEdge?

**4.1. Experimental Dataset.** To evaluate the effectiveness of LG-DropEdge, we conducted experiments on three public datasets: Gowalla, Yelp2018, and Amazon-book. The Gowalla dataset uses the login dataset from Gowalla [31], where users share their location by logging in. Yelp2018 comes from the 2018 version of the Yelp Challenge, in which companies such as hotels and coffee shops were used as items. Amazon-book [32] comes from book data in Amazon reviews. There are differences in data entities, numbers of interactions, and sparsity, which can meet the needs of different data characteristics as required by the algorithm. Table 1 summarizes the statistics of the three datasets.

Inspired by mainstream recommendation algorithms [23, 25], for each dataset, 80% of each user’s historical interactions were randomly selected to form the training set, and the rest were used as the test set. From the training set, 10% of the interactions were randomly selected as the validation set to adjust the hyperparameters. Each observed user-item interaction was treated as a positive instance. A negative sampling strategy was then used to pair it with a negative item that the user had not previously consumed.

TABLE 1: Statistics of the tested datasets.

Dataset	Gowalla	Yelp2018	Amazon-book
User	29,858	31,668	52,643
Item	40,981	38,048	91,599
Interaction	1,027,370	1,561,406	2,984,108
Density	0.000,84	0.001,30	0.000,62

## 4.2. Experimental Setup

**4.2.1. Evaluation Index.** The performance assessment of the recommendation algorithm has many aspects, divided into a quantitative calculation and qualitative description. This section describes the performance of the algorithm from the perspective of prediction precision. When calculating this indicator, an offline dataset is required that contains historical user behavior data. The dataset is then divided into a training set and a test set, and finally the user's behavior on the test set is predicted by establishing a user's behavior and interest algorithm on the training set. The coincidence degree of the predicted behavior and the actual behavior on the test set are calculated as the prediction precision. For users' Top-N recommendations, algorithm [14, 29] used evaluation indicators, such as precision, recall, and Normalized Discounted Cumulative Gain (NDCG) for  $N = \{20, 100\}$ .

Let  $R(u)$  denote the recommendation list given to the user through training and let  $T(u)$  denote the behavior list on the test set.

The precision represents the proportion of the number of samples that are correctly predicted to the total number of samples and is defined as

$$\text{precision@N} = \frac{\sum_{u \in N} |R(u) \cap T(u)|}{\sum_{u \in N} |T(u)|}. \quad (13)$$

The recall of the recommended result represents the probability that the sample is correctly predicted to occupy the actual sample, which is defined as

$$\text{recall@N} = \frac{\sum_{u \in N} |R(u) \cap T(u)|}{\sum_{u \in N} |R(u)|}. \quad (14)$$

The NDCG of the recommended result is defined as

$$\begin{aligned} \text{ndcg@N} &= \frac{\text{dcg}}{\text{idcg}}, \\ \text{dcg} &= \sum_{i=1}^N \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}, \\ \text{idcg} &= \sum_{i=1}^{|\text{REL}_N|} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}, \end{aligned} \quad (15)$$

where  $\text{dcg}$  is the cumulative gain of loss,  $\text{idcg}$  is the maximum value of  $\text{dcg}$  under ideal conditions, and  $\text{rel}_i \in \{0, 1\}$  indicates the user's rating for the  $i$ -th item. Finally,  $|\text{REL}_N|$  indicates that the results are sorted in descending order of relevance, where the set consisting of the previous  $N$  results is adopted; that is, according to the most, sort the results in an optimal way.

**4.2.2. Baseline.** To demonstrate the effectiveness of the proposed algorithm, we compared it with the following methods:

NeuMF [2]: this method is an advanced neural collaborative filtering algorithm that uses multiple hidden layers above elements and the concatenation of user and item embeddings to capture their nonlinear characteristic interactions.

Hop-Rec [22]: this method combines matrix decomposition and graphs through random walks on the graph, combined with the degree of the vertices, sampling different positive samples with a certain probability, and assigning attenuation coefficients to the ranking pairs obtained in different orders.

NGCF [23]: this method explicitly introduces the collaborative signal into the collaborative filtering algorithm and achieves this by using the high-level connectivity in the user-item interaction graph.

LightGCN [25]: based on NGCF, this method removes feature changes and nonlinear activation through ablation experiments and adds a weight factor to the final aggregation, which is greatly improved. It is the latest direction of graph convolution.

**4.2.3. Parameter Settings.** With reference to NGCF [23] and LightGCN [25], the LG-DropEdge algorithm is implemented in PyTorch. Considering the settings of the comparison experiment, choose the same hyperparametric settings as NGCF and LightGCN, etc. To ensure the accuracy and fairness of the comparison results, the embedding size of all algorithms is set to 64; the learning rate is set to  $1e^{-3}$ ; the number of layers is set to 3; the batch size is set to 2048; the regularization coefficient is set to  $1e^{-3}$ ; considering the convergence of loss, the number of training is set to 1500. Grid search is performed on the hyperparameters and the edge loss rate is determined in  $\{0.8, 0.6, 0.5, 0.4, 0.2\}$ .

## 4.3. Performance Comparison (Q1)

**4.3.1. Performance Comparison with LightGCN.** Compared with the LightGCN algorithm, Table 2 records the algorithm performance for different datasets and different indicators and shows the percentage improvement of each indicator. It reveals the clear improvement made by the proposed LG-DropEdge.

In the three sets of comparative experiments, LG-DropEdge performs better than LightGCN. In the three datasets, the algorithm performance on the Yelp2018 dataset has been greatly improved; the precision index has increased by 6.04%

TABLE 2: Performance comparison of LG-DropEdge and LightGCN.

Dataset	Method	Precision@20	Recall@20	ndcg@20
Gowalla	LightGCN	0.0558	0.1821	0.1545
	LG-DropEdge	<b>0.0575 (+3.04%)</b>	<b>0.1867 (+2.53%)</b>	<b>0.1582 (+2.39%)</b>
Yelp2018	LightGCN	0.0283	0.0632	0.0520
	LG-DropEdge	<b>0.0301 (+6.36%)</b>	<b>0.0671 (+6.17%)</b>	<b>0.0549 (+5.58%)</b>
Amazon-book	LightGCN	0.0171	0.0415	0.0321
	LG-DropEdge	<b>0.0180 (+5.26%)</b>	<b>0.0435 (+4.82%)</b>	<b>0.0336 (+4.67%)</b>

The bold values represent the experimental results of the algorithm proposed in this paper and the improvement effect on the classical experiment.

on average. The main reason for this is that the Yelp2018 datasets are sparser than the other two datasets. A dataset with high sparseness has a large number of user interaction items, and the data characteristics can be better retained after multiple algorithm training phases. In addition, the average precision of the three datasets increased by 4.89%, recall increased by 4.51%, ndcg increased by 4.21%, and overall performance improved.

**4.3.2. Overall Comparison.** To verify the advantages of the LG-DropEdge recommendation algorithm in terms of precision, we implemented performance comparisons with other classic algorithms (NeuMF [2], Hop-Rec [22], NGCF [23], and LightGCN [25]). The experimental results are shown in Table 3.

The following can be seen from Table 3:

- (1) NeuMF performs better than Hop-Rec on the Amazon-book dataset. Since Hop-Rec is implemented by combining MF and graphs, its performance largely depends on the random walk algorithm, and the effect is not very obvious because it does not make full use of high-order connectivity of graphs.
- (2) Since Hop-Rec is implemented by combining MF and graphs, it does not make full use of high-order connectivity. Its performance largely depends on the random walk, and the effect is not very obvious.
- (3) NGCF yields a significantly better performance than NeuMF and Hop-Rec because it explicitly introduces the collaborative signal into the system filtering algorithm and spreads it on the interactive graph. However, its algorithm directly inherits GCN, which leads to increased algorithm complexity and training time.
- (4) LightGCN, as a simplification of NGCF, yields a powerful performance, but its algorithm does not solve the problem of deep-network overfitting. Further, it uses relatively simple aggregation functions, which limit the effect of improvement.
- (5) LG-DropEdge yields the best performance on all datasets, particularly Yelp2018, which showed an increase of more than 5%. Thus, the LG-DropEdge algorithm can be used to improve the precision of a recommendation system.

**4.4. Ablation Experiment (Q2).** To demonstrate the feasibility of LG-DropEdge, the algorithm is subjected to ablation experiments. To verify the effectiveness of the combined modules, we performed several sets of experiments on the three datasets: removing the improved aggregation function and keeping DropEdge, named LightGCN + DropEdge; removing DropEdge and keeping the improved aggregation function, named LightGCN + f; removing both DropEdge and the improved aggregation function, which is the original LightGCN; and the proposed method, LG-DropEdge.

The results are as shown in Table 4.

The following findings can be made from Table 4:

- (1) Compared with LightGCN, precision, recall, and ndcg increased by 2.33%, 1.48%, and 1.23%, respectively, in the LightGCN + f algorithm
- (2) Compared with LightGCN, precision, recall, and ndcg decreased by 0.72%, 0.49%, and 0.19% respectively, in the LightGCN + DropEdge algorithm
- (3) Compared with LightGCN, precision, recall, and ndcg increased by 3.04%, 2.53%, and 2.39%, respectively, in the LG-DropEdge algorithm

The results are as shown in Table 5.

The following findings can be made from Table 5:

- (1) Compared with LightGCN, precision, recall, and ndcg increased by 5.65%, 5.38%, and 5.00%, respectively, in the LightGCN + f algorithm
- (2) Compared with LightGCN, precision, recall, and ndcg decreased by 2.12%, 2.58%, and 2.69%, respectively, in the LightGCN + DropEdge algorithm
- (3) Compared with LightGCN, precision, recall, and ndcg increased by 6.36%, 6.17%, and 5.58%, respectively, in the LG-DropEdge algorithm

The results are as shown in Table 6.

The following Findings can be made from the Table 6:

- (1) Compared with LightGCN, precision, recall, and ndcg increased by 2.92%, 1.69%, and 1.56%, respectively, in the LightGCN + f algorithm
- (2) Compared with LightGCN, precision, recall, and ndcg increased by 2.34%, 2.65%, and 2.49%, respectively, in the LightGCN + DropEdge algorithm
- (3) Compared with LightGCN, precision, recall, and ndcg increased by 5.26%, 4.82%, and 4.67%, respectively, in the LG-DropEdge algorithm

TABLE 3: Overall performance comparison.

Dataset method	Gowalla		Yelp2018		Amazon-book	
	Recall@20	ndcg@20	Recall@20	ndcg@20	Recall@20	ndcg@20
NeuMF	0.1339	0.1050	0.0445	0.0359	0.0327	0.0248
Hop-Rec	0.1399	0.1201	0.0517	0.0428	0.0296	0.0211
NGCF	0.1547	0.1313	0.0562	0.0459	0.0324	0.0250
LightGCN	0.1821	0.1545	0.0632	0.0520	0.0415	0.0321
LG-DropEdge	<b>0.1867</b>	<b>0.1582</b>	<b>0.0671</b>	<b>0.0549</b>	<b>0.0435</b>	<b>0.0336</b>

The bold values represent the experimental results of the algorithm proposed in this paper.

TABLE 4: Performance of the Gowalla dataset under the four algorithms.

Algorithm	Precision	Recall	ndcg
LightGCN	0.0558	0.1821	0.1545
LightGCN + $f$	0.0571	0.1848	0.1564
LightGCN + DropEdge	0.0554	0.1812	0.1542
LG-DropEdge	0.0575	0.1867	0.1582

TABLE 5: Performance of the Yelp2018 dataset under the four algorithms.

Algorithm	Precision	Recall	ndcg
LightGCN	0.0283	0.0632	0.0520
LightGCN + $f$	0.0299	0.0666	0.0546
LightGCN + DropEdge	0.0277	0.0614	0.0506
LG-DropEdge	0.0301	0.0671	0.0549

TABLE 6: Performance of the Amazon-book dataset under the four algorithms.

Algorithm	Precision	Recall	ndcg
LightGCN	0.0171	0.0415	0.0321
LightGCN + $f$	0.0176	0.0422	0.0326
LightGCN + DropEdge	0.0175	0.0426	0.0329
LG-DropEdge	0.0180	0.0435	0.0336

Combining Tables 4–6, the following can be obtained:

- (1) Compared with LightGCN, LightGCN +  $f$  algorithm has different degrees of improvement in the three datasets of precision, recall, and ndcg
- (2) Compared with LightGCN, the LightGCN + DropEdge algorithm has decreased in the precision, recall, and ndcg indicators of the Gowalla and Yelp2018 datasets, but there has been a small increase in the Amazon-book dataset. It can be seen that adding DropEdge has less impact on the performance of less sparse datasets
- (3) Compared with LightGCN, LG-DropEdge has a greater improvement over LightGCN-f on the three indicators of precision, recall, and ndcg in three datasets, which fully demonstrates that modifying the aggregation function and adding DropEdge make the algorithm more accurate

To explain the effectiveness of the DropEdge, we compared the loss tested under the three datasets at the same ndcg level. The performance on the test set measures the true performance of the algorithm. Under the same test set index (ndcg@20), the training performance was the same, and the train\_loss produced by adding DropEdge with the training set was higher, indicating

that it is at the same level and there is no mitigation of overfitting. Figure 2 compares the train\_loss for 23 sets of data in Gowalla dataset, Figure 3 compares the train\_loss for 55 sets of data in Yelp2018 dataset, and Figure 4 compares the train\_loss for 41 sets of data in Amazon-book dataset (ndcg is arranged in increasing order), showing that it has increased to different degrees. It can be seen that adding DropEdge is effectively mitigating overfitting. Furthermore, as ndcg increases, LG-DropEdge performs more smoothly in terms of loss, in contrast to the irregular fluctuations of LightGCN. Therefore, the algorithm proposed in this paper has many advantages in mitigating overfitting and oversmoothing problems.

**4.5. Hyperparameter Experiment (Q3).** To verify the degree of influence of the edge loss rate on the algorithm, the edge loss rate  $p$  is determined in  $\{0.8, 0.6, 0.5, 0.4, 0.2\}$  (using three datasets), as in [26].

Figure 5 shows the performance for precision, recall, train\_time, and ndcg under different edge loss rates. As seen in Figure 5(a), starting from 0.0, precision and recall maintain the same upward trend as the edge loss rate increases. The first peak is reached at 0.4, after which the two indicators decrease steadily and slightly between 0.4 and 0.5, increase significantly between 0.5 and 0.6, and achieve their best performance at 0.6. After 0.6,

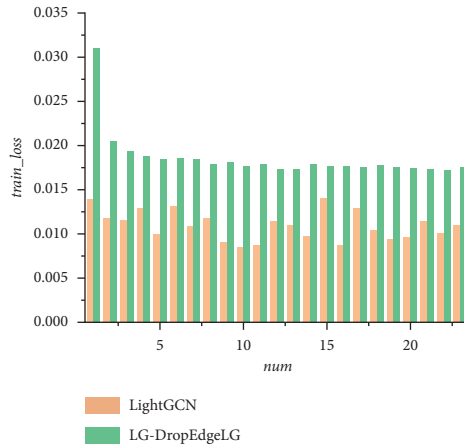


FIGURE 2: Train\_loss comparison at the same ndcg level (Gowalla).

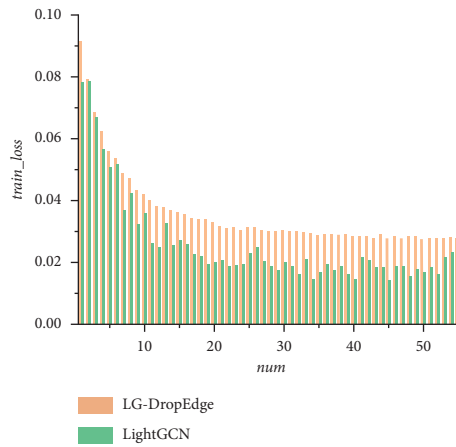


FIGURE 3: Train\_loss comparison at the same ndcg level (Yelp2018).

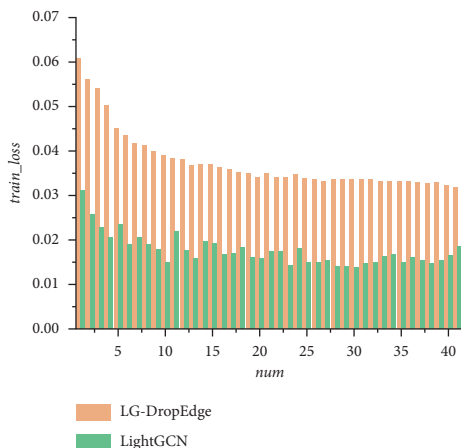


FIGURE 4: Train\_loss comparison at the same ndcg level (Amazon-book).

they begin to decline sharply. The precision index is at its worst value at 0.8. Meanwhile, as seen in Figure 5(b), the ndcg index maintains a steady increase from 0.0, achieves maximum value at 0.6, after which it drops sharply, and reaches its worst value at

0.8. The train\_time index starts at the lowest value at 0.0 and then increases rapidly to 0.2, before declining slowly. Therefore, the overall trends for precision, recall, and ndcg are increasing from the start 0.0, achieving a maximum at 0.6, and thereafter declining rapidly. Time is expressed as the average train\_time for an epoch. It is obvious that when DropEdge is increased, the train\_time increases significantly. Undoubtedly, it increases the difficulty of training. From Figure 5, when the train\_time is relatively short, the precision index performs the best and the edge loss rate  $p$  of DropEdge is finally determined to be 0.6.

Figure 6 shows the performance for precision, recall, train\_time, and ndcg under different edge loss rates. As can be seen from the figure, in Figure 6(a), the precision and recall both increase between 0.0 and 0.2, and they begin to decline between 0.2 and 0.4, increase between 0.5 and 0.6, and then continue to decline. The only trend difference is between 0.4 and 0.5, precision increases, and recall decreases, but the overall trend is the same, first increasing to the highest point ( $p = 0.2$ ) and then slightly lowering, then slightly increasing, and finally falling to the lowest. In Figure 6(b), the ndcg indicator increases between 0.0 and 0.2, reaches the highest point at that time, drops to the lowest point between 0.2 and 0.4, and then increases sharply between 0.4 and 0.5; it

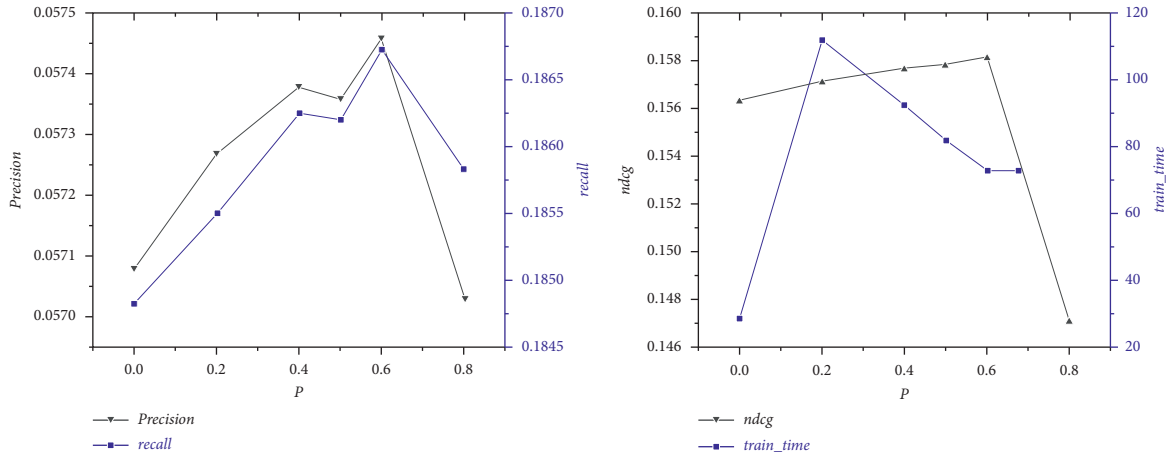


FIGURE 5: Algorithm performance under different edge drop rates (Gowalla).

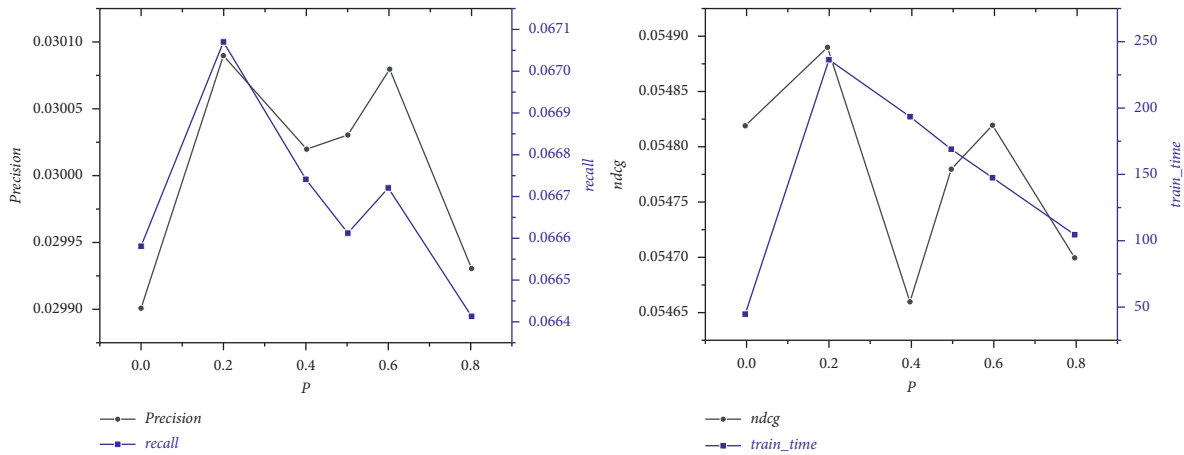


FIGURE 6: Algorithm performance under different edge drop rates (Yelp2018).

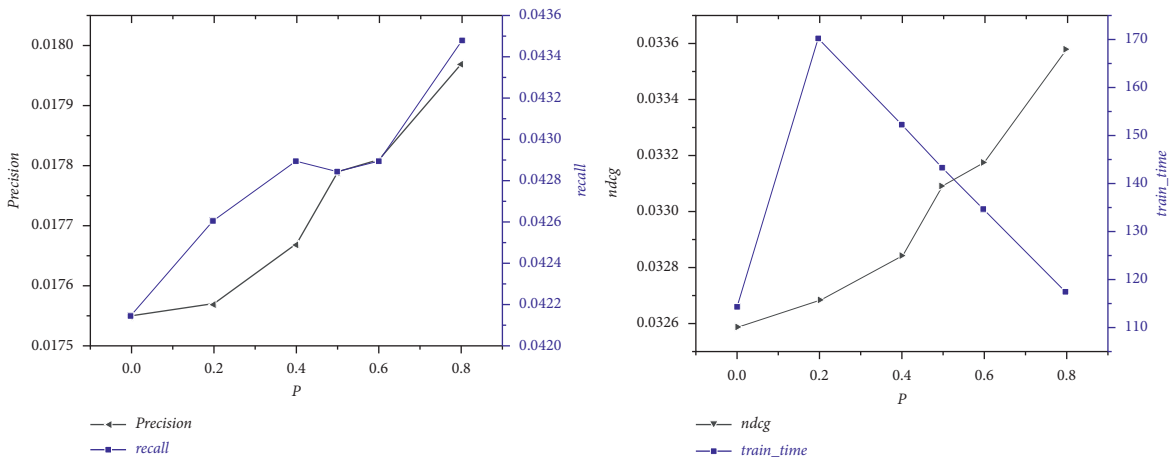


FIGURE 7: Algorithm performance under different edge drop rates (Amazon-book).

increases slowly between 0.5 and 0.6 and begins to decline after 0.6; while train\_time indicator starts to decline after rising from 0.0 to 0.2 to the highest point. Combined with the

two subgraphs, all three accuracy indicators reach the best state at that time ( $p = 0.2$ ), then begin to decline, then rise and reach the second peak at  $p = 0.6$ , and then begin to decline.

According to the rule of relatively little train\_time and priority in accuracy index performance, the edge loss rate  $p$  of DropEdge is finally determined to be 0.2.

Figure 7 shows the performance for precision, recall, train\_time, and ndcg under different edge loss rates. It can be seen from the figure that, in Figure 7(a), the two indicators generally show a continuous increase trend. Only precision has a small decline between 0.4 and 0.5, both of which reach the maximum value at  $p = 0.8$ ; in Figure 7(b), ndcg also shows an increase trend, reaching the maximum value at  $p = 0.8$ , while train\_time at  $p = 0.2$  reaches the maximum value and then continues to decline. The two indicators have an intersection between 0.5 and 0.6. After the intersection, they both develop in a positive direction and reach the optimal state of the algorithm at  $p = 0.8$ , especially after  $p = 0.6$  showing a rapid positive growth, so the edge loss rate  $p$  of DropEdge is finally determined to be 0.8.

The purpose of this section is to fully verify the improvement of the performance of the algorithm after adding the DropEdge module. It can be seen that the final boundary edge loss rate for different datasets is different. It can be seen that, in machine learning, the algorithm's adaptability to different datasets is different. Based on the results of the above three datasets, there is reason to believe that the performance of arithmetic has been greatly improved by adding the DropEdge module.

## 5. Conclusion

This paper proposed a joint light graph convolutional network and DropEdge recommendation algorithm (LG-DropEdge). The DropEdge was developed based on the LightGCN framework, which improves the final multilayer fusion aggregation function, mitigates the overfitting problem caused by the deep network algorithm, and improves the precision, recall, and ndcg, and enhances the interpretability of the recommendation algorithm. The proposed algorithm has improved the recommendation accuracy in the offline experiments of the three datasets, but it has not been verified in practical application. In the later research, we will continue to apply other public datasets to our algorithm and gradually apply them to the actual recommendation system to improve the generalization of the algorithm. The existing recommendation algorithm based on graph convolution network generally controls the number of convolution layers at 3 layers considering the attenuation degree of propagation factor, while DropEdge fully demonstrates that its performance is generally in deeper convolutional networks. This paper sets the number of convolution layers to be 3 layers considering the sparseness of datasets and the needs of contrast algorithms. Therefore, future research is to select more sparse datasets for comparative experiments and further analyse the relationship between the number of convolutional layers and the recommendation accuracy.

## Data Availability

The data used to support the findings of the study are included in the supplementary files.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This paper was supported by Young Scientists' Fund of National Natural Science Foundation of China and Natural Science Foundation of Liaoning Province.

## Supplementary Materials

gowalla-test/train.txt;yelp-test/train.txt;amazon-test/train.txt includes data used to support the findings of this study. run-gowalla/yelp2018/amazon.txt includes data values given in Tables 2–6 and figures, NGCF and LightGCN results, the datasets needed for the experiment, and part of the log files of the experiment. The results of NeuMF and Hop-Rec in Table 3 are quoted from Xiang Wang, Xiangnan he, Meng Wang, Fuli Feng, and Tat Seng Chua, "neural graph collaborative filtering," in SIGIR, pp. 165-174. The code data used to support the findings of this study have been deposited in the [Baidu Netdisk] repository (link: <https://pan.baidu.com/s/1JUFyBpoc3RPoKkPsQXvXkQ>; extraction code: gaz1). (*Supplementary Materials*)

## References

- [1] Y. J. Zhang, Z. Dong, and X. W. Meng, "Research on personalized advertising recommendation systems and their applications," *Chinese Journal of Computers*, vol. 44, no. 3, pp. 531–561, 2021.
- [2] X. He, L. Liao, H. Zhang, L. Nie, and X. S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, Perth, Australia, April 2017.
- [3] D. Valcarce, A. Landin, J. Parapar, and Á. Barreiro, "Collaborative filtering embeddings for memory-based recommender systems," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 347–356, 2019.
- [4] M. Gao, B. Ling, L. Yang, J. Wen, Q. Xiong, and S. Li, "From similarity perspective: a robust collaborative filtering approach for service recommendations," *Frontiers of Computer Science*, vol. 13, no. 2, pp. 231–246, 2019.
- [5] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, pp. 1257–1264, 2008.
- [6] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [7] Y. He, C. Wang, and C. Jiang, "Correlated matrix factorization for recommendation with implicit feedback," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 451–464, 2019.
- [8] Y. Cai, T. Luan, H. Gao et al., "YOLOv4-5D: an effective and Efficient Object Detector for Autonomous Driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, Article ID 4503613, 2021.
- [9] D. Ma, X. Song, and P. Li, "Daily Traffic Flow Forecasting through a Contextual convolutional Recurrent neural network Modeling inter- and Intra-Day Traffic Patterns," *IEEE*



- Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2627–2636, 2021.
- [10] F. Monti, D. Boscai, J. Masci, E. Rodolà, J. Svoboda, and M. Bronstein, “Geometric Deep Learning on Graphs and Manifolds Using Mixture Model CNNs,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5425–5434, Honolulu, HI, USA, July 2017.
- [11] F. Scarselli, M. Gori, G. Monfardini, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network Model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.
- [12] J. Bruna, W. Zaremba, A. D. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” 2014, <https://arxiv.org/abs/1312.6203>.
- [13] M. Henaff, J. Bruna, and Y. LeCun, “Deep Convolutional Networks on Graph-Structured Data,” 2015, <https://arxiv.org/abs/1506.05163>.
- [14] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with Fast Localized spectral filtering,” *Advances in Neural Information Processing Systems*, vol. 29, pp. 3844–3852, 2016.
- [15] T. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” 2017, <https://arxiv.org/abs/1609.02907>.
- [16] R. Li, S. Wang, F. Zhu, and J. Huang, “Adaptive graph convolutional neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, LA, USA, February 2018.
- [17] W. L. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, Red Hook, NY, United States, December 2017.
- [18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph Attention Networks,” 2018, <https://arxiv.org/abs/1710.10903>.
- [19] X. Wang, H. Ji, C. Shi et al., “Heterogeneous Graph Attention Network,” in *Proceedings of the World Wide Web Conference*, pp. 2022–2032, CA, San Francisco, USA, May 2019.
- [20] R. Ying, R. He, K. Chen, P. Eksombatchai, L. H. William, and J. Leskovec, “graph convolutional neural networks for Web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983, London, United Kingdom, August 2018.
- [21] R. V. D. Berg, T. Kipf, and M. Welling, “Graph Convolutional Matrix Completion,” 2017, <https://arxiv.org/abs/1706.02263>.
- [22] J. H. Yang, C. M. Chen, C. J. Wang, and M. F. Tsai, “HOP-rec: high-order proximity for implicit recommendation,” in *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 140–144, British Columbia, Vancouver, Canada, October 2018.
- [23] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, “Neural graph collaborative filtering,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR’19*, pp. 165–174, Paris, France, July 2019.
- [24] F. Wu, T. Zhang, A. Souza, C. Fifty, Y. Tao, and K. Q. Weinberger, “Simplifying Graph Convolutional Networks,” pp. 6861–6871, 2019, <https://arxiv.org/abs/1902.07153>.
- [25] X. He, K. Deng, X. Wang, Y. Li, Y. D. Zhang, and M. Wang, “LightGCN: simplifying and powering graph convolution network for recommendation,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event*, pp. 639–6498, ACM Press, New York, USA, July 2020.
- [26] Y. Rong, W. Huang, T. Xu, and J. Huang, “DropEdge: Towards Deep Graph Convolutional Networks on Node Classification,” 2020, <https://arxiv.org/abs/1907.10903>.
- [27] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, CA, USA, December 2017.
- [28] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka, “Representation learning on graphs with Jumping Knowledge networks,” in *ICML*, vol. 80, pp. 5453–5462, 2018.
- [29] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “B. P. R.: Bayesian personalized ranking from implicit feedback,” pp. 452–461, 2009, <https://arxiv.org/abs/1205.2618v2>.
- [30] P. K. Diederik and J. Ba, “Adam: a method for Stochastic Optimization,” 2015, <https://arxiv.org/abs/1412.6980>.
- [31] D. Liang, L. Charlin, J. McInerney, and D. M. Blei, “Modeling user Exposure in recommendation,” in *Proceedings of the 25th International Conference on World Wide Web*, pp. 951–961, Québec, Montréal, Canada, April 2012.
- [32] R. He and J. McAuley, “Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering,” in *Proceedings of the 25th International Conference on World Wide Web*, pp. 507–517, Québec, Montréal, Canada, April 2016.

## Research Article

# Online Traffic Accident Spatial-Temporal Post-Impact Prediction Model on Highways Based on Spiking Neural Networks

Duowei Li , Jianping Wu , and Depin Peng

Department of Civil Engineering, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Jianping Wu; [jianpingwu@tsinghua.edu.cn](mailto:jianpingwu@tsinghua.edu.cn)

Received 2 August 2021; Accepted 29 October 2021; Published 2 December 2021

Academic Editor: Alain Lambert

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

Traffic accident management as an approach to improve public security and reduce economic losses has received public attention for a long time, among which traffic accidents post-impact prediction (TAPIP) is one of the most important procedures. However, existing systems and methodologies for TAPIP are insufficient for addressing the problem. The drawbacks include ignoring the recovery process after clearance and failing to make comprehensive prediction in both time and space domain. To this end, we build a 3-stage TAPIP model on highways, using the technology of spiking neural networks (SNNs) and convolutional neural networks (CNNs). By dividing the accident lifetime into two phases, i.e., clean-up phase and recovery phase, the model extracts characteristics in each phase and achieves prediction of spatial-temporal post-impact variables (e.g., clean-up time, recovery time, and accumulative queue length). The framework takes advantage of SNNs to efficiently capture accident spatial-temporal features and CNNs to precisely represent the traffic environment. Integrated with an adaptation and updating mechanism, the whole system works autonomously in an online manner that continues to self-improve during usage. By testing with a new dataset CASTA pertaining to California statewide traffic accidents on highways collected in four years, we prove that the proposed model achieves higher prediction accuracy than other methods (e.g., KNN, shockwave theory, and ANNs). This work is the introduction of SNNs in the traffic accident prediction domain and also a complete description of post-impact in the whole accident lifetime.

## 1. Introduction

People's living standards have increased all over the world, leading to an increase in the ownership of private vehicles [1]. While private vehicles have improved people's traveling experience, they have also contributed to several traffic problems, where traffic safety is one of the main concerns. According to statistical data released by World Health Organization in 2004, road traffic accidents are among the main causes of deaths and injuries all over the world, leading to 1.2 million deaths and 50 million injuries each year [2]. In 2019, the number of traffic accidents in China was 248,000, and the direct property loss was as high as 1.35 billion yuan (China Statistical Yearbook).

Besides the costs of fatalities and injuries, indirect effects brought by traffic accidents such as congestion and energy consumption also have tremendous impacts on socioeconomic development. Typically, accident congestion can

spread rapidly and even cause a chain breakdown in the entire system, especially in closed roads like highways. In severe cases, the delays of clearance for an accident may increase the occurrence probability of a secondary accident [3]. In 1988, the total extra travel time and fuel consumption caused by traffic congestion, both regular and occasional, in 50 major cities in the United States was estimated at 35 billion U.S. dollars [4].

Traffic accident management (TAM) is of great importance to government and transportation agencies, which can not only improve public security but also reduce economic losses. There are usually two main tasks in TAM: analysis and prediction. By analyzing the features of historical accident records, accident prediction can be realized to foresee the time and place of the accident in advance. Furthermore, traffic accidents post-impact prediction (TAPIP) is also an essential part in TAM but have not caught enough attention. When an accident occurs, if the spatial and temporal impact scope can

be quickly determined, then the transportation agency can induce vehicles to avoid unnecessary congestion, and travelers can also plan their routes in advance.

The lifetime of a traffic accident is usually divided into five phases, namely, detection, verification, response, clearance, and recovery [5]. Most TAPIP-related research studies till now concentrate only on the first four phases, which means the duration from accident occurrence to the reopening of all blocked traffic lanes. The existing research studies use traffic wave theory, regression model, and other methods to analyze the influence of various factors on accident indicators, but have not established a complete relationship between the influencing factors and the spatial-temporal impact of the accidents. Furthermore, the clearance is never the end of an accident, and the process that the traffic gradually recovers to normal conditions cannot be ignored. For the convenience of expression, this work proposes a two-phase accident lifetime. As shown in Figure 1, the four phases before clearance are grouped into clean-up phase, while the recovery phase remains unchanged. The reason for this division is that in the clean-up phase, at least one lane will be blocked, while after that, the whole lanes will be reopened.

As the third-generation neuron model, spiking neural networks (SNNs) [6] are closer to biological principle compared with other artificial neural networks (ANNs). Due to the advantage of being sensitive to spatial-temporal characteristics of information [7], SNNs already improve the traditional neural models on accuracy in many application areas. However, SNNs till now are rarely used in traffic domain and have even never been applied to accident prediction related research.

This study proposes a traffic accident spatial-temporal post-impact prediction model on highways, using the technology of SNNs and CNNs. In the 3-stage model, the duration of clean-up phase (clean-up time) is first estimated, and then, several post-impact variables such as recovery time and accumulative queue length are predicted. Additional online stage is used to make adaptation and update the model. In this regard, the main contributions of this work can be summarized as follows:

- (1) The development of a framework to predict post-impact in the complete accident lifetime, which contains the recovery process.
- (2) The prediction of highway accident post-impact in both spatial and temporal domain, and the outputs are specific and effective variables that can help both transportation agencies and travelers.
- (3) The first use of SNNs in the traffic accident prediction domain and the implementation of encoding the traffic spatial-temporal features into time spikes.
- (4) The processing of a new highway spatial-temporal traffic accident dataset CASTA and the use of it in testing the performance of the proposed model.
- (5) The proposal of an online adaptation and updating method to revise the error and modify the model, for self-improvement during usage.

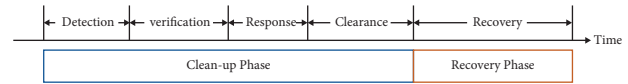


FIGURE 1: Two-phase traffic accident lifetime.

The rest of this study is organized as follows. Section 2 describes literature review on SNNs and TAPIP. Section 3 defines the general framework of the 3-stage prediction model, including the specific input, output, and structure of SNNs and CNNs. The processing of the new dataset is introduced in Section 4, and Section 5 presents the experiment results. Finally, Section 6 provides concluding statements on our work.

## 2. Literature Review

*2.1. Spiking Neural Networks in Traffic Domain.* With the rapid development of artificial intelligence and computer technology, deep learning has been widely applied in traffic domain, including traffic state prediction [8, 9], traffic signal control [1, 10], and driving model development [11].

The works mentioned above are all based on artificial neural networks (ANNs), where the neurons use differentiable, nonlinear activation functions. Although ANNs have achieved great performance in wide areas including classification, pattern recognition, and prediction, they are fundamentally different in structure, neural computations, and learning rule compared to the brain [7]. The way of information propagation between units is one of the most obvious differences. While ANNs rely on 32 bit or 64 bit continuous messages sent between units [12], the neurons in biological brain broadcast trains of action potentials, also known as spike trains to downstream neurons. As the third-generation neuron model, spiking neural networks (SNNs) are close to biological principle by utilizing spikes. When dealing with complex or large datasets, while ANNs face the concern of huge computational consumption, SNNs achieve energy efficiency since output spike trains can be made sparse in time [13]. Besides that, SNNs improve the traditional neural models on accuracy due to the advantage of being sensitive to the spatial-temporal characteristics of information [7]. Although still in the early stages of development, SNNs have become the focus of many application areas and made progress in visual processing [12], speech recognition [14], and medical diagnosis [15].

However, SNNs till now are rarely used in traffic domain, despite their specialized capability in representing spatial-temporal data. Laña et al. [16] presents an approach for spatial-temporal road traffic forecasting that relies on the adoption of the NeuCube architecture based on SNNs. Taking advantages of the NeuCube platform, this work focuses on the spatially-aware traffic variable forecasts and on the exploration of the spatial-temporal relationships among different sensor locations within a traffic network. In another work of Laña et al. [17], an evolving spiking neural networks (eSNNs) based adaptive long-term traffic state estimation model is proposed. By using similarity-based clustering of daily traffic volume data and monitoring them

in real time, the model encodes traffic data into spikes over the time domain effectively and achieves high accuracy in online prediction when new data samples arrive.

**2.2. Post-Impact Prediction of Traffic Accident.** At present, the main research methods in predicting the impact of highway accidents are the traffic wave theory [18, 19], vehicle arriving-departure model [20, 21], regression model [22, 23], decision tree model [5], and ANNs [23, 24]. Yu et al. [18] analyzed the accumulation and dissipation process of the accident location on two-lane highway, using the shockwave theory as a foundation, and finally estimated the spatial-temporal impact of the accident. However, such methods based on theoretical analysis considered only limited influencing factors (flow and density of upstream) and only showed general laws that lead to poor applicability. Zhu et al. [23] established two kinds of forecasting models of spatial-temporal impact for traffic accidents based on nonlinear regression and BP neural networks, where the influence of factors were analyzed such as upstream flowrate, ratio of cargos, and accident handling time. The results showed that the BP neural network achieves higher accuracy. Lin and Li [24] embedded three machine learning algorithms in a hierarchical scheme to perform sequential prediction. The result outperformed others by achieving a MAPE range of 5.5–53.8%. However, it just considered the time duration of an accident rather than spatial impact. In contrast, Lee et al. [25] only focused on estimation of vehicle accident queue length using the ANN model with relatively abundant influencing factors.

The methods above theoretically described and inferred the characteristics of incidental accidents and analyzed the influence of various factors on accident indicators. But they have not established a complete relationship model between the influencing factors and the spatial-temporal impact of the accidents. Furthermore, related works all used one model to output all prediction variables regardless of the characteristics in different phases of accidents. However, the five phases are closely related and can have great influence on each other. Thus, in order to fully capture the spatial-temporal features in the whole process, the prediction should be made separately according to the order of phase.

Since the prediction of accident impact highly relies on feature extraction of spatial and temporal data, SNNs are considered to be a more suitable method due to its characteristics compared with the models mentioned above. However, no such SNNs-related models have been proposed till now.

### 3. Methodology

This work proposes a spatial-temporal post-impact prediction model of traffic accidents. The 3-stage process is shown in Figures 2 and 3, where Figure 2 corresponds to the training process using historic accident records (training set) and Figure 3 corresponds to the predicting process using new accident records (test set). Stage 1 is the prediction of accident clean-up time; it uses clusters to define length

hierarchy of clean-up time and then extract accident features to train the SNNs classifier. Stage 2 concerns with a prediction model of spatial-temporal impact based on CNNs and backpropagation neural networks (BPNNs), feeds with flow and speed data, and outputs the recovery time and accumulative queue length. When a new accident record comes to the model after being processed with the above two stages, another online updating stage will be promoted to correct error and revise model. The notations used frequently in the 3-stage model are given in Table 1.

**3.1. Stage 1: Accident Clean-Up Time Prediction.** According to the two-phase accident lifetime defined in Section 1, phases are not completely independent and can influence each other. Therefore, in order to figure out the spatial-temporal impact in the whole process, we need to predict the duration of clean-up phase first.

Mean-shift clustering is first performed on the clean-up time, which groups the clean-up time into several intervals. Thus, the accidents with similar cleaning difficulty are labeled as a cluster. Furthermore, the accident records with attributes and corresponding class labels are input to train an SNNs classifier. Whenever a new accident record comes in, the accident attributes are extracted and then classified using the trained classifier. The predicting clean-up time is represented by the centroid value of its corresponding cluster.

**3.1.1. Clustering Using Mean-Shift.** According to the statistics in Section 4, the clean-up time of the accidents distributes unevenly throughout the interval. It exists the case where some of the interval lack records. Thus, to improve the prediction accuracy and remove exceptional, clustering the clean-up time into groups and using the centroid value to present the predicted value is better than a direct prediction model.

The mean-shift algorithm [26] is chosen to cluster the clean-up time for two main reasons: it does not require a predefined number of clusters; and such density-based algorithms are less affected by the mean value compared with the  $K$ -means algorithm. The key operation of the algorithm is to calculate the drift vector of the centroid through the data density change in the region of data, so as to move the centroid in the next iteration until it reaches the maximum density. Algorithm 1 shows the procedure of clustering the traffic records according to the clean-up time. The bandwidth is set as 10.

**3.1.2. Accident Characterization.** When a new accident record comes to the model, in order to predict the clean-up time, the cluster that the records belongs to needs to be figured out. Thus, a number of attributes that represent the characteristics of accidents are used to train a classifier, which outputs the corresponding cluster.

In a previous work, three kinds of attributes are usually extracted to demonstrate the accident: accident information (e.g. accident type [24, 25, 27–29], blocked area size [5, 25], number of vehicles involved [5, 25, 28], and casualties

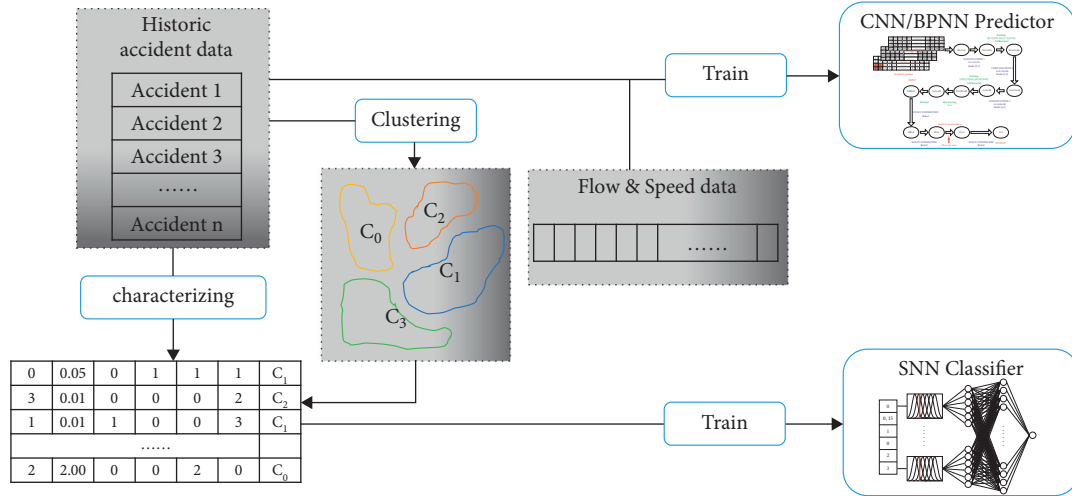


FIGURE 2: 3-stage model for accident post-impact prediction-training process.

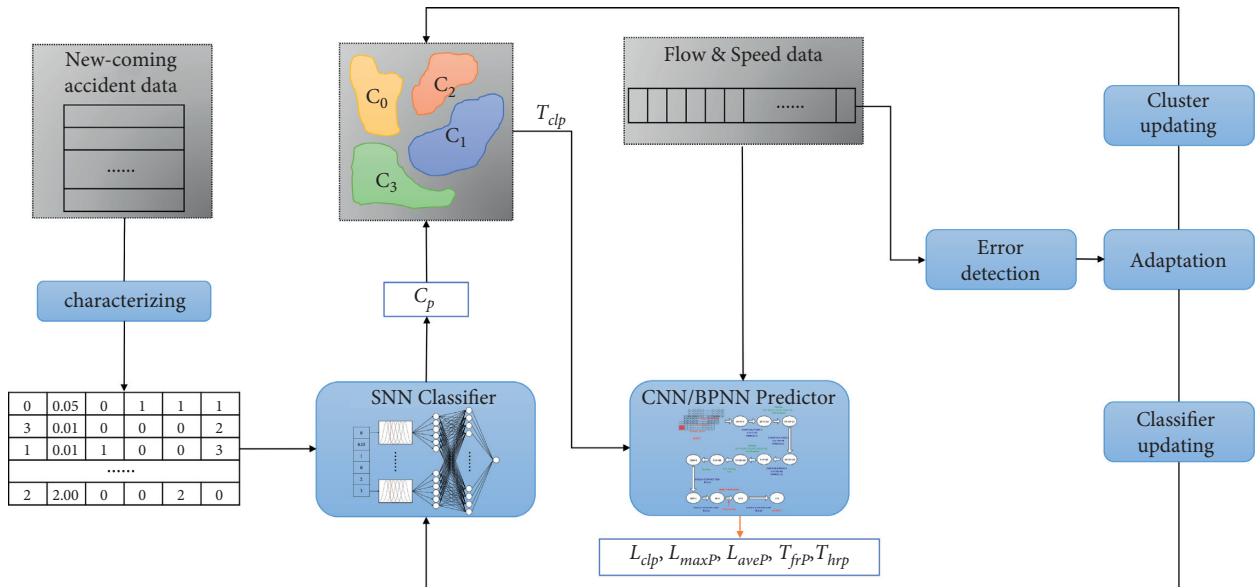


FIGURE 3: 3-stage model for accident post-impact prediction-predicting process.

TABLE 1: Notations.

Notation	Meaning	Units
$C_A$	The actual cluster that the clean-up time of an accident belongs to	—
$C_P$	The predicted cluster that the clean-up time of an accident belongs to	—
$T_{clA}$	The actual clean-up time	Minute
$T_{clP}$	The predicted clean-up time	Minute
$L_{clA}$	The actual accumulative queue length in the clean-up phase	Mile
$L_{clP}$	The predicted accumulative queue length in the clean-up phase	Mile
$L_{maxA}$	The actual maximum accumulative queue length	Mile
$L_{maxP}$	The predicted maximum accumulative queue length	Mile
$L_{aveA}$	The actual average accumulative queue length	Mile
$L_{aveP}$	The predicted average accumulative queue length	Mile
$T_{frA}$	The actual full-recovery time	Minute
$T_{frP}$	The predicted full-recovery time	Minute
$T_{hrA}$	The actual half-recovery time	Minute
$T_{hrP}$	The predicted half-recovery time	Minute

- (1) Input
- (2)  $\{T_c\}$  := dataset of clean-up time over which the procedure is performed
- (3) Initialization
- (4)  $r$  := the radius of a region
- (5)  $d$  := the minimum distance between clusters
- (6)  $F(i, C^j)$  := probability that point  $i$  belongs to Cluster  $C^j$
- (7) Repeat
- (8) Randomly select an unvisited point from the dataset as the starting centre point  $o^c$
- (9) Repeat
- (10) Find all the points that appear in the circle  $\odot(o^c, r)$  and group into cluster  $C^{o^c}$
- (11) For point  $i$  in  $C^{o^c}$ ,  $F(i, C^{o^c}) \leftarrow F(i, C^{o^c}) + 1$
- (12) Calculate the shift vector of  $o^c$  as  $M_c = 1/K \sum_{x_i \in C} (x_i - o^c)$
- (13) Update  $o^c$  towards the shift vector as  $o^c \leftarrow o^c + M_c$
- (14) **Until**  $M_c$  reaches minimum threshold
- (15) If  $\exists C^{o^j}$ , distance( $C^{o^c}, C^{o^j}$ )  $< d$ , then merge  $C^{o^c}$  and  $C^{o^j}$
- (16) Else, label  $C^{o^c}$  as a new cluster
- (17) **Until** all points are visited
- (18) For point  $i$  in  $\{T_c\}$ ,  $i$  belongs to cluster  $C$  with maximum probability  $F(i, C) = \max_{C^j} F(i, C^j)$
- (19) Output
- (20)  $\{C^{o^j}\}$  := all clusters
- (21)  $\{C_{T_i}\}$  := clusters that each point in  $\{T_c\}$  belongs to

ALGORITHM 1: Mean-shift clustering of clean-up time.

[5, 25, 29]), spatial-temporal attributes (e.g. lane location [25, 27, 29], road type [24, 25], and time [5, 24, 25, 27–29]), and weather information (e.g. humidity [24, 25, 28], wind [24, 25], and visibility [24, 25]). In this model, considering the dataset used, 6 attributes are selected as the input in this stage:

- (i) Block lanes: a number between 1 and 6 represents the number of lanes blocked by the accident. The value is determined according to TMC event code [30].
- (ii) Block distance: a value represents the length of road blocked by the accident in miles.
- (iii) Day or night: a binary digit, where 0 represents daytime while 1 represents night.
- (iv) Workday or not: a binary digit, where 0 represents workday while 1 represents holiday or weekend.
- (v) Peak hour or not: a number between 0 and 2, where 0 represents morning peak (8 a.m.–10 a.m.), 1 represents evening peak (4 p.m.–6 p.m.), and 2 represents the normal period [31].
- (vi) Weather condition: a number between 0 and 4, which represents sunny/rain/snow/thunderstorm/fog, respectively, when accident happens.

More detailed information of data processing can be found in Section 4.

**3.1.3. Classification Using SNNs.** A classifier is trained to match the accident attributes to the cluster it belongs to. Due to the characteristics of being sensitive to the spatial-temporal features, SNNs are applied in building the classifier. It is the first time of applying SNNs in traffic accident prediction domain.

After being first proposed in 1997 by Maass [6], SNNs have attracted researchers to explore different research directions. The main technique directions of SNNs can be concluded in four main categories: encoding and decoding of spiritual information, neuron models and network simulation strategy, similarity measurement of spiking sequence, and synaptic weight learning rules [32].

The classification problem in this work concerns with the supervised learning in SNNs, which refers to finding a suitable synaptic weight matrix for a series of given input spike trains and target spike trains, so that the output of the neurons is as close as the corresponding target spike trains, that is, the error evaluation function of the two reaches minimum [33].

(1) *Leaky Integrate-and-Fire Neural Model.* Neural computing relies on the construction of the spiking neuron model. In order to do numerical analysis, an efficient mathematical neural model should be built based on the geometry and electrical properties of neuron. Leaky Integrate-and-Fire neural (LIF) model [34], as the single compartment model with fixed threshold which has been widely used in neural computing, can directly describe the relationship between membrane potential  $V$  and input electric current  $I$ . When  $V$  is between the resting potential  $V_{rest}$  and the threshold  $V_{thre}$ , it is given by the following equation:

$$\tau_m \frac{dV}{dt} = -(V - V_{rest}) + R_m I, \quad (1)$$

where  $\tau_m$  is a time constant, and  $I$  is the sum of current released by presynaptic neurons. Once  $V$  crosses  $V_{thre}$  from below, a spike is generated and  $V$  is reset to  $V_{rest}$ . The LIF model, with the advantages of using simple structure to realize accurate simulation of SNNs by event-driven simulation strategy, is chosen as the basic neuron model in our work.

(2) *Gaussian Tuning Curve Coding*. In the biological neural system, spikes are transmitted between neurons with different combinations. The corresponding rules and mechanisms of conversion between the stimulation signal and the spike trains are called neural coding [35]. When neuron feels external stimuli, encoding module encodes the stimulation signal to specific spike trains and responses; in turn, decoding module can estimate the stimulation signal from spike trains.

Researchers put forward various coding methods, which can be divided into two main categories: frequency-based coding [36] and time-based coding [37]. Evidence confirms that time-based coding is the more effective, since the time structure of the spike trains carries stimulus signals on a millisecond or even smaller scale, not just the average firing frequency [38]. With the introduction of more complex systems, single-neuron coding cannot effectively present the huge amount of information. Population coding [39] is thus proposed to solve complex information coding problem, where each neuron has a unique spike response distribution for a given stimulus, and the responses of the neuron population are combined to represent the overall information.

The population coding method used in this model is Gaussian tuning curve coding [40], as shown in Figure 4. A neuron covers a certain range of analog quantity in the form of Gaussian function, while the height of the corresponding Gaussian function to a certain value of the analog quantity determines the time for the neuron to emit spike. In this work, a population containing six neurons is used to encode the six accident attributes mentioned above. Figure 4 shows an example of how “block distance” of 0.18 miles is encoded into spike trains {0.79, 0.04, 0.52, 0.96, 1.00, 1.00} (ms).

(3) *Multi-ReSuMe Training Algorithm*. The supervised learning algorithm in SNNs can be divided into three categories based on the learning rules of synaptic weight: gradient-descent based method, synaptic plasticity-based method, and convolution-based method [41]. Gradient-descent based methods such as SpikeProp [40] learn from the error back propagation algorithm of traditional ANNs, where the state variables of the neuron model must have analytical expressions and lack biological realism. From a biological perspective, a training algorithm should update synaptic weights based on the temporal correlation of presynaptic and postsynaptic spikes, in keeping with the spike timing-dependent plasticity (STDP) theory [42]. In this work, a synaptic plasticity-based method called the multilayer remote supervised method (Multi-ReSuMe) [43] is used to train the classifier. Multi-ReSuMe extends the single-layer impulse neural network ReSuMe algorithm [41] to the multilayer network and overcomes some limitations as it can be applied to neurons firing multiple spikes, and it can in principle be applied to any linear neuron model.

The main contribution of Multi-ReSuMe is the combination of the STDP and anti-STDP mechanism in one process, where synaptic weight update between hidden layer  $H$  and output layer  $O$  satisfies the following equation:

$$\begin{aligned} \frac{d}{dt}w_{oh}(t) = & \frac{1}{n_h}S_h(t) \left[ \int_0^\infty a^{\text{pre}} [S_o^d(t) - S_o^a(t)] ds \right] \\ & + \frac{1}{n_h} [S_o^d(t) - S_o^a(t)] \left[ a + \int_0^\infty a^{\text{post}}(s)S_h(t-s)ds \right]. \end{aligned} \quad (2)$$

For any excitatory synaptic connection from hidden neuron  $h$  to output neuron  $o$ , a synaptic strength  $w_{oh}$  is potentiated whenever a target spike  $S_o^d(t)$  is observed and depressed whenever the trained neuron fires  $S_o^a(t)$ . Here,  $s$  denotes a delay between the presynaptic and postsynaptic firing times, where  $s = t_o^f - t_h^f$ . The role of the noncorrelative factor  $a$  in equation (2) is to adjust the average strength of the synaptic inputs so as to impose on a neuron a desired level of activity. The kernels  $a^{\text{pre}}$  and  $a^{\text{post}}$  are STDP-related terms that define the shape of a learning window  $W(s)$  [44],

$$W(s) = \begin{cases} a^{\text{pre}}(-s) = -A_- \cdot \exp\left(\frac{s}{\tau_-}\right), & s \leq 0, \\ a^{\text{post}}(s) = +A_+ \cdot \exp\left(\frac{-s}{\tau_+}\right), & s > 0, \end{cases} \quad (3)$$

where parameters  $A_+$ ,  $A_- > 0$  are the amplitudes and  $\tau_+$ ,  $\tau_- > 0$  are the time constants of the learning process.  $a^{\text{pre}}(-s)$  gives the weight change if the presynaptic spike (the spike of the hidden neuron occurs) comes after the postsynaptic spike (the spike of the output and target neurons), while  $a^{\text{post}}(s)$  gives the weight change if the presynaptic spike comes before the postsynaptic spike [43].

Similarly, synaptic weight update between input layer  $I$  and hidden layer  $H$  satisfies the following equation:

$$\begin{aligned} \frac{d}{dt}w_{hi}(t) = & \frac{1}{n_i n_h} S_i(t) \sum_{o \in O} \left[ \int_0^\infty a^{\text{pre}} [S_o^d(t) - S_o^a(t)] \right] w_{oh} \\ & + \frac{1}{n_i n_h} \sum_{o \in O} [S_o^d(t) - S_o^a(t)] \\ & \left[ a + \int_0^\infty a^{\text{post}}(s)S_i(t-s)ds \right] w_{oh}. \end{aligned} \quad (4)$$

As Figure 5 shows, the network has 36 input neurons corresponding to six accident attributes, and an output neuron representing the cluster that the input sample belongs to. The hidden layer contains 80 neurons which are fully connected with other layers. The output neuron's spike train contains a single spike, where the timing differs from each other, as given in Table 2. The process of building and training a Multi-ReSuMe-based classifier is given in Algorithm 2. The parameters are initialized as follows:  $A_+ = 1.2$ ,  $A_- = 0.5$ ,  $\tau_+ = \tau_- = 0.5$ , and  $a = 0.05$  [45].

**3.2. Stage 2: Accident Spatial-Temporal Post-Impact Prediction.** After the clean-up phase of an accident, the traffic flow will begin to recover from congestion to normal condition. Recovery phase is an important process which is often

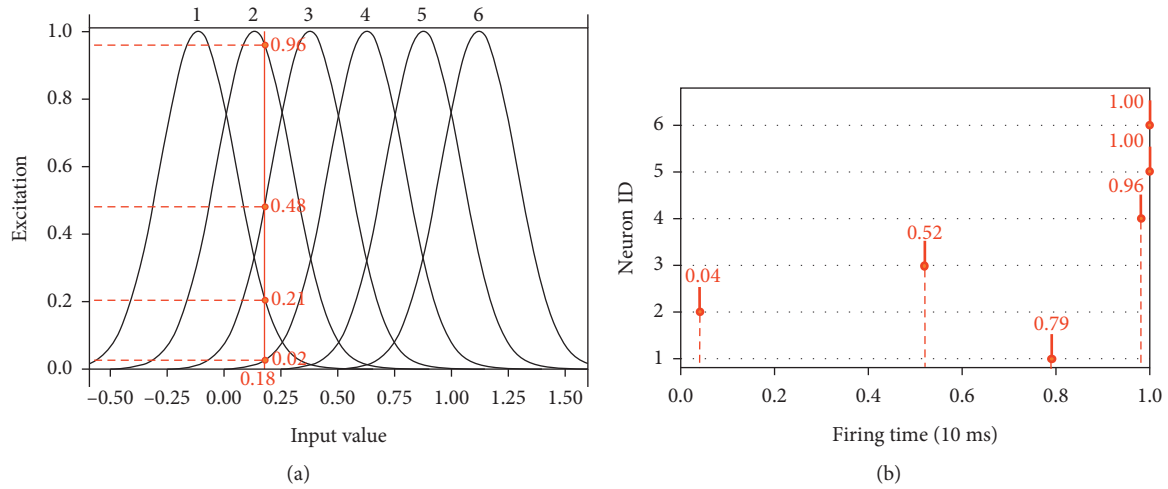


FIGURE 4: Accident attribute coding with Gaussian tuning curve. (a) The corresponding Gaussian curve of six neurons. (b) Firing time of six neurons.

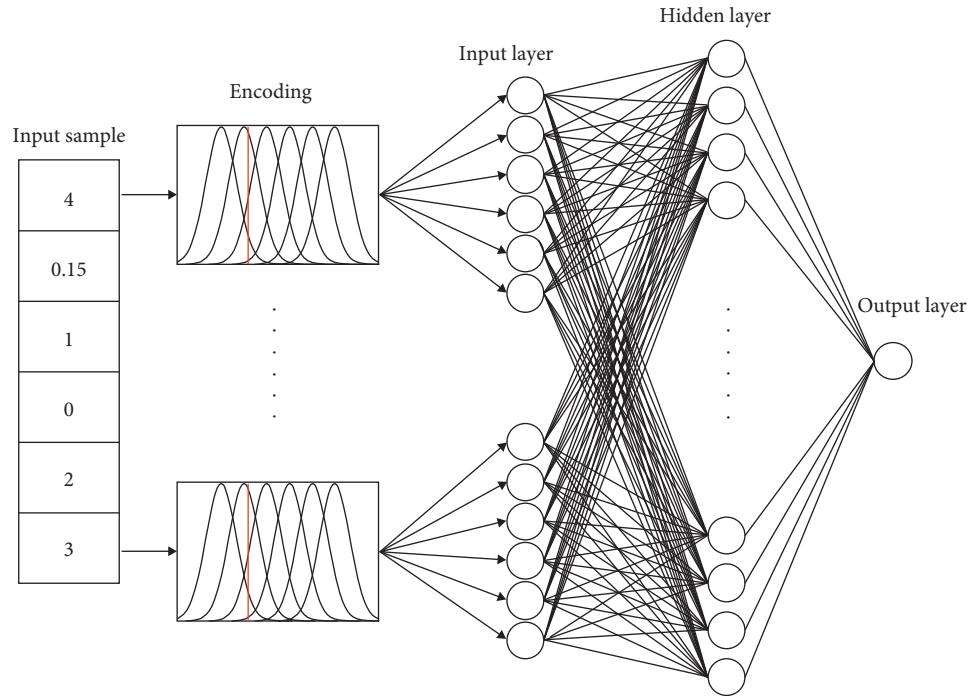


FIGURE 5: A Multi-ReSuMe-based SNNs classifier.

TABLE 2: The target spike-time of output neuron corresponding to each cluster.

Cluster no.	Output spike-time (ms)
0	0
1	1
2	2
3	3
⋮	⋮
$n$	$n$



```

(1) Definition
(2)  $\{A\} :=$  Historic accident dataset
(3)  $A_i :=$  accident data in  $\{A\}$  where  $A_i = (a_1, a_2, a_3, a_4, a_5, a_6, c)$ 
(4)  $e_{\text{target}} :=$  the minimum value of network error to reach when learning is considered converged
(5)  $W :=$  synaptic weight between all neurons
(6)  $M :=$  maximum number of iterations
(7) Initialization
(8)  $W \leftarrow$  Initial synaptic weight with random value  $\theta \sim U(0.2, 0.8)$ 
(9) For iteration = 1,  $M$  do
(10)   For  $A_i \in \{A\}$  do
(11)     Encode  $A_i = (a_1, a_2, a_3, a_4, a_5, a_6, c)$  to a series of spike trains  $A_i = (S_i^1, S_i^2, S_i^3, S_i^4, S_i^5, S_i^6, S_o^d)$ 
(12)     Set membrane potential of all neurons to the resting potential (set to 0)
(13)     Pass input spike trains to the network and find actual output spike  $S_o^a$ 
(14)     Calculate network error  $e_{A_i} = 1/2(S_o^d - S_o^a)^2$ 
(15)     Compute weight modifications  $\Delta W$  for all layers according to equation (2) and equation (4)
(16)     Update synaptic weight  $W \leftarrow W + \Delta W$ 
(17)   End for
(18)   Calculate summed network error  $e = \sum e_{A_i}$ 
(19) While  $e \geq e_{\text{target}}$ 

```

ALGORITHM 2: Multi-ReSuMe-based classifier.

neglected by the travelers and traffic management department, since people always regard clearance as the end of an accident, while the evacuation process afterwards plays a vital role in reasonable path planning. Thus, in this phase, several important values in the recovery phase will be predicted.

First, the predicted clean-up time, matrixes of traffic volume, and speed at the occurrence of the accident are used to estimate the queue length at the end of the clean-up phase (or the beginning of recovery phase). A CNNs-based predictor is used in order to better capture the spatial features of the congested road. Second, the analysis of the recovery phase is based on a backpropagation neural networks (BPNNs), where the estimated queue length together with the historical average traffic condition are used as input. The final output of this stage can include full-recovery time, half-recovery time, maximum accumulative queue length, and average accumulative queue length. Whenever a new accident record comes in, stage 2 will start when it receives the output clean-up time of stage 1.

**3.2.1. Queue Length Estimation at the End of Clean-Up Phase Using CNNs.** During the clean-up phase, one or several lanes of the highway will be affected which leads to the reduction of capacity. According to the specific number of block lanes and block distance, there will be varying degrees of vehicle accumulation. If the block area is relatively small or the traffic volume is low, the traffic flow itself will absorb disturbance and cause little congestion. However, if high traffic volume meets the large block area, the accumulated queue will be very long, and it will be necessary for the travelers to change the travel plan.

Convolutional neural network (CNN) is a class of deep, feed-forward artificial neural network, which has been successfully employed to analyze visual imagery. Since the variables that influence the queue length in our model include

traffic flow, speed, and the geometry of the block area, they can all be better presented by pictures. Thus, CNNs are the best choice since they behave well in extracting spatial features from images so as to fully understand the spatial characteristics around the block area. A CNN consists of an input and an output layer, multiple convolutional layers, and optional hidden layers such as pooling layers, fully connected layers, and normalization layers. Figure 6 shows the demonstration of how these layers can be combined to build a CNN according to the requirement [1]. Convolutional layers apply a convolution operation to the input and pass the result to the next layer, so as to achieve feature extraction [46].

*(1) Traffic Environment Preprocessing.* The definitions and representations of the traffic environment are very important, as the accuracy of prediction is dependent on the effectiveness of the information received about the environment. To take advantage of the CNNs, the environment is processed as three pictures in the model: a picture of geometry showing the block area, a picture of the vehicle speed, and a picture of the traffic flow. A representation of this process is shown in Figure 7, where an accident blocks two lanes with 0.23 miles on a four-lane highway. Noticing that the red arrow shows the location of sensors (spread unevenly), and the grey dotted lines in Figure 7(a) represents how the picture is divided into grids that is long in 0.1 miles and wide in lane width. Figure 7(b) shows the trafficability of each grid, where the block area is set to 0, and other normal area is set to 1. The corresponding average flow (vehicles/hour) and average speeds (miles/hour) are shown in Figures 7(c) and 7(d). The data in each grid are always accessed from the nearest sensor and later normalized. The total length is set as 5 miles, which means  $4 \times 50$  grids. These settings ensure that the environment is accurately and sufficiently represented and also not too complex for the CNNs to understand.

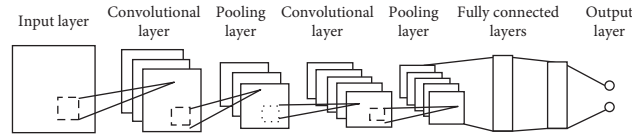


FIGURE 6: Architecture of CNNs.

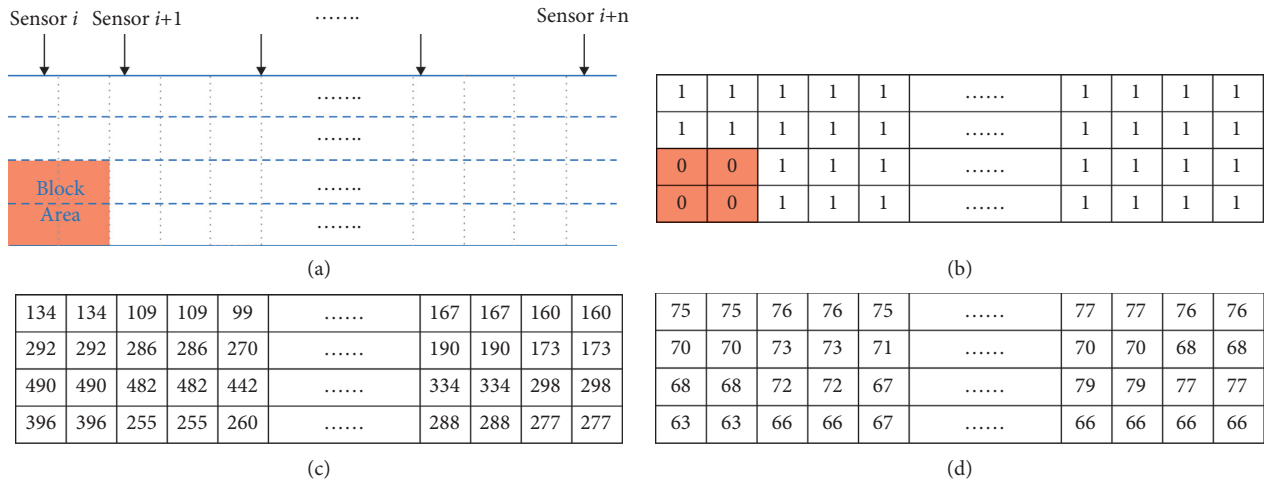


FIGURE 7: Example of processing traffic environment pictures. (a) Four-lane highway map. (b) Geometry picture. (c) Traffic flow (vehicles/hour). (d) Average speed (miles/hour).

(2) *Network Structure.* For highways with different number of lanes in the dataset, we separately trained three CNNs, namely, the 4-lane model, 5-lane model, and 6-lane model. Each CNN receives three traffic environment pictures mentioned in 3.2.1.1 together with the predicted clean-up time as input, and after processing through six layers (three convolutional layers and three fully connected layers), it outputs a number representing the predicted queue length  $L_{clp}$ . The structure of the 5-lane CNN model, including the processing method in each layer and the picture size before and after each layer, is shown in Figure 8. The network structure of the 4-lane model and 6-lane model is not presented here. Learning rate is set as 0.005, and the training ends when mean square error reaches 0.001.

3.2.2. *Prediction of Spatial-Temporal Impact in Recovery Phase Using BPNNs.* At the beginning of the recovery phase, the block area will be removed and the traffic will gradually return to normal condition. Thus, the environment in this phase can be described as a simple congestion propagation model in the straight closed road. Shock wave theory [47], proposed in 1955, regards traffic flow as a continuous fluid with a linear flow-density relationship and has been applied widely in the congestion propagation analysis. However, it only concerns constant flow and density and neglects stochastic effects. This work fully considers the uneven distribution of traffic in the upstream and downstream and uses a BPNNs-based predictor to capture the nonlinear relationship between model input and output that estimate spatial-temporal impact variables such as full-recovery time,

half-recovery time, maximum accumulative queue length, and average accumulative queue length.

Similar to Section 3.2.1, we also train three BPNNs models corresponding to 4-lane case, 5-lane case, and 6-lane case. Each BPNN receives inputs including the speed and flow of each lane at the beginning of recovery phase together with the predicted queue length, and after processing through a hidden layer, it outputs a number representing the spatial-temporal impact variable. Note that a separate BPNN is trained for each variable. It is worth mentioning that in the predicting process shown in Figure 3, the instant traffic data are represented by the historical average value to improve accuracy. For example, if the predicted recovery phase of an accident starts at 2021/6/20, 8:00 (Sunday), the speed and flow data will be accessed through calculating the average value at the four most recent “Sunday 8:00.” Figure 9 shows the structure of the 4-lane BPNN model, which has 17 input neurons and 9 hidden neurons. The 5-lane model has 21 input neurons and 11 hidden neurons, while the 6-lane model has 25 and 13. Learning rate is set as 0.005, and the training ends when mean square error reaches 0.001.

The process of acquiring actual recovery time and accumulative queue length is detailed in Section 4.

3.3. *Stage 3: Online Updating.* After stage 1 and stage 2, the whole lifetime of an accident has already been predicted. It is expected that most accidents will be classified accurately, and the predicted spatial-temporal variables will match the actual one within a fault tolerance. However, some of the predictions can have long time span using the information at

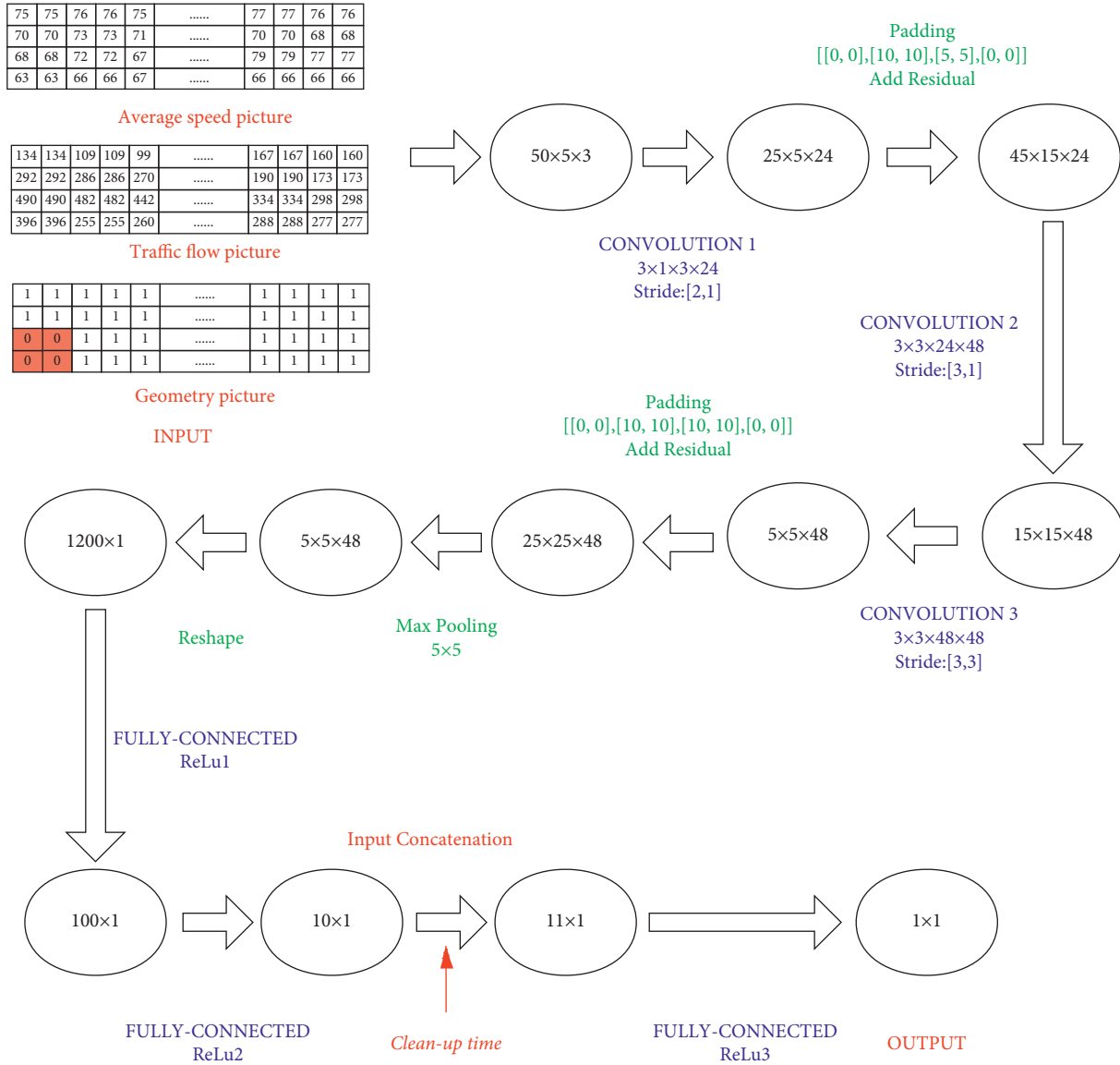


FIGURE 8: Structure of the CNNs in the 5-lane model.

the first beginning. Long-term prediction has always been a difficulty and errors must exist. The most influencing part would be the clustering and classification accuracy, since the error will continue to propagate and cause more serious errors in the following stages. That is the reason for proposing stage 3: detecting the error during the development of the accident and making timely adaptation and revision. Furthermore, the revised values can also help to modify and update the models in stage 1, namely, mean-shift-based clustering and SNNs-based classification.

**3.3.1. Error Detection and Adaptation.** The first step in the online updating stage is to detect if the predicted clean-up time is excessively deviating from the true value. As Figure 10 shows, the detection process is carried out every 5 minutes starting from the occurrence of an accident. The cause of errors can be divided into two situations:

- (i)  $T_{clA} > T_{clP}$ : the accident has not been totally cleaned in the predicted clean-up time; then, the stage 2 is repeated every 5 minutes until the accident is cleaned.
- (ii)  $T_{clA} < T_{clP}$ : the accident is cleaned earlier than predicted; then, the current time is set as the actual clean-up time and is used to repredict in stage 2.

This process makes up for the shortcomings of long-term prediction and helps the model to predict more accurately.

**3.3.2. Clustering and Classification Updating.** Another benefit of online stage is the modification of the clustering and prediction model when new knowledge is found along with the incoming accidents. As Figure 11 shows, the process starts with finding the actual closest cluster  $C_A$  for  $T_{clA}$ . Then, it can be divided into the following two situations:

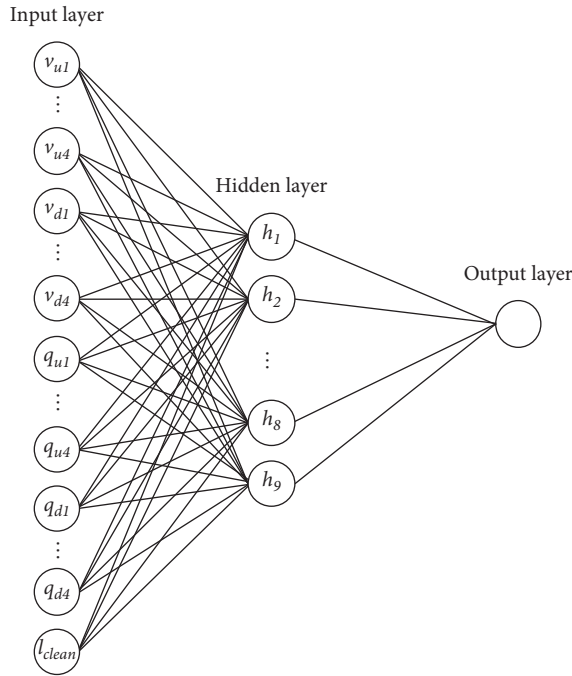


FIGURE 9: Structure of the BPNNs in the 4-lane model.

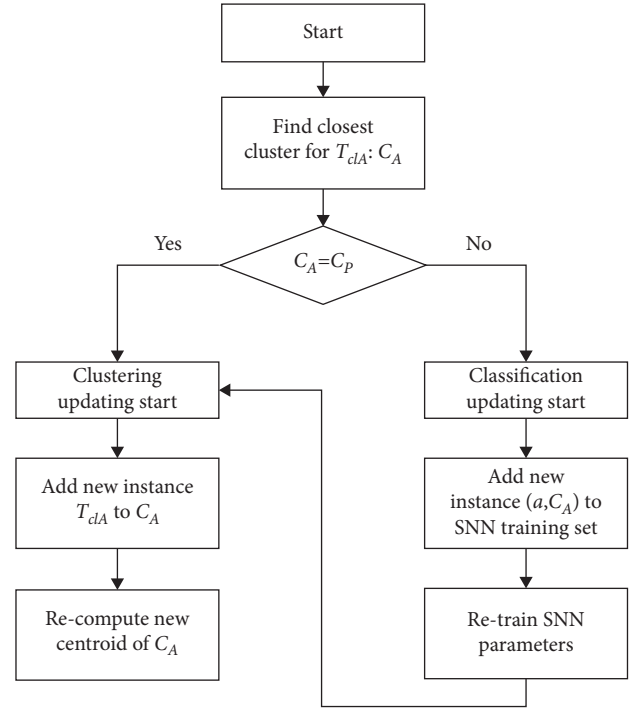


FIGURE 11: Clustering and classification updating process.

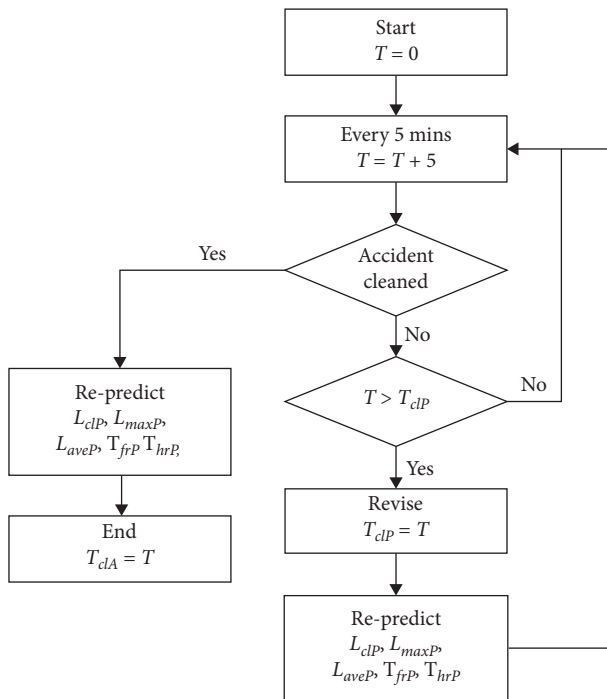


FIGURE 10: Error detection and adaptation process.

- (i)  $C_A = C_P$ : the accident is classified into the proper cluster and the error is caused by the difference from the centroid value. In this case, only a clustering updating is carried out to add a new instance and recompute new centroid of  $C_A$ ;

- (ii)  $C_A \neq C_P$ : the accident is not classified into the proper cluster and the error is caused by the misclassification of the SNNs classifier. Then, another classification updating is needed where the accident attributes and its actual cluster  $C_A$  is added to the SNNs training set and later used to retrain the SNNs model.

This process helps to describe accidents that are not available in the training dataset and further improve the overall accuracy of the model.

#### 4. Dataset

This section describes the process of constructing a California statewide spatial-temporal traffic accident dataset (CASTA), using two datasets named US accident [48] and California Department of Transportation (Caltrans) Performance Measurement System (PeMS) [49].

US accident is a countrywide traffic accident dataset, which covers 49 states of the United States. The data are continuously being collected from February 2016, containing about 3.5 million accident records currently.

PeMS provides access to real-time and historical performance data in many useful formats and presentation styles, including a consolidated database of traffic data collected by Caltrans placed on state highways throughout California, as well as other Caltrans and partner agency datasets.

To figure out the spatial-temporal impact of each accident, we use the process shown in Figure 12 to match and

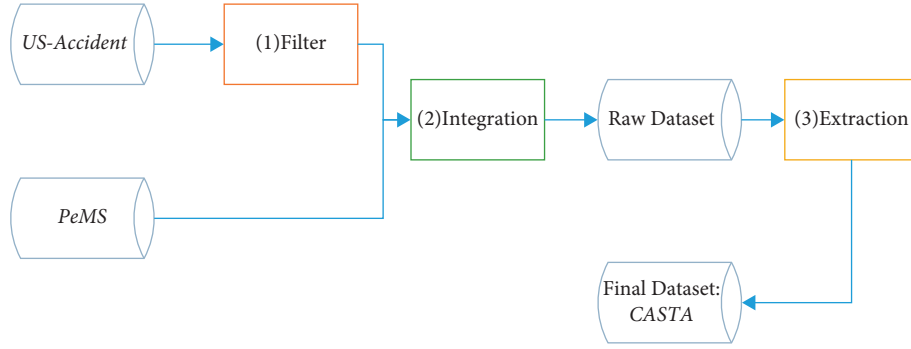


FIGURE 12: Process of creating spatial-temporal accident dataset CASTA.

merge the two datasets—US accident and PeMS. It includes three important steps: data filter, integration, and extraction.

**4.1. Data Filter.** 12,000 traffic accident records in California between February 2016 and February 2020 are filtered from the US accident datasets. The locations of the samples are all state motorways in California, and their distributions are given in Table 3. For better representing the spatial characteristics of accidents, the statistical distribution of clean-up time (sum of detection, verification, response, and clearance time) of the selected records is shown in Figure 13, from 16 min to 8 h 40 min. It can be figured out that the clean-up time is not distributed evenly in the range, where more than 96% of the records fall in (25, 120], and (25, 30] is the interval with the highest amount. However, the records with clean-up time less than 25 min or higher than 240 min are not representative and not enough to train the model. Thus, we only include 11940 pieces of records with clean-up time fallen in (25, 240] minutes in the experiment.

**4.2. Data Integration.** The data integration process is about matching the traffic flow and speed data with traffic accident records. First, we extract the latitude and longitude coordinates and time of each accident record, denoted as  $\langle \text{lat}, \text{lng}, t \rangle$ . Second, we find the nearest sensor to that coordination in PeMS and extract the flow and speed data after  $t_e$ . Third, concatenate the PeMS and US accident to form the new raw dataset. Note that disabled detectors have already been filtered out.

**4.3. Data Extraction.** By comparing with the upstream flow/speed after  $t_e$ , figure out the accident full-recovery time  $T_{\text{frA}}$ , half-recovery time  $T_{\text{hrA}}$ , maximum accumulative queue length  $L_{\text{maxA}}$ , and average accumulative queue length  $L_{\text{aveA}}$  according to Algorithm 3. The final dataset CASTA consists of 20 attributes that fall in 6 main types as given in Table 4.

## 5. Experiments and Results

Methods proposed in Section 3 have been tested with the CASTA dataset, and the results are discussed in this section. Among the 11940 pieces of records, 9940 of them are randomly chosen as the training set, while the rest 2000 are

used as the test set. Since no similar previous work that predicts the whole accident process has been proposed yet, the comparison is carried out in each stage, respectively. First, the outcomes of the first two stages without the online adaptation are presented. Second, the online stage is added to test how the outcomes have been improved. Experiments have been implemented on the server with four CPU (Intel Xeon (R) CPU E5-2650 v2 @ 2.60 GHz), 32 GB RAM.

### 5.1. Offline Prediction Analysis

**5.1.1. Clean-Up Time Analysis.** The efficiency of the initial clean-up time prediction is essential to the following stages, since the error will be further propagated and lead to larger mistakes. According to the cluster results, the clean-up time has been divided into 13 groups, where group 13 is a noise cluster. Table 5 presents the interval of each group and its corresponding centroid value.

When comparing the performance of classifiers, the classification accuracy which means the proportion that predicted class is equal to the real class is usually calculated. However, in this case, since the specific value of clean-up time is more important than the class, the classification accuracy is replaced by the MAPE and RMSE in the following equations, respectively, by comparing the predicted clean-up time  $T_{\text{clP}}$  and actual clean-up time  $T_{\text{clA}}$ .

$$\text{MAPE} = \sum_{i=1}^n \left| \frac{T_{\text{clA}} - T_{\text{clP}}}{T_{\text{clA}}} \right| \times \frac{100}{n}, \quad (5)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (T_{\text{clA}} - T_{\text{clP}})^2}. \quad (6)$$

In order to verify the suitability of our proposed model (mean-shift clustering and SNNs classifier), several methods have been selected as comparison benchmarks: nonlinear regression [50], KNN [51], ANNs, and mean-shift clustering and ANNs classifier. Noticing that the first three benchmarks directly build the relationship between accident attributes and clean-up time without the clustering process, while the fourth replaces the SNNs classifier by ANNs. The experiment results are given in Table 6. Our proposed method outperforms all other benchmarks in accuracy and

TABLE 3: Locations of traffic accident records.

County	Motorway											Total
	I-5	I-805	I-8	I-15	I-405	I-605	I-10	I-880	I-80	I-280	I-680	
San Diego	694	484	391	258	0	0	0	0	0	0	0	1827
Orange	523	0	0	0	410	25	0	0	0	0	0	958
Riverside	0	0	0	284	0	0	178	0	0	0	0	462
Los Angeles	708	0	0	0	1320	690	1741	0	0	0	0	4459
San Bernardino	0	0	0	259	0	0	863	0	0	0	0	1122
Santa Clara	0	0	0	0	0	0	0	307	0	317	216	840
Alameda	0	0	0	0	0	0	0	818	242	0	379	1439
Solano	0	0	0	0	0	0	0	0	605	0	57	662
San Francisco	0	0	0	0	0	0	0	0	137	94	0	231
Total	1925	484	391	801	1730	715	2782	1125	984	411	652	12000

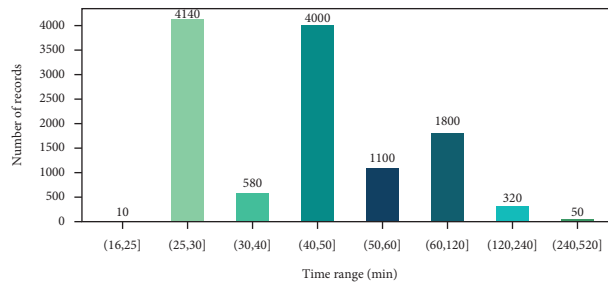


FIGURE 13: Distribution of accident clean-up time.

TABLE 4: Attributes in CASTA.

Attribute category	Attributes	Value and description
Accident attribute	Id	Unique identifier of the accident record 202 serious accidents 203 multivehicle accidents
	TMC event code	245 two-lane blocked 246 three-lane blocked .....
	Block lanes	The number of lanes affected by the accident
	Block distance	The length of the road extent blocked by the accident
Spatial attribute	Latitude	Latitude in GPS coordinate of the accident point
	Longitude	Longitude in GPS coordinate of the accident point
	County	County in address field
	Motorway ID	Motorway name in address field
Temporal attribute	Sensor ID	Unique identifier of the nearest sensor to the accident point
	Day or night	0 day 1 night
	Workday or not	0 workday 1 holiday or weekend
	Peak hour or not	0 morning peak hour 1 evening peak hour 2 normal period
Weather attribute		0 sunny 1 rain 2 snow 3 thunderstorm 4 fog
	Traffic flow characteristic attribute	Flow data series Speed data series
		A matrix of flowrate of upstream sensors after accident happens A matrix of speed of upstream sensors after accident happens

TABLE 4: Continued.

Attribute category	Attributes	Value and description
Impact attribute	Accident clean-up time	Sum of detection, verification, response, and clearance time
	Full-recovery time	Time duration from clearance to the moment when no vehicle is accumulated
	Half-recovery time	Time duration from clearance to the moment when queue length is half of maximum accumulative queue length
	Maximum accumulative queue length	—
	Average accumulative queue length	—

```

(1) Input
(2)  $s_e$  := the nearest sensor in PeMS to the location of traffic accident  $e$ 
(3)  $S$  := upstream sensors set of  $s_e$  denoted by  $S = \{s_1, s_2, \dots, s_n\}$ 
(4)  $t_e$  := the closest time point in PeMS to the start time of traffic accident  $e$ 
(5)  $T$  := time point set in PeMS after  $t_e$  denoted by  $T = \{t_1, t_2, \dots, t_n\}$ , where  $t_i - t_{i-1} = 5$  min
(6)  $V_i$  := speed of sensor  $s_j$  in  $T$  denoted by  $V_i = \{v_{s_i, t_1}, v_{s_i, t_2}, \dots, v_{s_i, t_n}\}$ 
(7)  $T_c$  := sum of detection, verification, response, and clearance time of traffic accident  $e$ 
(8) Initialization
(9)  $F_{\text{dissipate}} \leftarrow$  false means if the traffic flow is in dissipating process
(10)  $v_{\text{threshold}} \leftarrow$  10 mile/h means the vehicle is low-speed driving or waiting
(11) For  $t_i$  in  $T$  do
(12)   For  $s_j$  in  $S$  do
(13)     If  $v_{s_j, t_i} > v_{\text{threshold}}$ 
(14)       Then If  $L_{\text{maxA}} < \text{distance}(s_e, s_j)$ 
(15)         Then update  $L_{\text{maxA}} = \text{distance}(s_e, s_j)$ 
(16)       Else
(17)          $F_{\text{dissipate}} = \text{true}$ 
(18)         If  $\text{distance}(s_e, s_j) < L_{\text{maxA}}/2$ 
(19)           Then  $T_{\text{hrA}} = t_i - t_e$ 
(20)         End if
(21)       End if
(22)     Update  $L_{\text{aveA}} \leftarrow ((j-1) * L_{\text{aveA}} + \text{distance}(s_e, s_j))/j$ 
(23)     Break
(24)   End for
(25)   If  $j = 1$  and  $F_{\text{dissipate}} = \text{true}$ 
(26)     Then update  $T_{\text{total}} = t_i - t_e$ 
(27)   Break
(28) End for
(29)  $T_{\text{frA}} = T_{\text{total}} - T_c$ 
(30) Output
(31)  $T_{\text{frA}}$  := full-recovery time of traffic accident  $e$ 
(32)  $T_{\text{hrA}}$  := half-recovery time of traffic accident  $e$ 
(33)  $L_{\text{maxA}}$  := maximum accumulative queue length
(34)  $L_{\text{aveA}}$  := average accumulative queue length

```

ALGORITHM 3: Data extraction from dataset.

TABLE 5: Clustering results.

Cluster no.	Clean-up time interval	Centroid value
1	[25, 30)	28.73
2	[30, 34)	30.89
3	[34, 40)	36.65
4	[40, 48)	43.02
5	[48, 53)	49.22
6	[53, 58)	56.87
7	[58, 62)	60.11

TABLE 5: Continued.

Cluster no.	Clean-up time interval	Centroid value
8	[62, 76)	68.35
9	[76, 91)	84.25
10	[91, 108)	98.37
11	[108, 116)	113.03
12	[116, 125)	118.08
13 (noise cluster)	[125, 240)	130.90

TABLE 6: Prediction performance comparison of clean-up time.

Method	MAPE (%)	RMSE	Training time (h)
Nonlinear regression	29.33	30.97	—
KNN	30.02	28.05	—
ANNs	28.58	23.64	5.50
Mean-shift clustering and ANNs classifier	14.30	11.19	4.88
Mean-shift clustering and SNNs classifier (ours)	12.07	8.13	3.20

training time. Mean-shift clustering and ANNs classifier ranks first in the comparing methods, but still needs more running time and higher MAPE and RMSE. The other three benchmarks achieves unsatisfied prediction accuracy due to the lacks of combination of clustering and classification. The results prove that clustering the records into classes and using the centroid value to present the predicted value is better than direct prediction model when the data are dispersive distributed, as well as the suitability of applying SNNs in capturing spatial and temporal features.

For better analysis of the model performance, the accuracy in each interval is calculated, respectively, as given in Table 7. It can be figured out that MAPE and RMSE are relatively low in intervals with abundant records, e.g., [25, 30) and [40, 50). In contrast, the prediction difficulty of intervals that clustered as noise (e.g., [120, 240)) is relatively high and the accuracy is not satisfied due to insufficient training. Fortunately, the probability of such severe accidents is low enough and will not cause frequent impact on traffic. Furthermore, with the increase of historic accident data, those outliers can be removed from the noise cluster and the accuracy can be further improved.

**5.1.2. Clean-Up Phase Queue Length Analysis.** In this section, the performance of CNNs based queue length prediction is analyzed, using the output of the previous stage. Comparison benchmarks include the shockwave theory [18], nonlinear regression, and ANNs. Shockwave theory assumes that the traffic flow and speed is constant at the accident site and upstream. It takes traffic flow as a continuous fluid with a flow ( $q$ )-density ( $k$ ) relationship and uses equation (7) to calculate the shockwave when the state of a traffic stream changes from  $(q_i, k_i)$  to  $(q_j, k_j)$  [18]. The length of block area is neglected in the shockwave theory.

$$W_{ij} = \frac{q_j - q_i}{k_j - k_i} \quad (7)$$

Due to inability of capturing picture input, other two benchmarks set up relationships between constant flow, speed, block length, and queue length.

In Table 8, MAPE and RMSE of the proposed model and other benchmarks are listed based on number of lanes. 4-lane cases, 5-lane cases, and 6-lane cases account for 19%, 27%, and 54%, respectively. The proposed CNNs based model still outperforms other benchmarks in MAPE and RMSE in all three cases. Shockwave theory is unable to make efficient prediction due to unrealistic assumption, while nonlinear regression and ANNs make similar performance but are still not satisfied due to constant traffic information. Considered together with the distribution of queue length shown in Figure 14, the 4-lane model and 5-lane model have simple environment that leads to high prediction accuracy. With the increase of lanes, the traffic itself can absorb the disturbance better; thus, the accumulated queue length becomes shorter. That explains the high MAPE but low RMSE in 6-lane case.

The MAPE of 6-lane cases grouped according to the length of the queue is also calculated and shown in Figure 15. It can be concluded that the MAPE of queue length in [0.4, 2.2) miles is below 20%, while the model performs relatively poor when the queue length is below 2 miles and above 2.2 miles. The cause of error in low queue length is the quality of the PeMS dataset, where the distance between sensors is not small enough to give accurate queue length. For example, if the actual queue length of an accident is 0.06 miles while the distance between the closet sensors is 0.1 miles, then the queue length will be regarded as 0. That kind of errors will greatly affect the model performance, especially in low queue length cases. Meanwhile, when the accumulative queue length is greater than 2.2 miles, the model needs larger input to capture the environment, and thus, 5-mile is not enough to make accurate prediction.

Though the proposed model has already achieved better performance than other methods, the accuracy would still improve if trained by a more complete dataset. The merging



TABLE 7: MAPE and RMSE in clean-up time intervals.

Clean-up time interval (min)	MAPE (%)	RMSE
[25, 30)	3.62	2.64
[30, 40)	9.06	4.32
[40, 50)	5.30	3.08
[50, 60)	10.87	6.37
[60, 120)	9.01	18.26
[120, 240)	43.90	50.20

TABLE 8: Prediction performance comparison of queue length.

Method	4-lane (19%)		5-lane (27%)		6-lane (54%)		Average	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
Shockwave theory	30.03	0.17	28.98	0.15	36.79	0.15	33.40	0.15
Nonlinear regression	22.30	0.13	21.07	0.13	27.09	0.12	24.55	0.12
ANNs	22.09	0.12	23.50	0.10	26.85	0.08	25.04	0.09
CNNs (ours)	16.81	0.07	15.06	0.06	20.77	0.05	18.48	0.06

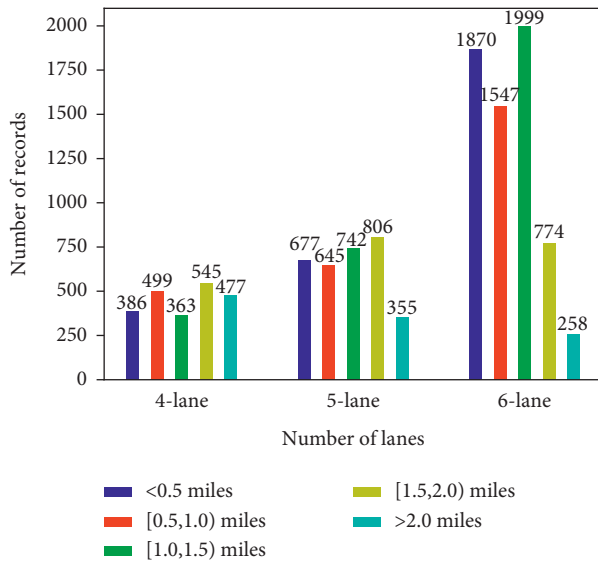


FIGURE 14: Distribution of clean-up phase queue length.

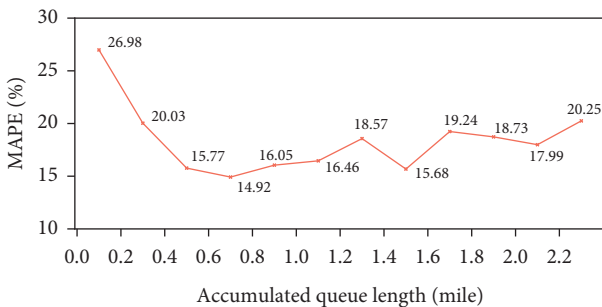


FIGURE 15: MAPE and RMSE in queue length intervals of 6-lane cases. The cross-point represents the average value in that interval.

process of US accident and PeMS can expand the error in some situation. For example, the coordination bias errors in US accident will result in an offset when matching the closet sensor in PeMS.

**5.1.3. Recovery Phase Spatial-Temporal Impact Analysis.** In the last stage of the offline prediction, the model will give out the final predicted spatial-temporal impact during the recovery process. Comparison benchmarks include shockwave theory, nonlinear regression, and BPNNs (base). The proposed model will use the output data of the CNNs model in the previous stage, while shockwave theory and nonlinear regression will still be based on the result using the same method, which means the performance analysis already includes the error of previous stages. Comparing to our proposed model that uses historical average flow/speed as input, the BPNN (base) only uses the value at the beginning of the recovery phase. According to the results given in Table 9, the error of the shockwave theory and nonlinear regression continues to expand. The MAPE of maximum accumulative queue length even reaches 40.47%, which losses its reference value. The BPNN (base) stably ranks the second under all assessment criteria, just behind our proposed method. Due to the comprehensive representation of traffic environment and the use of historical average value, the proposed model achieves satisfying performance, where the MAPE of four impact variables are all below 20%.

Although there is no existing work considering the whole accident process that can be used as comparing benchmark, the above separate comparison in each offline stage still illustrates the effectiveness of our proposed method. The offline part can be regarded as the combination of three different models developed for the characteristics in each stage, and the advantages are further expanded after the combination.

**5.2. Online Prediction Analysis.** Although the accuracy of offline prediction is already much higher than other comparing benchmarks, it is still not satisfying enough for providing precise information and guidance for governments and travelers. Especially for accidents with long time span, as shown in Figure 16, the error can be as large as 40%.

TABLE 9: Prediction performance comparison of spatial-temporal impact.

Method	4-lane (19%)		5-lane (27%)		6-lane (54%)		Average	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
Full-recovery time (minute)								
Shockwave theory	30.88	33.96	30.05	32.77	27.28	29.88	28.71	31.44
Nonlinear regression	29.07	31.54	27.96	31.08	26.03	26.03	27.13	28.44
BPNNs (base)	24.55	26.05	25.37	25.73	24.07	22.36	24.51	23.97
BPNNs (ours)	19.30	18.03	20.05	17.99	17.93	15.02	18.76	16.39
Half-recovery time (min)								
Shockwave theory	36.52	22.58	32.95	20.30	33.84	20.17	34.11	20.66
Nonlinear regression	27.33	18.07	27.74	17.94	26.32	17.72	26.90	17.85
BPNNs (base)	21.80	16.33	22.93	15.02	21.01	13.22	21.68	14.30
BPNNs (ours)	18.71	11.26	17.97	10.38	18.02	9.96	18.14	10.32
Maximum accumulative queue length (mile)								
Shockwave theory	38.74	0.52	38.55	0.50	42.03	0.47	40.47	0.49
Nonlinear regression	29.75	0.37	28.03	0.33	32.55	0.34	30.80	0.34
BPNNs (base)	21.74	0.31	23.09	0.31	24.11	0.27	23.38	0.29
BPNNs (ours)	18.24	0.24	18.01	0.21	21.39	0.19	19.88	0.20
Average accumulative queue length (mile)								
Shockwave theory	35.52	0.43	36.04	0.36	40.54	0.35	38.37	0.37
Nonlinear regression	26.22	0.26	22.34	0.19	25.62	0.25	24.85	0.23
BPNNs (base)	18.41	0.20	20.03	0.16	22.57	0.19	21.09	0.18
BPNNs (ours)	16.28	0.19	17.26	0.12	20.09	0.13	18.60	0.14

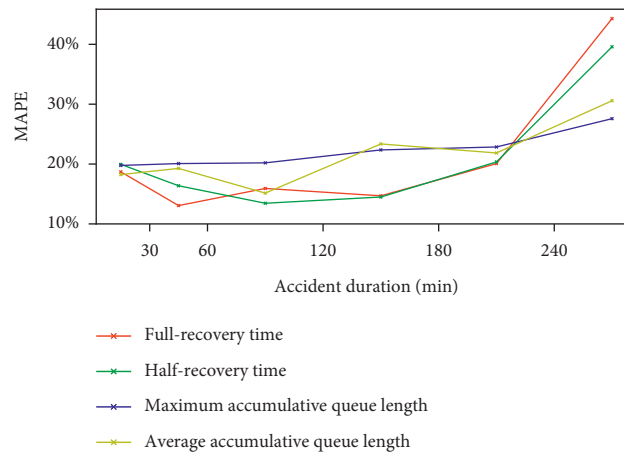


FIGURE 16: MAPE in spatial-temporal impact in accident duration intervals of 6-lane cases. The cross-point represents the average value in that interval.

To compensate for the difficulty in long-term prediction, the online prediction phase is carried out and analyzed to see how much improvement it can bring.

Taking the 6-lane cases as examples, 10 accidents in each clean-up time interval given in Table 10 are chosen as the test set (60 accidents in total) in this stage. According to statistical results given in Table 10, the actual clean-up time of 29 accidents is larger than the predicted ones, while 28 is smaller, and the rest 3 are equal. Apply error detection and adaptation mechanism to these test cases, and an averaged result per interval is given in Table 11. Reduction of MAPE is visible from any interval and impact variable when the final adaptation is done.

To further find out how the improvement is made in each test case, the detailed MAPE difference in full-recovery

time of 60 cases is shown in Figure 17, where values above 0 represent cases with less error, while negative values represent cases with more error. It can be concluded that most of the cases except for one accident (in red) experienced lower MAPE, and the differences range from  $-0.28\%$  to  $32.98\%$ . Along with the increase of accident clean-up time, the improvement becomes more evident. That is reasonable since cases with longer clean-up time always experience more times of adaptation.

In Figure 18, 6 accidents are randomly picked to show how MAPE of full-recovery time changes along with the increase in adaptation times, where the adaptation interval is 5 minutes. The overall trend for all cases is gradually decreasing, where the least of them has experienced one adaptation, while the most has experienced 8 times of

TABLE 10: Clean-up time differences distribution in intervals.

Clean-up time interval	[25, 60)	[60, 90)	[90, 120)	[120, 150)	[150, 180)	[180, 240)	Total
$T_{clA} > T_{clP}$	5	4	6	5	2	7	29
$T_{clA} = T_{clP}$	2	0	1	0	0	0	3
$T_{clA} < T_{clP}$	3	6	3	5	8	3	28

TABLE 11: Average MAPE in clean-up time intervals before and after adaptation.

Clean-up time interval		[25, 60) (%)	[60, 90) (%)	[90, 120) (%)	[120, 150) (%)	[150, 180) (%)	[180, 240) (%)	Average (%)
$T_{fr}$	Offline	18.40	13.36	15.62	14.39	21.07	44.80	21.27
	Online	16.33	10.52	10.07	12.35	17.09	24.37	15.12
$T_{hr}$	Offline	20.95	16.07	12.45	14.89	20.05	38.58	20.50
	Online	17.86	15.01	10.99	11.37	17.29	20.55	15.51
$L_{max}$	Offline	19.37	20.28	20.80	22.36	22.55	27.64	22.17
	Online	17.70	17.25	16.83	18.77	17.05	18.42	17.67
$L_{ave}$	Offline	18.14	19.47	16.14	23.76	21.65	30.48	21.61
	Online	15.17	17.93	14.82	19.90	18.95	21.07	17.97

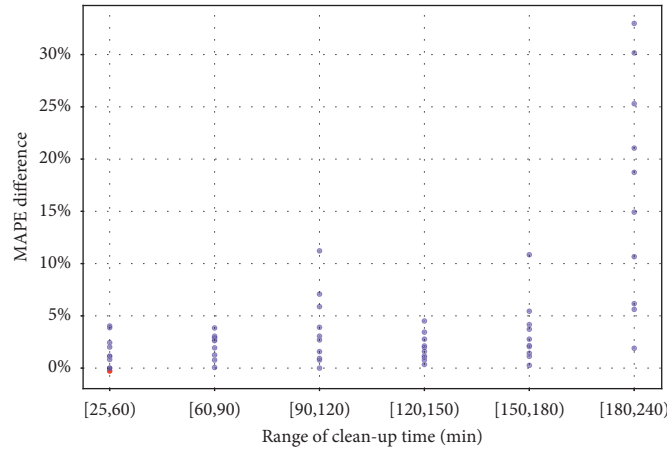


FIGURE 17: MAPE difference distribution of full-recovery time.

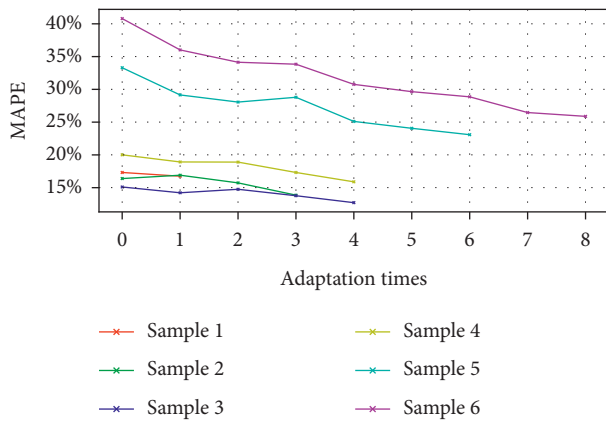


FIGURE 18: MAPE of full-recovery time along with the increase in adaptation times.

adaptation. Even after a few times of adaptation, the model can achieve a much better accuracy, which conveys the effectiveness of this mechanism.

In the second step of online stage, the clustering and prediction model will be modified and updated to improve the future prediction. To reduce workload, this process should be carried out until a batch of data has been gathered. Due to the dataset size limit, there is not enough cases to support to show the obvious change in model parameters and improvements after retraining. Thus, the result analysis of this step is not represented in this work, but can be supplemented when applied to more datasets in the future.

All of the results and discussion above point to the same conclusion: the introduction of online stage can greatly improve the model performance.

## 6. Conclusions

Traffic accidents post-impact prediction (TAPIP) plays an important role in traffic management. In this study, a 3-stage TAPIP model on highway is proposed and tested by the dataset CASTA. Using the advantages of SNNs, the model builds a comprehensive relationship between the spatial-temporal features of accidents and clean-up time. The

combination of clustering and classification improves the prediction accuracy. Due to the capability of CNNs to fully capture the spatial characteristics of traffic environment and the use of historical average data, the model achieves effective prediction of post-impact variables (e.g., fully-recovery time, half-recovery time, maximum accumulative queue length, and average accumulative queue length). A new dataset CASTA that describes California statewide spatial-temporal traffic accidents is constructed and used to test the model. Experiment results prove that our model performs better in accuracy and running time than several existing benchmarks, including nonlinear regression and KNN. The introduction of online adaptation and updating the mechanism further expands the advantages of the model.

This work implements a framework that describes the whole process of a traffic accident and captures the spatial-temporal characteristics in each phase. When applied in real world, whenever an accident takes place, the whole process can be predicted immediately. It proves the reliability and efficiency in applying SNNs in traffic domain, especially in accident prediction research. With regards to future work, we acknowledge that this work is not perfect and that there are still many aspects that can be improved upon. First, the dataset with more detailed accident information and higher quality traffic data can be used to improve the model performance. Second, accidents with clean-up time longer than 240 or shorter than 25 minutes can be further added into the model when enough records have been gathered. Third, clustering and classification updating in online stage needs further experimental demonstration. Last, the model can be extended to city roads rather than limited to highways.

## Data Availability

All data and program files included in this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1709212) and “Research on frontiers of intelligent transport system” funded by China Association for Science and Technology, National Natural Science Foundation of China (U1509205).

## References

- [1] D. Li, J. Wu, M. Xu, Z. Wang, and K. Hu, “Adaptive traffic signal control model on intersections based on deep reinforcement learning,” *Journal of Advanced Transportation*, vol. 2020, Article ID 6505893, 14 pages, 2020.
- [2] T. Beshah and S. Hill, “Mining road traffic accident data to improve safety: role of road-related factors on accident severity in Ethiopia,” in *Proceedings of the 2010 AAAI Spring Symposium Series*, Stanford, CA, USA, March 2010, <https://www.aaai.org/ocs/index.php/SSS/SSS10/paper/view/1173>.
- [3] J. Tang, L. Zheng, C. Han et al., “Statistical and machine-learning methods for clearance time prediction of road incidents: a methodology review,” *Analytic Methods in Accident Research*, vol. 27, Article ID 100123, 2020.
- [4] K. A. Small, “Economics and urban transportation policy in the United States,” *Regional Science and Urban Economics*, vol. 27, no. 6, pp. 671–691, 1997.
- [5] C. Zhan, A. Gan, and M. Hadi, “Prediction of lane clearance time of freeway incidents using the M5P tree algorithm,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1549–1557, 2011.
- [6] W. Maass, “Networks of spiking neurons: the third generation of neural network models,” *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [7] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, “Deep learning in spiking neural networks,” *Neural Networks*, vol. 111, pp. 47–63, 2019.
- [8] F. Su, H. Dong, L. Jia, Y. Qin, and Z. Tian, “Long-term forecasting oriented to urban expressway traffic situation,” *Advances in Mechanical Engineering*, vol. 8, no. 1, 2016.
- [9] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, “Short-term traffic prediction based on dynamic tensor completion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2123–2133, 2016.
- [10] L. Zhihui, C. Qian, Z. Yonghua, and Z. Rui, “Signal cooperative control with traffic supply and demand on a single intersection,” *IEEE Access*, vol. 6, pp. 54407–54416, 2018.
- [11] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, “A data-driven lane-changing model based on deep learning,” *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 41–60, 2019.
- [12] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Frontiers in Computational Neuroscience*, vol. 9, 2015.
- [13] Q. Yu, H. Tang, K. C. Tan, and H. Yu, “A brain-inspired spiking neural network model with temporal encoding and learning,” *Neurocomputing*, vol. 138, pp. 3–13, 2014.
- [14] A. Tavanaei and A. Maida, “Bio-inspired multi-layer spiking neural network extracts discriminative features from speech signals,” in *Neural Information Processing*, pp. 899–908, Springer, Cham, Switzerland, 2017.
- [15] N. Kasabov, V. Feigin, Z.-G. Hou et al., “Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke,” *Neurocomputing*, vol. 134, pp. 269–279, 2014.
- [16] I. Laña, E. Capecci, J. Del Ser, J. L. Lobo, and N. Kasabov, “Road traffic forecasting using NeuCube and dynamic evolving spiking neural networks,” *Intelligent Distributed Computing XII*, Springer, Cham, Switzerland, pp. 192–203, 2018.
- [17] I. Laña, J. L. Lobo, E. Capecci, J. Del Ser, and N. Kasabov, “Adaptive long-term traffic state estimation with evolving spiking neural networks,” *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 126–144, 2019.
- [18] G. Yu, Y. Liu, Y. Wang, and X. Zhu, “Analysis of traffic accident temporal and spatial impact based on shockwave theory,” in *Proceedings of the 12th International Conference of Transportation Professionals (CICTP 2012)*, pp. 557–567, Beijing, China, August 2012.
- [19] S. Chandana Wirasinghe, “Determination of traffic delays from shock-wave analysis,” *Transportation Research*, vol. 12, no. 5, pp. 343–348, 1978.

- [20] Y. He, Z. Liu, and S. Du, "Prediction of recovery time of urban traffic accident based on active flow-split," in *Proceedings of the 2016 International Conference on Smart City and Systems Engineering (ICSCSE)*, pp. 7–9, Hunan, China, November 2016.
- [21] E. C. Sullivan, "New model for predicting freeway incidents and incident delays," *Journal of Transportation Engineering*, vol. 123, no. 4, pp. 267–275, 1997.
- [22] S. Cohen and C. Nouveliere, "Modelling incident duration on an urban expressway," *IFAC Proceedings Volumes*, vol. 30, no. 8, pp. 297–301, 1997.
- [23] B. Zhu, X. Fu, and S. Yang, "Prediction model of space-time impact of traffic accidents based on nonlinear regression and BP neural network," *Highway Engineer*, vol. 43, no. 6, pp. 134–139, 2019.
- [24] Y. Lin and R. Li, "Real-time traffic accidents post-impact prediction: based on crowdsourcing data," *Accident Analysis & Prevention*, vol. 145, Article ID 105696, 2020.
- [25] Y. Lee, C.-H. Wei, and K.-C. Chao, "Evaluating the effects of highway traffic accidents in the development of a vehicle accident queue length estimation model," *International Journal of Intelligent Transportation Systems Research*, vol. 16, no. 1, pp. 26–38, 2018.
- [26] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
- [27] C.-H. Wei and Y. Lee, "Sequential forecast of incident duration using artificial neural network models," *Accident Analysis & Prevention*, vol. 39, no. 5, pp. 944–954, 2007.
- [28] L. Kuang, H. Yan, Y. Zhu, S. Tu, and X. Fan, "Predicting duration of traffic accidents based on cost-sensitive Bayesian network and weighted K-nearest neighbor," *Journal of Intelligent Transportation Systems*, vol. 23, no. 2, pp. 161–174, 2019.
- [29] B. Yu, Y. T. Wang, J. B. Yao, and J. Y. Wang, "A comparison of the performance of ANN and SVM for the prediction of traffic accident duration," *Neural Network World*, vol. 26, no. 3, pp. 271–287, 2016.
- [30] "TMC/Event Code List-OpenStreetMap Wiki," 2021, [https://wiki.openstreetmap.org/wiki/TMC/Event\\_Code\\_List](https://wiki.openstreetmap.org/wiki/TMC/Event_Code_List).
- [31] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMULATION (NGSIM) program data," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1243–1262, 2011.
- [32] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *International Journal of Neural Systems*, vol. 19, no. 4, pp. 295–308, 2009.
- [33] X. Lin, X. Wang, and Z. Hao, "Supervised learning in multilayer spiking neural networks with inner products of spike trains," *Neurocomputing*, vol. 237, pp. 59–70, 2017.
- [34] J. Feng, "Is the integrate-and-fire model good enough?—A review," *Neural Networks*, vol. 14, no. 6–7, pp. 955–975, 2001.
- [35] C. F. Stevens and A. Zador, "Neural coding: the enigma of the brain," *Current Biology*, vol. 5, no. 12, pp. 1370–1371, 1995.
- [36] G. D. Lewen, W. Bialek, and R. R. d. R. v. Steveninck, "Neural coding of naturalistic motion stimuli," *Network: Computation in Neural Systems*, vol. 12, no. 3, pp. 317–329, 2001.
- [37] G. Buzsaki, R. Llinas, W. Singer, A. Berthoz, and Y. Christen, *Temporal Coding in the Brain*, Springer Science & Business Media, Berlin, Germany, 2012.
- [38] D. A. Butts, C. Weng, J. Jin et al., "Temporal precision in the neural code and the timescales of natural vision," *Nature*, vol. 449, no. 7158, pp. 92–95, 2007.
- [39] A. E. Orhan and W. J. Ma, "Neural population coding of multiple stimuli," *Journal of Neuroscience*, vol. 35, no. 9, pp. 3825–3841, 2015.
- [40] S. M. Bohte, J. N. Kok, and H. La Poutré, "Error-back-propagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1, pp. 17–37, 2002.
- [41] A. Kasiński and F. Ponulak, "Comparison of supervised learning methods for spike time coding in spiking neural networks," *International Journal of Applied Mathematics and Computer Science*, vol. 16, pp. 101–113, 2005.
- [42] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: a Hebbian learning rule," *Annual Review of Neuroscience*, vol. 31, no. 1, pp. 25–46, 2008.
- [43] I. Sporea and A. Grüning, "Supervised learning in multilayer spiking neural networks," *Neural Computation*, vol. 25, no. 2, pp. 473–509, 2013.
- [44] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, Cambridge, UK, 2002.
- [45] I. Sporea and A. Grüning, "Classification of distorted patterns by feed-forward spiking neural networks," in *Artificial Neural Networks and Machine Learning-ICANN 2012*, pp. 264–271, Springer, Berlin, Germany, 2012.
- [46] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: integrated recognition, localization and detection using convolutional networks," 2013, <https://arxiv.org/abs/1312.6229>.
- [47] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London-Series A: Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.
- [48] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath, "Accident risk prediction based on heterogeneous sparse data," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 33–42, New York, NY, USA, November 2019.
- [49] Caltrans PeMS," 2021, [https://pems.dot.ca.gov/?controller\\_id=714907&content=detector\\_health&tab=&s\\_time\\_id=1625443200&dnode=VDS&station\\_id=717033&st\\_cd=on&st\\_ch=on&st\\_ff=on&st\\_hv=on&st\\_ml=on&st\\_fr=on&st\\_or=on](https://pems.dot.ca.gov/?controller_id=714907&content=detector_health&tab=&s_time_id=1625443200&dnode=VDS&station_id=717033&st_cd=on&st_ch=on&st_ff=on&st_hv=on&st_ml=on&st_fr=on&st_or=on).
- [50] T. Amemiya, "Chapter 6 non-linear regression models," in *Handbook of Econometrics*, vol. 1, pp. 333–389, Elsevier, Amsterdam, Netherlands, 1983.
- [51] J. H. Friedman, F. Baskett, and L. J. Shustek, "An algorithm for finding nearest neighbors," *IEEE Transactions on Computers*, vol. 24, no. 10, pp. 1000–1006, 1975.

## Research Article

# MDGCN: Multiple Graph Convolutional Network Based on the Differential Calculation for Passenger Flow Forecasting in Urban Rail Transit

Chenxi Wang, Huizhen Zhang , Shuilin Yao, Wenlong Yu, and Ming Ye

College of Computer Science and Technology, Huaqiao University, Xiamen, China

Correspondence should be addressed to Huizhen Zhang; zhanghz@hqu.edu.cn

Received 23 July 2021; Revised 24 October 2021; Accepted 5 November 2021; Published 1 December 2021

Academic Editor: Geqi Qi

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

Passenger flow forecasting plays an important role in urban rail transit (URT) management. However, complex spatial and temporal correlations make this task extremely challenging. Previous work has been done by capturing spatiotemporal correlations of historical data. However, the spatiotemporal relationship between stations not only is limited to geospatial adjacency, but also lacks different perspectives of station correlation analysis. To fully capture the spatiotemporal correlations, we propose a deep learning model based on graph convolutional neural networks called MDGCN. Firstly, we identify the heterogeneity of stations under two spaces by the Multi-graph convolutional layer. Secondly, we designed the Diff-graph convolutional layer to identify the changing trend of heterogeneous features and used the attention mechanism unit with the LSTM unit to achieve adaptive fusion of multiple features and modeling of temporal correlation. We evaluate this model on real datasets. Compared to the best baselines, the root-mean-square errors of MDGCN are improved by 1%–15% for different prediction intervals.

## 1. Introduction

With the expansion of urban traffic congestion, URT has become one of the important solutions to alleviate the congestion problem. As an important research topic in URT, short time passenger flow forecasting can help decision-makers to make timely emergency plans and enhance security forces [1], as well as providing an important reference basis for the optimization of bus line networks.

Accurate, real-time passenger flow forecasting in URT is a challenging task. This is because the accuracy of forecasting can be affected by various aspects, such as the URT network's development, its topology, and spatiotemporal correlation. The short-term prediction models of URT passenger flow range from the initial mathematical-statistical models such as Autoregressive Integrated Moving Average (ARIMA) [2] to traditional machine learning models such as Support Vector Machines (SVM) [3] to various deep learning models. However, mathematical-statistical models cannot mine the nonlinear features of traffic flow data, and traditional machine learning models rely on feature extraction, so

their adaptability to data is not stable. Compared with traditional mathematical-statistical models and machine learning models, deep learning models perform better in the extraction of spatiotemporal features and are mainly classified into three categories: CNN-based, RNN-based, and GCN-based.

In the earliest stage, many scholars treated traffic data as image pixels and used a convolutional neural network (CNN) for local spatiotemporal feature extraction. For example, Luo et al. [4] modeled the traffic flow as a two-dimensional spatiotemporal matrix to describe the images of spatiotemporal relationships of traffic flow and used CNN to predict the traffic speed. Similarly, based on converting the passenger flow data into images, Zhang et al. [5] introduced the residuals into CNN. However, applying CNN to the passenger flow prediction cannot accurately capture the spatiotemporal correlation. Regarding this problem, Zhang et al. [6] proposed a multitask learning prediction model called MTL-TCNN that considered the relevance of multiple regions for the prediction problem, but CNN is only suitable for Euclidean space, and traffic data is typical non-Euclidean

data. Recurrent neural network (RNN) and its variant models have good performance in handling time-series data because of their self-circulation mechanism. For example, Fu et al. [7] used gate recurrent unit (GRU) for traffic flow prediction for the first time. Ma et al. [8] and Yang et al. [9] used long-short term memory (LSTM) to capture the trend of long and short time series. Abduljabbar et al. [10] extended the one-way LSTM into a two-way LSTM; thus, the accuracy was further improved by training the data twice by forward and backward. However, RNN and CNN cannot reasonably extract the correlation of spatial dimensions and are not suitable for traffic flow data processing.

Researchers gradually realize the advantages of GCN in dealing with non-Euclidean data such as traffic flow. Many scholars tend to apply GCN to traffic flow prediction with other models that perform well in capturing temporal correlations. Some studies [11–13] integrated GCN- and RNN-based models to predict. For example, Zhao et al. [14] and Lv et al. [15] combined GCN with GRU; Geng et al. [16] combined GCN with RNN; Zhang et al. [1], Chai et al. [17], and He et al. [18] combined GCN with LSTM to make up for the shortcoming of GCN in capturing temporal dependence poorly. Zhang et al. [19] proposed a model including attention mechanism, GCN, and sequence-to-sequence for multistep prediction. Besides, some studies tried to propose novel models, such as Wu et al. [20], who proposed an adaptive adjacency matrix to change the structure of a fixed graph and used extended convolution to capture temporal correlation. Park et al. [21] used the transformer model [22] and the self-attention mechanism with encoder-decoder architecture. Hao et al. [23] constructed a sequence-to-sequence architecture with attentional mechanisms. Also, a hybrid prediction model [24] based on kernel ridge regression (KRR) and Gaussian process regression (GPR) is proposed to predict the short-term passenger flow. However, these studies only consider the topological structure of the URT network. More correlations between stations should be explored. Effectively exploring correlations among stations could further improve the prediction performance. For example, Geng et al. [16], Chai et al. [17], Wang et al. [25], and Wang et al. [26] reconstructed the relationship graph from different perspectives. However, they ignored the correlation hidden in historical data and the changing trend which could help capture the movement patterns of passenger flow.

The above research indicates that there are still two problems to be solved in the current research on URT passenger flow prediction: the correlation between stations and the extraction of changing trends. To be exact, the relationship between stations is not limited to the adjacency relationship of geographical space; there is a lack of station correlation analysis from different perspectives. And it is necessary to construct an appropriate model to mine the changing trends.

In this paper, we propose MDGCN (Multi-Graph Differential Convolutional Network). It can mine the correlations between stations in heterogeneous space for URT passenger flow prediction. The contributions of this paper are as follows:

- (i) The correlation of boarding and alighting passenger flows between stations that are physically distant from each other is also important for passenger flow prediction. Therefore, to more accurately represent the connectivity between stations, we reconstruct the connectivity of stations by historical passenger flow data and station geographic information.
- (ii) We design a layer of Multi-GCN that can jointly mine the correlation between stations in physical space and nonphysical space to gain the heterogeneous spatial correlation between stations.
- (iii) We introduce the concept of difference and construct the difference feature extraction layer called Diff-GCN to extract the changing trends of heterogeneous spatial features.
- (iv) We conducted experiments on two datasets. The experimental results show that the prediction error of the proposed model is reduced by 1%–15% compared with the optimal baseline.

The arrangement of the paper is shown below. Section 2 introduces the basic problem definition of passenger flow forecasting. Section 3 presents the model framework. Section 4 discusses the experimental results. Conclusions are made in Section 5.

## 2. Preliminary

In this section, we define some key concepts and give the problem definition of the studied content.

*Definition 1 (urban rail transit network).* We define the URT network using the graph as  $G(V, E, A)$ , where the set of stations is used as nodes  $V = \{v_1, v_2, \dots, v_n\}$  of  $G$ , and  $n$  is the number of stations. Figure 1 illustrates sample lines related to the URT network. The connection relationships between stations are used as edges  $E = \{e_k: (v_i, v_j) \in V \times V\}$  of  $G$ . We use the matrix  $A \in R^{n \times n}$  to record the adjacent relationships between stations, and the value  $\delta_{i,j}$  represents the strength of the association between stations. So, we reconstruct the spatial relationships of the URT network from two perspectives.

*Definition 2 (node characteristics).* We take the passenger flow of a station as the characteristics of a network node; i.e., given a station  $V_i$  and a period  $t$ , the passenger flow of the station at that period is denoted as  $X_t^{V_i}$ , and the passenger flow of all stations in the past  $m$  periods is denoted as  $X = \{X_t, X_{t-1}, X_{t-2}, \dots, X_{t-m-1}\}$ ; then the temporal characteristics of the set  $V$  of stations in the past  $m$  periods can be written as  $X \in R^{n \times m}$ .

*Problem. Definition.* This problem can be defined as learning a mapping function to predict the passenger flow data at the next moment given the URT network  $G(V, E, A)$  with the historical passenger flow data  $X$ . The process can be referred to as

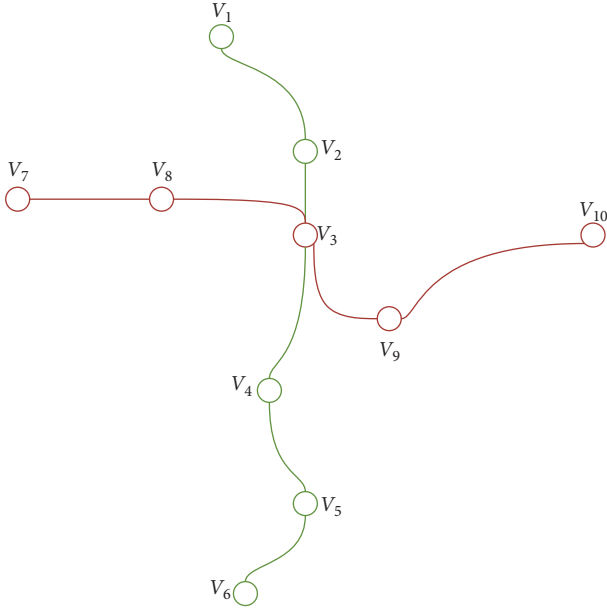


FIGURE 1: A sample of an urban rail transit network.

$$X_{t+1} = f(G, X). \quad (1)$$

### 3. Study Methodology

This section introduces the framework of our model, research methods, and pseudocode.

**3.1. Modeling Framework.** This model (Figure 2) can be divided into four main parts: the reconstruction of the relationship of station, the layer of Multi-GCN, the layer of Diff-GCN, and the layer of output.

- (i) The first part is based on the relationship between the physical space and the nonphysical space of stations and combines the historical passenger flow data as the input of the subsequent model.

- (ii) The second part is the Multi-GCN (Multiple-Graph Convolutional Network) layer, which can jointly model the correlation between stations in both physical space and nonphysical space; i.e., the Multi-GCN layer is constructed to capture the heterogeneous spatial correlation between stations.
- (iii) The third part is a Diff-GCN (Differential Graph Convolutional Network) layer, a differential feature extraction module, to extract the changing trends of heterogeneous spatial features. Besides, the attention mechanism is used to adaptively fuse the features of the Multi-GCN layer and the Diff-GCN layer.
- (iv) The fourth part is the LSTM unit for global temporal correlation extraction and the fully connected network layer for prediction result output.

**3.1.1. Reconstruction of the Correlation of Stations.** The key for a neural network being able to be trained is in the input data. And the input to a GCN consists of two parts: the adjacency matrix and the node features. Whether the adjacency matrix correctly encodes the relationships of the network nodes is related to the performance of the model. Therefore, we interpret the URT network from two perspectives. Stations in close geographical proximity may have similar passenger flow patterns. Given this, we construct the topology of the physical space  $G_D(V, E_D, A_D)$  based on the physical adjacency and distances between station  $i$  and station  $j$ .

Firstly, we obtain the adjacency matrix  $A$  (in (2)) between stations using the physical adjacency relationship of stations. Then, we build the distance matrix  $D$  (in (3)) based on the inverse of the distance between stations. Finally, we reconstruct the topology matrix  $A_D$  using the Hadamard product formula [1] (in (4)).

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & a_{ij} & \vdots \\ a_{n1} & \cdots & \cdots & \cdots & a_{nn} \end{pmatrix}, \quad a_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ is adjacent,} \\ 0, & \text{if } i \text{ and } j \text{ is not adjacent,} \end{cases} \quad (2)$$

$$D = \begin{pmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & d_{ij} & \vdots \\ d_{n1} & d_{n2} & \cdots & \cdots & d_{nn} \end{pmatrix}, \quad d_{ij} = \begin{cases} \frac{1}{\text{distance}_{ij}}, & i \neq j, \\ 0, & i = j, \end{cases} \quad (3)$$



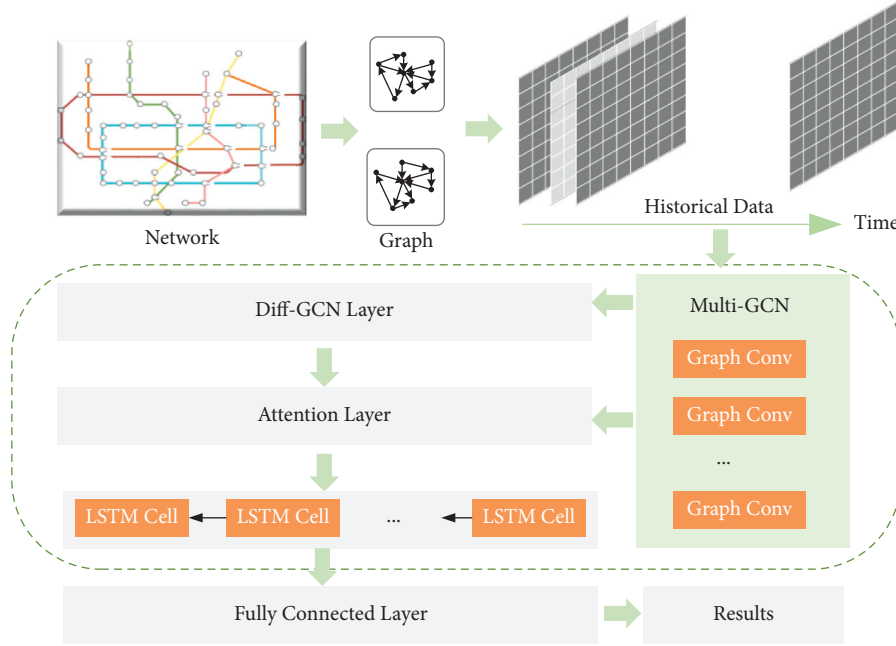


FIGURE 2: The architecture of MDGCN.

$$A_D = A \circ D. \quad (4)$$

However, the flow pattern could make two stations with no spatial adjacency have a certain association, and the strength of the association is determined by the passenger flow. This flow pattern truly reflects the trend of passenger flow movement between stations. Therefore, we describe the passenger flow movement pattern and movement intensity between stations based on the data of historical passenger flow (in (5)) to construct the associated graph  $G_C(V, E_C, A_C)$ . Denote  $p_{ij}$  as the passenger flow from station  $i$  to  $j$ .

$$A_C = \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ p_{31} & p_{32} & p_{33} & \cdots & p_{3n} \\ \vdots & \vdots & \vdots & p_{ij} & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} \end{pmatrix}. \quad (5)$$

**3.1.2. Multi-GCN Network.** Two different network topologies are obtained by spatial relationship reconstruction of stations. We propose the layer of Multi-GCN to extract heterogeneous spatial correlation by GCN. The propagation rule of GCN can be expressed as

$$H_{(l+1)} = \delta\left(\widehat{D}^{-(1/2)} \widehat{A} \widehat{D}^{-(1/2)} H_{(l)} W_{(l)}\right), \quad (6)$$

where  $H_{(l)}$  is the input of  $l$ -layer and its original value is  $X$ .  $\widehat{A} = \text{Normalization}(A + I)$  is the adjacent of a graph with added self-connections,  $\widehat{D} = \sum_j \widehat{A}_{ij}$  is degree matrix,  $\delta$  represents the sigmoid function for a nonlinear model, and  $W_{(l)}$  is a weight matrix of the current layer.

In particular, GCN has a powerful ability to model higher-order neighborhood interactions by stacking multiple layers. As shown in Figure 3, taking the red site as an example, the first layer of GCN can capture the influence of two neighboring stations of the red one. As the number of network layers deepens, the correlation of all stations can be captured completely. The spatial features in the URT network, i.e., the topology  $A_D$  based on the physical space and the station association matrix  $A_C$  in the passenger space, will be extracted by the  $N$ -layer GCN. So we input them into the layer of Multi-GCN to obtain the outputs  $Z_D$  (in (7)) and  $Z_C$  (in (8)) of the multigraph convolutional layers, respectively.

$$\begin{aligned} Z_D &= Z_{D(N)} \\ &= \delta\left[\widehat{D}^{-(1/2)} \widehat{A}_D \widehat{D}^{-(1/2)} Z_{D(N-1)} W\right], \end{aligned} \quad (7)$$

$$\begin{aligned} Z_C &= Z_{C(N)} \\ &= \delta\left[\widehat{D}^{-(1/2)} \widehat{A}_C \widehat{D}^{-(1/2)} Z_{C(N-1)} W\right]. \end{aligned} \quad (8)$$

**3.1.3. Diff-GCN Network.** The difference reflects the variation between two discrete quantities. We introduce the concept of differencing to construct the layer of Diff-GCN to extract the tendency of heterogeneous spatial features. As shown in Figure 4, the passenger flow data  $X_t, X_{t-1}$  based on time  $t$  and  $t-1$ , and the adjacent matrix  $A$  are put into Multi-GCN, respectively, and the respective feature matrices  $Z_{(N,t)}$  and  $Z_{(N,t-1)}$  are obtained after Multi-GCN, the difference is calculated, and the final difference factor  $\Delta Z$  (in (9)) is obtained by using GCN for spatial feature extraction.

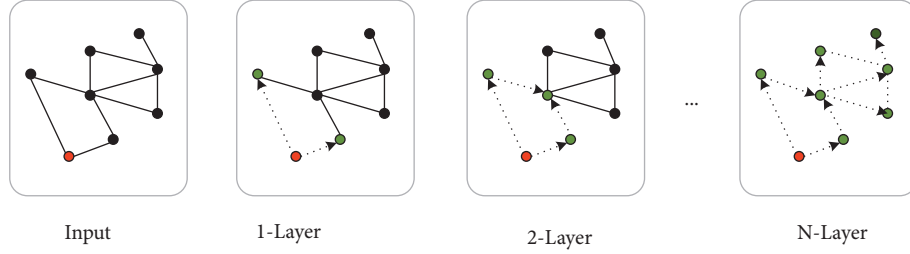


FIGURE 3: The process of capturing the spatial correlation of passenger flows using GCNs (taking red station as an example).

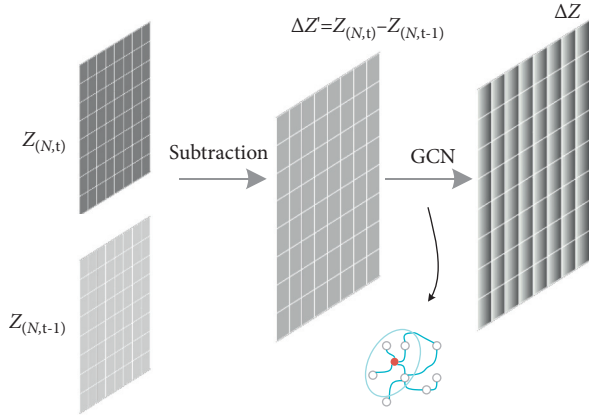


FIGURE 4: The architecture of Diff-GCN.

Here, we obtain two factors  $\Delta Z_D$  (in (10)) and  $\Delta Z_C$  (in (11)) based on two spatial relationships.

$$\Delta Z = \delta \left[ \hat{D}^{-(1/2)} \hat{A} \hat{D}^{-(1/2)} (Z_{(N,t)} - Z_{(N,t-1)}) W \right], \quad (9)$$

$$\begin{aligned} \Delta Z_D &= \Delta Z_{D(N)} \\ &= \delta \left[ \hat{D}^{-(1/2)} \hat{A}_D \hat{D}^{-(1/2)} (Z_{D(N,t)} - Z_{D(N,t-1)}) W \right], \end{aligned} \quad (10)$$

$$\begin{aligned} \Delta Z_C &= \Delta Z_{C(N)} \\ &= \delta \left[ \hat{D}^{-(1/2)} \hat{A}_C \hat{D}^{-(1/2)} (Z_{C(N,t)} - Z_{C(N,t-1)}) W \right]. \end{aligned} \quad (11)$$

**3.1.4. Attention.** For  $Z_D$  and  $Z_C$  obtained from the layer of Multi-GCN and  $\Delta Z_D$  and  $\Delta Z_C$  obtained from the layer of Diff-GCN, we use the attention mechanism to learn the importance  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  of these four features as follows (in (12)). Firstly, we calculate the hidden state vector scoring  $S_1$  of  $Z_D$  using a nonlinear transformation and one shared attention vector  $Q$  (in (13)). Here,  $W$  is the weight matrix and  $b$  is the bias vector. Similarly, the embedding  $Z_C, \Delta Z_D, \Delta Z_C$  do the same operation to get  $S_2, S_3, S_4$ . Secondly, we perform a normalization operation (in (14)) to obtain the coefficient ( $\alpha_i$ ) of the individual embedding vector. Finally, the linear operation is used to combine the four components to obtain  $Z$  (in (15)).

$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \text{Attention}(Z_D, Z_C, \Delta Z_D, \Delta Z_C), \quad (12)$$

$$S_1 = Q^T \cdot \tanh(W \cdot (Z_D)^T + b), \quad (13)$$

$$\begin{aligned} \alpha_i &= \text{softmax}(S_i) \\ &= \exp(S_i) / \sum_{i=1}^4 \exp(S_i), \end{aligned} \quad (14)$$

$$Z = \alpha_1 \cdot Z_D + \alpha_2 \cdot Z_C + \alpha_3 \cdot \Delta Z_D + \alpha_4 \cdot \Delta Z_C. \quad (15)$$

**3.1.5. Prediction.** After obtaining  $Z$  by attention, we use LSTM that is shown in Figure 5 to capture temporal features. LSTM is one of the most common neural network structures in time series forecasting problems. It uses the concept of gates to control the selective passage of information. As can be seen in (16) and Figure 5, it has three gates: Forget Gate, Input Gate, and Output Gate. Forget Gate filters the input information. Input Gate decides what new memory is stored. And Output Gate gets the processed results. The processing of each gate can be seen in (16).

As can be seen in Figure 5, after  $\{Z|Z_1, Z_2, \dots, Z_t\}$  is inputted into LSTM, the result  $\{h|h_1, h_2, \dots, h_t\}$  is gained. Then we obtain the predicted output  $\{X_{t+1}|X_{t+1}^V, X_{t+1}^V, \dots, X_{t+1}^V\}$  via the layer of fully connected with  $n$  neurons which means the number of stations.

$$\begin{cases} \text{Forget Gate: } f_t = \sigma(W_f \cdot [h_{t-1}, z_t] + b_f), \\ \text{Input Gate: } i_t = \sigma(W_i \cdot [h_{t-1}, z_t] + b_i), \\ \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, z_t] + b_c), \\ C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \\ \text{Output Gate: } o_t = \sigma(W_o \cdot [h_{t-1}, z_t] + b_o), \\ h_t = o_t \cdot \tanh(C_t). \end{cases} \quad (16)$$

**3.2. Pseudocode.** MDGCN pseudocode is shown in Algorithm 1. Besides, to evaluate the complexity of this deep learning algorithm, we use two indicators, FLOPs and Params. The time complexity is determined by the number of operations of the model, i.e., FLOPs (0.0126 G), and the space complexity is determined by the number of parameters, i.e., Params (2, 147, 895).

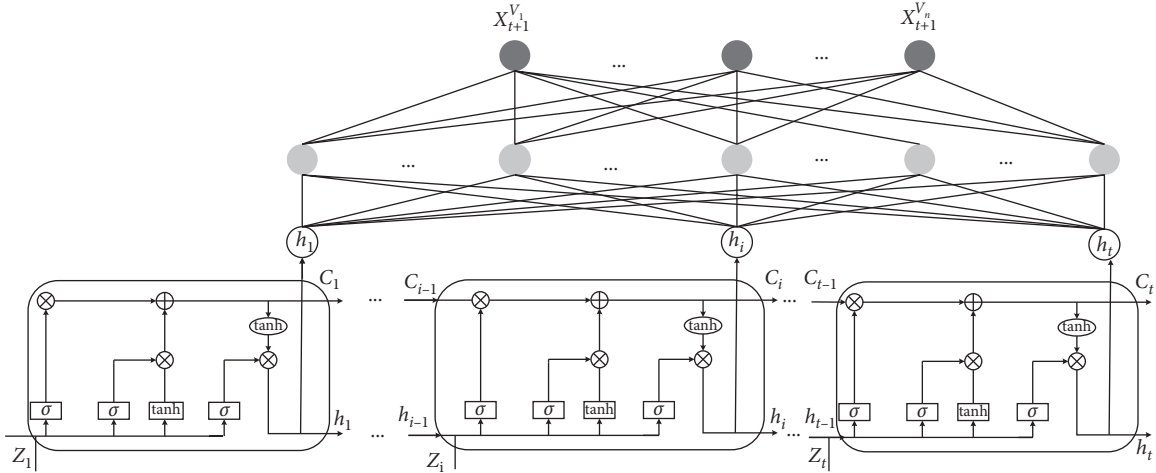


FIGURE 5: The architecture of prediction.

Input: Historical passenger flow  $\{X|X_1, X_2, \dots, X_t\}$  (described in Section 4.2.);  $G_C, G_D$  (described in Section 3.1.1).  
Output: Passenger flow  $X_{t+1}$  of  $t + 1$ .  
Process:  
(1) for  $i = 1, 2, \dots, n$  do  
(2)  $\{Z_{(D,t-1)}, Z_{(D,t)}\} \leftarrow \text{Multi\_GCN}(\{X_1^{V_i}, X_2^{V_i}, \dots, X_t^{V_i}\}, A_D)$  by equation (7)  
(3)  $\{Z_{(C,t-1)}, Z_{(C,t)}\} \leftarrow \text{Multi\_GCN}(\{X_1^{V_i}, X_2^{V_i}, \dots, X_t^{V_i}\}, A_C)$  by equation (8)  
(4)  $\{\Delta Z_D, \Delta Z_C\} \leftarrow \text{Diff\_GCN}(\{(Z_{(D,t-1)}, Z_{(D,t)}), (Z_{(C,t-1)}, Z_{(C,t)})\})$  by equations (10) and (11)  
(5)  $Z \leftarrow \text{Attention}(Z_{(D,t)}, Z_{(C,t)}, \Delta Z_D, \Delta Z_C)$   
(6)  $X_{t+1}^{V_i} \leftarrow \text{FC}(\text{LSTM}(Z))$   
(7) end  
(8)  $\{X_{t+1}|X_{t+1}^{V_1}, X_{t+1}^{V_2}, \dots, X_{t+1}^{V_n}\}$  (Forecasted passenger flow of all stations)

ALGORITHM 1: MDGCN.

## 4. Results and Discussion

**4.1. Experimental Settings.** This model is implemented in Tensorflow and Keras. All experiments were run on a graphics processing unit (GPU) platform with an NVIDIA GeForce GTX 1070 Ti graphics card and 8GB GPU memory. We train our model using Adam optimizer with a learning rate of 0.0001. The training epoch is 150. Two kinds of evaluation metrics are adopted, including root mean squared errors (RMSE), mean absolute errors (MAE), and the evaluation indicators which are shown in

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (17)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |(y_i - \hat{y}_i)|, \quad (18)$$

where  $y_i$  is the true value,  $\hat{y}_i$  is the predicted value, and  $n$  is the number of samples.

**4.2. Dataset.** We used two real-world datasets from China to evaluate our model, i.e., data collected from Xiamen and Shanghai subway Automatic Fare Collection (AFC) System.

SH dataset was collected from 1 April to 31 April 2015. This dataset includes records from 14 lines and 313 stations. The example of this dataset is shown in Table 1. Each record contains the card number (ID), travel date (DATE), card-swiping time (TIME), the name of stations (STATION\_NAME), the number of lines (LINE\_ID), and TPYE. Passenger flow is calculated by formula (19).  $x_{t-b}^a$  represents the number of passenger flows of station  $a$  in period  $t - b$ . Here,  $a$  is determined by "STATION\_NAME" and  $t - b$  is determined by "DATE" and "TIME."

XM dataset was collected from 1 December to 31 December 2019. This dataset includes records from two lines and 52 stations. The example of this dataset is shown in Table 2. We use the same procedure of data preprocessing to gain passenger flow.

$$X = \begin{pmatrix} x_t^1 & x_{t-1}^1 & x_{t-b}^1 & \dots & x_{t-m-1}^1 \\ x_t^2 & x_{t-1}^2 & x_{t-b}^2 & \dots & x_{t-m-1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_t^n & x_{t-1}^n & x_{t-b}^n & \dots & x_{t-m-1}^n \end{pmatrix}. \quad (19)$$

TABLE 1: The sample of Shanghai dataset.

ID	DATE	TIME	STATION_NAME	LINE_ID	FARE	TYPE
6021**128	2015/4/17	9:34:30	Xinzhuang	1	2	Discount
2801**273	2015/4/17	18:23:59	Dalian Road	4	1	Nondiscount
1014**819	2015/4/17	9:16:18	Guanglan Road	2	4	Discount

TABLE 2: The sample of Xiamen dataset.

ID	DATE	TIME	STATION_ID	STATION_NAME	LINE_ID	KHTYPE
17**25	20191224	51090	21	DongFu	2	E-card
82**47	20191210	73725	11	HuoJuYuan	1	E-card
94**60	20191231	68397	26	GuanYin	2	E-card

4.3. *Experiment Results.* In this section, we conducted four main parts of experiments to demonstrate the validity of the proposed model and the reasonableness of each module.

4.3.1. *Evaluation of the Number of GCN Layers.* To determine the optimal value of the number of GCN stacks, we conduct experiments at different stacking values to obtain the corresponding prediction values, and the prediction results are shown in Figure 6. We found that the experiment worked best when  $N = 3$ , so we set the number of stacks to 3 in the experiment.

4.3.2. *The Results in Baselines.* To evaluate the competitive performance of the proposed method (i.e., MDGCN), we compared it with two types of models, covering the most basic mathematical-statistical models and the deep learning models. All baselines are optimized to output the best performance.

- (i) HA [27]: it means historical average model, which uses the average of several previous periods as the prediction. In this paper, we use the average of the last ten periods as the forecast value.
- (ii) ARIMA [28]: we used SPSS software to make predictions so that we could get the best predictions.
- (iii) SVR: we used support vector regression (SVR) to predict passenger flow. Here, the kernel function is rbf, epsilon is 0.2.
- (iv) GBDT: we used gradient boosting decision tree (GBDT) to predict passenger flow. Here,  $n\_estimators$  is 100,  $min\_samples\_split$  is 2, and  $learning\_rate$  is 0.1.
- (v) LSTM: Ma et al. [8] used this method to predict traffic flow. In this paper, we use the same method to predict passenger flow. We used a stacked LSTM which has 128 and 276 neurons for the first and second layers.
- (vi) GCN + LSTM: Chai et al. [17] used this method to predict bike flow. Likewise, we use the same method to predict passenger flow. We used a 2-layer GCN and a layer of LSTM which has 64 neurons.

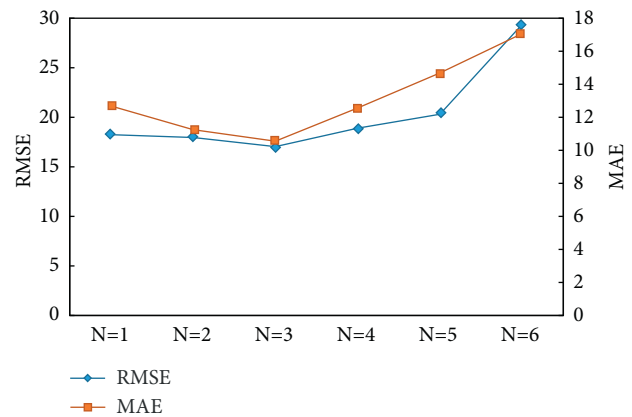


FIGURE 6: Performance with different number of GCN layer.

- (vii) ResLSTM: it refers to the method in [29], which combines ResNet and GCN with LSTM to predict the traffic flow. Here we use the same setting as it is without branch 4.
- (viii) Conv-GCN: it refers to the method in [1], which combines the 3D-Conv with GCN to predict the traffic flow. Also, we use the same setting as is.

The results are shown in Table 3. The following trends can be seen from the results. Firstly, deep learning models (including LSTM, Conv-GCN, ResLSTM, GCN + LSTM, and MDGCN) have a better performance than traditional mathematical-statistical (including HA and ARIMA) and shallow machine learning models (including SVR and GBDT). This result shows that deep learning methods can better capture nonlinear spatiotemporal correlations. Second, the variant of recurrent neural network (including LSTM and ResLSTM) outperforms machine learning methods (including SVR and GBDT). It indicates these time-series models can capture temporal correlation deeply. By comparison, the spatial deep learning-based models (including Conv-GCN, GCN + LSTM, and MDGCN) outperform the variant of LSTM. It demonstrates the effectiveness of spatial correlation in passenger flow prediction. Also, MDGCN mines the relationship of all the stations deeply to have the best performance compared to all baselines. Thirdly, almost all the models behave much better on XM than SH. The biggest reason must be the complexity of the dataset. SH dataset has more lines and stations than

TABLE 3: The performance in different baselines.

Model	SH					
	RMSE			MAE		
	10 min	15 min	30 min	10 min	15 min	30 min
HA	69.812	89.210	95.574	35.954	49.034	54.601
ARIMA	51.044	81.012	156.256	29.057	44.257	82.964
SVR	30.454	36.214	39.754	17.207	19.903	21.169
GBDT	29.935	39.364	32.542	16.561	20.001	17.342
LSTM	28.059	26.541	29.354	15.705	14.672	15.584
Conv-GCN	22.154	21.097	25.645	11.254	14.360	14.391
ResLSTM	26.387	29.548	29.642	14.087	15.265	16.005
GCN + LSTM	20.165	21.364	24.608	13.904	14.367	13.927
MDGCN	17.041	20.962	22.354	10.552	12.786	13.452

Model	XM					
	RMSE			MAE		
	10 min	15 min	30 min	10 min	15 min	30 min
HA	67.241	85.455	95.147	35.264	47.751	53.813
ARIMA	50.695	79.568	175.265	28.564	43.561	95.556
SVR	29.365	35.524	36.524	16.454	18.428	19.658
GBDT	33.635	37.652	50.562	17.058	19.952	26.895
LSTM	28.545	26.421	30.241	14.214	16.754	19.241
Conv-GCN	19.421	20.145	24.245	12.256	13.978	15.842
ResLSTM	24.007	23.557	26.352	14.545	15.812	18.731
GCN + LSTM	18.954	19.145	22.296	11.925	10.770	13.754
MDGCN	16.713	16.927	18.560	9.924	10.247	11.155

TABLE 4: The evaluation results of spatial relationship reconstruction.

Model	SH					
	RMSE			MAE		
	10 min	15 min	30 min	10 min	15 min	30 min
MDGCN-A	35.548	30.674	29.541	19.066	16.078	15.597
MDGCN-AD	25.685	35.547	36.545	14.950	16.635	18.654
MDGCN-AC	20.365	26.654	23.315	11.027	14.028	12.389
MDGCN	17.041	20.962	22.354	10.552	12.786	13.452

Model	XM					
	RMSE			MAE		
	10 min	15 min	30 min	10 min	15 min	30 min
MDGCN-A	27.241	27.624	30.148	18.279	18.698	23.039
MDGCN-AD	22.148	55.512	27.539	14.208	26.384	14.257
MDGCN-AC	18.214	19.012	22.351	10.852	11.109	13.691
MDGCN	16.713	16.927	18.560	9.924	10.247	11.155

XM. Thus, the traffic condition patterns in Shanghai would be much more complex than in Xiamen. Finally, MDGCN has the best performance in these two datasets. Compared with the optimal baseline, it reduced error by 1% to 15%. This shows that it can get better performance by capturing various correlations (including spatial and temporal) and deeply mining the historical passenger flow.

*4.3.3. Evaluation of Spatial Relationship Reconstruction of MDGCN.* In the third experiment, we try to evaluate the effectiveness of spatial relationship reconstruction. Specifically, we compare the performance of the following three variants of MDGCN. Here, AM represents the adjacent matrix that is put into the Multi-GCN layer.

- (i) MDGCN-A: it concludes only the most basic adjacency matrix (i.e.,  $AM = A$ )

- (ii) MDGCN- $A_D$ : it concludes only the physical topology matrix based on the adjacency matrix with distance inverse recoding (i.e.,  $AM = A_D$ )
- (iii) MDGCN- $A_C$ : it concludes only the topological matrix based on the passenger space (i.e.,  $AM = A_C$ )
- (iv) MDGCN: it combines the topological matrix of both physical space and passenger space (i.e.,  $AM = A_D \cup A_C$ )

The experiment results are shown in Table 4. On the one hand, MDGCN has the best overall performance and MDGCN-A has the worst. It demonstrates the importance and effectiveness of recoding. On the other hand, MDGCN- $A_C$  outperforms MDGCN-A and MDGCN- $A_D$ , and it has almost the same level of accuracy as MDGCN. It suggests that the topology of the passenger space more accurately describes the correlations between stations.

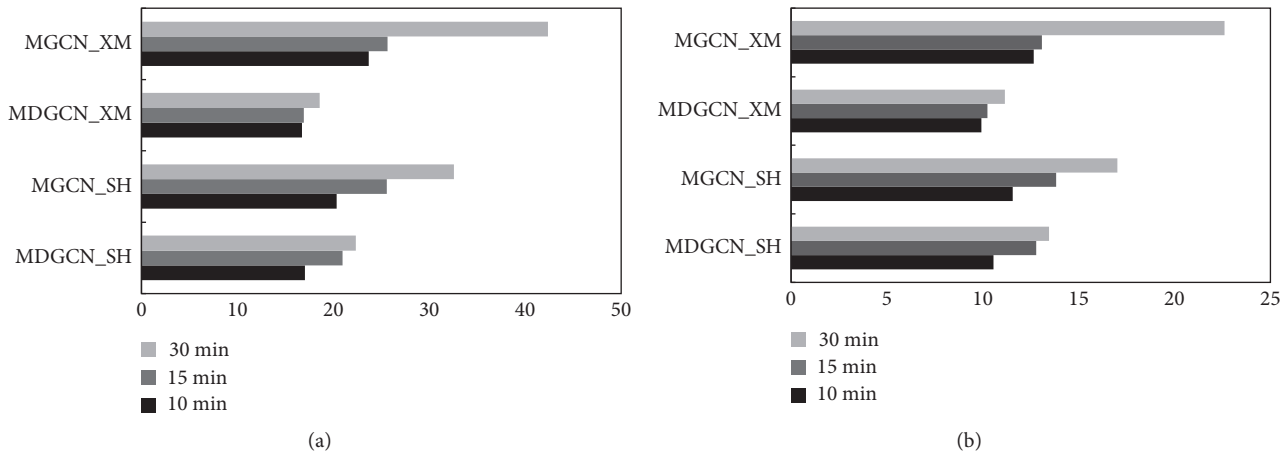


FIGURE 7: The evaluation of Diff-GCN under different time intervals. (a) RMSE. (b) MAE.

**4.3.4. Evaluation of the Diff-GCN Layer of MDGCN.** In the last experiment, we evaluate the performance of the layer of Diff-GCN. We compare the performance of MDGCN with its variant called MGCN which does not conclude the module of Diff-GCN. The experiment results are shown in Figure 7.

It is clear from the results on these two datasets are consistent with each other. The layer of Diff-GCN reduces the prediction error because it can fully extract the trend information among passenger flow.

## 5. Conclusions

In this paper, we propose a deep learning model called MDGCN to predict passenger flow. We construct the Multi-GCN layer to extract heterogeneous correlations under two spatial relationships, and we use the Diff-GCN layer to extract the changing trend of heterogeneous features to fully capture the spatiotemporal correlations. Eventually, it is validated in the actual datasets and obtained better results. However, there are still many points for improvement in the above work, for example, how to construct dynamic connectivity of stations using historical passenger flow data and how to consider the impact of various factors such as weather and unexpected accidents on the prediction results. Therefore, future work will focus on addressing these issues.

## Data Availability

The original data set cannot be provided due to the confidentiality of the data.

## Conflicts of Interest

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

## Acknowledgments

This work was supported by the General Project of Natural Science Foundation of Fujian Province of China (2017J01111), the Science and Technology Plan Guiding

Project of Fujian Province (2020H0016), the National Nature Science Foundation of China (61802133), and the Fundamental Research Funds for the Central Universities (ZQN-910).

## References

- [1] J. Zhang, F. Chen, Y. Guo, and X. Li, "Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit," *IET Intelligent Transport Systems*, vol. 14, no. 10, pp. 1210–1217, 2020.
- [2] L. M. Moreira, J. Gama, M. Ferreira, J. M. Mendes, and L. Damas, "Predicting taxi-passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.
- [3] H. Su and S. Yu, "Hybrid GA based online support vector machine model for short-term traffic flow forecasting," in *Proceedings of the International Workshop on Advanced Parallel Processing Technologies*, pp. 743–752, Guangzhou, China, November 2007.
- [4] J. Luo, J. Li, J. Wang, Z. Jiang, and Y. Chen, "Deep attributes from context-aware regional neural codes," 2015, <https://arxiv.org/abs/1509.02470v1>.
- [5] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Hilton San Francisco Union Square, CA, USA, February 2017.
- [6] K. Zhang, Z. Liu, and L. Zheng, "Short-term prediction of passenger demand in multi-zone level: temporal convolutional neural network with multi-task learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1480–1490, 2020.
- [7] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proceedings of the 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, Wuhan, China, November 2016.
- [8] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.
- [9] D. Yang, K. Chen, M. Yang, and X. Zhao, "Urban rail transit passenger flow forecast based on LSTM with enhanced long-

- term features,” *IET Intelligent Transport Systems*, vol. 13, no. 10, pp. 1475–1482, 2019.
- [10] R. L. Abduljabbar, H. Dia, and P. W. Tsai, “Unidirectional and bidirectional LSTM models for short-term traffic prediction,” *Journal of Advanced Transportation*, vol. 2021, Article ID 5589075, 16 pages, 2021.
- [11] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2019.
- [12] L. Lin, Z. He, and S. Peeta, “Predicting station-level hourly demand in a large-scale bike-sharing network: a graph convolutional neural network approach,” *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 258–276, 2018.
- [13] T. Bogaerts, A. D. Masegosa, J. S. Z. Angarita, E. Onieva, and P. Hellinckx, “A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data,” *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 62–77, 2020.
- [14] L. Zhao, Y. Song, and C. Zhang, “T-GCN: a temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [15] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, “Temporal multi-graph convolutional network for traffic flow prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3337–3348, 2021.
- [16] X. Geng, Y. Li, L. Wang et al., “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 3656–3663, Honolulu, HI, USA, February 2019.
- [17] D. Chai, L. Wang, and Q. Yang, “Bike flow prediction with multi-graph convolutional networks,” in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 397–400, Seattle, WA, USA, November 2018.
- [18] Y. He, Y. Zhao, H. Wang, and K. L. Tsui, “G. C.-L. S. T. M.: A deep spatiotemporal model for passenger flow forecasting of high-speed rail network,” in *Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, Rhodes, Greece, September 2020.
- [19] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, “Multistep speed prediction on traffic networks: a deep learning approach considering spatio-temporal dependencies,” *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 297–322, 2019.
- [20] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph wavenet for deep spatial-temporal graph modelling,” 2019, <https://arxiv.org/abs/1906.00121>.
- [21] C. Park, C. Lee, and H. Bahng, “S. T. G. R. A. T.: A spatio-temporal graph attention network for traffic forecasting,” 2019, <https://arxiv.org/abs/1911.13181>.
- [22] A. Vaswani, N. Shazeer, and N. Parmar, “Attention is all you need,” in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 5998–6008, CA, USA, December 2018.
- [23] S. Hao, D.-H. Lee, and D. Zhao, “Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system,” *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 287–300, 2019.
- [24] Z. Guo, X. Zhao, and Y. Chen, “Short-term passenger flow forecast of urban rail transit based on GPR and KRR,” *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1374–1382, 2019.
- [25] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, “Am-Gcn: Adaptive multi-channel graph convolutional networks,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1243–1253, CA, USA, August 2020.
- [26] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, “Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modelling,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1227–1235, Anchorage, AK, USA, August 2019.
- [27] J. Liu and W. Guan, “A summary of traffic flow forecasting methods,” *Journal of Highway and Transportation Research and Development*, vol. 3, pp. 82–85, 2004.
- [28] M. M. Hamed, H. R. A. Maseid, and Z. M. B. Said, “Short-term prediction of traffic volume in urban arterials,” *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.
- [29] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, “Deep learning architecture for short-term passenger flow forecasting in urban rail transit,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7004–7014, 2021.

## Research Article

# Prediction of Road Network Traffic State Using the NARX Neural Network

Ziwen Song , Feng Sun , Rongji Zhang , Yingcui Du , and Chenchen Li 

*Department of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo 255000, China*

Correspondence should be addressed to Feng Sun; sunfeng@sdut.edu.cn

Received 7 September 2021; Revised 29 October 2021; Accepted 5 November 2021; Published 28 November 2021

Academic Editor: Ling Huang

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

To provide reliable traffic information and more convenient visual feedback to traffic managers and travelers, we proposed a prediction model that combines a neural network and a Macroscopic Fundamental Diagram (MFD) for predicting the traffic state of regional road networks over long periods. The method is broadly divided into the following steps. To obtain the current traffic state of the road network, the traffic state efficiency index formula proposed in this paper is used to derive it, and the MFD of the current state is drawn by using the classification of the design speed and free flow speed of the classified road. Then, based on the collected data from the monitoring stations and the weighting formula of the grade roads, the problem of insufficient measured data is solved. Meanwhile, the prediction performance of NARX, LSTM, and GRU is experimentally compared with traffic prediction, and it is found that NARX NN can predict long-term flow and the prediction performance is slightly better than both LSTM and GRU models. Afterward, the predicted data from the four stations were integrated based on the classified road weighting formula. Finally, according to the traffic state classification interval, the traffic state of the road network for the next day is obtained from the current MFD, the predicted traffic flow, and the corresponding speed. The results indicate that the combination of the NARX NN with the MFD is an effective attempt to predict and describe the long-term traffic state at the macroscopic level.

## 1. Introduction

Transportation is a key link to social development. According to the latest statistics from the Traffic Management Bureau of the Ministry of Public Security, the number of motor vehicles in China reached 384 million until June 2021, and this figure is still growing. It is difficult to meet the increasing demand for comfort in traveling, and the negative effects of congestion on various routes within the road network in urban areas are even more serious. In this context, accurate and reliable traffic information is very important for intelligent transportation systems (ITS), advanced traffic management systems (ATMS), and advanced traveler information systems (ATIS) [1]. In detail, it is important to use traffic information wisely and accurately determine the future traffic situation of the regional road network to support traffic managers and travelers in their management or travel decisions. The key to the problem is

how to accurately predict traffic trips and induce travelers to choose travel routes, thus spreading the pressure of traffic trips within the road network.

To tackle the traffic problems mentioned above, some scholars have proposed forecasting traffic events [2, 3] and traffic demand [4–6] in the study of traffic state prediction. Others have proposed short-term prediction of various traffic state parameters and have designed various prediction models [7, 8]. As a key component in achieving the purpose of traffic congestion mitigation, short-term prediction of traffic state is a combination of traffic information and mathematical algorithms. The future short-term traffic state parameters are predicted by building the corresponding prediction models. Usually, the time interval of short-term prediction does not exceed 15 min [9, 10]. However, due to the randomness and complexity of the traffic state, the traffic flow shows strong nonlinearity as the time interval is shortened [11]. It will add difficulty to the short-term traffic



state prediction. Meanwhile, although the parameter prediction of short-term traffic flow can enable traffic managers to discover abnormal traffic operation states in time and take corresponding control measures, it is difficult to satisfy the demand for long-term traffic trips, and the information obtained by users is limited by the form of information output. In contrast, long-term traffic state prediction can provide the most direct traffic operation information for travelers and has greater application value.

In the studies on the state of regional road networks, many studies only focused on the prediction target on a single section of varying length within the road network. It failed to consider the complex road conditions at the regional level and cannot demonstrate generalisability. And there are fewer studies on the traffic state prediction of road networks, which need to be improved and supplemented. Therefore, the challenges of current research in both traffic state prediction and highway network state studies are as follows:

- (i) With limited data collection devices and the impact on the COVID-19 pandemic, how can traffic parameters and corresponding states of the entire road network be obtained?

Despite the more advanced and diverse means of data collection on the road network at the present stage, some cities are relatively underdeveloped and lack sufficient data collection equipment and services (as in the area selected for this paper) to represent the effects of the data. The key in this situation is the effective mining and processing of the available collection data.

- (ii) Expanding from both temporal and spatial dimensions, how can the traffic state of a road network be represented and predicted by using cross-sectional flows when faced with a larger road network?

From a spatial perspective, basic traffic flow parameters are required as quantitative criteria to characterize traffic, whether it is a section, road, or network. However, the distinction is that the analytical methods utilized, the forms of representation, and their meanings are similar but different at the microscopic scale and the macroscopic scale. From a temporal perspective, it is unclear what time scales are defined for long-term traffic prediction and short-term traffic prediction, and what duration of time can achieve optimum performance.

To address the above challenges, we have proposed a long-term prediction model that combines nonlinear autoregressive with exogenous inputs (NARX) neural network and Macroscopic Fundamental Diagram (MFD) for the state of the road network. First, the traffic state needs to be classified by the proposed formula of traffic state efficiency index and the design speed of the classified highway, and the MFD of the current state is drawn. To achieve the best prediction performance, the performance parameters were adjusted, such as the number of nodes and delay parameters of the NARX neural network (NARX NN). Finally,

with the traffic state efficiency index and the predicted traffic output, the predicted MFD is drawn to express the traffic operation state. In this model, the capabilities of MFD to characterize traffic state relationships at the macroscopic scale and the capabilities of the NARX NN to handle traffic flows with time-series characteristics are utilized, respectively. To the best knowledge of the authors, this paper is one of the first attempts to combine a neural network forecasting model with the MFD and employ it for long-term traffic flow prediction on a regional road network.

The main contributions of the paper are summarised as follows: (1) The NARX-MFD method for predicting long-term traffic state parameters is proposed and the NARX NN is verified to be capable of long-term traffic prediction. (2) The formula of the traffic state efficiency index is proposed. With the analysis of the traffic flow parameter curves, the traffic state of the regional road network is classified into four categories by using the smooth speeds in the curves and the design speeds of the classified highway. This will be used as an evaluation classification for the predicted state. (3) With the predicted traffic flow parameters and traffic state efficiency indicators, the predicted traffic state curve is produced and provides the traveler with more intuitive feedback on the future state of the road network.

Our work is presented as follows. The second section introduces the related works of prediction methods of traffic state parameters and macroscopic evaluation of traffic state and analyzes the reasons for choosing NARX NN and MFD. The third section addresses the theoretical system of traffic state evaluation. It introduces the formulas of traffic state efficiency index and the evaluation grading of regional road networks derived from it. The fourth section mainly focuses on data description and experimental design and describes several time-series prediction models involved in the experiments. Section 5 then presents the results of the comparison experiments of the prediction models and the outputs of the predicted traffic states and analyzes the reasons for them. Finally, in Section 6, we conclude our paper and outlook future research.

## 2. Related Work

In this chapter, the related work is reviewed and summarised in terms of two aspects: prediction methods of traffic state parameters and macroscopic evaluation of traffic state.

*2.1. Traffic State Parameters Prediction.* In the context of the ITS, numerous prediction methods for traffic state parameters have emerged. In the past research, prediction models for various types of traffic state parameters have been proposed in the relevant literature. According to the type of parameter, these include traffic flow, speed, density, occupancy, and the resulting travel time and travel time index. Based on the traffic flow parameter relationships proposed by Greenshields, when conducting the selection of indicators, some have argued that the selection of volumes is important because of the stability of volumes [12].

Some have argued that speed prediction is more critical [1, 13], while others have argued that prediction accuracy can be improved by the combined effect of multiple traffic metrics, such as simultaneous characterization of volume and density [14]. In recent years, with the continuous improvement of algorithms and the development of data collection equipment, travel time and travel time index have received increasing attention as predictors. Li et al. [15] proposed a model based on an ensemble empirical mode decomposition and random vector functional link network to predict the travel time of a highway network. Xu et al. [16] chose the travel time index when dealing with the prediction of regional road networks. In contrast, due to the limitations of the collection equipment, other indices are difficult to obtain. So for the current work, we use traffic data for consideration. However, when these data are available, a multi-index prediction approach is taken.

A comparative study by Smith et al. [17] found that studies of flow prediction models can be divided into parametric and nonparametric methods, with parametric methods being more time consuming and nonparametric methods being more effective when dealing with the random data.

Research on parametric methods is much earlier and more established. Jensen [18] used linear regression methods for traffic forecasting in 1972, and Hogberg [19] used nonlinear regression methods to estimate the parameters in traffic forecasting models in 1973. Lee et al. [20] used the ARIMA model for short-term freeway traffic volume forecasting and proved to be able to accurately predict traffic variations with time-series characteristics. The method contains fewer parameters but has better forecasting results. ARIMA is widely applied as a parametric model for time-series forecasting, which is obtained by introducing hidden variables through moving average (MA). Tatsuya [21] eliminated the seasonal factor based on ARIMA and applied the SARIMA model to forecast both short-term and long-term traffic flows, which proved that SARIMA is more suitable for long-term prediction. On the freeway, the sudden changes in traffic conditions are often caused by the effects of weather, accidents, and surges in demand. Thus Bezuglov [22] tested three grey system theory models and demonstrate that grey system models are better adapted to the sudden changes in parameters. Cai et al. [23] then formulated a noise-immune Kalman filter model applied to short-term traffic flow prediction, which solved the problem that the Kalman filter is insensitive to traffic flows satisfied by non-Gaussian noises.

The research on nonparametric methods mainly includes neural network models, support vector regression (SVM), and K-nearest neighbor (KNN) algorithms. Among them, the first application of KNN in traffic flow prediction was in [24]. Meanwhile, Harrou et al. [25] monitored the traffic congestion problem with the KNN algorithm, which was very effective and was based on the advantages of the Kalman filter (KF) and a piecewise switched linear traffic (PWSL). Support vector regression (SVR) models were applied and proven to be appropriate for traffic flow

forecasting on urban roads, e.g., [26]. And Castro-Neto et al. [27] demonstrated that the OL-SVR model can well deal with the atypical conditions when they occur and is better than other prediction models. The Gauss-SVR model proposed by Li et al. [28] has better predictive performance than the neural network when it comes to urban traffic flow forecasting, but it should be noted that the structure of the BP neural network used in the paper is simpler.

Although neural network models have been considered to be better suited for short-term traffic forecasting [29–31], and it has been argued that the accuracy of predictions gradually decreases as the range of predictions increases, neural networks do not work well for long-term traffic forecasting. However, Çetiner et al. [32] applied an artificial neural network (ANN) to predict long-term (1h) data and short-term (5 min) data for one day in a historical dataset for the city of Istanbul and discovered that better results were obtained with the long-term dataset. Hao et al. [33] demonstrated that Long Short-Term Memory (LSTM) neural networks can consistently and effectively achieve long-term traffic forecasting based on dynamic traffic flow probability graphs and alleviate the effect of missing data on the forecasting results. Therefore, the neural network can enable long-time forecasting. It has been demonstrated that NARX NN is also capable of predicting long time-series tasks [34].

While parametric and nonparametric methods have their advantages and disadvantages, both should be tested and compared with the baseline model to obtain the optimal model. However, the use of combined a model, or algorithms, or heuristic approach for forecasting seems to be the current rage. This is very common in complex traffic forecasting, e.g., [23, 28]. But some problems have remained. In [35], Vlahogianni has discussed comparing the use of baseline and combined models and raised the question of when to combine predictions. It addressed the situation where combined models may make it difficult to control errors in long-term prediction.

In the studies of traffic parameter prediction mentioned above, most of them were for single sites or individual road sections. In contrast, when analyzing macroscopic characteristics, traffic flow parameters have to be predicted for multiple sections on the road and the entire road network in an integrated manner. In a previous study, Williams [36] used the ARIMA model to predict traffic flows at several test sections within the city of Bonn, Germany. Kamanakis [37, 38] applied the STARIMA model to traffic flow forecasting on major arterial road sections in the city of Athens, Greece, and described the topological distance relationship with the sections by using a weight matrix. A comparison with ARIMA and VARMA revealed similar prediction performance. In a recent study, Li et al. [39] proposed a short-term multistep traffic prediction model for macroscopic road networks, Dynamic Graph Convolutional Network (DGCN), which is based on the Dynamic Graph Convolutional (DGC) model. Therefore, the above study shows that it is feasible to forecast the traffic state of the regional road network.

**2.2. Macroscopic Evaluation of Traffic State.** In macroscopic traffic state evaluation, the range of study is often extended to multiple cross sections or the entire road network based on the fundamental diagram theory of traffic flow. To describe the traffic state of urban road networks, scholars have proposed various theories, such as dimensional analysis theory [40] and two-fluid theory [41–43]. The MFD was proposed by Godfrey in 1969 [44]. Daganzo et al. [45, 46] in 2008 revealed the characterization of flows and trip completion rates by using GPS data of taxis in Yokohama, Japan, which has a high level of congestion. The existence of the MFD was formulated and verified, and a new paradigm was initiated. With the MFD, it is possible to significantly reduce the complexity of the traffic flow model, thus achieving a controlled design of the model for the urban network.

After that, analytical modeling and control researches on MFD have been continuously proposed. The main difference between these models is that the controlling strategies are different and the controlled models are summarised as follows.

- (1) Three-parameter baseline model: according to Edie's [47] simple definition of road network traffic flow, the arithmetic mean or weighted mean of the section traffic flow parameters can be obtained, as shown in the following equation:

$$\left\{ \begin{array}{l} \bar{Q} = \frac{1}{N} \sum_i Q_i, \bar{K} = \frac{1}{N} \sum_i K_i, \bar{V} = \frac{\bar{Q}}{\bar{K}} \end{array} \right. \quad (1)$$

where  $\bar{Q}$ ,  $\bar{K}$ , and  $\bar{V}$  are the averages of the volume (pcu/h), density (pcu/km), and speed (km/h) in the road network, while  $N$  is the number of sections or road segments detected.

- (2) Perimeter control models [48]: By defining a generic mathematical model for multireservoir networks with well-defined MFDs for each reservoir, maximizing the throughput of the system without the necessity to calculate workload and future demand data is achieved.
- (3) Adaptive control models [49]: These models optimize the controller through real-time observation of changes to adapt the model to uncertainties and state delays under different parameters.
- (4) Hierarchical perimeter control models [50]: This model is proposed to address the problem of traffic congestion with high traffic demand and heterogeneous distribution. The hierarchical control strategy is introduced, and each layer solves the corresponding problem according to different demands to achieve the stability of the traffic system.
- (5) Robust control models [51, 52]: This model mainly considers the parameter uncertainty of single or multiple regional networks and sets different algorithms for robust controllers according to various demands to ensure stable operation of the model.

And in addition to the MFD, new progress has been introduced to the study of macroscopic traffic flow

analysis. Thonfer et al. [53] modularized the urban traffic network and proposed a model that contains signal control and is adapted to large-scale urban road networks, which can improve operational efficiency through parallel computation. Xiao et al. [54] presented traffic speed cloud maps that can dynamically capture the process of congestion formation and degradation, but the scale of data resources required is larger than that of the MFD.

In summary, scholars have demonstrated in previous studies that traffic data is highly nonlinear, periodic, and time-varying, and that the NARX model also supports the prediction of traffic data with time-series features compared to the SARIMA model [55]. Although NARX NN has long-term forecasting capability, whether it can fulfill the task of long-term traffic state parameter prediction or not still needs to be verified [56]. In this paper, we decided to conduct comparative experiments using the NARX model with the other baseline models. While the MFD is better than other methods of macroscopic state evaluation in presenting the traffic state characteristics in terms of data relationships, we chose MFD.

### 3. Evaluation Model of Regional Road Network's Traffic State Based on the MFD

For our study, considering the large scope of road network evaluation, the traditional microscopic traffic models are not effective, so the MFD is used to reflect the traffic conditions of the road network.

**3.1. Traffic State Classification and Index Calculation.** As multiple indicators can reflect traffic conditions, the selection of the appropriate indicators is a key step. Therefore, the selection steps of indicators in this paper will follow the following principles: (1) Clear purpose, (2) comprehensive scope of the evaluation, and (3) being practical and operable. After reviewing the relevant literature [57, 58], the current commonly used traffic state evaluation indicators were compiled and shown in the following figure.

The parameters of flow, speed, and occupancy were the most commonly used (see Figure 1). Considering the difficulty of index acquisition, evaluation criteria, and model applicability, the traffic flow parameters of flow, density, and speed were selected in this paper to reflect the condition of the road network.

The essence of road network condition evaluation follows the basic evaluation principles and discriminates the traffic state through scientific and reasonable methods. In traditional traffic evaluation systems, the traffic flow fundamental diagram model is based on the analysis of historical data. It is established with statistical methods and characterizes the relationship between the parameters of a single section of road and has a limited scope of application. The relationships between spatially averaged flow variables of multiple sections or regional road networks are analyzed with MFDs, which provide an intuitive and systematic description of the overall structure of the road network.

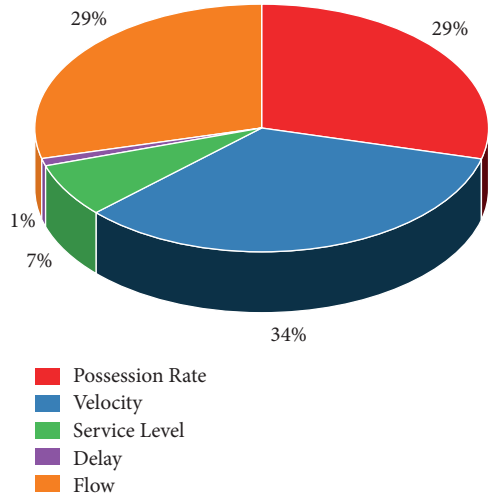


FIGURE 1: Usage of common indicators for traffic state assessment.

To discern whether the road network is congested or not, it is assumed that the road network is homogeneous and all the cross-sectional traffic parameters are compatible with the flow relationship curve. According to the critical speed in the flow model proposed by Greenshields, it is divided into congested and noncongested areas.

Let  $\partial Q/\partial V = 0$ ; then,  $V_m = V/2$  corresponding to  $Q_{max}$  can be obtained and thus the point  $(V_f/2, Q_m)$  is the maximum value of the flow-velocity curve.

The extreme value of the flow-velocity curve  $Q_m$  can be used as a dividing point between the uncongested and congested areas (see Figure 2).

In this paper, we have introduced  $M$ , the road network's traffic condition efficiency index, which characterizes the state of vehicles per time in the road network. Its calculation is shown as follows:

$$M = QVL = K_j \left( V^2 - \frac{V^3}{V_f} \right) L, \quad (2)$$

where  $L$  is the length of the road section in the road network.

From equation (2), it can be seen that there is a threefold functional relationship between  $M$  and  $V$ , and there must be an  $M_{max}$  corresponding to a  $V$ .

When  $\partial M/\partial V = 0$ ,  $V_A = 2/3V_f$  can be obtained, which corresponds to  $M_{max}$ . At this time, the point  $(2/3V_f, V_A)$  can be regarded as the dividing point between the free flow (F) state and the harmonic flow (H) state in uncongested areas. Similarly,  $V_A$  corresponds to  $V_D = 1/3V_f$  of the same flow; then the point  $(1/3V_m, Q_D)$  can be regarded as the dividing point between the synchronous flow (S) state and the blocked flow (B) state in the congested area, as shown in Figure 3.

In summary, the traffic state of the road network can be divided into four classes based on the traffic state curve. The intervals of each class are shown in Table 1.

**3.2. Evaluation Model of Road Network Traffic State.** The road network is composed of highways of many different technical grades. The state of the road network cannot be

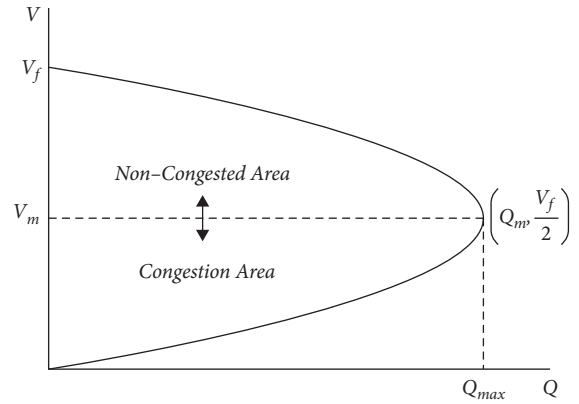


FIGURE 2: Traffic-velocity model congestion area division.

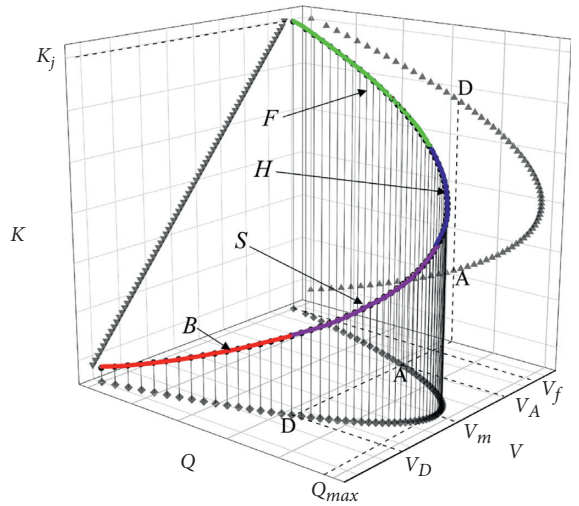


FIGURE 3: Traffic state curve.

TABLE 1: Traffic state classification intervals.

State	Interval	State	Interval
Blocked flow	$[1/3, 0)V_f$	Harmonic flow	$[2/3, 1/2)V_f$
Synchronous flow	$[1/2, 1/3)V_f$	Free flow	$[1, 2/3)V_f$

generalized due to the technical variability of the different highway grades. Therefore, it is necessary to evaluate the traffic state of the same highway levels first, then divide the weights according to the different technical highway grades and calculate them, and finally formulate the traffic state model of the road network.

- (1) The traffic state evaluation model for road networks of the same level uses the average speed of all vehicles in the road network as the state evaluation index, as shown in the following equation:

$$V_{jk} = \frac{\sum_{i=1}^n V_{ijk}}{n}, \quad (3)$$

where  $V_{jk}$  (km/h) is speed state for all roads of grade  $j$  at the  $k$ -th inspection time interval;  $V_{ijk}$  (km/h) is the speed state for the  $i$ -th road monitoring station of the grade  $j$  roads at the  $k$ -th inspection time interval, and  $n$  is the number of  $j$ -grade road monitoring stations in the road network.

(2) Equation for calculating road weights:

The importance of a road section is related to its capacity and length. Variations in the operational state of road sections with higher capacities or smaller lengths will have a greater impact on the operational state of the road network. The importance weights of intersections and sections are modeled as follows in equation (4), where traffic monitoring stations are calculated as intersections:

$$\omega_c(L_{ij}) = \frac{C(L_{ij})}{l(L_{ij})} \approx \frac{Q(L_{ij})}{l(L_{ij})}, \quad (4)$$

where  $\omega_c(L_{ij})$  is the section importance weight;  $C(L_{ij})$  is the section capacity, in  $veh/h$ ; and  $l(L_{ij})$  is the section length, in  $m$ .

Using the road section level and weights, a comprehensive weighting model for road sections is established using the following equation:

$$\omega_k(L_{ij}) = \frac{\omega_c(L_{ij})}{\sum_{i=1}^h \omega_c(L_{ij})} = \frac{\sum_{i,j=1}^n Q_k(L_{ij})}{\sum_{j=1}^m Q_k(L_j)}, \quad (5)$$

where  $\omega_k(L_{ij})$  is the comprehensive weight for the  $i$ -th road monitoring station of the grade  $j$  roads at the  $k$ -th inspection time interval;  $Q_k(L_{ij})$  is the flow for the  $i$ -th road monitoring station of the grade  $j$  roads at the  $k$ -th inspection time interval, in  $veh/h$ ;  $Q_k(L_j)$  is the flow for all roads of grade  $j$  at the  $k$ -th inspection time interval, in  $veh/h$ ;  $m$  is the number of technical grade types and  $h$  is the number of road sections in the road network.

(3) Road network traffic state evaluation model:

The traffic state evaluation model of the road network within the evaluation range is obtained by multiplying and adding each grade of the road according to the correspondence in equations (4) and (5), as shown in the following equation:

$$F_k = \sum_{j=1}^m \omega_k(L_{ij}) \bar{V}_{jk}, \quad (6)$$

where  $F_k$  is the traffic state value of the road network in the  $k$ -th time interval, in (km/h).

In this study, we only include primary roads and secondary roads due to the limitations of road network grades and monitoring devices in the selected areas. The higher the grade is, the more the transport functions it corresponds to undertake, and the more the weights that need to be assigned. In terms of the capacity of each road class, the weights of these two classes of roads were determined to be 0.6 and 0.4.

## 4. Experiment

After analyzing and obtaining the above method, the experimental flowchart of this paper is summarised as follows (see Figure 4).

**4.1. Data Description.** To test the operational effectiveness of the state prediction model over a large area, the location chosen for this study was in Linzi District, Zibo City, Shandong Province, which has an area of 672.58 km<sup>2</sup> for the study. The area has long latitudinal boundaries as it is crossed by many roads of various grades to Qingzhou City. The traffic flow is inevitably high as vehicles must pass through the area to enter Qingzhou. Considering the large area of collection, the monitoring stations were selected to monitor four sites, Zi River, Bei Liu, Reed River, and Wu Tai, which are distributed on the main national roads (e.g., G308, G309, and G233) and provincial roads (e.g., S227, S228, and S102) in the region (see Figure 5(a)). Among them, there are more factories near the Reed River monitoring stations, and large vehicles will pass through the station, so the sectional traffic volume is larger than the other stations. The locations are relatively dispersed to reflect the intrinsic links of the regional road network. The weights are then divided according to the grade of each road, as shown in Figure 5(b).

To reflect the correlation and weight expression of adjacent links in the road network more intuitively, a spatial weight matrix  $W$  (0.6,0.4) was established, as shown in Figure 6. In the figure, "0.6" and "0.4" indicate the corresponding road level weights between two nodes, and "0" indicates that the two nodes are not connected.

The total length of the roads involved is 122.135 km. From 12/01/2019 to 02/28/2020, the raw data was provided by the Shandong Provincial Highway Traffic Investigation and Management Institute in Zibo, Shandong Province, China. All vehicles passing on these roads are recorded in the station's database with radar-based detectors and video collecting equipment at the monitoring stations. The traffic flow data was measured every 1 hour and the speed data is measured every 5 min. In addition, due to the epidemic, some roads were closed and the communication of the detector was faulty during the collecting period and resulting in some traffic data being missing (1.94%). To minimize the influence, the missing values were recovered by the average values of other days measured at the same time during data processing. To avoid any loss of monitor accuracy during rain and snow, which could have an impact on the accuracy of the acquired data, the data was recalibrated using video equipment to ensure accuracy. After the above steps and data processing and data imputation, the traffic flow data and average speeds obtained from each of the four detection points were grouped by hour, with 8,640 sets of data each (see Figure 7). It implies that 90 days \* 24 h of data were recorded for each monitoring station.

**4.2. Traffic Flow Prediction Modeling and Baseline.** As previously mentioned, one of the main objectives of this paper aims to investigate the feasibility of NARX NN in long-term

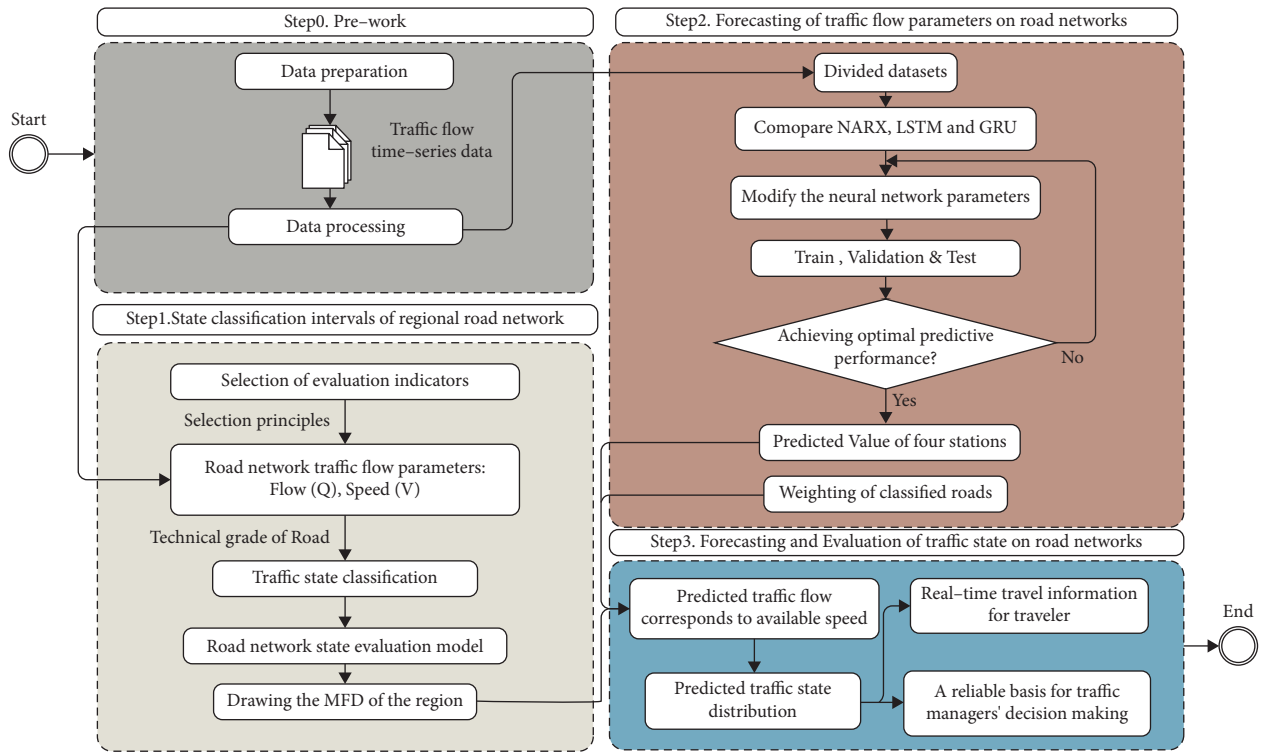
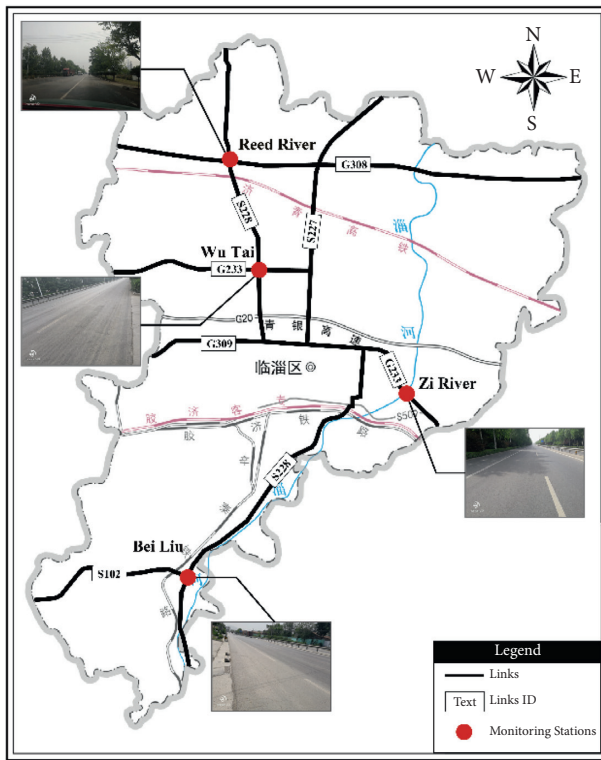
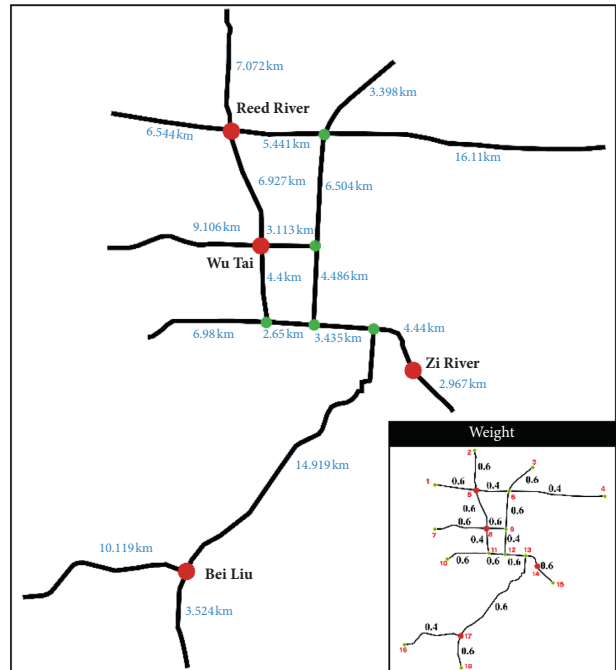


FIGURE 4: Flowchart of road network's traffic state prediction.



(a)



(b)

FIGURE 5: Selected road network in Linzi District. (a) Station location map in Linzi District. (b) Network diagram of links with the division of weights diagram.

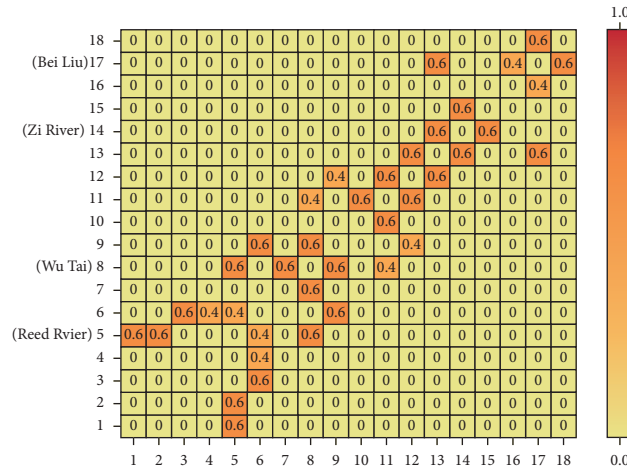


FIGURE 6: Spatial weight adjacency matrix  $W$  (0.6,0.4) for classified roads.

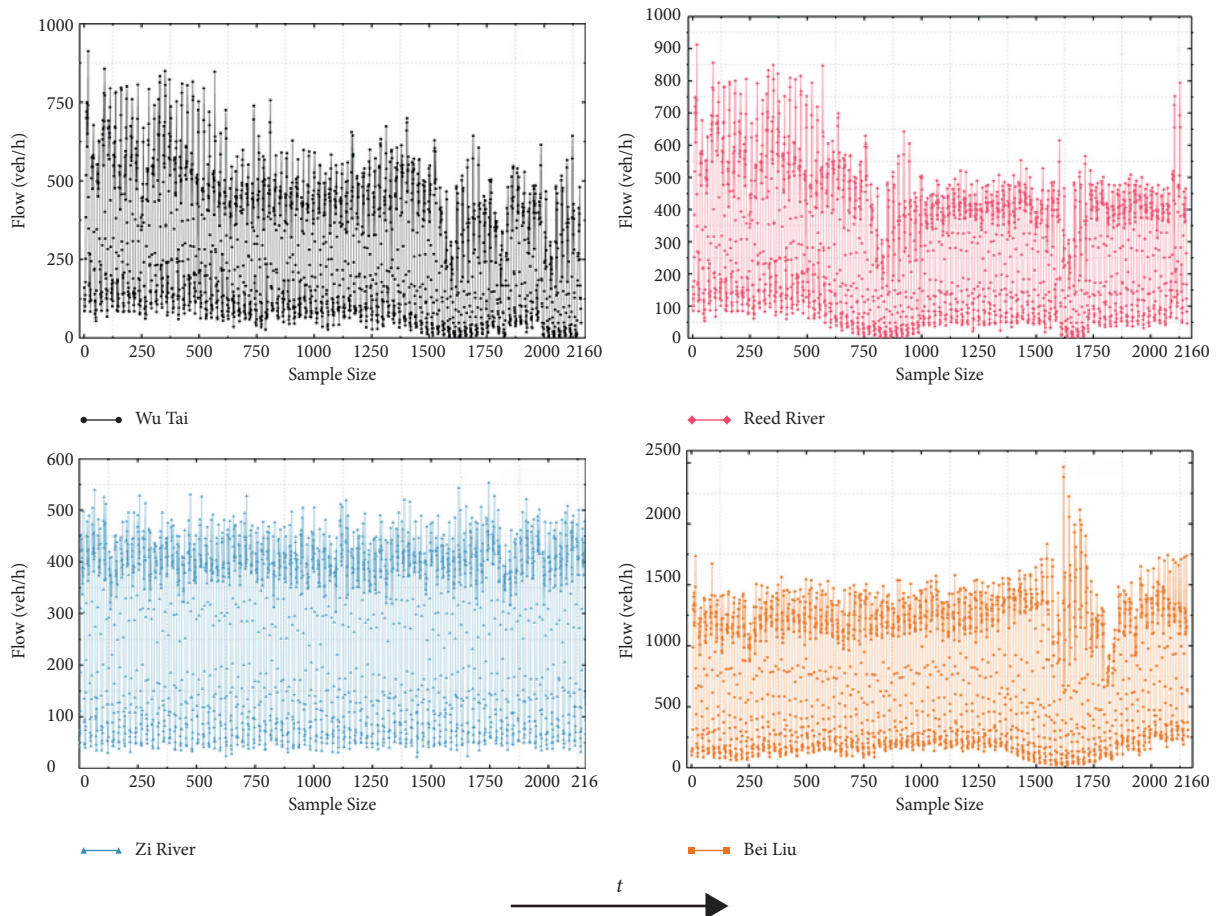


FIGURE 7: Traffic flow raw data from four monitoring stations.

traffic flow forecasting and compares it with other types of artificial intelligence methods. To make a fair comparison, the LSTM NN and GRU NN were chosen as the two methods.

4.2.1. NARX NN. The NARX NN can be seen as a neural network version of the time-series model, which can consider external time-series inputs. Its predictions are based on

historical data from the same series. As a standard NARX neural network architecture, the parallel architecture feeds the output back into the input of the feedforward neural network, where TDL is the tapped delay line. The other architectural mode is a series-parallel architecture that uses the real output rather than the output estimated by the feedback. It enables the input to the feedforward network to be more accurate and can be trained by using static

backpropagation [59, 60]. The simple structure is shown in Figure 8.

As a typical nonlinear prediction model with input and output delays, in the NARX model, the traffic flow  $\hat{y}(t+1)$  is obtained from the input and output flow predictions before  $t$ . The model's expression is shown in the following equation:

$$\hat{y}(t+1) = f(x(t-1), \dots, x(t-d), y(t), y(t-1), \dots, y(t-d)), \quad (7)$$

where  $f$  is a nonlinear function and  $d$  is the order of delay of the input and output.

**4.2.2. LSTM NN.** As a special form derived from the RNN model, LSTM NN can better deal with data of longer sequences. Long-term memory is preserved through gating and cell state updates, and time-related information is retained through storage cells to solve the gradient disappearance and gradient explosion problems in RNN models.

**4.2.3. GRU NN.** The GRU NN, as well as the LSTM NN mentioned above, is excellent variant of the RNN [61]. However, differing from the LSTM NN, the GRU network simplifies the structure by merging the forgetting gate and the input gate into an update gate. The updated state of the cell is controlled through the update gate, and the reset gate determines how to combine the new input information with the previous memory to make convergence faster.

To compare the traffic flow prediction effect of the NARX NN, LSTM NN, and GRU NN models, the same traffic flow datasets are chosen. The root means square error (RMSE), mean absolute percentage error (MAPE), and  $R^2$  were selected to evaluate the difference between the real results  $Y(t) = \{y_1, y_2, \dots, y_n\}$  and predicted results  $\hat{Y}(t) = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$ , as shown in equations (8)-(10):

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}, \quad (8)$$

$$\text{MAPE} = \frac{1}{p} \sum_{i=1}^p \frac{|y_i - \hat{y}_i|}{y_i}, \quad (9)$$

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y)^2}{\sum_i (\bar{y} - y)^2}, \quad (10)$$

where  $p$  is the number of samples,  $y_i$  is the flow's measured value, and  $\hat{y}_i$  is the corresponding predicted value.

**4.3. Experimental Settings.** The purpose of this experiment would be to predict future day traffic values with three-month data and then obtain the predicted traffic status from MFD and traffic status evaluation intervals. For the structure of the dataset described in Section 4.1 and the baseline model in Section 4.2, the dataset of each monitoring station is partitioned into a training set, a validation set, and a test set, corresponding to a ratio of 7 : 1.5 : 1.5. It is

worth noting that the datasets of the methods used were uniform to be fair for comparisons between different methods. Moreover, some of the parameters were taken as constant or default values to facilitate the control of variables. The open-closed loop control (see Figure 8(a)) is chosen for the training of the NARX NN, while the closed loop control (see Figure 8(b)) is chosen for its prediction with a learning rate of 0.005, which is intended to obtain better prediction results in long-term traffic flow prediction. The corresponding training algorithm used the Levenberg-Marquardt (LM) algorithm to obtain a faster iteration rate. The training process for both LSTM NN and GRU NN utilized the Adam optimizer, with the Max-Epochs set to 250, the gradient threshold of 1, and an initial learning rate of 0.01. After the 125th training round, the learning rate is multiplied by 0.2 to reduce the learning rate. As it is a sequence of input and output data, both input and output are set to 1 dimension.

To compare different prediction methods and to ensure the uniformity of the platform, the experimental equipment is composed of a 6-core AMD3600 X 3.80 GHz CPU, 16 GB of RAM, a GEFORCE RTX 2060S 8 GB GPU, and MATLAB 2021a is chosen for the experimental platform.

## 5. Results

According to the above progress, the corresponding steps of Figure 4 and the obtained results are as follows.

*Step 1.* In this step, when classifying the regional states, the traffic state classification intervals in Table 1 and the design speeds of the corresponding roads are used. We can obtain the smooth flow speed  $V_f$ , consequently  $2V_f/3$ ,  $V_f/2$ ,  $V_f/3$ , and the boundary value of each state. In this study, we used the data for the three months from December 2019 to February 2020 as the basis to obtain the actual state classification intervals of the regional road network by the above method (see Table 2).

According to the relationship between the traffic flow parameters within the regional road network and the speed classification interval in the table above, we can obtain the MFD of the road network (see Figure 9). At this time, the traffic flow of the overall network is classified into four different states according to the average speed values, with different colors representing different traffic flow states. As there are a large number of primary roads in the network and the monitoring stations are mainly located at the intersection of the two grades of roads, the classification intervals will be based on the primary roads.

*Step 2.* After Step 1, we first evaluated and compared the performance of several of the above prediction models on the datasets from the four monitoring stations based on the experimental design in Section 4.3. By varying the number of hidden layers or units, we were able to filter out the best method of setting the model parameters to ensure higher prediction accuracy. The specific parameter data are shown in Tables 3–5.



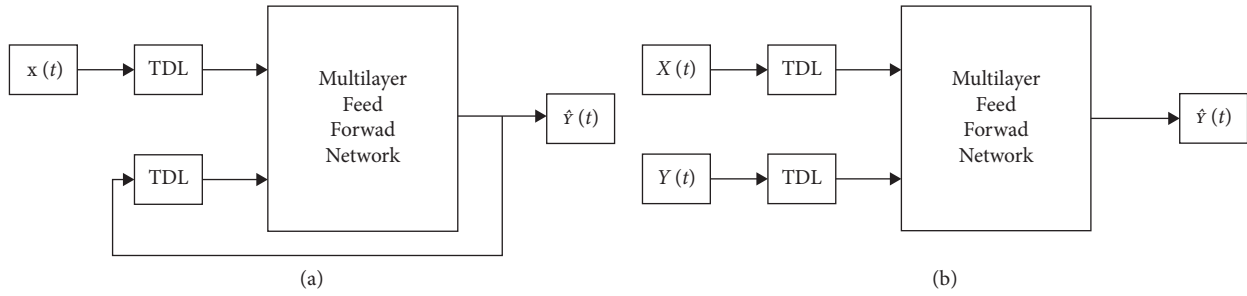
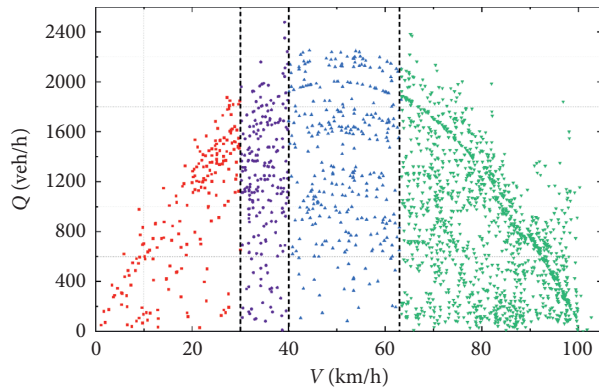


FIGURE 8: Two architectures of NARX networks. (a) Parallel architecture. (b) Series-parallel architecture.

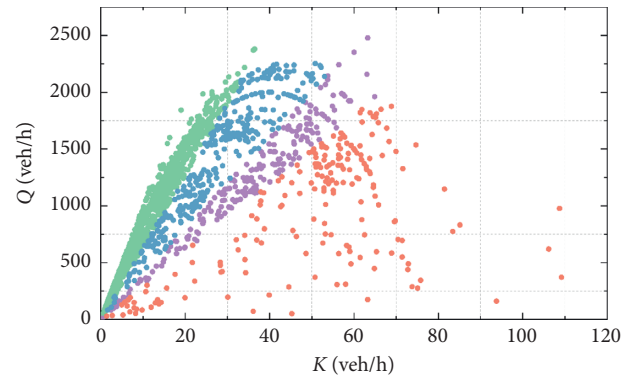
TABLE 2: State classification intervals of the regional road network.

Technical grade	Blocked flow	Synchronous flow	Harmonic flow	Free flow
Primary roads	[0, 30)	[30, 40)	[40, 190/3)	[190/3, 100]
Secondary roads	[0, 20)	[20, 30)	[30, 40)	[40, 60]



- Blocked flow
- Synchronous flow
- ▲ Harmonic flow
- ▼ Free flow

(a)



- Free flow
- Harmonic flow
- Synchronous flow
- Blocked flow

(b)

FIGURE 9: MFD of the overall road network: (a) Q-V curve, (b) Q-K curve.

TABLE 3: Traffic flow prediction accuracy evaluation of NARX NN

Stations	NARX NN					
	Hidden layer	8	9	10	11	12
Parameters	Delay layer	2	2	2	2	2
Wu Tai	$R^2$	76.749%	85.321%	<b>88.433%</b>	79.343%	77.545%
	RMSE	122.759	97.463	<b>73.736</b>	114.381	123.269
	MAPE	7.53%	6.93%	<b>4.98%</b>	8.24%	14.18%
Zi River	$R^2$	92.784%	93.494%	<b>94.983%</b>	93.923%	89.621%
	RMSE	62.047	56.492	<b>39.835</b>	58.277	75.53
	MAPE	1.77%	2.21%	<b>1.61%</b>	1.82%	1.98%
Bei Liu	$R^2$	93.442%	94.249%	<b>96.600%</b>	92.101%	92.509%
	RMSE	183.368	171.490	<b>131.763</b>	210.467	195.711
	MAPE	1.11%	2.87%	<b>1.09%</b>	1.35%	2.54%
Reed River	$R^2$	88.094%	94.853%	<b>95.047%</b>	94.756%	89.006%
	RMSE	245.781	167.633	<b>174.916</b>	166.907	252.857
	MAPE	2.47%	1.13%	<b>1.22%</b>	1.18%	1.61%

TABLE 4: Traffic flow prediction accuracy evaluation of LSTM NN.

Stations		LSTM NN				
Parameters	HiddenUnites	276	282	288	294	300
	SequenceInputLayer	1	1	1	1	1
	FullyConnectedLayer	1	1	1	1	1
Wu Tai	$R^2$	92.979%	94.380%	<b>94.036%</b>	92.305%	91.237%
	RMSE	75.93	65.582	<b>72.511</b>	78.963	82.293
	MAPE	5.27%	5.81%	<b>5.02%</b>	5.50%	6.35%
Zi River	$R^2$	99.039%	<b>99.081%</b>	99.112%	99.105%	99.007%
	RMSE	21.846	<b>21.499</b>	21.292	21.158	22.365
	MAPE	1.64%	<b>1.59%</b>	1.67%	1.60%	1.74%
Bei Liu	$R^2$	93.340%	94.916%	<b>95.774%</b>	94.480%	95.665%
	RMSE	198.882	174.465	<b>159.654</b>	181.866	162.147
	MAPE	1.789%	1.295%	<b>1.105%</b>	1.368%	1.122%
Reed River	$R^2$	86.063%	87.124%	<b>91.011%</b>	90.481%	86.132%
	RMSE	275.21	180.68	<b>162.572</b>	198.024	228.13
	MAPE	2.86%	2.31%	<b>1.98%</b>	2.38%	2.15%

TABLE 5: Traffic flow prediction accuracy evaluation of GRU NN.

Stations		GRU NN				
Parameters	HiddenUnites	270	276	282	288	294
	SequenceInputLayer	1	1	1	1	1
	FullyConnectedLayer	1	1	1	1	1
Wu Tai	$R^2$	81.181%	<b>93.552%</b>	88.85%	79.53%	84.01%
	RMSE	118.076	<b>73.332</b>	94.207	147.014	113.782
	MAPE	6.617%	<b>5.063%</b>	5.45%	7.44%	6.10%
Zi River	$R^2$	98.971%	98.966%	98.865%	99.068%	<b>99.091%</b>
	RMSE	22.805	22.642	24.005	21.542	<b>21.242</b>
	MAPE	1.807%	1.727%	1.863%	1.626%	<b>1.604%</b>
Bei Liu	$R^2$	95.736%	96.024%	<b>96.145%</b>	95.882%	96.117%
	RMSE	160.169	154.73887	<b>152.6051</b>	157.25165	153.57241
	MAPE	1.144%	1.097%	<b>1.090%</b>	1.149%	1.085%
Reed River	$R^2$	79.014%	85.961%	<b>88.093%</b>	81.664%	83.526%
	RMSE	281.021	190.011	<b>170.425</b>	236.447	200.634
	MAPE	3.02%	2.54%	<b>2.12%</b>	2.83%	2.51%

The optimal selection of parameters for the different prediction models was collated for each monitoring station as shown in the figure below (see Figure 10).

As a result of the above comparison experiments, it can be seen from the comparison of the indicators in Tables 3–5 and Figure 10 that, due to the similar trend of the flow data at each monitoring station, it resulted in a very similar selection of parameter and evaluation indices for each location in each method. By comparing the performance evaluation metrics of each model with the best parameters, it was found that the  $R^2$  values of NARX NN are larger than both LSTM NN and GRU NN, but the RMSE values were smaller at some monitoring stations, and the MAPE values of NARX NN were smaller than both models.

Furthermore, it is surprising that when the number of nodes in the hidden layer of the LSTM NN is continuously increased, there is no gain in performance, but rather a decrease. The reason for this is that the method is already

approaching the optimal number of nodes for this dataset, and increasing the number of nodes will complicate the network and tend to “overfit” or prolong the training time. Due to its simpler structure, the GRU NN converges faster than the LSTM model during training but has lower prediction accuracy than the LSTM.

Therefore, the traffic flow of the four observation stations was predicted based on the optimal number of hidden layers or units for the above-mentioned models. The results obtained are shown in Figure 11 below.

After observing and analyzing the above prediction curves, it can be seen that all three models fit well at all four monitoring stations and that the NARX NN converges faster and with relatively smaller errors than the other two models. However, in some time intervals, the NARX model oscillates less than the other two models. This indicates that while the model is well adapted to the time-varying characteristics of the traffic flow data, it is not able to cope with sudden variations in the flow at some sites.

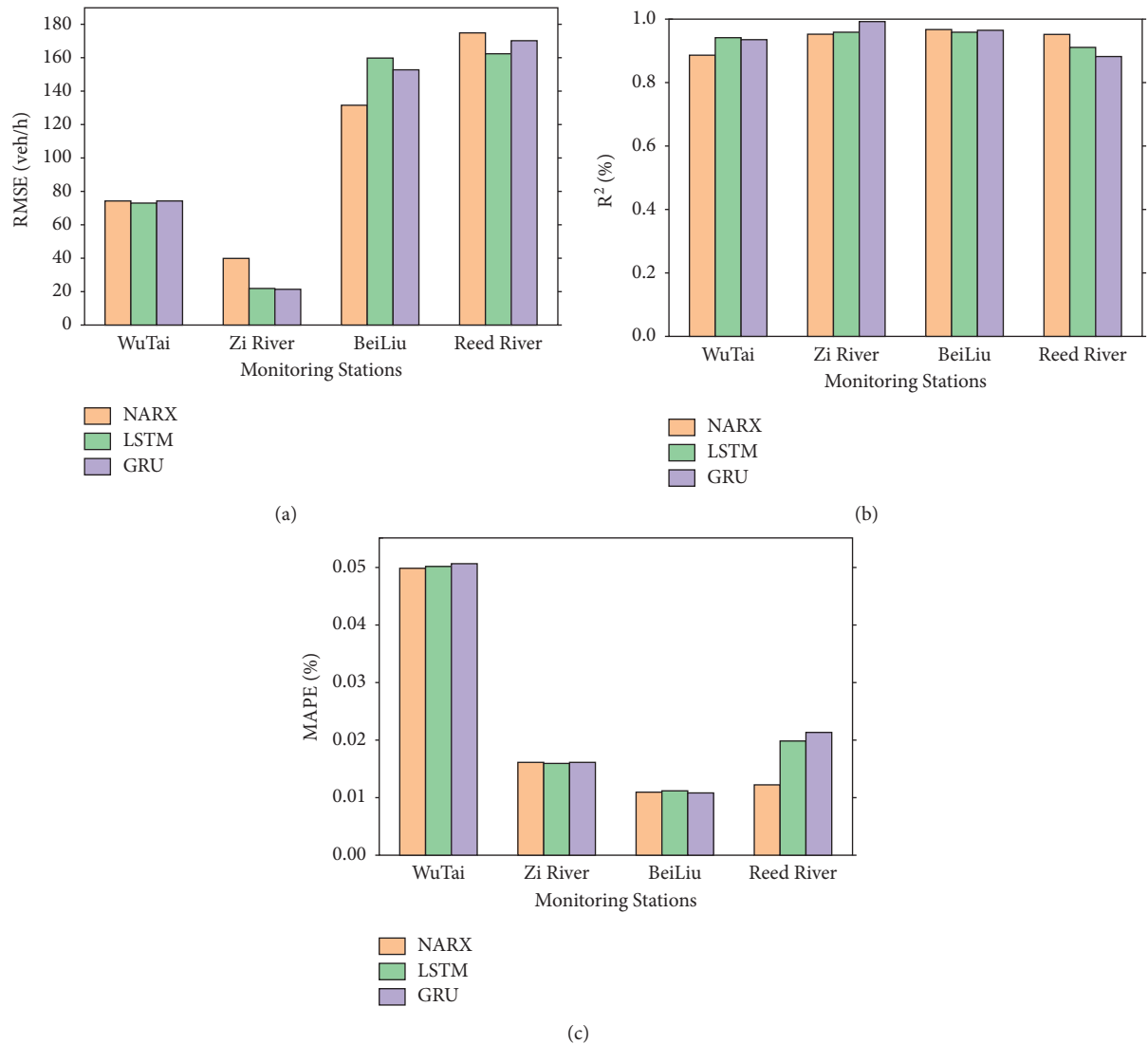


FIGURE 10: Comparison of evaluation indicators for predicted results at the four monitoring stations.

*Step 3.* After the first two steps, the flow for the entire network is calculated with equations (4) and (5), the classified road weights, and the predicted flows for the four monitoring stations obtained. After processing the traffic data for the next day, a comparison of the three neural network models shows that the NARX model is highly adaptable and real-time for nonlinear traffic data, while the LSTM has better lagging performance (see Figure 12). The original values of traffic flow matched well with the predicted values. The predicted results are then evaluated. As can be seen from Table 6, the three evaluation indicators of the NARX model are slightly better than the other two models for predicting traffic flow for the next day. Though similar, it converges more quickly.

Finally, according to the traffic state classification interval, the traffic state of the road network for the next day is obtained from the current MFD, the predicted traffic flow, and the corresponding speed (see Figure 13). The analysis in the figure shows that the free flow is distributed during the low peak hours of the day. The blocked flows are mainly distributed during the morning and evening peak hours, and the synchronic flows are distributed after them with the temporal progression, as the main state after the peak hours. Harmonic flows occur less frequently and occupy fewer hours. The main reason for the above phenomenon is that most of the selected monitoring stations are located in the surroundings of the central city. Residents have to pass through these points

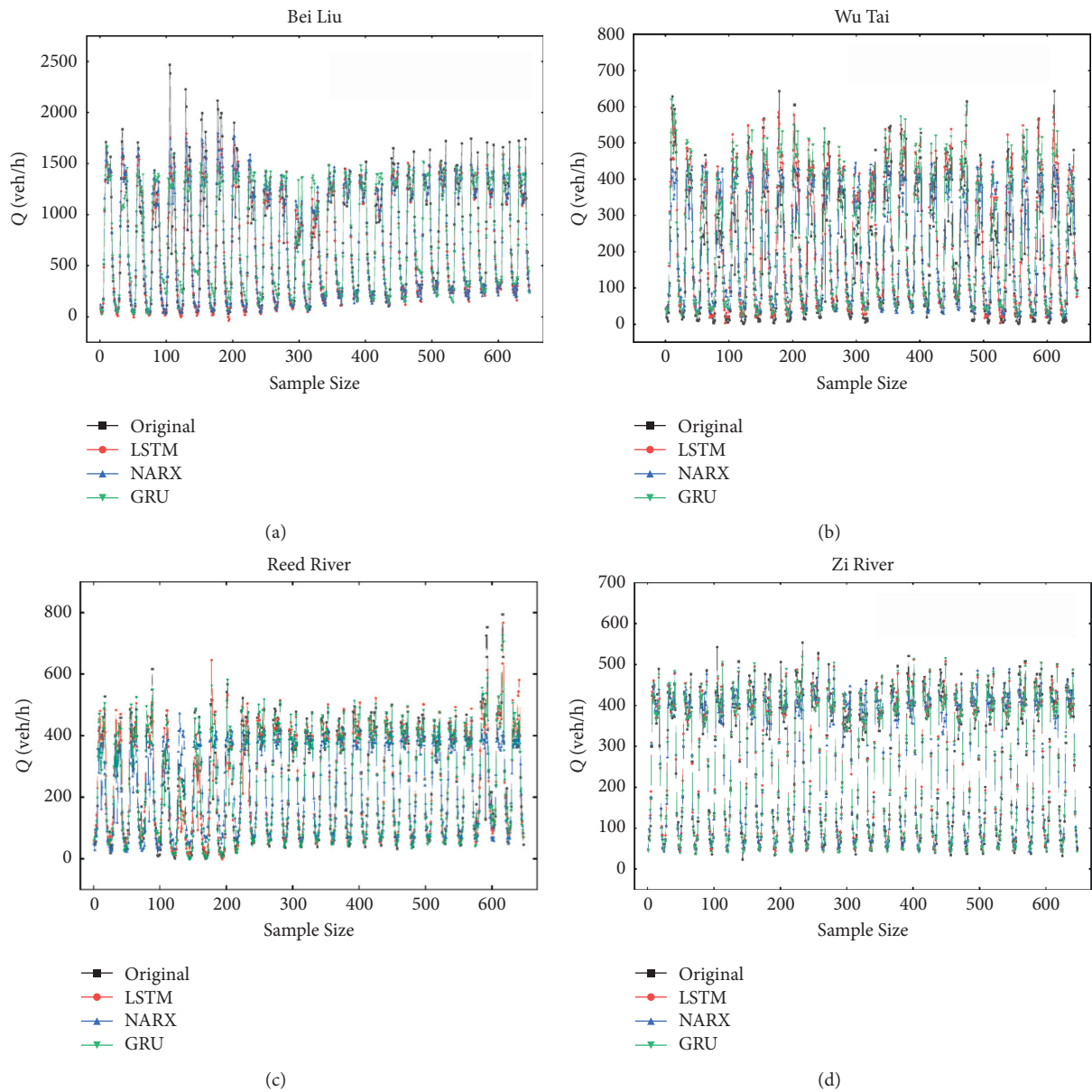


FIGURE 11: Flow prediction curves for each monitoring station under different prediction models.

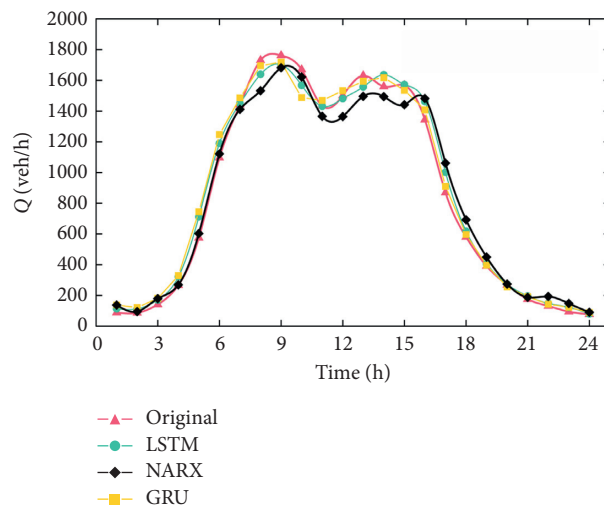


FIGURE 12: Comparison of prediction results.

TABLE 6: Accuracy evaluation of traffic flow forecasting with three models.

	$R^2$ (%)	MAPE (%)	RMSE
NARX NN	99.576	0.374	60.060
LSTM NN	99.579	0.395	65.034
GRU NN	99.518	0.403	69.039

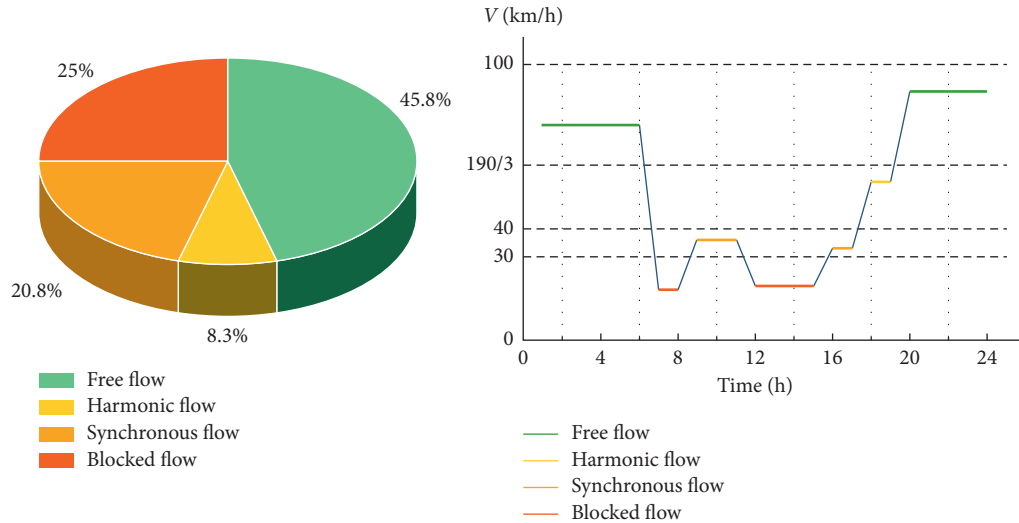


FIGURE 13: Predicted traffic distribution by the time of day.

when entering and leaving the city. Due to a large number of factories in the selected areas, the frequent presence of large vehicles caused the blocked flow from midday until the evening peak.

## 6. Conclusions and Prospects

In this paper, we developed a state prediction model for regional road networks (NARX-MFD) and proposed a traffic state efficiency index formula. The traffic state of the regional road network is classified into four categories by using and analyzing the free flow speed and the design speed of the classified road in the traffic flow parameter curves. This will be used as the evaluation classification of the predicted state. Then, according to the traffic state parameters measured from the four monitoring stations, the MFD of the road network in Linzi District of Zibo City is obtained for the selected period.

Afterward, a comparison experiment of LSTM, GRU, and NARX with the same dataset showed that NARX had slightly better prediction performance than LSTM and GRU and converged fastest and can cope well with long-time traffic data. However, it did not cope well with the sudden phenomenon of traffic changing and the oscillation amplitude was not large. The main reason for this is that only the test set has mutation data, and the model trained using the training set with regular variation can only cope well with such data. Finally, a prediction diagram of the future day's traffic state is obtained with MFD, predicted flows, and corresponding speeds based on the state classification intervals of the regional road network. It realizes the functions

of data quality control, prediction, and visualization of the operation state of the road network. In summary, the road network operation state prediction model can provide a reliable basis for traffic managers' decision-making and provide effective real-time traffic information for travelers, thus reducing travel time and improving travel efficiency.

However, as there are still shortcomings in both macroscopic traffic state evaluation and predicted models in this paper, our future work will be carried out from these two aspects. In the macroscopic traffic evaluation section, the number of class categories affects the calculation of weights, the calculation of capacity, and even the classification of the road network state. And there are only three selected road class categories in the network, which could be further refined. In the prediction model section, firstly, as it is difficult to obtain data and the amount of data is not large enough, considering the scale of data under the practical application of the model, further research is needed to see whether the NARX model can maintain the current prediction effect when dealing with a large dataset. Secondly, trying to combine the NARX model with other models to improve the prediction accuracy of traffic flow is a part of future research.

## Data Availability

The data used to support the results of this study were obtained from the Shandong Provincial Highway Traffic Investigation and Management Institute in Zibo, Shandong Province, China, and are available from the corresponding author upon request.

## Conflicts of Interest

The authors have no conflicts of interest.

## Authors' Contributions

Study conception and design were done by Ziwen Song and Feng Sun; data collection was done by Ziwen Song and Yingcui Du; analysis and interpretation of results were done by Ziwen Song and Chenchen Li; draft manuscript preparation was done by Ziwen Song and Rongji Zhang. All the authors reviewed the results and approved the final version of the manuscript.

## Acknowledgments

This research was supported in part by the Pacific Northwest 13 Transportation Consortium, US Department of Transportation University Transportation Center for 14 Federal Region 10, the Science and Technology Planning Project of the Shandong Province (Grant no. 2016GGB01539) and the Science and Technology Planning Project of the Zibo City (Grant no. 2019ZBXC515).

## References

- [1] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.
- [2] Y. Lin, L. Li, H. Jing, B. Ran, and D. Sun, "Automated traffic incident detection with a smaller dataset based on generative adversarial networks," *Accident Analysis & Prevention*, vol. 144, Article ID 105628, 2020.
- [3] M. Kuwahara, A. Takenouchi, and K. Kawai, "Traffic state estimation by backward moving observers: an application and validation under an incident," *Transportation Research Part C: Emerging Technologies*, vol. 127, Article ID 103158, 2021.
- [4] Y. Zhao, H. Zhang, L. An, and Q. Liu, "Improving the approaches of traffic demand forecasting in the big data era," *Cities*, vol. 82, pp. 19–26, 2018.
- [5] L. Cheng, X. Lai, X. Chen, S. Yang, J. De Vos, and F. Witlox, "Applying an ensemble-based model to travel choice behavior in travel demand forecasting under uncertainties," *Transportation Letters*, vol. 12, no. 6, pp. 375–385, 2020.
- [6] K. Kumar, M. Parida, and V. K. Katiyar, "Short term traffic flow prediction for a non urban highway using artificial neural network," *Procedia - Social and Behavioral Sciences*, vol. 104, pp. 755–764, 2013.
- [7] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transportmetrica: Transportation Science*, vol. 15, no. 2, pp. 1688–1711, 2019.
- [8] X. Dong, T. Lei, S. Jin, and Z. Hou, "Short-term traffic flow prediction based on XGBoost," in *Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 854–859, IEEE, Enshi, China, May 2018.
- [9] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm," *Knowledge-Based Systems*, vol. 172, pp. 1–14, 2019.
- [10] J. Sun, M. Ruo-jin, and Y.-H. Liu, "Study on short-term traffic state forecasting model of SVM considering proportion of large vehicles," *Journal of Highway and Transportation Research and Development*, vol. 35, no. 10, pp. 126–132, 2018.
- [11] Z. Huang, Y. Wang, J. Han, and Z. Wei, "Short-term urban traffic flow prediction based on MLS SVM and spatiotemporal features," *Journal of Hefei University of Technology*, vol. 1, 2020.
- [12] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)," *Transportation Research Record*, vol. 773, 1980.
- [13] M. Tayyab, J. Dauwels, C. Y. Goh et al., "Spatiotemporal patterns in large-scale traffic speed prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 794–804, 2013.
- [14] M. S. Dougherty and M. R. Cobbett, "Short-term inter-urban traffic forecasts using neural networks," *International Journal of Forecasting*, vol. 13, no. 1, pp. 21–31, 1997.
- [15] L. Li, X. Qu, J. Zhang, H. Li, and B. Ran, "Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network," *Applied Soft Computing*, vol. 73, pp. 921–932, 2018.
- [16] X. Xu, Y. Sun, Y. Bai, K. Zhang, Y. Liu, and X. Zhao, "Seq2Img-DRNET: a travel time index prediction algorithm for complex road network at regional level," *Expert Systems with Applications*, vol. 185, Article ID 115554, 2021.
- [17] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002.
- [18] T. Jensen and S. K. Nielsen, "Calibrating a gravity model and estimating its parameters using traffic volume counts," in *Proceedings of the 5th Conference of Universities' Transport Study Groups*, University College, London, UK, January 1973.
- [19] P. Högborg, "Estimation of parameters in models for traffic prediction: a non-linear regression approach," *Transportation Research*, vol. 10, no. 4, pp. 263–265, 1976.
- [20] S. Lee and D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1678, no. 1, pp. 179–188, 1999.
- [21] T. Otsoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, and K. Shiimoto, "Traffic prediction for dynamic traffic engineering," *Computer Networks*, vol. 85, pp. 36–50, 2015.
- [22] A. Bezuglov and G. Comert, "Short-term freeway traffic parameter prediction: application of grey system theory models," *Expert Systems with Applications*, vol. 62, pp. 284–292, 2016.
- [23] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin, "A noise-immune Kalman filter for short-term traffic flow forecasting," *Physica A: Statistical Mechanics and Its Applications*, vol. 536, Article ID 122601, 2019.
- [24] S. Clark, "Traffic prediction using multivariate nonparametric regression," *Journal of Transportation Engineering*, vol. 129, no. 2, pp. 161–168, 2003.
- [25] F. Harrou, A. Zeroual, and Y. Sun, "Traffic congestion monitoring using an improved kNN strategy," *Measurement*, vol. 156, Article ID 107534, 2020.
- [26] Z. Yao, C. Shao, and Y. Gao, "Research on methods of short-term traffic forecasting based on support vector regression [J]," *Journal of Beijing Jiaotong University*, vol. 30, no. 3, pp. 19–22, 2006.
- [27] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, L. D. Han, and 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.
- [28] M.-W. Li, W.-C. Hong, and H.-G. Kang, “Urban traffic flow forecasting using Gauss-SVR with cat mapping, cloud model and PSO hybrid algorithm,” *Neurocomputing*, vol. 99, pp. 230–240, 2013.
- [29] G. Fusco, C. Colombaroni, L. Comelli, and N. Isaenko, “Short-term traffic predictions on large urban traffic networks: applications of network-based machine learning models and dynamic traffic assignment models,” in *Proceedings of the 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pp. 93–101, IEEE, Budapest, Hungary, June 2015.
- [30] J. Park, L. Dai, Y. L. Murphey et al., “Real time vehicle speed prediction using a neural network traffic model,” in *Proceedings of the 2011 International Joint Conference on Neural Networks*, pp. 2991–2996, IEEE, California, CA, USA, July 2011.
- [31] C. Chen, K. Li, S. G. Teo, X. Zou, K. Li, and Z. Zeng, “Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks,” *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 4, pp. 1–23, 2020.
- [32] B. G. Çetiner, M. Sari, and O. Borat, “A neural network based traffic-flow prediction model,” *Mathematical and Computational Applications*, vol. 15, no. 2, pp. 269–278, 2010.
- [33] H. Peng, B. Du, M. Liu et al., “Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning,” *Information Sciences*, vol. 578, pp. 401–416, 2021.
- [34] P. José Maria and G. A. Barreto, “Long-term time series prediction with the NARX network: an empirical evaluation,” *Neurocomputing*, vol. 71, pp. 16–18, 2008.
- [35] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Short-term traffic forecasting: where we are and where we’re going,” *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.
- [36] B. M. Williams, “Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1776, no. 1, pp. 194–200, 2001.
- [37] Y. Kamarianakis and P. Prastacos, “Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches,” *Transportation Research Record*, vol. 1857, pp. 74–84, 2003.
- [38] Y. Kamarianakis and P. Prastacos, “Space-time modeling of traffic flow,” *Computers & Geosciences*, vol. 31, no. 2, pp. 119–133, 2005.
- [39] G. Li, V. L. Knoop, and H. van Lint, “Multistep traffic forecasting by dynamic graph convolution: interpretations of real-time spatial correlations,” *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103185, 2021.
- [40] R. J. Smeed, “The road capacity of city centers,” *Highway Research Record*, vol. 169, 1967.
- [41] R. Herman and I. Prigogine, “A two-fluid approach to town traffic,” *Science (New York, N.Y.)*, vol. 204, no. 4389, pp. 148–151, 1979.
- [42] R. Herman and S. Ardekani, “Characterizing traffic conditions in urban areas,” *Transportation Science*, vol. 18, no. 2, pp. 101–140, 1984.
- [43] H. S. Mahmassani, J. C. Williams, and R. Herman, “Investigation of network-level traffic flow relationships: some simulation results,” *Transportation Research Record*, vol. 971, pp. 121–130, 1984.
- [44] J. W. Godfrey, “The mechanism of a road network,” *Traffic Engineering and Control*, vol. 8, p. 8, 1969.
- [45] C. F. Daganzo, “Urban gridlock: macroscopic modeling and mitigation approaches,” *Transportation Research Part B: Methodological*, vol. 41, no. 1, pp. 49–62, 2007.
- [46] N. Geroliminis and C. F. Daganzo, “Existence of urban-scale macroscopic fundamental diagrams: some experimental findings,” *Transportation Research Part B: Methodological*, vol. 42, no. 9, pp. 759–770, 2008.
- [47] L. C. Edie, *Discussion of Traffic Stream Measurements and Definitions*, Port of New York Authority, New York, NY, USA, 1963.
- [48] K. Aboudolas and N. Geroliminis, “Perimeter and boundary flow control in multi-reservoir heterogeneous networks,” *Transportation Research Part B: Methodological*, vol. 55, pp. 265–281, 2013.
- [49] J. Haddad and Z. Zheng, “Adaptive perimeter control for multi-region accumulation-based models with state delays,” *Transportation Research Part B: Methodological*, vol. 137, pp. 133–153, 2020.
- [50] H. Fu, N. Liu, and G. Hu, “Hierarchical perimeter control with guaranteed stability for dynamically coupled heterogeneous urban traffic,” *Transportation Research Part C: Emerging Technologies*, vol. 83, pp. 18–38, 2017.
- [51] J. Haddad, “Robust constrained control of uncertain macroscopic fundamental diagram networks,” *Transportation Research Procedia*, vol. 7, pp. 669–688, 2015.
- [52] K. Ampountolas, N. Zheng, and N. Geroliminis, “Macroscopic modelling and robust control of bi-modal multi-region urban road networks,” *Transportation Research Part B: Methodological*, vol. 104, pp. 616–637, 2017.
- [53] E. Thonhofer, T. Palau, A. Kuhn, S. Jakubek, and M. Kozek, “Macroscopic traffic model for large scale urban traffic network design,” *Simulation Modelling Practice and Theory*, vol. 80, pp. 32–49, 2018.
- [54] J. Xiao and Z. Wang, “Traffic speed cloud maps: a new method for analyzing macroscopic traffic flow,” *Physica A: Statistical Mechanics and Its Applications*, vol. 508, pp. 367–375, 2018.
- [55] P. Duan, G. Mao, C. Zhang, and S. Wang, “STARIMA-based traffic prediction with time-varying lags,” in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1610–1615, IEEE, Rio de Janeiro, Brazil, November 2016.
- [56] N. Dengen, “Comparison of SARIMA, NARX and BPNN models in forecasting time series data of network traffic,” in *Proceedings of the 2016 2nd International Conference on Science in Information Technology (ICSITech)*, pp. 264–269, IEEE, Balikpapan, Indonesia, October 2016.
- [57] X. Jieni and Z. Shi, “Short-time traffic flow prediction based on chaos time series theory,” *Journal of Transportation Systems Engineering and Information Technology*, vol. 8, no. 5, pp. 68–72, 2008.
- [58] J. Guo, W. Huang, B. M. Williams, and Williams, “Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.
- [59] H. Xie, H. Tang, and Y.-H. Liao, “Time series prediction based on NARX neural networks: an advanced approach,” in

*Proceedings of the 2009 International Conference on Machine Learning and Cybernetics*, pp. 1275–1279, IEEE, Baoding, China, July 2009.

- [60] X. Zeng and Y. Zhang, “Development of recurrent neural network considering temporal-spatial input dynamics for freeway travel time modeling,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 28, no. 5, pp. 359–371, 2013.
- [61] K. Cho, B. Van Merriënboer, C. Gulcehre et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” 2014, <https://arxiv.org/abs/1406.1078>.



## Research Article

# Multichannel Speech Enhancement in Vehicle Environment Based on Interchannel Attention Mechanism

Xueli Shen <sup>1,2</sup>, Zhenxing Liang <sup>2</sup>, Shiyin Li <sup>1</sup> and Yanji Jiang <sup>2,3</sup>

<sup>1</sup>School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China

<sup>2</sup>School of Software, Liaoning Technical University, Huludao 125105, China

<sup>3</sup>Suzhou Automotive Research Institute, Tsinghua University, Suzhou 215100, China

Correspondence should be addressed to Yanji Jiang; [jjvip@126.com](mailto:jjvip@126.com)

Received 6 August 2021; Revised 27 September 2021; Accepted 8 October 2021; Published 15 November 2021

Academic Editor: Geqi Qi

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

Speech enhancement in a vehicle environment remains a challenging task for the complex noise. The paper presents a feature extraction method that we use interchannel attention mechanism frame by frame for learning spatial features directly from the multichannel speech waveforms. The spatial features of the individual signals learned through the proposed method are provided as an input so that the two-stage BiLSTM network is trained to perform adaptive spatial filtering as time-domain filters spanning signal channels. The two-stage BiLSTM network is capable of local and global features extracting and reaches competitive results. Using scenarios and data based on car cockpit simulations, in contrast to other methods that extract the feature from multichannel data, the results show the proposed method has a significant performance in terms of all SDR, SI-SNR, PESQ, and STOI.

## 1. Introduction

In the process of driving, the speech signals recorded by a microphone are often corrupted by reverberation and background noise, such as wind noise, engine noise, and tire noise, leading to considerable degradation in speech quality, particularly at low signal-to-noise ratios (SNRs) [1]. Speech enhancement technology can improve the speech quality of the interphone system and the ability of the speech recognition system. Multichannel enhancement in vehicle scenarios uses microphone arrays that are convenient and flexible for speech-enabled applications [2]. The multichannel structure could provide more spatial information from the interchannel data and better results than the signal channel.

Although the technology of microphone array has been developed for a long time, multichannel speech enhancement is still a great challenge in the field of speech recognition. The methods can be divided into two categories: one is based on the frequency domain, and the other is based on the time domain. Researchers mostly use the frequency-domain methods, which are based on the short-time spectrum estimation. Chakrabarty and Habets [3] proposed

a multichannel online speech enhancement method based on time-frequency masking. Convolutional recurrent neural network (CRNN) is used to estimate the mask, and the effects of the ideal ratio mask (IRM) and ideal binary mask (IBM) on the results are discussed. The results show that the method is robust to different angles of sound sources. In [4], a multichannel speech enhancement system based on a deep neural network is proposed. Firstly, the audio signal is transformed into the frequency domain by STFT, the time-frequency mask is estimated by DNN, and the multichannel Wiener filtering is performed by using the power spectral density of speech and noise. The experimental results show that the method is effective. A beamforming method different from the traditional DNN is proposed in [5]. The spectrum of each channel is mapped to the non-Euclidean space, usually using the phase information to improve real-time performance, and the graph neural network is used for end-to-end training. Compared with the existing methods, the experiment result is better. A time-domain beamforming method named FaSNet (Filter and Sum Network) suitable for the low delay is proposed in [6]. The author selects the reference channel for filtering, calculates the filter of other channels by the reference channel, and then adds the filtered

speech of each channel as the denoised speech. The model size of the algorithm is small, and the performance is better than that of traditional beamforming methods. In [7], a streaming speech enhancement system is proposed, which adopts the Wave-U-Net framework, adds temporal convolution and attention mechanism into the encoding and decoding structure, and explores the history caching mechanism. This method achieves almost the same noise reduction effect as the nonstreaming model. The time-domain convolutional denoising autoencoders (TCDAEs) method is proposed in [8]. It is used to learn the mapping structure between noisy speech waveform and clean speech waveform and solve the problem of speech signal delay between different channels effectively. Compared with the traditional denoising autoencoder, the effect has been significantly improved.

The multichannel speech enhancement model has the most significant advantage of obtaining abundant information between channels compared with the single channel. Therefore, for the multichannel model, the way that extracts the spatial features between channels more effectively becomes the key to achieving better performance. In [9], a multichannel convolution sum (MCS) is used to extract features between channels. On the contrary, in [9], inspired by the IPD [10] feature, the interchannel convolution feature (ICD) is proposed. The method is to perform one-dimensional convolution subtraction on a pair of microphones. Based on GCC-PHAT, [6, 11] considered the normalized cross-correlation (NCC) method, which uses cosine similarity to calculate the information between channels. All the above methods achieve better performance improvement for multichannel speech enhancement. To address the problem of speech enhancement in the car cockpit, this paper proposes a novel method based on interchannel attention mechanism frame by frame (IAF), which helps analyse the influence of each channel on speech signal by using the characteristic information of the channel. Moreover, the proposed method also explores interchannel relationships and achieves more information representation on channel structure. It provides a new idea for multichannel speech enhancement based on vehicle environment and can also be applied to smart homes, teleconference, and other scenes.

The main contents of this paper are as follows: Section 1 introduces the related research work in this field. The structure of the multichannel speech enhancement model based on IAF is proposed in Section 2. The algorithm performance in the vehicle environment is evaluated in Section 3. The experimental results of the algorithm on several microphone arrays are analysed and discussed in Section 4, and Section 5 draws the conclusion and points out the future of the research work.

## 2. Problem Formulation

The proposed method aims to obtain an accurate estimate of the features for all the channels of a single time frame, given the input feature representation of the corresponding frame. The multichannel speech enhancement process of vehicle data is divided into four successive steps. First, spatial features

from multichannel data added context information is extracted by IAF. Then, the frame-level beamforming filters are estimated by a well-trained two-stage BiLSTM model using spatial features, and the original waveforms computed by 1-dimensional convolution, for  $N (N > 2)$  microphones,  $N$  beamforming filters are estimated. Next, the filters are adopted to filter the noisy speech in every channel, thereby obtaining the beamformed speech. Finally, add the beamformed speech as the denoised speech. The detail is presented in the following sections. A block diagram of the proposed multichannel enhancement framework is shown in Figure 1.

**2.1. Data Preprocessing.** It is assumed that the input signal corresponding to each microphone is represented as (2). Here, the frame length is  $M$ , the frameshift is  $K \in [0, M - 1]$ , the total length of the speech signal is  $l$ , and the total number of frames is  $Z$ :

$$Z = \frac{l}{K} + 2, \quad (1)$$

$$x_t^i = x^i[tK : tK + M - 1], \quad t \in [0, Z], \quad i = 1, \dots, N, \quad (2)$$

where  $t$  is the frame index value and  $i$  is the channel index.  $x_t^i \in \mathbb{R}^{1 \times M}$  indicates that the signal vector of frame  $t$  is collected by microphone  $i$ .

Due to the different distance between each microphone and the sound source, there is a time delay between the signals received by each microphone. Add a context window to make sure the model can capture interchannel delays of signal samples [12]. We add a group of contextual speech information in  $x_t^i$  and define it as  $\bar{x}_t^i$ :

$$\bar{x}_t^i = x^i[tK - W : tK + W + M - 1], \quad (3)$$

where  $W$  is the size of the context window and  $\bar{x}_t^i \in \mathbb{R}^{2W+M}$  is the signal vector of the microphone  $i$  containing the context information at frame  $t$ . The input sequence to these networks consists of  $W$  past and  $W$  future time frame.

**2.2. Interchannel Attention Mechanism Frame by Frame.** We calculate the corresponding weights of different parts of the channel by constructing the score function to describe the transmission characteristics of the signal in the channel. The principle of interchannel attention mechanism frame by frame is shown in Figure 2.

In order to extend the context information  $\bar{x}_t^i$ , firstly, average pooling is performed in the frame length dimension:

$$F_a = z_t^i = \frac{1}{2W + M} \sum_{j=tK-W}^{tK+W+M-1} \bar{x}^i[j], \quad i = 1, \dots, N. \quad (4)$$

$z_t^i \in \mathbb{R}^1$  is the average value of microphone  $i$  at the number of frames  $t$ . Then, the results are input into multiple fully connected layers:

$$F_b = G_t = S(P([z_t^1, z_t^2, \dots, z_t^N])), \quad G_t \in [0, 1]. \quad (5)$$

$G_t \in \mathbb{R}^{1 \times N}$  is the microphone array feature at the frame  $t$ .  $P(*)$  is a set of fully connected layers with parameter

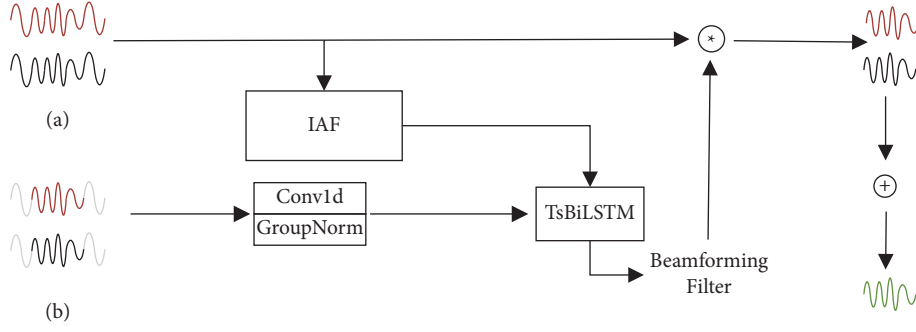


FIGURE 1: The overall process of multichannel speech enhancement model based on IAF. (a) The speech signal with context. (b) The original speech signal. “IAF” and “TsBiLSTM” denote the feature extraction and feature filter using the TsBiLSTM model, respectively.

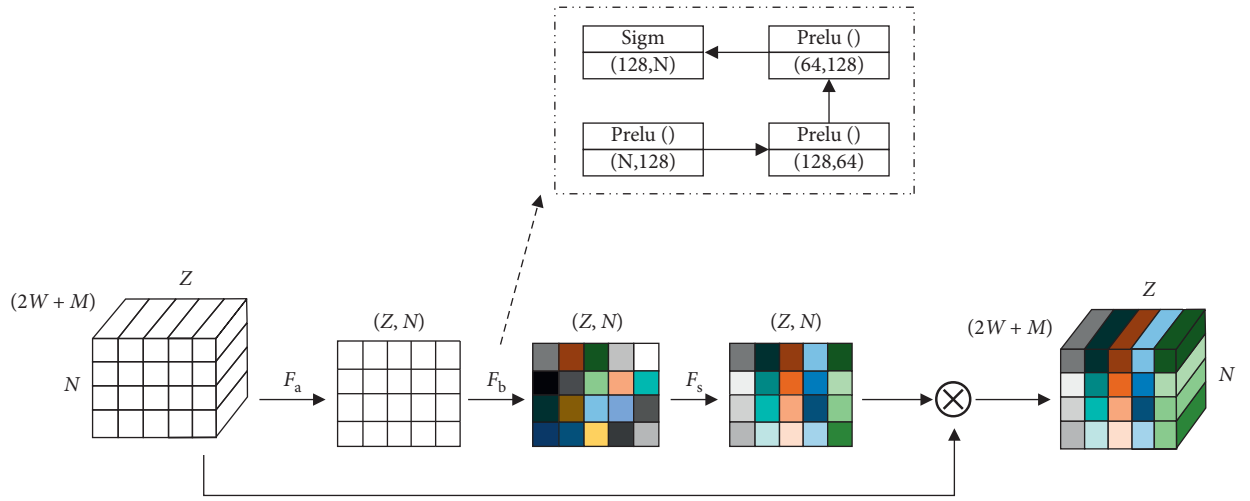


FIGURE 2: The module of the frame-level interchannel attention mechanism: different colors represent different weight values, and multiply the weight values with the original data.

modified linear unit (PReLU) activation function,  $S(\cdot)$  is a set of fully connected layers with sigmoid activation function, and the output of  $P(\cdot)$  and  $S(\cdot)$  are  $[128, 64, 128]$  and  $[N]$ , respectively. Then, input  $G_t$  into the softmax activation function:

$$F_s = \bar{F}_t = \text{Softmax}(G_t), \quad (6)$$

where  $\bar{F}_t$  is a vector whose sum is 1. The final output  $out$  is obtained by multiplying with  $\bar{F}_t$  and  $x_t$ :

$$out_t^i = \bar{F}_t^i \times x_t^i, \quad i = 1, \dots, N. \quad (7)$$

$out_t^i \in \mathbb{R}^{1 \times (2W+M)}$  presents the speech feature sequence of the  $t$ -th frame data in the  $i$ -th channel.

By using the attention mechanism frame by frame of the speech signal in multiple channels, the model could learn the characteristics of each channel and capture spatial features between channels more accurately.

**2.3. Two-Stage BiLSTM Network.** The two-stage bidirectional LSTM (TsBiLSTM) is used to derive a beamformer as BiLSTM is adopted to estimate the global feature. For the beamformer, the approach aims to improve the SNR without destroying the target speech.

Figure 3 shows the TsBiLSTM architecture employed in this work. We divide the data into blocks, consider using the BiLSTM network model to obtain local and global features of the blocks and establish the timing relationship of the signal, and use the residual connections to alleviate the gradient dispersion problem.

In this work, we combine the speech signal with context information in the first stage. The observed signal can be expressed as follows:

$$y_t = \text{GroupNorm}(\text{Conv1 } d(x_t)), \quad (8)$$

$$xb_t = \text{concat}([out_t, y_t]). \quad (9)$$

$xb_t \in \mathbb{R}^{N \times 2(M+W)}$  represents the speech features of frame  $t$  and  $xb \in \mathbb{R}^{Z \times N \times 2(M+W)}$  represents all the speech features; then, we do the one-dimensional convolution on  $xb$ :

$$c = \text{Conv1 } d(xb), \quad (10)$$

where  $c \in \mathbb{R}^{Z \times (N \times M)}$ , then divide  $c$  into  $S$  blocks of the same size. Each block presents  $B_s \in \mathbb{R}^{U \times (N \times M)}$ ,  $s \in [1, S]$ , and all the blocks will be connected to form a four-dimensional vector  $O \in \mathbb{R}^{N \times M \times S \times U}$ .

We transform the shape  $O \in \mathbb{R}^{N \times M \times S \times U}$  to  $O \in \mathbb{R}^{(S \times N) \times U \times M}$  and input the first BiLSTM:

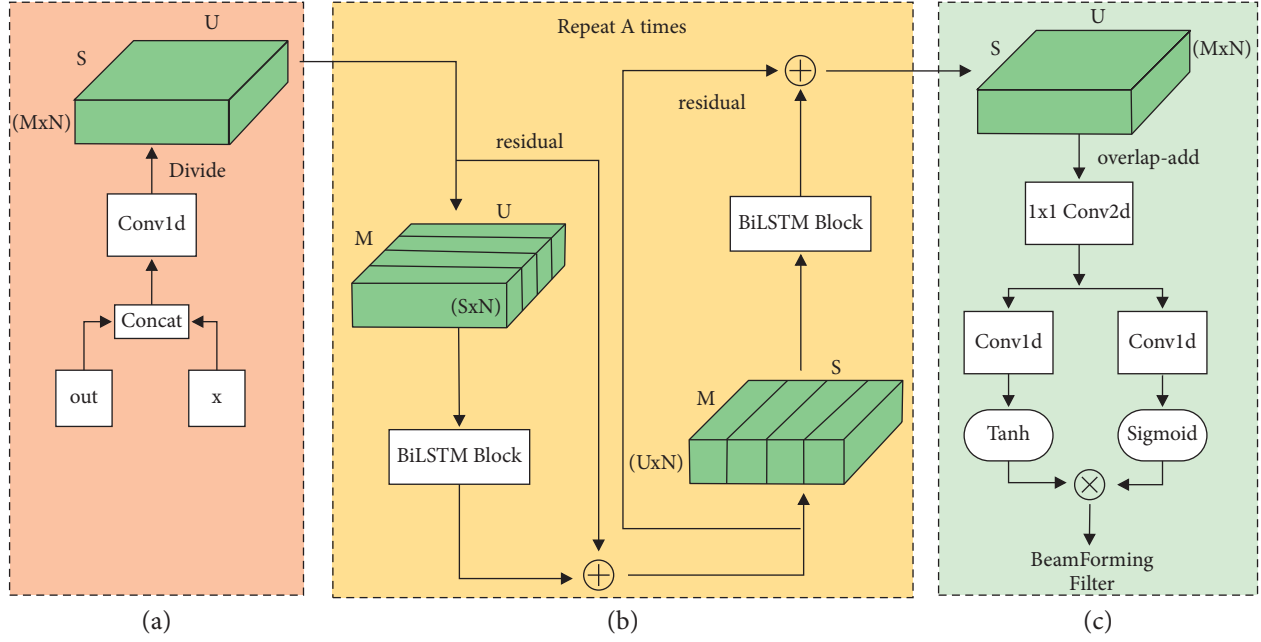


FIGURE 3: The structure of the TsBiLSTM module and illustration of the proposed module architecture. (a) The input of the TsBiLSTM includes interchannel features and original waveform. (b) The processing chain shows the two-stage BiLSTM with residual connections. (c) Carry out  $1 \times 1$  convolution on the output after overlap-add method operation, then operate conv1d layer with sigmoid and tanh activation function, respectively, and obtain beamforming filter.

$$\begin{aligned} \text{out} &= \text{reshape}(O), \\ \text{out}_1 &= \text{GroupNorm}(\text{Linear}(\text{BiLSTM}(\text{out}))), \\ \text{out} &= \text{reshape}(\text{out}_1) + O. \end{aligned} \quad (11)$$

The output of BiLSTM passes through the linear layer and GroupNorm operation and then output  $\text{out}_1 \in \mathbb{R}^{(S \times N) \times U \times M}$ . Reshape  $\text{out}_1 \in \mathbb{R}^{(S \times N) \times U \times M}$  to  $\text{out}_1 \in \mathbb{R}^{N \times M \times S \times U}$ , add the vector using the residual connection to reduce the problem of gradient disappearance or gradient explosion, and finally obtain  $\text{out} \in \mathbb{R}^{N \times M \times S \times U}$ .

In the next stage, change  $\text{out} \in \mathbb{R}^{N \times M \times S \times U}$  into  $\text{out} \in \mathbb{R}^{(U \times N) \times S \times M}$ , then input the next BiLSTM, as the first BiLSTM block, and finally, obtain  $\text{out} \in \mathbb{R}^{N \times M \times S \times U}$ . Because the signals are transmitted to the BiLSTM model in different block forms, we can obtain the local and global features of the signals, respectively:

$$\begin{aligned} \text{out}_1 &= \text{reshape}(\text{out}), \\ \text{out}_2 &= \text{GroupNorm}(\text{Linear}(\text{BiLSTM}(\text{out}_1))), \\ \text{out} &= \text{reshape}(\text{out}_2) + \text{out}. \end{aligned} \quad (12)$$

Then, use the overlap-add operation to convert the segmented block back to the original sequence:

$$\text{out}_3 = O D(\text{out}), \quad (13)$$

where  $\text{out}_3 \in \mathbb{R}^{Z \times (N \times M)}$ ,  $O D(\cdot)$  is the overlap-add method, which means to restore the partitioned data. Then, convolve  $\text{out}_3 \in \mathbb{R}^{Z \times (N \times M)}$  in two dimensions with the convolution kernel of size set one:

$$\text{out}_4 = \text{Conv2 } d(\text{out}_3), \quad (14)$$

where  $\text{out}_4 \in \mathbb{R}^{Z \times (N \times M)}$ . We perform twice one-dimensional convolution operations on  $\text{out}_4$ , then use the activation function of Tanh and Sigmoid, respectively, and multiply the results to get the filters for each channel:

$$h = \text{Tanh}(\text{Conv1 } d(\text{out}_4)) \otimes \text{Sigmoid}(\text{Conv1 } d(\text{out}_4)), \quad (15)$$

where  $h \in \mathbb{R}^{N \times Z \times (2W+1)}$ ,  $\otimes$  is the Hadamard product symbol,  $\text{Tanh}(\cdot) \otimes \text{Sigmoid}(\cdot)$  is the gating mechanism of filter that controls the output data.

Figure 4 shows the structure of the BiLSTM block. The input layer is the feature vector of noisy speech with dimension 64, which is input into the BiLSTM layer with dimension 128. The output dimension is 256 since bidirectional LSTM is used. Then, input the linear hidden layer of 64, and get the output after the GroupNorm operation.

**2.4. Summation.** Integrating the signals of multiple channels into one signal output is an important step in the multi-channel speech enhancement problem. After passing the signals of each channel through the channel filter, the results obtained are summed and averaged, that is, the final enhanced speech signal:

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N h^i \otimes \bar{x}^i, \quad i = 1, \dots, N, \quad (16)$$

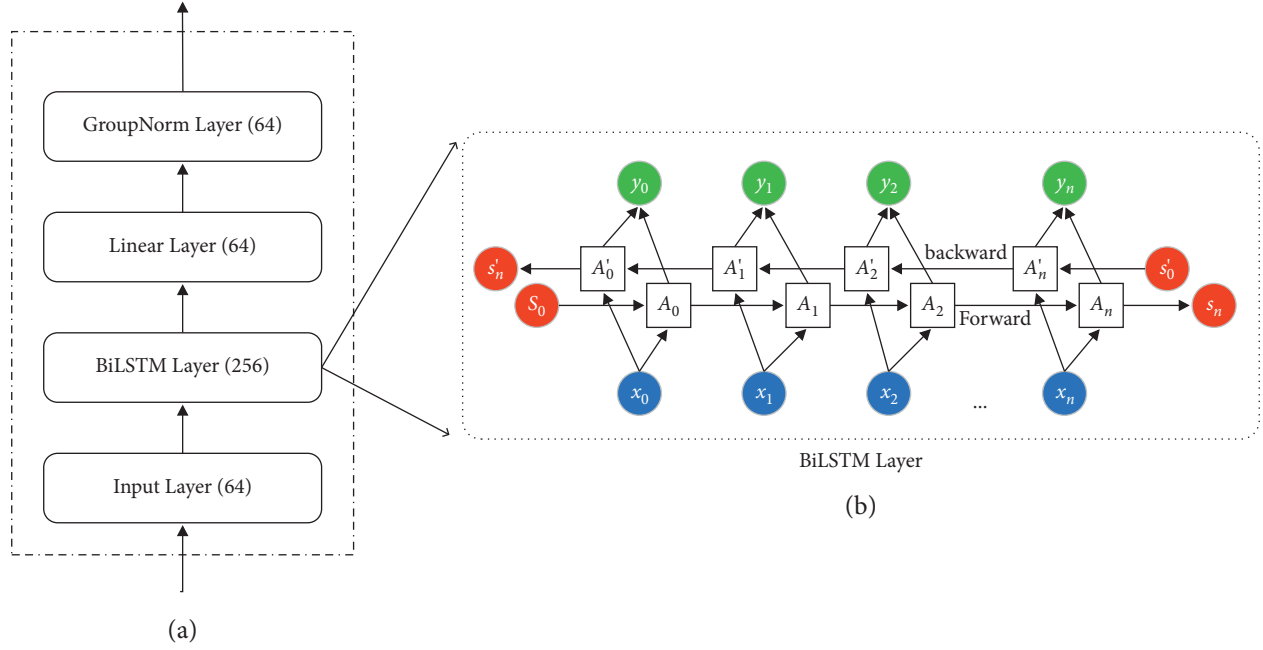


FIGURE 4: The structure of the BiLSTM block. (a) The module contains BiLSTM, linear, and GroupNorm layers, where the numbers in parentheses indicate the size of the output dimension. (b) The internal structure diagram of the BiLSTM layer, where A refers to the LSTM module,  $x$  is the input data,  $s$  is the output of the hidden layer, and  $y$  is the output result BiLSTM.

where  $\bar{y} \in \mathbb{R}^{N \times Z \times M}$  and  $\otimes$  is convolution operation. Ultimately,  $\bar{y}$  is inverted from segmentation into an enhanced speech waveform by overlapping.

**2.5. Loss Function.** In training and evaluation, the scale-invariant source-to-noise ratio (SI-SNR) is used as the loss function.

$$s_{\text{target}} = \frac{\langle \bar{x}, x \rangle x}{\|x\|^2},$$

$$e_{\text{noise}} = \bar{x} - s_{\text{target}},$$

$$SI - \text{SNR} = 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{noise}}\|^2},$$
(17)

where  $\bar{x}$  is the denoised speech and  $x$  is pure speech signal.

### 3. Experiment Section

The speech enhancement tasks are evaluated in four kinds of microphone array structures to simulate the location of the microphone in the car. The speech source and locations of the noise source are shown in Figure 5, where the black circle represents the microphones, the green square represents the speech source, and the red five-pointed star represents the noise source. The design of the microphone array is as follows:

- (i) Consider a uniform linear array with 2 microphones with intermicrophone distance of 3 cm, and the microphone array is located in the front of the car cockpit, as Figure 5(a)

- (ii) Consider a uniform linear array with 2-uniform linear distributed 2-channel microphone array with intermicrophone distance of 3 cm, and the microphone array is located in the front and middle of the car cockpit, respectively, as shown in Figure 5(b)
- (iii) Consider a uniform linear array with 4 microphones with intermicrophone distance of 3 cm, and the microphone array is located in the front of the car cockpit, as shown in Figure 5(c)
- (iv) Consider a distributed array with 4 microphones with intermicrophone distance of 80 cm, and the microphone is located around the car cockpit, as shown in Figure 5(d)

Different microphone array structures can reflect different spatial characteristics. In order to make the method independent of the spatial position of the required speech source, each microphone array position and source-array distance are considered under the training condition.

**3.1. Datasets Building.** For training, we used 3000 randomly chosen speech utterances from the LibriSpeech [13] dataset which are open and well-studied dataset used for speech enhancement, each 4 s long, with sampling frequency of 16 kHz, and 500 were used as a validation set. Volvo car noise [14] was added to the training data as noisy speech in the car cockpit with randomly chosen SNRs between  $-10$  dB and  $-5$  dB. Additionally, since the number of noise is small, spsquare noise [15] as a noise source, with randomly chosen SNRs between  $-10$  dB and  $-5$  dB, was also added. All dataset are divided into frames with 64 sampling points length, 50% overlapping, and the context window is 256.

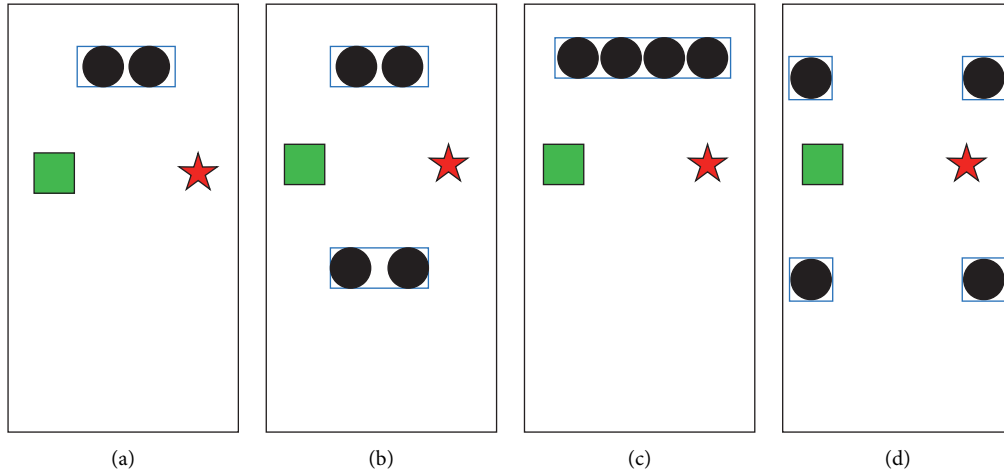


FIGURE 5: The distribution map of four microphone arrays. (a) Uniform linear 2-channel microphone array. (b) Two-uniform linear distributed 2-channel microphone array. (c) Uniform linear 4-channel microphone array. (d) Distributed 4-channel microphone array.

To simulate a car cockpit, we designed the space size to be 3.4 meters long, 1.8 meters wide, and 1.4 meters high. The vehicle cockpits impulse responses required to simulate real acoustic conditions are generated by `gpuRIR` toolbox [16], with the reverberation time ( $T_{60}$ ) selected from 0.1 seconds to 0.3 seconds randomly.

**3.2. Experiment Settings.** The experiment aims to verify the generalization capability of the proposed method over different microphone arrays and compare the performance to that of traditional beamformers. For a fair comparison, we make the comparison as all models, including NCC, MCS, and ICD, are based on the same two-stage BiLSTM modules presented in Section 2.3 for each microphone channel. The architecture of each BiLSTM network consists of 128 hidden layers. Set layer number 4. For MCS and ICD, the size of the convolution kernel is 64, the number of convolution kernels is 16, the step size is 2, and the expansion number is 2, leading to the output where the filter estimate for each microphone is obtained.

The BiLSTM network was trained using the Adam-based optimizer, with minibatches of 128 input signals and a learning rate of 0.001. Meanwhile, the L2 norm of 5 is used for gradient pruning to prevent gradient explosion. During training, if the loss value of the latest 10 epoch model does not decrease on the validation sets, the training will stop automatically. Dynamic strategy warmup [17] is used to adjust the learning rate during the training. This operation can warm up the model at a small learning rate in the initial stage to increase the stability of the model and then gradually reduce it with a decay rate of 0.98 every 2 epochs. The specific approach is similar to [18]. All the implementations were done in PyTorch:

$$\begin{aligned} lr &= a_1 \cdot n \cdot d_{\text{model}}^{-0.5} \cdot n_{\text{warmups}}^{-1.5}, n \leq n_{\text{warmups}}, \\ lr &= a_2 \cdot 0.98^{\lfloor \text{epoch}/2 \rfloor}, n > n_{\text{warmups}}, \end{aligned} \quad (18)$$

where  $n$  is the number of training steps and  $a_1$ ,  $a_2$ ,  $n_{\text{warmups}}$ , and  $d_{\text{model}}$  present the hyperparameter. In the experiment, we set  $a_1 = 0.2$ ,  $a_2 = 1e^{-3}$ ,  $n_{\text{warmups}} = 4000$ , and  $d_{\text{model}} = 64$ .

## 4. Results and Discussion

Following the common speech enhancement metrics, we adopt average SI-SNR, SDR, PESQ, and STOI improvement to evaluate the performance of multichannel speech enhancement. For a more comprehensive evaluation of the speech quality, we also report the performances under different SNRs of speech and noise to give a more comprehensive model assessment. The experimental results are summarized in Table 1, where the highlighted numbers with black are the best scores for each model. The results indicate that the performance of proposed method is better than other methods when tested at different SNRs, which verifies the effectiveness of the model. By assigning weight values to each channel frame by frame, using attention mechanism to learn the feature expression between channels, the proposed method leads to the best improvement in terms of four metrics. It learns from the magnitude spectrum and phase spectrum of the individual microphone signals and exploits the difference in the spatial characteristics of the speech and noise sources.

In the four kinds of microphone array structures designed in the experiment, we obtain 13.60 dB improvement in SI-SNR in the structure of 2 microphones with  $-10$  dB SNR and 14.76 dB improvement in SI-SNR in the distributed 4-microphone array structure.

Another conclusion from the experimental results is that the array structure with four microphones is better than that with the two microphones, indicating that the more the channels are, the more the feature information can be provided to the speech enhancement model. In addition, compared with the other structures, the 4-channel distributed microphone array has the optimum performance. The SDR increase [15.24, 13.87], respectively, in SNR =  $-10$  dB and  $-5$  dB, and the performance improvement of the other

TABLE 1: Evaluation results of our proposed model compared with other methods on the same dataset. Four metrics and two SNRs are considered.

Method	# Mics	SDR		SI-SNR		PESQ		STOI	
		-10 dB	-5 dB	-10 dB	-5 dB	-10 dB	-5 dB	-10 dB	-5 dB
Noise	2 linear	-6.72	0.34	-7.19	0.24	1.07	1.13	0.40	0.64
	4 linear	-5.93	0.11	-6.40	0.02	1.06	1.11	0.40	0.64
	2 × 2 dB	-6.72	0.32	-6.18	0.22	1.07	1.13	0.40	0.64
	4 dB	-6.89	0.68	-7.17	0.60	1.07	1.13	0.46	0.70
+NCC	2 linear	7.45	11.98	6.42	11.30	1.20	1.76	0.73	0.87
	4 linear	8.89	12.44	7.86	11.75	1.28	<b>1.83</b>	<b>0.75</b>	<b>0.88</b>
	2 × 2 dB	7.63	13.14	7.50	12.37	1.23	1.85	0.71	0.89
	4 dB	8.38	14.58	7.60	13.96	1.30	2.02	<b>0.76</b>	<b>0.91</b>
+ICD	2 linear	7.42	11.95	6.39	11.27	1.19	1.75	0.72	0.86
	4 linear	8.80	12.38	7.79	11.68	1.27	1.82	0.74	0.87
	2 × 2 dB	7.55	13.06	7.44	12.30	1.22	1.83	0.71	0.89
	4 dB	8.33	14.51	7.56	13.93	1.27	2.01	<b>0.76</b>	<b>0.91</b>
+MCS	2 linear	7.40	11.92	6.36	11.25	1.19	1.75	0.72	0.86
	4 linear	8.83	12.40	7.83	11.72	1.27	1.82	0.74	0.87
	2 × 2 dB	7.52	13.04	7.42	12.28	1.21	1.82	0.70	0.88
	4 dB	8.28	14.46	7.52	13.79	1.25	2.00	0.75	0.90
Proposed	2 linear	<b>7.52</b>	<b>12.06</b>	<b>6.48</b>	<b>11.36</b>	<b>1.22</b>	<b>1.77</b>	<b>0.74</b>	<b>0.88</b>
	4 linear	<b>8.94</b>	<b>12.49</b>	<b>7.91</b>	<b>11.83</b>	<b>1.29</b>	<b>1.83</b>	<b>0.75</b>	<b>0.88</b>
	2 × 2 dB	<b>7.72</b>	<b>13.23</b>	<b>7.58</b>	<b>12.45</b>	<b>1.24</b>	<b>1.86</b>	<b>0.72</b>	<b>0.90</b>
	4 dB	<b>8.41</b>	<b>14.66</b>	<b>7.66</b>	<b>14.07</b>	<b>1.33</b>	<b>2.04</b>	<b>0.76</b>	<b>0.91</b>

The best evaluation results are shown in bold, comparing the results of the four speech enhancement methods used in the four microphone arrays.

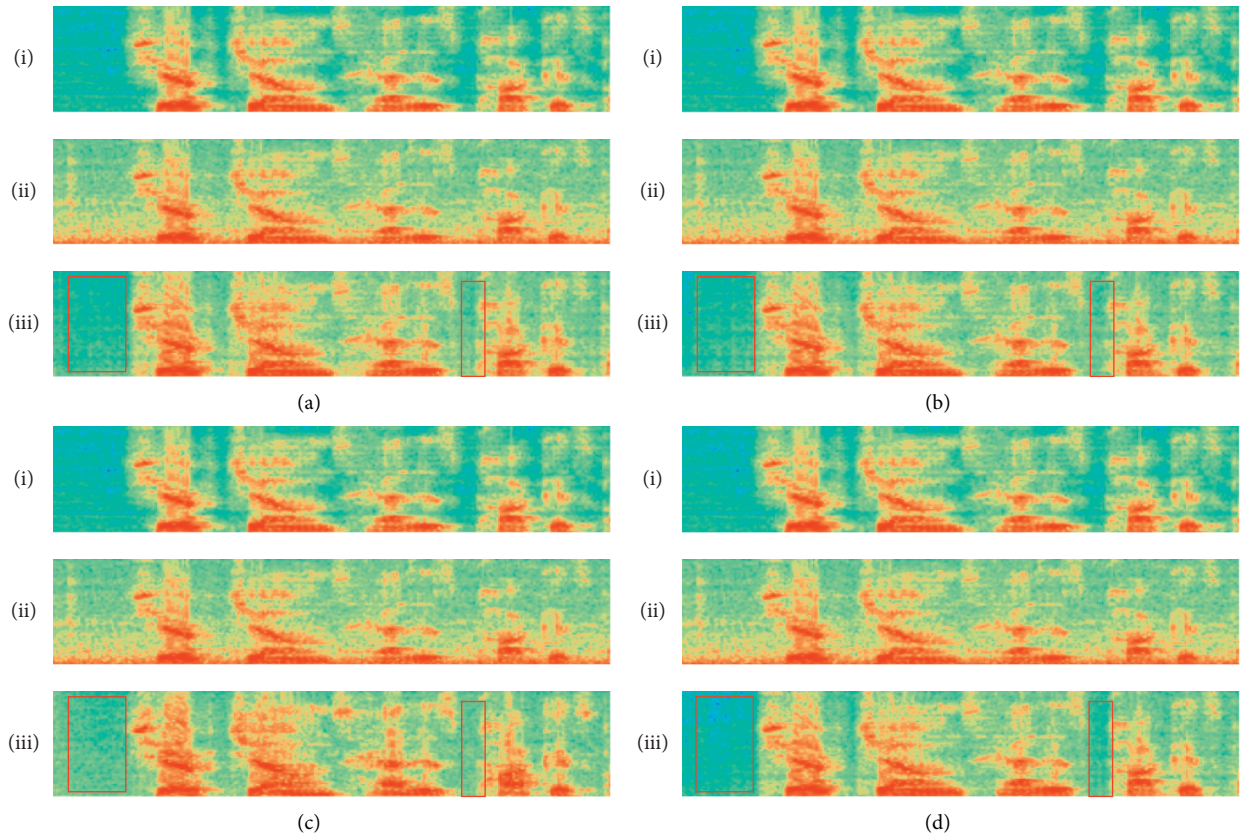


FIGURE 6: The spectrum of four methods under distributed 4-channel microphone array. (a) NCC method. (b) ICD method. (c) MCS method. (d) IAF method, where (i) represents clear speech, (ii) represents noisy speech with SNR = -10 dB, and (iii) represents denoised speech.

structures are [14.16, 11.63], [14.79, 12.31], [14.32, 12.79], respectively. The distributed microphone array structure has advantages in obtaining the spatial characteristics of the entire cockpit due to the difference in the location of the speech source and the noise source, which is helpful to train a better beamforming filter.

Figure 6 is the speech spectrogram, including the pure speech spectrogram, the noisy speech spectrogram with SNR = -10 dB, and the speech spectrogram enhanced by four methods. The four methods have good noise reduction effects. Compared with the enhanced noise energy spectrum in the box, the method proposed in this paper has significant advantages. At the same time, compared with the enhanced speech spectrogram and pure speech spectrogram, the method did not cause speech damage and ensured the integrity of the speech signal.

## 5. Conclusions

This work proposed an interchannel attention mechanism frame by frame (IAF) method and jointed with the two-stage BiLSTM network to learn the spatial features directly from multichannel waveforms to solve the problem of multichannel speech enhancement in the car cockpit. Experimental results show the IAF method is more effective than the traditional NCC, MCS, and ICD method in learning spatial features directly from the multichannel speech waveforms. The proposed model based on four distributed microphone arrays obtains the optimal enhancement performance in terms of SDR, SI-SNR, STOI, and PESQ. The results indicated that the method is suitable for different structures of the microphone array and has good robustness. This work provided valuable conclusions for improving the performance of multichannel speech enhancement in the vehicle cockpit. In future work, we will explore the effect of the position of the voice source on the performance using the proposed method.

## Data Availability

In order to facilitate the further research of other researchers, the LibriSpeech data in this article can be found at <http://www.openslr.org/12/>, the Volvo car noise data can be found at <http://spib.linse.ufsc.br/noise.html>, and the spsquare noise data can be found at <https://zenodo.org/record/1227121#.YP0sjo4zZhG>.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This work was supported by the fund project of Education Department of Liaoning Province (nos. LJKZ0338 and LJ2020FWL001) and the Undergraduate Innovation and Entrepreneurship Training Project (no. 202110147019).

## References

- [1] P. Lei, M. Chen, and J. Wang, "Speech enhancement for in-vehicle voice control systems using wavelet analysis and blind source separation," *IET Intelligent Transport Systems*, vol. 13, no. 4, pp. 693–702, 2019.
- [2] M. Vollrath, "Speech and driving - solution or problem?" *IET Intelligent Transport Systems*, vol. 1, no. 2, pp. 89–94, 2007.
- [3] S. Chakrabarty and E. A. P. Habets, "Time-frequency masking based online multi-channel speech enhancement with convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 787–799, 2019.
- [4] Y. Masuyama, M. Togami, and T. Komatsu, "Consistency-aware multi-channel speech enhancement using deep neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 821–825, Barcelona, Spain, May 2020.
- [5] P. Tzirakis, A. Kumar, and J. Donley, "Multi-channel speech enhancement using graph neural networks," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Ontario, Canada, June 2021, <http://arxiv.org/abs/2102.06934>.
- [6] Y. Luo, E. Ceolini, C. Han, C. Enea, and S.-C. Liu, "FaSNet: low-latency adaptive beamforming for multi-microphone audio processing," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 260–267, Singapore, December 2019.
- [7] V. Kuzmin, F. Kravchenko, A. Sokolov, and G. Jie, "Real-time streaming wave-u-net with temporal convolutions for multichannel speech enhancement," *Signal Processing*, <https://arxiv.org/abs/2104.01923>, 2021.
- [8] N. Tawara, T. Kobayashi, and T. Ogawa, "multi-channel speech enhancement using time-domain convolutional denoising autoencoder," in *Proceedings of the 20th Annual Conference of the International Speech Communication Association: Crossroads of Speech and Language, INTERSPEECH 2019*, pp. 86–90, Graz, Austria, September 2019.
- [9] R. Gu, S. X. Zhang, L. Chen et al., "Enhancing end-to-end multi-channel speech separation via spatial feature learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7319–7323, Barcelona, Spain, May 2020.
- [10] S. Araki, T. Hayashi, M. Delcroix, F. Masakiyo, T. Kazuya, and N. Tomohiro, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 116–120, South Brisbane, QLD, Australia, April 2015.
- [11] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, & Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.
- [12] M. S. Brandstein and H. F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 375–378, Munich, Germany, April 1997.
- [13] V. Panayotov, G. Chen, D. Povey, and K. Sanjeev, "Librispeech: an ASR corpus based on public domain audio books," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, South Brisbane, QLD, Australia, April 2015.
- [14] A. Varga, H. Steeneken, and M. Tomlinson, "The NOISEX-92 study on the effect of additive noise on automatic speech



- recognition,” *Speech Communication*, vol. 12, no. 3, pp. 247–251, 1992, technical report speech research unit defense research agency.
- [15] J. Thiemann, N. Ito, and E. Vincent, “The diverse environments multi-channel acoustic noise database: a database of multichannel environmental noise recordings,” *Journal of the Acoustical Society of America*, vol. 133, no. 5, pp. 3591–3597, 2013.
- [16] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, “GPURIR: a python library for room impulse response simulation with GPU acceleration,” *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 1–19, 2021.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [18] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

## Research Article

# Graphical Optimization Method for Symmetrical Bidirectional Corridor Progression

Kai Lu <sup>1,2</sup>, Shuyan Jiang <sup>1</sup>, Yiming Zhao <sup>1</sup>, Yongjie Lin <sup>1</sup> and Yin Hai Wang <sup>3</sup>

<sup>1</sup>South China University of Technology, School of Civil Engineering and Transportation, Guangzhou, Guangdong, China

<sup>2</sup>Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Nanjing, China

<sup>3</sup>University of Washington, Department of Civil and Environmental Engineering Seattle, Seattle, Washington, USA

Correspondence should be addressed to Yongjie Lin; [linyjscut@scut.edu.cn](mailto:linyjscut@scut.edu.cn)

Received 13 August 2021; Accepted 1 October 2021; Published 2 November 2021

Academic Editor: JingXin Dong

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

The graphical progression method can obtain grand coordinated schemes with minimal computational complexity. However, there is no standardized solution for this method, and only a few related studies have been found thus far. Therefore, based on the in-depth discussion of the graphical optimization theory mechanism, a process-oriented and high-efficiency graphical method for symmetrical bidirectional corridor progression is proposed in this study. A two-round rotation transformation optimization process of the progression trajectory characteristic lines (PTC lines) is innovatively proposed. By establishing the updated judgment criteria for coordinated mode, the first round of PTC line rotation transformation realizes the optimization of coordinated modes and initial offsets. Giving the conditions for stopping rotation transformation and determining rotation points, rotation directions, and rotation angles, the second round of PTC line rotation transformation achieves the final optimization of the common signal cycle and offsets. The case study shows that the proposed graphical method can obtain the optimal progression effect through regular graphing and solving, although it can also be solved by highly efficient programming.

## 1. Introduction

Intersections are the road network nodes that frequently cause traffic disruption, severe delays, and accidents in a city [1]. Therefore, corridor progression has always been an effective way to ensure traffic safety and efficiency at intersections [2, 3]. Generally speaking, the solution methods of the corridor progression design scheme can be roughly divided into three types: model method, algebraic method, and graphical method.

In comparison, the model method can obtain multiple different ideal coordination design schemes. However, its modeling process is complex, requiring a long solution time for the sizeable calculated amount. The algebraic method, which can meet the requirements of corridor progression in various situations, has the advantages of good operability and robust reproducibility. However, practitioners often need to have a relatively complete theoretical knowledge of coordination planning. Meanwhile, the graphical method

obtains coordination planning using time-space diagrams, illustrating the relationship between intersection spacing, signal timing, and vehicle movement. Although the graphical method is challenging to ensure the optimal corridor progression effect and measure the solution efficiency by manual drawing, it is suitable for engineering applications due to its slight computational complexity, strong operability, and intuitive reflection of the coordinated optimization process.

The model method constructs a linear or nonlinear programming optimization model based on the relationships between the progression bandwidth and signal timing parameters, travel time, and progression boundary trajectory [4]. Moreover, the mixed-integer linear programming method is usually used to realize the optimal solution of signal timing parameters. The most classic models are the MAXBAND model proposed by Morgan and Little [5] and the MULTIBAND model constructed by Gartner et al. [6]. The MAXBAND model optimized common signal cycle,

offsets, vehicle speed, and the order of left-turn phases to maximize the bandwidths, ensuring that most vehicles can drive through the downstream signal intersections without stopping [7]. The MULTIBAND model designed an individually weighted bandwidth for each directional road section, considering the traffic volumes and flow capacities. Extensive research has been constantly provided based on the MAXBAND and MULTIBAND models [8–10]. Optimizing the left-turn phase sequence of the MAXBAND model, Chang et al. [11] further established the MAXBAND-86 model. Lu et al. [12] introduced a bandwidth prorotation impact factor to improve the bandwidth maximization model further. Defining the signal phase sequence as a decision variable, Yang et al. [13] constructed three types of multipath progression models based on the MAXBAND model. Zhang et al. [14] established an asymmetrical multiband (AM-BAND) model and successfully achieved an excellent coordination effect. Zhang et al. [15] constructed the MAXBANDLA model, and numerical tests proved that the model could significantly reduce the number of stops and the average delay of vehicles on long corridors. Zhou et al. [16] built two multiobjectives uneven double cycles. By doing this, they allowed minor intersections to adopt uneven double cycling models to ensure the coordinated efficiency of minor intersections. Aiming to provide progression bands for all major O-D flows, Arsava et al. [17] constructed the OD-BAND model, which considered the significant turning traffic flows from and to cross streets. Arsava et al. [18] expanded the OD-BAND model to the OD-NETBAND model, solving the OD-based traffic signal coordination problem in multiarterial grid networks.

The algebraic method utilizes the numerical calculation method to obtain the optimal corridor progression design schemes, realizing the comprehensive optimization of signal phases. Xu [19] expounded on the classical algebraic method, that is, to determine the optimal common signal cycle and offsets according to the ideal intersection distance that best matches the actual one. Moreover, this method is only applicable under symmetrical phasing. On this basis, the related research of the algebraic method has gradually gained traction. Lu et al. [20] first addressed the theoretical limitations of the classical algebraic method and improved the algebraic method for symmetrical corridor progression. Analyzing the advantages and disadvantages of the classical algebraic method, Wang et al. [21] further improved the calculation method of the bandwidths and the offsets and the location matching method of actual intersection and ideal intersection. As research progressed, the research focus of the corridor progression has been gradually expanded from the algebraic method under the symmetrical phasing to split phasing and asymmetrical phasing. Lu et al. [22] established a bidirectional algebraic method under the split phasing, which can adapt to different corridor conditions, such as asymmetric geometry, large left-turn traffic volume, uneven traffic flow. Ji and Song [23] proposed an asymmetrical bidirectional algebraic method and analyzed the effects of vehicle speed and queued vehicles during the red light. Using the speed transformation and phase combination method, Lu et al. [24] also constructed a bidirectional algebraic

method under asymmetrical phasing, effectively solving each road section's asymmetric bidirectional distance and unequal speed.

The graphical method establishes a set of easy-to-understand drawing rules and utilizes the transformation of basic elements on the time-space diagram to obtain ideal progression bandwidths. However, scholars have done little research on the graphical method of corridor progression until now. China Highway Association [25] defined the intuitive and straightforward graphical method as one of the earliest timing design methods for corridor progression. Xu [19] provided the basic solution idea of the graphical method, but there are still many deficiencies during the solution process. It lacked a specific design principle and well-defined rules for optimizing the common signal cycle and offsets. Lu and Cheng [26] introduced the concept of the NEMA phase based on the graphical method and optimized the phase sequence and offsets by the graphical way. Although an ideal bidirectional bandwidth was obtained, there was no amendment to the shortcomings of the existing graphical method.

In this study, we define the trajectory lines that reflect the unrestrained movement of characteristic vehicles in the progression as progression trajectory characteristic lines (PTC lines). Then, based on the rotation transformation of the PTC line, a graphical method for symmetrical bidirectional corridor progression is proposed to improve further and optimize the overall process of the graphical method. The corresponding progression coordination design process will also be presented in detail. The remainder of the article is organized as follows: Section 3 introduces the design principle of the proposed method. Section 4 presents a case study, which helps to compare the coordinated optimization effect between the proposed method, the improved algebraic method, and the MAXBAND model. Concluding remarks are provided in the last section.

## 2. Materials and Methods

*2.1. Design Process.* The graphical method proposed can be solved by a normal step-by-step graphical process and programming. We define the horizontal scaling of the time-space diagram to convert the progression design speed of the road section to the corridor progression design speed  $V$ . Meanwhile, we also define the vertical scaling of the time-space diagram to realize the transformation of the optimization of the common signal cycle into the optimization of the corridor progression speed. The design process of the proposed graphical method is refined and organized as shown in Figure 1, which is mainly divided into four parts: initialization, the first round of rotation transformation, the second round of rotation transformation, and scheme generation.

### 2.2. Initialization

*2.2.1. Initial Common Signal Cycle.* Let us suppose that there are  $n$  signalized intersections along the corridor and number the intersections in the ascending order in the outbound

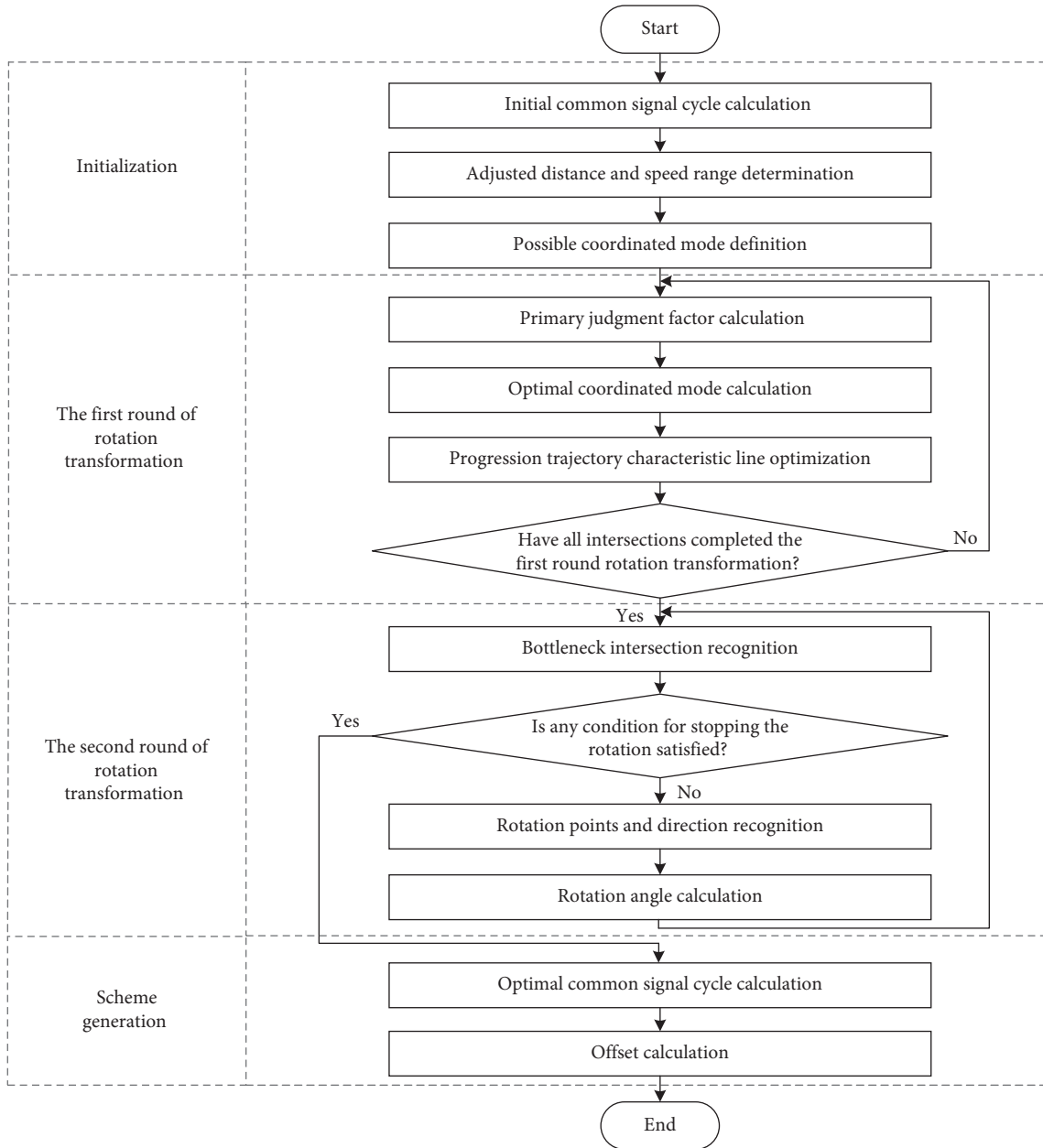


FIGURE 1: The design flow chart of the proposed method.

direction. Then, the  $p$ th ( $1 \leq p \leq n$ ) intersection is defined as  $I_p$  in sequence along the outbound direction.

The common signal cycle range will be determined by the signal cycle range of each intersection, denoted as  $[C_{\min}, C_{\max}]$ . The initial common signal cycle  $C_1$  can be valued as the midpoint between the minimum common signal cycle  $C_{\min}$  and the maximum common signal cycle  $C_{\max}$ .

**2.2.2. Adjusted Distance and Speed.** Let  $d_p$  and  $v_p$  be the actual distance and the progression design speed between intersection  $I_p$  and  $I_{p+1}$  ( $1 \leq p \leq n-1$ ), respectively. Then, we find that the optimal effect of progression coordination does not change in the time-space diagram as long as the travel time of the road section remains unchanged. For example, as

shown in Figure 2, when we simultaneously change the progression design speed and distance of the road section highlighted by the blue and purple arrows in Figure 2(a), but keep the vehicle travel time unchanged, the progression bandwidth of the corridor remains unchanged in Figure 2(b). Therefore, it can be said that the proposed method can be applied to coordinate corridors with inconsistent progression design speeds across road segments. We define it as the horizontal scaling of the time-space diagram.

Therefore, to facilitate the diagrammatic design process, when the design speed of the road section is inconsistent with the corridor progression design speed  $V$ , we will convert the progression design speed of the road section to  $V$  by adjusting the intersection distance. The adjusted distance

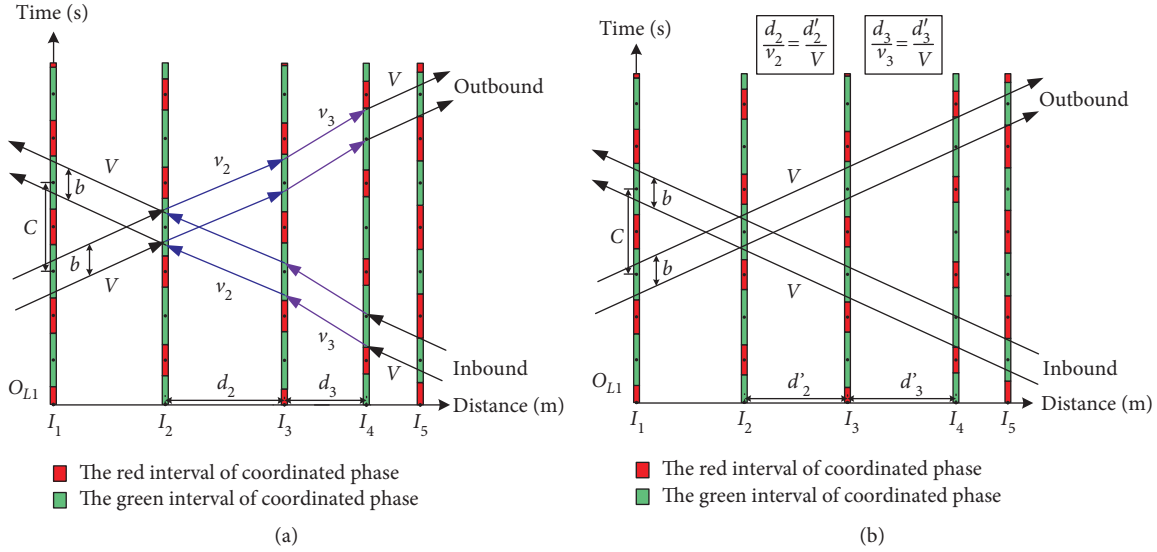


FIGURE 2: The horizontal scaling of the time-space diagram. (a) Before the horizontal scaling. (b) After the horizontal scaling.

$D_p$  between intersection  $I_p$  and intersection  $I_1$  can be calculated by the following equation:

$$D_p = V \cdot \sum_{k=1}^{p-1} \frac{d_k}{v_k}. \quad (1)$$

On the other hand, we find that the optimal effect of progression coordination does not change in the time-space diagram as long as the product of  $V$  and  $C$  remains unchanged. For example, as shown in Figure 3, when we change the common signal cycle and corridor progression design speed but keep their product unchanged in Figure 3(a), the percentage of progression bandwidth of the corridor remains unchanged in Figure 3(b). We define it as the vertical scaling of the time-space diagram.

Then, we can first keep the common signal cycle  $C_1$  unchanged and optimize the adjusted speed to find their optimal product value to obtain the best coordination effect. Finally, the final adjusted speed is adjusted to  $V$ , and the optimal common signal cycle  $C_B$  can be obtained.

According to the corridor progression design speed  $V$  and the range of the common signal cycle  $[C_{\min}, C_{\max}]$ , the adjusted speed  $V_{(i,j)}$  obtained after the  $j$ th rotation transformation in the  $i$ th round should satisfy the following equation:

$$\frac{V \cdot C_{\min}}{C_1} \leq V_{(i,j)} \leq \frac{V \cdot C_{\max}}{C_1}. \quad (2)$$

**2.2.3. Possible Coordinated Mode.** To balance the effect of bidirectional corridor progression, the coordinated mode of each intersection must adopt synchronous coordination or backstepping coordination. Synchronous coordination means that the green center point of the coordinated phase is consistent with intersection  $I_1$ , whereas backstepping coordination means that the red center point is consistent with the green center point of the coordinated phase at  $I_1$ . The

progressions in the outbound direction and the inbound direction have the characteristic of time symmetry, and then, only the outbound direction needs to be considered. Then, we define a Boolean variable  $F_p$  as the coordinated mode judgment factor. When  $F_p$  is 0, the optimal coordinated mode of the intersection  $I_p$  is synchronous coordination. When  $F_p$  is 1, the optimal coordinated mode of  $I_p$  is backstepping coordination.

**2.3. The First Round of Rotation Transformation.** The optimization of the intersection coordinated mode can be realized by defining the rules of the first round of PTC line rotation transformation. First, we define the green center point of the coordinated phase at  $I_1$  as the reference point  $O_{L1}$  of the time-space diagram coordinate system. A horizontal line can be drawn from  $O_{L1}$  and rotated until the cotangent of its angle with the  $x$ -axis equals the corridor progression design speed  $V$ . This process is defined as the first rotation transformation of the PTC line in the first round. We define the rotated ray as the initial PTC line  $L_1$  and then assign  $V$  to adjusted speed  $V_{(1,1)}$ .

The following steps have to be executed cyclically until the coordinated modes of all intersections have been determined.

**2.3.1. Primary Judgment Factor Calculation.** The intersection  $I_p$  is selected as the current coordinated intersection during the first round's  $p$ th ( $2 \leq p \leq n$ ) rotation transformation. According to PTC line  $L_{p-1}$  obtained by the last rotation transformation, the crossing point of  $L_{p-1}$  and the timeline of  $I_p$  is marked as  $O_{Lp}$  as shown in Figure 4(a). The coordinated phase horizontal red center lines at  $I_1$  have crossing points with the timeline of  $I_p$  and so we define the point closest to  $O_{Lp}$  as  $O_{Rp}$ . The coordinated phase horizontal green center lines at  $I_1$  have crossing points with the timeline of  $I_p$  and so we define the point closest to  $O_{Lp}$  as  $O_{Gp}$ .

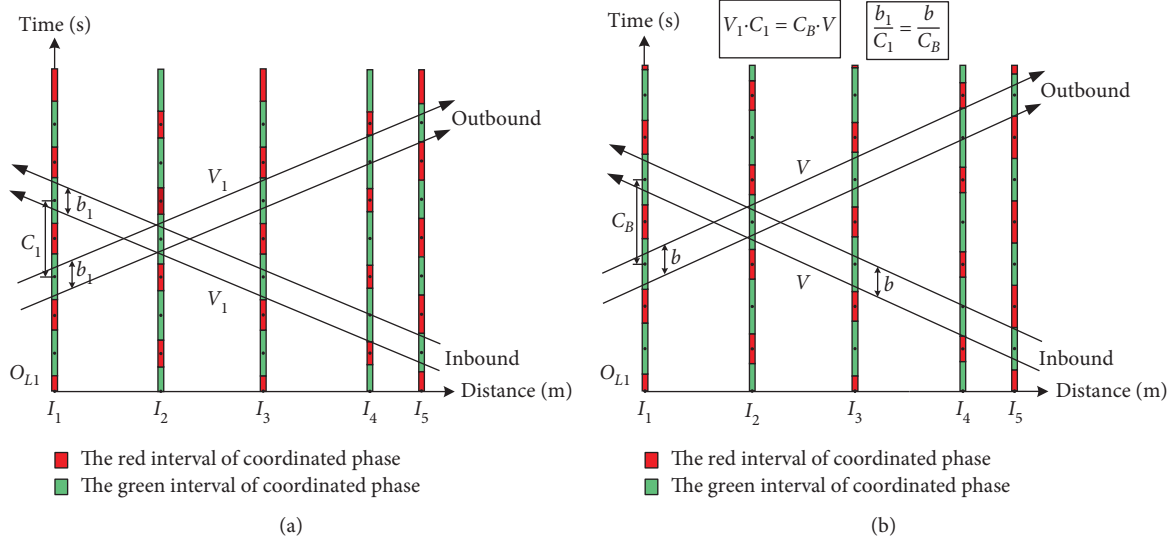


FIGURE 3: The vertical scaling of the time-space diagram. (a) Before the vertical scaling. (b) After the vertical scaling.

Our first step is to define and calculate some basic parameters, and the initialization routine can be presented below.

Initialization procedure

**Step0.** Determine the range of the common signal cycle  $[C_{\min}, C_{\max}]$  and calculate the initial common signal cycle  $C_1 = 0.5 \cdot (C_{\min} + C_{\max})$ .

**Step1.** Make a judgment of whether the progression design speeds of all road sections equal the corridor progression design speed or not. If there is an inequality, calculate the intersection adjusted distance  $D_p$  for the corresponding road section as shown in Equation (1).

**Step2.** Determine the range of adjusted speed  $V_{(i, j)}$  obtained after the  $j$ th rotation transformation in the  $i$ th round, as shown in Equation (2).

ALGORITHM 1: Initialization procedure.

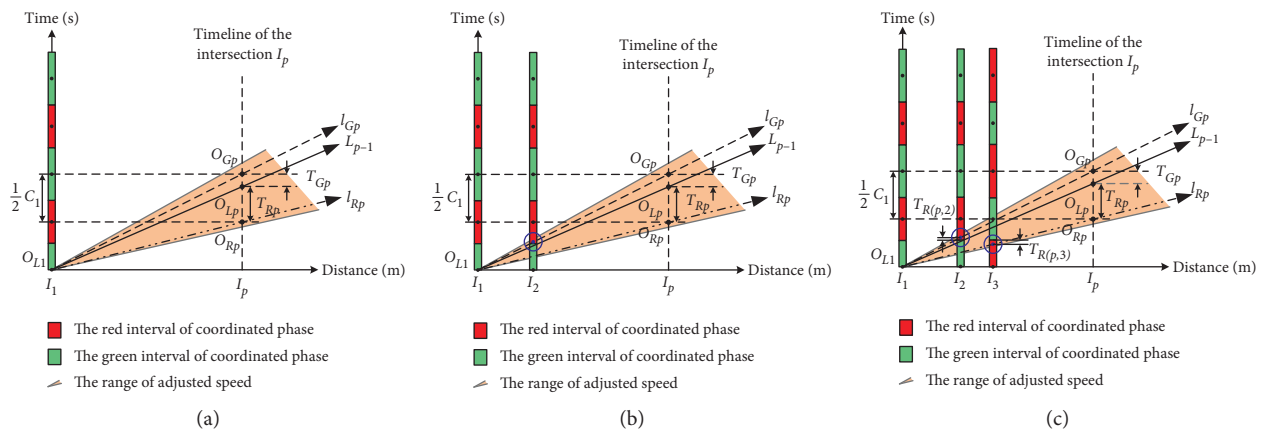


FIGURE 4: The determination of the coordinated mode when  $f_p \geq 0$ .

Starting at the point  $O_{L1}$ , we can make a ray  $l_{Rp}$  ( $l_{Gp}$ ) passing through the point  $O_{Rp}$  ( $O_{Gp}$ ), and the corresponding adjusted speed  $V_{Rp}$  ( $V_{Gp}$ ) satisfies the following:

$$V_{Rp}(V_{Gp}) = \frac{D_p}{y_{Rp}} \left( \frac{D_p}{y_{Gp}} \right). \quad (3)$$

The time difference between points  $O_{Lp}$  and  $O_{Rp}$  ( $O_{Gp}$ ), which corresponds to the distance between the ordinate of points  $y_{Lp}$  and  $y_{Rp}$  ( $y_{Gp}$ ), is recorded as  $T_{Rp}$  ( $T_{Gp}$ ). Then, the primary judgment factor  $f_p$  can be constructed according to  $T_{Rp}$  and  $T_{Gp}$ :

$$f_p = T_{Rp} - T_{Gp} = |y_{Lp} - y_{Rp}| - |y_{Lp} - y_{Gp}|. \quad (4)$$

**2.3.2. Optimal Coordinated Mode Calculation.** We define  $f_{R(p,q)}$  ( $f_{G(p,q)}$ ) ( $1 < q < p$ ) as a Boolean variable. When ray  $l_{Rp}$  ( $l_{Gp}$ ) does not pass through the red interval of the intersection  $I_q$  ( $1 < q < p$ ),  $f_{R(p,q)}$  ( $f_{G(p,q)}$ ) is 0. When ray  $l_{Rp}$  ( $l_{Gp}$ ) passes through the red interval of  $I_q$ ,  $f_{R(p,q)}$  ( $f_{G(p,q)}$ ) is 1. Also, the red interval crossing amount of  $l_{Rp}$  ( $l_{Gp}$ ) in  $I_q$  can be defined as  $T_{R(p,q)}$  ( $T_{G(p,q)}$ ).

When  $f_p \geq 0$ , the situation is shown in Figure 4. If ray  $l_{Gp}$  does not pass through any red interval of the coordinated phase, all  $f_{G(p,q)}$  equals 0. The current coordinated intersection can form a synchronous coordinated mode with  $I_1$ , where  $F_p = 0$ . If ray  $l_{Gp}$  passes through any red interval of the coordinated phase while ray  $l_{Rp}$  does not pass through any red interval, as shown in Figure 4(b), the backstepping coordinated mode can be chosen, where  $F_p = 1$ . If both  $l_{Gp}$  and  $l_{Rp}$  pass through the red interval of the coordinated phase, and if the maximum red interval crossing amount of  $l_{Gp}$  is less than or equal to  $l_{Rp}$ , as shown in Figure 4(c), these two intersections can form a synchronous coordinated mode, where  $F_p = 0$ . Otherwise, the backstepping coordinated mode will be chosen as  $F_p = 1$ .

Similarly, when  $f_p < 0$ , if ray  $l_{Rp}$  does not pass through any red interval of the coordinated phase, all  $f_{R(p,q)}$  equals 0,  $F_p = 1$ . If ray  $l_{Rp}$  passes through any red interval of the coordinated phase while ray  $l_{Gp}$  does not pass through any red interval, then  $F_p = 0$ . If both  $l_{Rp}$  and  $l_{Gp}$  pass through the red interval of the coordinated phase, and if the maximum red interval crossing amount of  $l_{Rp}$  is less than or equal to  $l_{Gp}$ ,  $F_p = 1$ . Otherwise,  $F_p = 0$ .

**2.3.3. PTC Line Optimization.** If the current coordinated intersection has formed a synchronous coordinated mode with  $I_1$  and  $V_{Gp}$  is within the rotation range of the PTC line, we can define  $l_{Gp}$  as PTC line  $L_p$  and assign the value of  $V_{Gp}$  to the adjusted speed  $V_{(1,p)}$ , as shown in Figure 5(a). When  $V_{Gp}$  is out of the rotation range, we can keep the PTC line unchanged, that is, as shown in Figure 5(b),  $L_p$  is the same as  $L_{p-1}$ , and  $V_{(1,p)}$  equals  $V_{(1,p-1)}$ .

Similarly, if the current coordinated intersection has formed a backstepping coordinated mode with  $I_1$  and  $V_{Rp}$  is within the rotation range of the PTC line, we can define  $l_{Rp}$  as PTC line  $L_p$  and assign the value of  $V_{Rp}$  to  $V_{(1,p)}$ . When  $V_{Rp}$  is out of the rotation range, we can keep the PTC line unchanged.  $L_p$  is the same as  $L_{p-1}$ , and  $V_{(1,p)}$  equals  $V_{(1,p-1)}$ .

**2.4. The Second Round of Rotation Transformation.** The final optimization of the common signal cycle and the offsets will be determined upon completing second round of PTC line rotation transformation.

After the first round of rotation transformation, the coordinated modes of all signalized intersections have been determined. We assign adjusted speed  $V_{(1,n)}$  to  $V_{(2,0)}$ . Then, the beginning PTC line  $L_{B0}$  and the end PTC line  $L_{E0}$  of the outbound progression can be obtained. Then, the initial bandwidth  $b_0$  and ratio  $R_0$  can also be calculated. Meanwhile, the bottleneck intersections of the beginning PTC line are put into the intersection set  $S_{B0}$ , and the bottleneck intersections of the end PTC line are put into the intersection set  $S_{E0}$ . The calculation of related parameters will be introduced in the next section.

We define a termination decision parameter  $F_A$ , which is a Boolean variable. The following steps have to be executed cyclically until the termination decision parameter  $F_A$  equals 1.

**2.4.1. Bottleneck Intersection Recognition.** We assume that the obtained beginning PTC line, end PTC line, adjusted speed, bandwidth, and bandwidth ratio after the  $m$ th rotation transformation in the second round are defined as  $L_{Bm}$ ,  $L_{Em}$ ,  $V_{(2,m)}$ ,  $b_m$ , and  $R_m$ , respectively. The parameters obtained from the  $m-1$ th rotation transformation need to be used during the  $m$ th rotation transformation in the second round.

According to  $F_p$ ,  $V_{(2,m-1)}$ ,  $C_1$ , and  $\lambda_p$ , we can draw the beginning and end PTC lines in the time-space diagram. We define the Boolean variables  $K_{B(m-1,p)}$  and  $K_{E(m-1,p)}$  as the judgment factors of the bottleneck intersection of the beginning and end PTC lines. When  $K_{B(m-1,p)}$  equals 1, intersection  $I_p$  is the bottleneck intersection of the beginning PTC line after the  $m-1$ th rotation transformation,  $I_p \in S_{Bm-1}$ , and the corresponding bottleneck point is defined as  $P_{Bp}$ . When  $K_{E(m-1,p)}$  equals 1,  $I_p$  is the bottleneck intersection of the end PTC line after the  $m-1$ th rotation transformation,  $I_p \in S_{Em-1}$ , and the corresponding bottleneck point is recorded as  $P_{Ep}$ .

Before the other calculation steps of the  $m$ th rotation transformation in the second round, it is necessary to judge whether the sets  $S_{Bm-1}$  and  $S_{Em-1}$  meet the conditions for stopping rotation. If the conditions for stopping rotation are unsatisfied, then it must be continued to complete the  $m$ th rotation transformation until the bottleneck intersections meet the conditions of stopping rotation, where  $F_A = 1$ .

#### 2.4.2. Conditions for Stopping the Rotation

**Condition 1.**  $\exists I_k \in (S_{Bm-1} \cap S_{Em-1})$ , that is, there are both bottleneck points of the beginning and end PTC lines at intersection  $I_k$ . At this time, the ratio of the bandwidth  $R_m$  equals the split of the coordinated phase at  $I_k$ , which means that it has already reached the maximum value of the bandwidth ratio. Therefore, there is no need to continue the rotation transformation of the PTC line.

**Condition 2.**  $\exists I_i \in S_{Em-1}$ ,  $I_j \in S_{Bm-1}$ ,  $I_k \in S_{Em-1}$ , and  $i < j < k$ , that is, there are two bottleneck intersections of the end PTC line located upstream and downstream of a bottleneck intersection of the beginning PTC line, respectively.

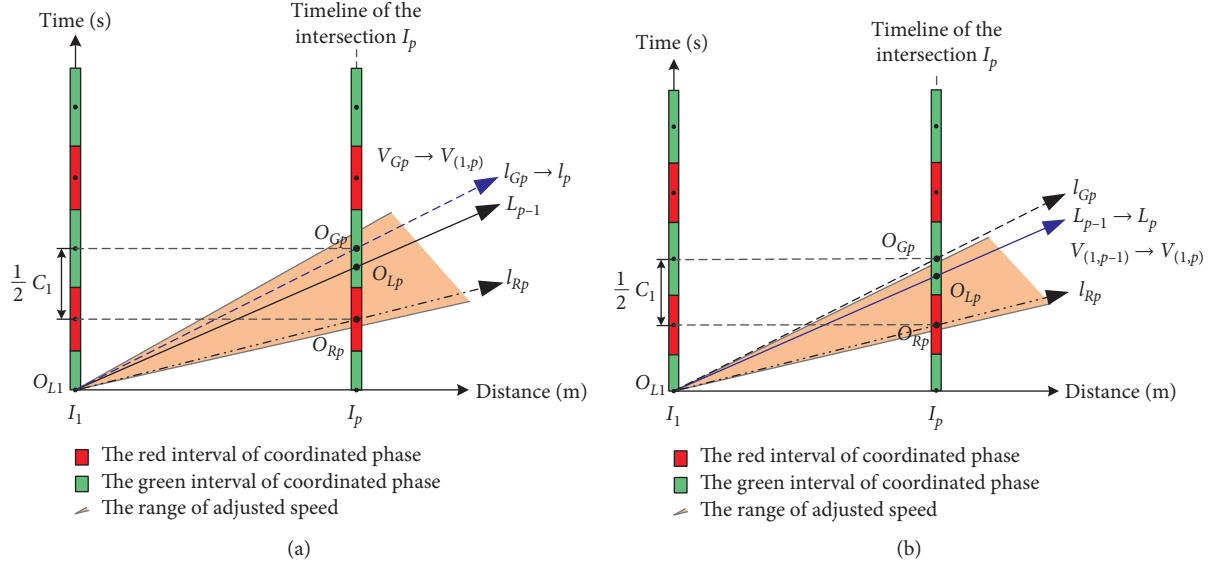


FIGURE 5: The PTC line updating under synchronous coordinated mode.

Our second step is to optimize the intersection coordinated modes, and the first round of the rotation transformation routine can be presented below.

First Round Rotation Transformation Procedure

**Step0.** Determine the crossing point  $O_{Lp}$ ,  $O_{Rp}$ , and  $O_{Gp}$ .

First, let  $y_{Lp} = D_p/V_{(1,p-1)}$ .

Then, if  $\text{mod}((y_{Lp} \lfloor (C_1/2) \rfloor), 2) = 0$ ,  $y_{Rp}$  equals  $C_1 \cdot (y_{Lp} \lfloor (C_1/2) \rfloor) / 2 + C_1/2$  and  $y_{Gp}$  equals  $C_1 \cdot (y_{Lp} \lfloor (C_1/2) \rfloor) / 2$ , else  $y_{Rp} = C_1 \cdot (y_{Lp} \lfloor (C_1/2) \rfloor) / 2 - C_1/2$  and  $y_{Gp} = C_1 \cdot (y_{Lp} \lfloor (C_1/2) \rfloor) / 2 + 1$ .

**Step1.** Calculate the primary judgment factor  $f_p$  as shown in Equation (4).

**Step2.** Make a judgment of whether ray  $l_{Rp}$  and  $l_{Gp}$  pass through the red interval of any intersection  $I_q$  ( $1 < q < p$ ) or not.

If  $|D_q/V_{Rp} - y_{R(p,q)}| \leq 0.5 \cdot C_1 \cdot \lambda_q$ ,  $f_{R(p,q)} = 0$ , otherwise  $f_{R(p,q)} = 1$  and  $T_{R(p,q)} = |D_q/V_{Rp} - y_{R(p,q)}| - 0.5 \cdot C_1 \cdot \lambda_q$ .

If  $|D_q/V_{Gp} - y_{G(p,q)}| \leq 0.5 \cdot C_1 \cdot \lambda_q$ ,  $f_{G(p,q)} = 0$ , otherwise  $f_{G(p,q)} = 1$  and  $T_{G(p,q)} = |D_q/V_{Gp} - y_{G(p,q)}| - 0.5 \cdot C_1 \cdot \lambda_q$ .

$y_{R(p,q)}$  ( $y_{G(p,q)}$ ) is the ordinate of the green center point of  $I_q$ , which is closest to the crossing point of  $l_{Rp}$  ( $l_{Gp}$ ) and the timeline of  $I_q$ , and  $\lambda_q$  is the green split of  $I_q$ .

**Step3.** Determine the coordinated mode of  $I_p$ .

If  $f_p \geq 0$  and  $\sum_{q=2}^{p-1} f_{G(p,q)} = 0$ , or  $f_p < 0$ ,  $\sum_{q=2}^{p-1} f_{R(p,q)} > 0$  and  $\sum_{q=2}^{p-1} f_{G(p,q)} = 0$ , or  $\sum_{q=2}^{p-1} f_{R(p,q)} > 0$ ,  $\sum_{q=2}^{p-1} f_{G(p,q)} > 0$  and  $\max(T_{G(p,2)}, \dots, T_{G(p,p-1)}) < \max(T_{R(p,2)}, \dots, T_{R(p,p-1)})$ , the coordinated mode judgment factor  $F_p$  equals 0.

If  $f_p < 0$  and  $\sum_{q=2}^{p-1} f_{R(p,q)} = 0$ , or  $f_p \geq 0$ ,  $\sum_{q=2}^{p-1} f_{G(p,q)} > 0$  and  $\sum_{q=2}^{p-1} f_{R(p,q)} = 0$ , or  $\sum_{q=2}^{p-1} f_{R(p,q)} > 0$ ,  $\sum_{q=2}^{p-1} f_{G(p,q)} > 0$  and  $\max(T_{G(p,2)}, \dots, T_{G(p,p-1)}) > \max(T_{R(p,2)}, \dots, T_{R(p,p-1)})$ , the coordinated mode judgment factor  $F_p$  equals 1.

**Step4.** PTC line optimization.

If  $F_p = 0$  and  $V_{Gp}$  is within the rotation range, calculate  $V_{Gp}$  as shown in Equation (3), and then let  $V_{(1,p)}$  equal  $V_{Gp}$ , else if  $F_p = 1$  and  $V_{Rp}$  is within the rotation range, calculate  $V_{Rp}$  and then let  $V_{(1,p)}$  equal  $V_{Rp}$ , else  $V_{(1,p)} = V_{(1,p-1)}$ .

**Step5.** Make a judgment of whether  $p$  equals  $n$  or not. If  $p < n$ , let  $p$  equal  $p+1$  and then return to Step0.

ALGORITHM 2: First round rotation transformation procedure.

The corresponding bandwidth will decrease when we increase the progression speed and rotate the PTC line clockwise, as shown in Figure 6(a). Furthermore, when we reduce the progression speed and rotate the PTC line counter-clockwise the corresponding bandwidth will decrease, as shown in Figure 6(b). Therefore, the further rotation transformation of the PTC line has to be finished.

*Condition 3.*  $\exists I_i \in S_{Bm-1}$ ,  $I_j \in S_{Em-1}$ ,  $I_k \in S_{Bm-1}$ , and  $i < j < k$ , that is, there are two bottleneck intersections of the beginning PTC line located upstream and downstream of a bottleneck intersection of the end PTC line, respectively. The

corresponding bandwidth will decrease when we increase the progression speed and rotate the PTC line clockwise, as shown in Figure 7(a). Furthermore, when we reduce the progression speed and rotate the PTC line counter-clockwise, the corresponding bandwidth will decrease, as shown in Figure 7(b). Therefore, the further rotation transformation of the PTC line has to be finished.

**2.4.3. Rotation Points and Direction Recognition.** It is necessary to determine the rotation points and direction according to the relationship between the rotation



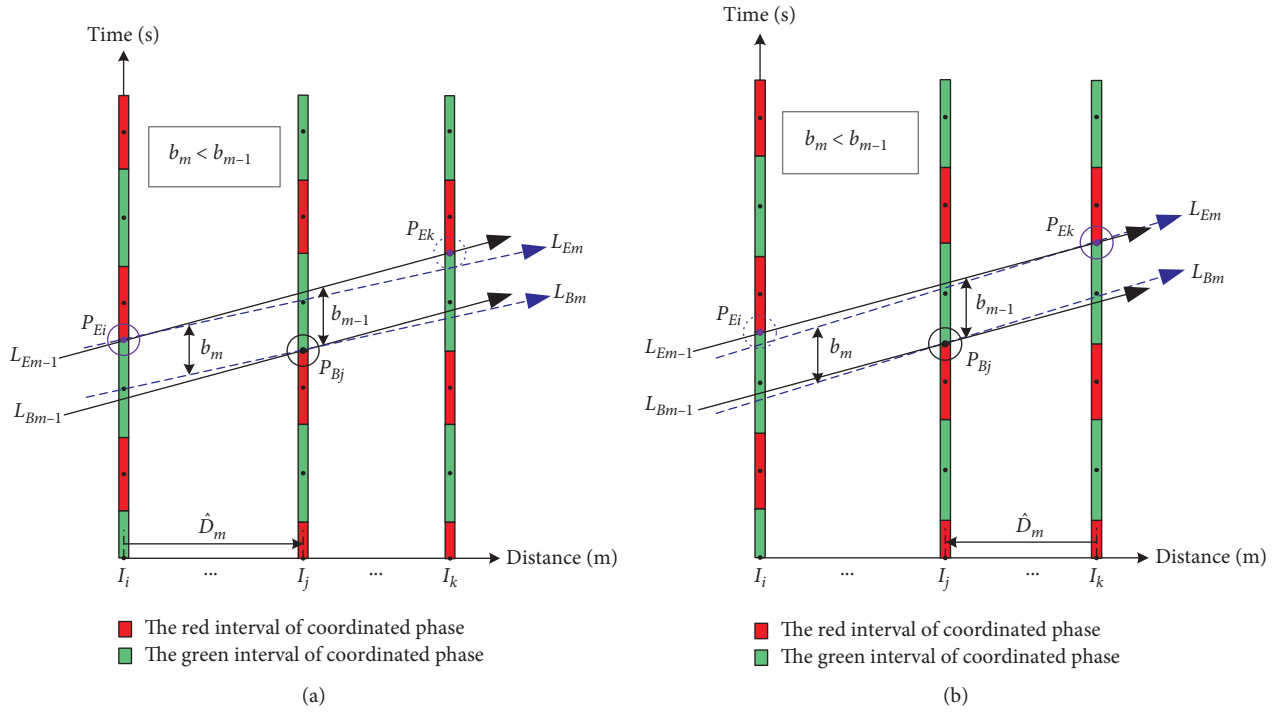


FIGURE 6: Condition 2 for stopping the rotation. (a)  $V_{(2,m)} > V_{(2,m-1)}$ . (b)  $V_{(2,m)} < V_{(2,m-1)}$ .

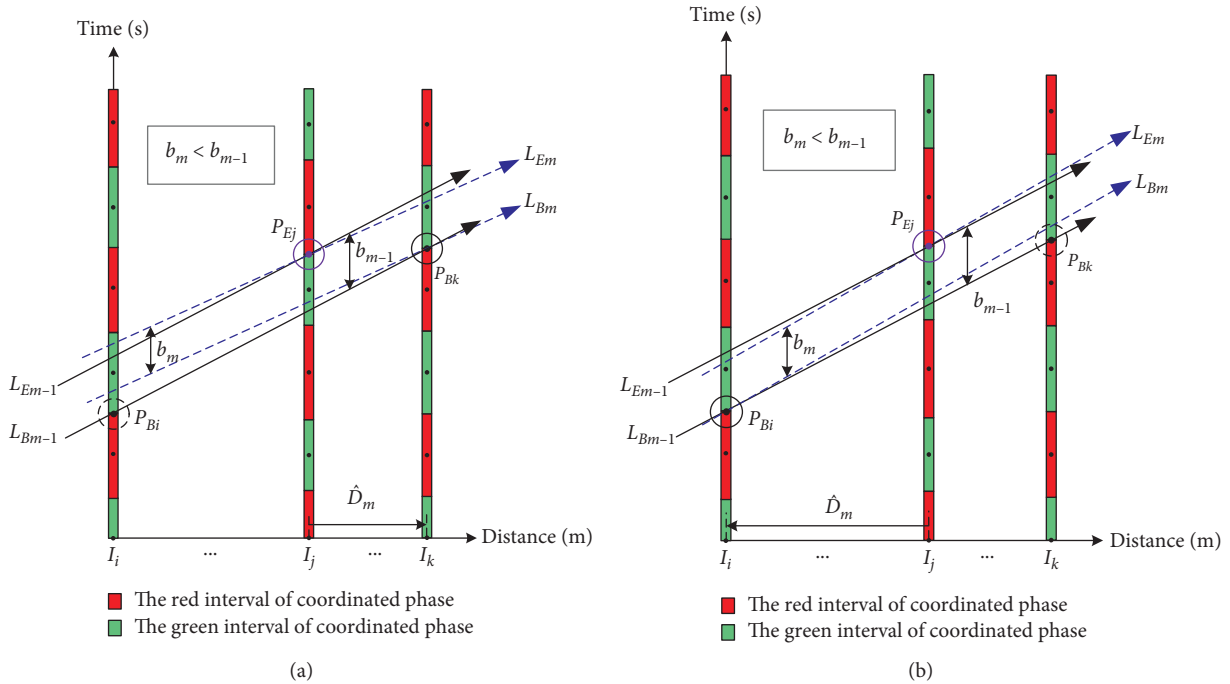


FIGURE 7: Condition 3 for stopping the rotation. (a)  $V_{(2,m)} > V_{(2,m-1)}$ . (b)  $V_{(2,m)} < V_{(2,m-1)}$ .

transformation and the progression bandwidth during the  $m$ th rotation transformation in the second round.

When increasing the progression speed, that is,  $V_{(2,m)} > V_{(2,m-1)}$ , the rotation point of  $L_{Bm}$  should fall at the most downstream bottleneck intersection in  $S_{Bm-1}$ , and the rotation point of  $L_{Em}$  should fall at the most upstream

bottleneck intersection in  $S_{Em-1}$ , as shown in Figures 8(a) and 8(b). Therefore, when rotating the PTC line clockwise, we should select intersections  $I_i \in S_{Em-1}$ ,  $I_j \in S_{Bm-1}$  to ensure that  $\forall I_k \in S_{Em-1}$  satisfies  $i \leq k$  and  $\forall I_l \in S_{Bm-1}$  satisfies  $j \geq l$ . Then, the bottleneck point  $P_{Ei}$  and  $P_{Bj}$  should be taken as the rotation points.

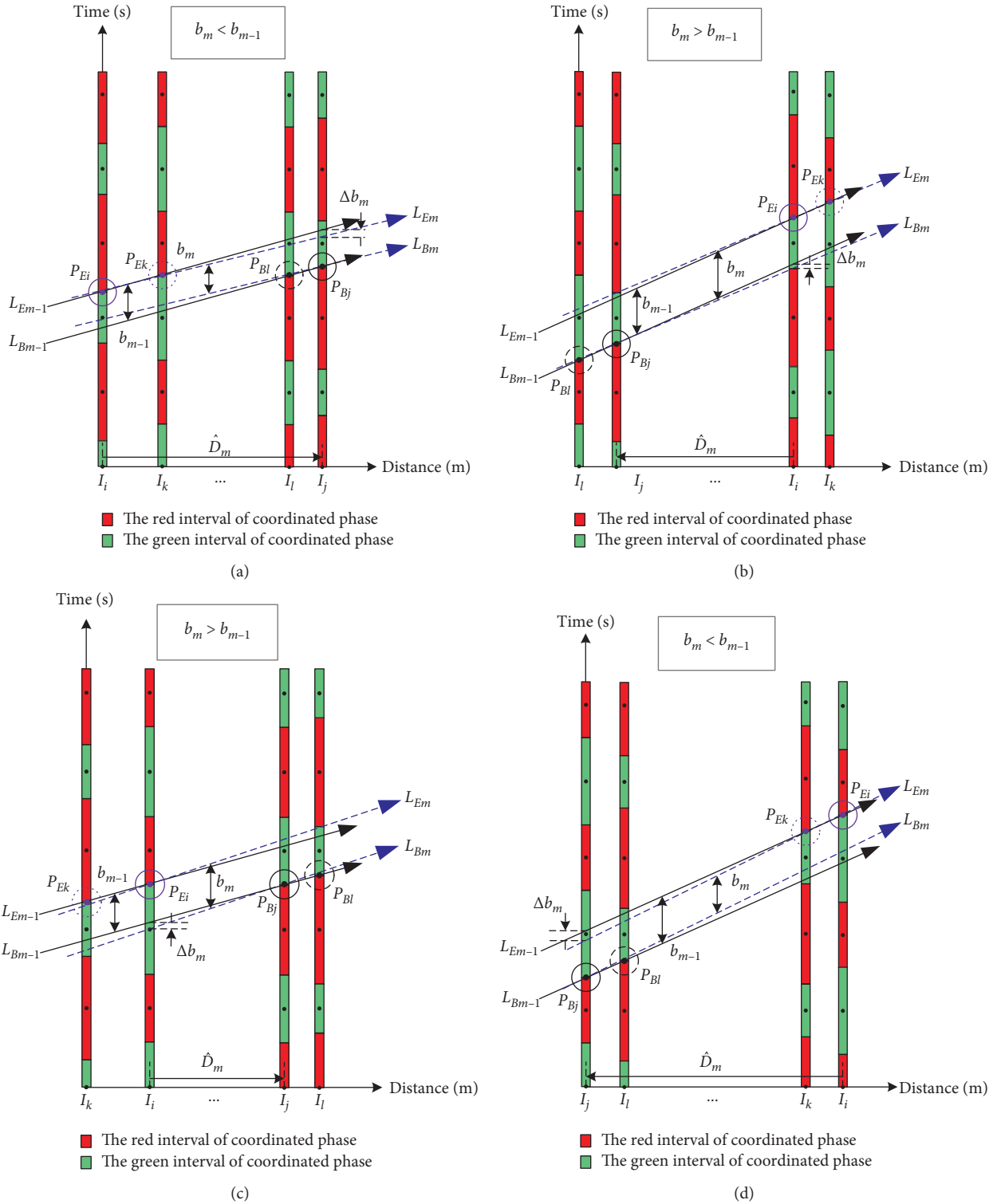


FIGURE 8: The relationship between the rotation transformation and the progression bandwidth. (a)  $V_{(2,m)} > V_{(2,m-1)}$  and  $\widehat{D}_m > 0$ . (b)  $V_{(2,m)} > V_{(2,m-1)}$  and  $\widehat{D}_m < 0$ . (c)  $V_{(2,m)} < V_{(2,m-1)}$  and  $\widehat{D}_m > 0$ . (d)  $V_{(2,m)} < V_{(2,m-1)}$  and  $\widehat{D}_m < 0$ .

When reducing the progression speed, that is,  $V_{(2,m)} < V_{(2,m-1)}$ , the rotation point of  $L_{Em}$  should fall at the most downstream bottleneck intersection in  $S_{Em-1}$ , and the rotation point of  $L_{Bm}$  should fall at the most upstream bottleneck intersection in  $S_{Bm-1}$ , as shown in

Figures 8(c) and 8(d). Therefore, when rotating the PTC line counter-clockwise, we should select intersections  $I_i \in S_{Em-1}$ ,  $I_j \in S_{Bm-1}$  to ensure that  $\forall I_k \in S_{Em-1}$  satisfies  $i \geq k$  and  $\forall I_l \in S_{Bm-1}$  satisfies  $j \leq l$ . Then, the bottleneck point  $P_{Ei}$  and  $P_{Bj}$  should be taken as the rotation points.

$\hat{D}_m$  is defined as the abscissa difference between the bottleneck rotation points  $P_{Bj}$  and  $P_{Ei}$  in the  $m$ th rotation transformation in the second round, that is,  $\hat{D}_m = x_j - x_i$ . Then, the bandwidth  $b_m$  can be calculated as follows:

$$b_m = b_{m-1} + \Delta b_m = b_{m-1} + \hat{D}_m \cdot \left( \frac{1}{V_{(2,m)}} - \frac{1}{V_{(2,m-1)}} \right), \quad (5)$$

where  $\Delta b_m$  is the bandwidth increment resulting from the transformation of the progression speed from  $V_{(2,m-1)}$  to  $V_{(2,m)}$ .

We define a Boolean variable  $K_{Dm}$  as the judgment factor of the rotation direction. When  $K_{Dm}$  equals 0, the rotation direction is the counter-clockwise rotation, whereas the rotation direction is the clockwise rotation when  $K_{Dm}$  equals 1.

*Scenario 1.* If  $\forall I_k \in S_{Em-1}$ , and  $\forall I_l \in S_{Bm-1}$  satisfy  $k < l$ , that is, all the bottleneck intersections of the end PTC line are located upstream of any bottleneck intersection of the beginning PTC line, then  $\hat{D}_m > 0$ . At this time, if the PTC line is rotated clockwise,  $V_{(2,m)} > V_{(2,m-1)}$ , the obtained bandwidth after the rotation will decrease according to Equation (5). If the PTC line is rotated counter-clockwise,  $V_{(2,m)} < V_{(2,m-1)}$ , the obtained bandwidth after the rotation will increase according to Equation (5). Therefore, the rotation direction in Scenario 1 should be determined as the counter-clockwise rotation,  $K_{Dm} = 0$ .

*Scenario 2.* If  $\forall I_k \in S_{Em-1}$ , and  $\forall I_l \in S_{Bm-1}$  satisfy  $k > l$ , that is, all of the bottleneck intersections of the end PTC line are located downstream of any bottleneck intersection of the beginning PTC line, then  $\hat{D}_m < 0$ . At this time, if the PTC line is rotated clockwise,  $V_{(2,m)} > V_{(2,m-1)}$ , the obtained bandwidth after the rotation will increase according to Equation (5). If the PTC line is rotated counter-clockwise,  $V_{(2,m)} < V_{(2,m-1)}$ , the obtained bandwidth after the rotation will decrease according to Equation (5). Therefore, the rotation direction in Scenario 2 should be determined as the clockwise rotation,  $K_{Dm} = 1$ .

**2.4.4. Rotation Angle Calculation.** Taking the bottleneck point of the end PTC line  $P_{Ei}$  and the bottleneck point of the beginning PTC line  $P_{Bj}$  as extreme points, we can calculate the rotation angle formed with other designated crossing points and determine the adjusted speed comprehensively after the rotation transformation.

The green endpoint of the coordinated phase crossed by the end PTC line  $L_{Em-1}$  at intersection  $I_k$  ( $1 \leq k \leq n$ ) is defined as  $P_{Fk}$ , and the green start point of the coordinated phase crossed by the beginning PTC line  $L_{Bm-1}$  at  $I_k$  ( $1 \leq k \leq n$ ) is defined as  $P_{Sk}$ .

The adjusted speed  $V_{Fk}(V_{Sk})$  corresponding to the rotation line  $L_{Fk}(L_{Sk})$  formed by connecting points  $P_{Ei}(P_{Bj})$  and  $P_{Fk}(P_{Sk})$  can be calculated by the following equation:

$$V_{Fk}(V_{Sk}) = \frac{x_k - x_i}{y_{Fk} - y_{Ei}} \left( \frac{x_k - x_j}{y_{Sk} - y_{Bj}} \right). \quad (6)$$

Here,  $y_{Fk}$  and  $y_{Ei}$  represent the ordinate of points  $P_{Fk}$  and  $P_{Ei}$ , respectively;  $y_{Sk}$  and  $y_{Bj}$  represent the ordinate of points  $P_{Sk}$  and  $P_{Bj}$ , respectively.

If  $K_{Dm}$  equals 0, we should connect points  $P_{Ei}$  and  $P_{Fg}$  ( $i + 1 \leq g \leq n$ ) in turn to form the rotation line  $L_{Fg}$ . When the corresponding adjusted speed  $V_{Fg}$  is within the adjusted speed range, the eligible  $V_{Fg}$  is incorporated into the optional vehicle speed set,  $S_{Vm}$ , obtained after the  $m$ th rotation transformation. Meanwhile,  $P_{Bj}$  and  $P_{Sh}$  ( $1 \leq h \leq j-1$ ) are connected to form the rotation line  $L_{Sh}$ . When the corresponding adjusted speed  $V_{Sh}$  is within the adjusted speed range, the eligible  $V_{Sh}$  is incorporated into  $S_{Vm}$ . Then, we have to select the maximum value in  $S_{Vm}$  as the adjusted speed  $V_{(2,m)}$  determined by the  $m$ th rotation transformation of the vehicle PTC line in the second round, as shown in Figure 9(a).

If  $K_{Dm}$  equals 1, we should connect points  $P_{Ei}$  and  $P_{Fg}$  ( $1 \leq g \leq i-1$ ) in turn to form the rotation line  $L_{Fg}$ . When  $V_{Fg}$  is within the adjusted speed range, the eligible  $V_{Fg}$  is incorporated into  $S_{Vm}$ . Meanwhile, points  $P_{Bj}$  and  $P_{Sh}$  ( $j + 1 \leq h \leq n$ ) are connected to form the rotation line  $L_{Sh}$ . When  $V_{Sh}$  is within the adjusted speed range, the eligible  $V_{Sh}$  is incorporated into  $S_{Vm}$ . Then, we have to select the minimum value in  $S_{Vm}$  as  $V_{(2,m)}$ , as shown in Figure 9(b).

According to  $V_{(2,m)}$ , the end PTC line  $L_{Em}$  and the beginning PTC line  $L_{Bm}$  in the outbound direction can be obtained after the rotation transformation. Before entering the next rotation transformation in the second round, the sets of the bottleneck intersections can be updated as  $S_{Em}$  and  $S_{Bm}$  according to  $L_{Em}$  and  $L_{Bm}$ .

## 2.5. Scheme Generation

**2.5.1. Optimal Common Signal Cycle Calculation.** The optimal adjusted speed  $V_B$  can be calculated according to the final PTC line obtained after the rotation transformations in the second round. According to the vertical scaling of the time-space diagram, the optimal common signal cycle  $C_B$  can be calculated by the following equation:

$$C_B = \frac{C_1 \cdot V_B}{V}. \quad (7)$$

**2.5.2. Offset Calculation.** When the coordinated mode of each intersection and the optimal common signal cycle have been determined, combined with the known split distribution scheme, the green interval of the coordinated phase and absolute offset of each intersection can be calculated. The absolute offset of the intersection  $I_p$  ( $1 \leq p \leq n$ ) is the beginning of the coordinated phase at  $I_p$  defined as  $O_p$ . While the green center point of the coordinated phase at intersection  $I_1$  is defined as the offset reference point with a value

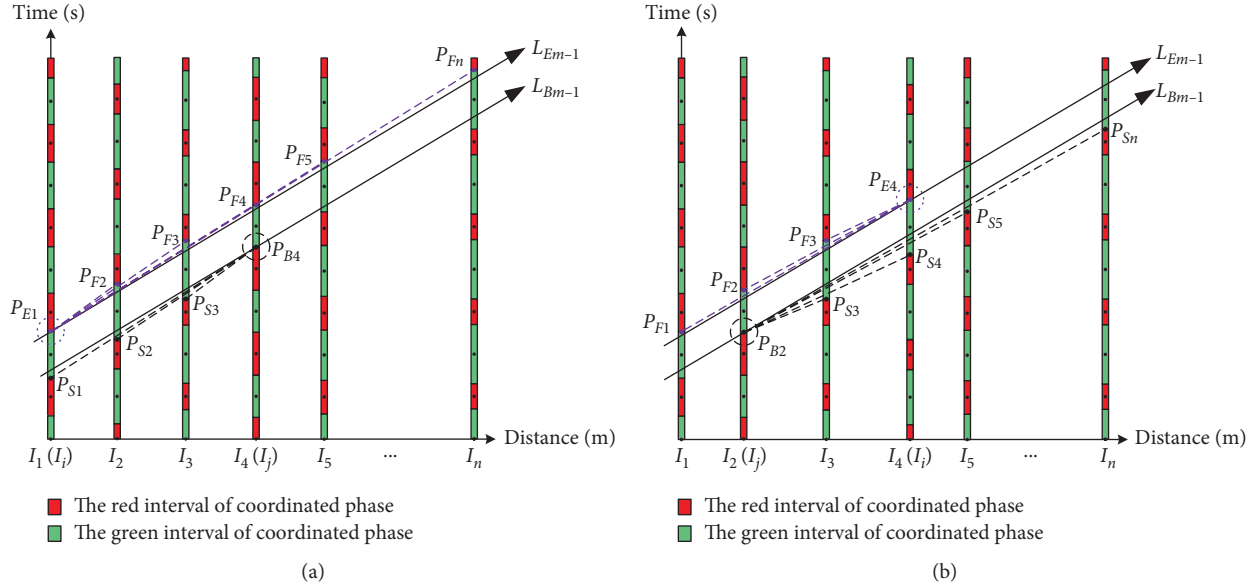


FIGURE 9: Rotation angle determination.

Our third step is to optimize the common signal cycle and offsets, and the second round of the rotation transformation routine can be presented below.

**Second Round Rotation Transformation Procedure**

**Step0.** Determine the bottleneck intersections and the set  $S_{B_{m-1}}$  and  $S_{E_{m-1}}$ .

According to  $F_p$ ,  $V_{(2,m-1)}$ ,  $C_1$ , and  $\lambda_p$ , draw the beginning and end PTC lines in the time-space diagram. If  $I_p$  is the bottleneck intersection of the beginning PTC line, let  $K_{B(m-1,p)} = 1$  and add  $I_p$  in set  $S_{B_{m-1}}$ , else  $K_{B(m-1,p)} = 0$ . And if  $I_p$  is the bottleneck intersection of the end PTC line, let  $K_{E(m-1,p)} = 1$  and add  $I_p$  in set  $S_{E_{m-1}}$ , else  $K_{E(m-1,p)} = 0$ .

**Step1.** Make a judgment of whether the sets  $S_{B_{m-1}}$  and  $S_{E_{m-1}}$  meet any condition for stopping rotation.

If  $\exists I_k \in (S_{B_{m-1}} \cap S_{E_{m-1}})$ , or  $\exists I_i \in S_{E_{m-1}}, I_j \in S_{B_{m-1}}, I_k \in S_{E_{m-1}}$  and  $i < j < k$ , or  $\exists I_i \in S_{B_{m-1}}, I_j \in S_{E_{m-1}}, I_k \in S_{B_{m-1}}$  and  $i < j < k$ , then  $F_A = 1$ , finish the second round of rotation transformation. Otherwise,  $F_A = 0$ , proceed to Step 2.

**Step2.** Determine the rotation direction.

If  $\forall I_k \in S_{E_{m-1}}, \forall I_l \in S_{B_{m-1}}$ , and  $k < l$ , let  $K_{D_m} = 0$ , else if  $\forall I_k \in S_{E_{m-1}}, \forall I_l \in S_{B_{m-1}}$ , and  $k > l$ , let  $K_{D_m} = 1$ .

**Step3.** Determine the rotation angle.

If  $K_{D_m} = 0$ , connect points  $P_{E_i}$  and  $P_{F_g}$  ( $i+1 \leq g \leq n$ ), points  $P_{B_j}$  and  $P_{S_h}$  ( $1 \leq h \leq j-1$ ) in turn, calculate the corresponding  $V_{F_g}$  and  $V_{S_h}$  as shown in Equation (6), add the eligible  $V_{F_g}$  and  $V_{S_h}$  in set  $S_{V_m}$ , then select the maximum value in  $S_{V_m}$  as the updated adjusted speed  $V_{(2,m)}$ .

If  $K_{D_m} = 1$ , connect points  $P_{E_i}$  and  $P_{F_g}$  ( $1 \leq g \leq i-1$ ), points  $P_{B_j}$  and  $P_{S_h}$  ( $j+1 \leq h \leq n$ ) in turn, calculate the corresponding  $V_{F_g}$  and  $V_{S_h}$  as shown in Equation (6), add the eligible  $V_{F_g}$  and  $V_{S_h}$  in set  $S_{V_m}$ , then select the minimum value in  $S_{V_m}$  as the updated adjusted speed  $V_{(2,m)}$ .

**Step4.** Let  $m = m + 1$ , and return to Step 0.

ALGORITHM 3: Second round rotation transformation procedure.

of zero, the absolute offset of each intersection can be calculated by the following equation:

$$O_p = \begin{cases} C_B - 0.5 \cdot C_B \cdot \lambda_p, & 1 \leq p \leq n \text{ and } F_p = 0, \\ 0.5 \cdot C_B - 0.5 \cdot C_B \cdot \lambda_p, & 1 \leq p \leq n \text{ and } F_p = 1. \end{cases} \quad (8)$$

**2.6. Case Study.** To facilitate the comparative analysis of the optimization effect of the graphical method proposed in this paper, we cite a case study that has been used in many books

and papers [19, 20, 25]. The corridor being analyzed is an east-west roadway, and there are eight signalized intersections along the corridor. The direction from west to east is defined as the outbound direction, and the western-most intersection is selected as intersection  $I_1$ . The schematic diagram of the selected corridor is shown in Figure 10. It has already known that each intersection adopts symmetrical phasing. The split of each intersection is 55%, 60%, 65%, 65%, 60%, 65%, 70%, and 50%, respectively. The optimization range of the common signal cycle is [60, 100] s, and

Our final step is to obtain the final optimized scheme, and the scheme generation routine can be presented below.

Scheme Generation Procedure

**Step0.** Calculate the optimal common signal cycle  $C_B$  as shown in Equation (7).

**Step1.** Calculate the absolute offset  $O_p$  of each intersection as shown in Equation (8).

**Step2.** Make the time-space diagram to check the coordination effect of the final optimized scheme obtained by the proposed graphical optimization method.

ALGORITHM 4: Scheme generation procedure.

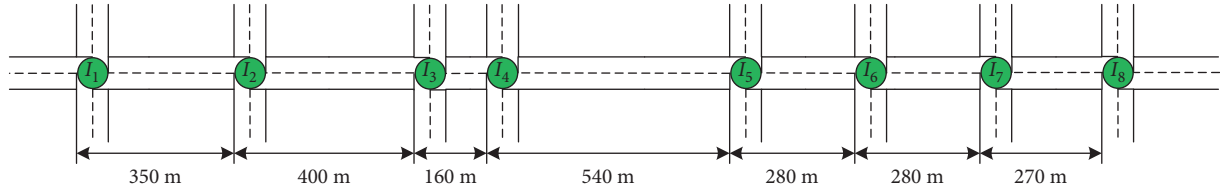


FIGURE 10: Structure diagram of the selected corridor.

the progression design speed is 40 km/h, which approximately equals 11 m/s.

**2.7. Scheme Generation.** The proposed method can obtain the scheme through a limited number of rule-guided drawings, and it is suitable for engineering applications. The related process can be shown as follows:

**2.7.1. Initialization.** The initial common signal cycle  $C_1$  is valued as the midpoint between the minimum common signal cycle  $C_{\min}$  and the maximum common signal cycle  $C_{\max}$ , which is 80 s. Because the progression design speeds of all the road sections are 11 m/s, there is no need to adjust the intersection distance. When the initial common signal cycle  $C_1$  is unchanged, the adjusted speed range can be determined as [8.25, 13.75] m/s.

**2.7.2. The First Round of Rotation Transformation.** The green center point of the coordinated phase at intersection  $I_1$  is defined as the reference point  $O_{L1}$  of the coordinate system of the time-space diagram. A horizontal line from the reference point  $O_{L1}$  can be drawn, and then, it is rotated until the cotangent of its angle with the  $x$ -axis equals the corridor progression design speed  $V$ . We can define the rotated ray as the initial PTC line  $L_1$  and assign the value of  $V$  to adjusted speed  $V_{(1, 1)}$ . Then, the first rotation transformation in the first round is completed.

The intersection  $I_2$  is selected as the current coordinated intersection. According to PTC line  $L_1$  obtained by the first rotation transformation, the crossing point of the PTC line  $L_1$  and the timeline of  $I_2$  is marked as  $O_{L2}$ . The nearest crossing point of the coordinated phase horizontal red (green) center line at  $I_1$ , and the timeline of  $I_2$  is identified and recorded as  $O_{R2}$  ( $O_{G2}$ ), as shown in Figure 11(a). The time difference  $T_{R2}$  between points  $O_{L2}$  and  $O_{R2}$  is 8 s, and the time difference  $T_{G2}$  between points  $O_{L2}$  and  $O_{G2}$  is 32 s. Then,  $f_2 = -24$  s.

Ray  $l_{R2}$  does not pass through the red interval of the coordinated phase, and  $f_2 < 0$ , and then,  $I_2$  can form a backstepping coordinated mode with  $I_1$ , where  $F_2 = 1$ .

The corresponding adjusted speed  $V_{R2}$  of  $l_{R2}$  is 8.75 m/s, within the adjusted speed range [8.25, 13.75] m/s. Therefore,  $l_{R2}$  is defined as PTC line  $L_2$ , and the value of  $V_{R2}$  is assigned to the adjusted speed  $V_{(1, 2)}$ . Up to now, the second rotation transformation in the first round is finished.

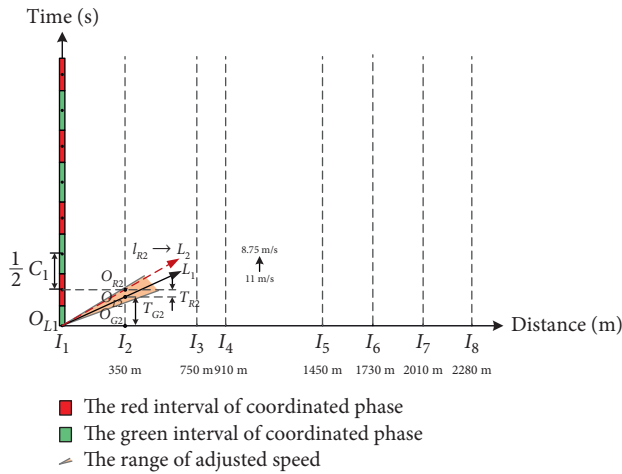
Repeat the above steps to complete the 3rd to 8th rotation transformation in the first round in turn. Then, the coordinated mode between  $I_1$  and  $I_3, I_4, I_5, I_6, I_7$ , and  $I_8$  can be determined, respectively. The coordination process is shown in Figures 11(b)–11(g).

After the first round of rotation transformation,  $I_2, I_5$ , and  $I_8$  form a backstepping coordinated mode with  $I_1$ ,  $F_2 = F_5 = F_8 = 1$ , whereas the intersections  $I_3, I_4, I_6$ , and  $I_7$  form a synchronous coordinated mode with  $I_1$ ,  $F_3 = F_4 = F_6 = F_7 = 0$ . The adjusted speed is 11.40 m/s, and the time-space diagram obtained after the first round of rotation transformation is shown in Figure 11(h). The corresponding bandwidth is 24 s, and the bandwidth ratio is 30.0%.

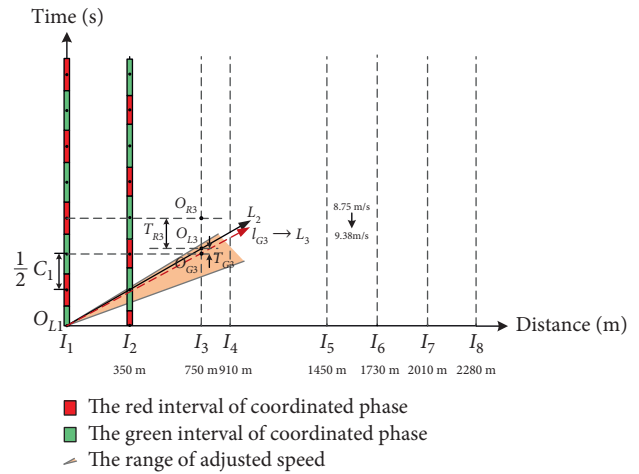
**2.7.3. The Second Round of Rotation Transformation.** According to the adjusted speed  $V_{(1, 8)}$  finally obtained after the first round of rotation transformation, the end PTC line of vehicle band  $L_{E0}$  and the beginning PTC line of vehicle band  $L_{B0}$  of the outbound progression can be drawn, as shown in Figure 12. Meanwhile, the bottleneck intersections of the end PTC line are put into the set  $S_{E0} = \{I_7\}$ . Also, the bottleneck intersections of the beginning PTC line are put into the set  $S_{B0} = \{I_3\}$ .

The set  $S_{B0}$  and  $S_{E0}$  do not meet any conditions for stopping rotation. Then, we have to complete the first rotation transformation in the second round.

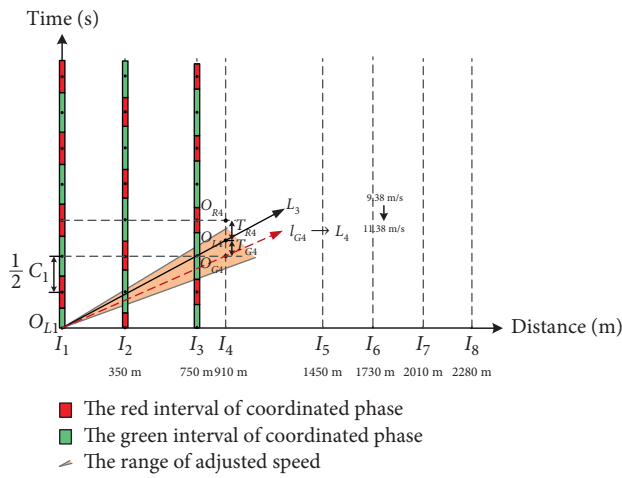
Because the only bottleneck intersection of the end PTC line  $I_7$  is located downstream of the only bottleneck intersection of the beginning PTC line  $I_3$ , it corresponds to the rotation direction judgment Scenario 2. Therefore, the PTC



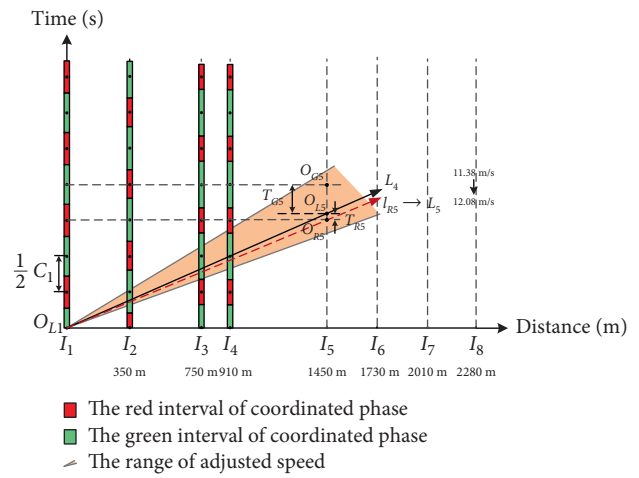
(a)



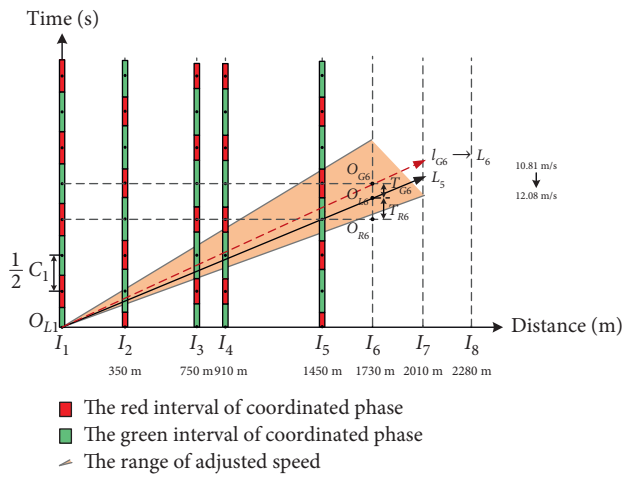
(b)



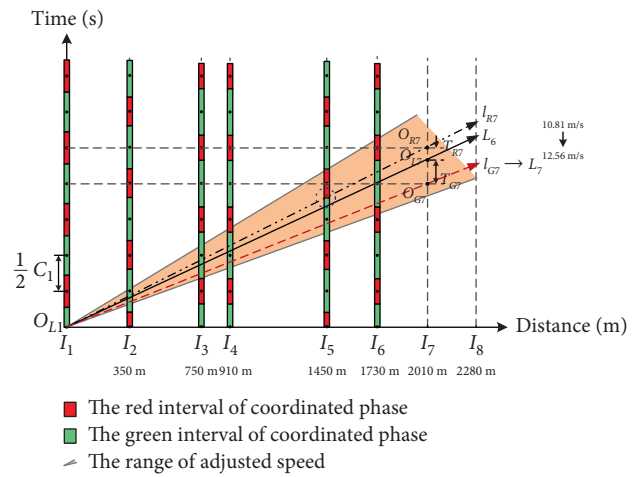
(c)



(d)



(e)



(f)

FIGURE 11: Continued.

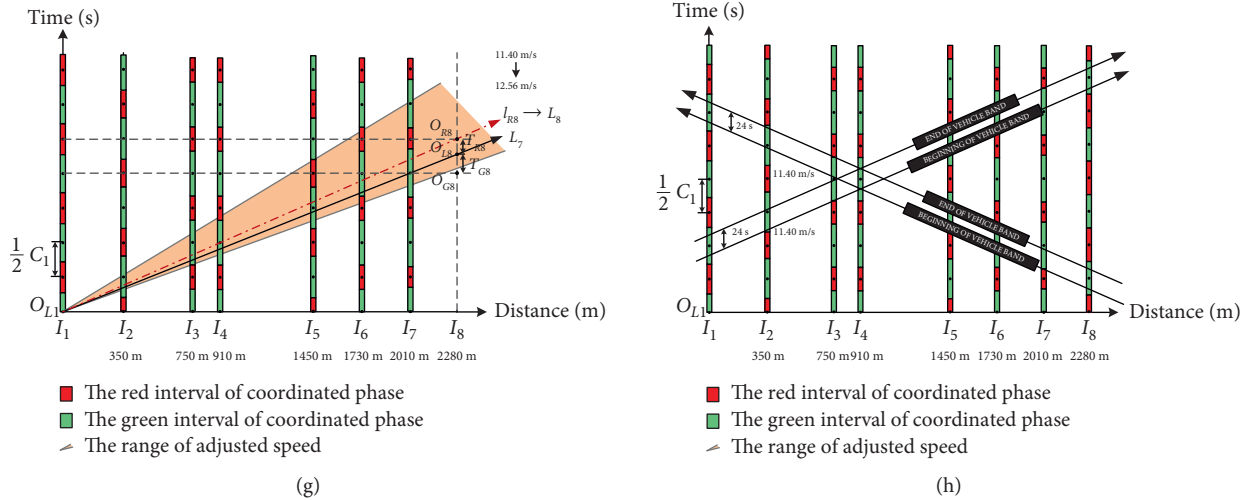


FIGURE 11: The coordination process of the first round of rotation transformation. (a) The 2nd rotation transformation. (b) The 3rd rotation transformation. (c) The 4th rotation transformation. (d) The 5th rotation transformation. (e) The 6th rotation transformation. (f) The 7th rotation transformation. (g) The 8th rotation transformation. (h) The obtained time-space diagram.

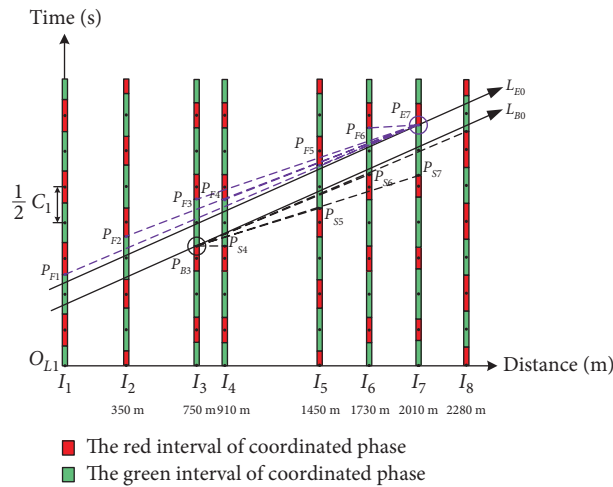


FIGURE 12: The second round of rotation transformation.

line can be rotated clockwise. Furthermore, the most upstream bottleneck point  $P_{E7}$  is selected as the rotation point of the end PTC line, whereas the most downstream bottleneck point  $P_{B3}$  is selected as the rotation point of the beginning PTC line.

As shown in Figure 12, we can connect points  $P_{E7}$  ( $P_{B3}$ ) and  $P_{Fg}$  ( $1 \leq g \leq 6$ ) ( $P_{Sh}$  ( $4 \leq h \leq 8$ )) to form the rotation line  $L_{Fg}$  ( $L_{Sh}$ ) and determine whether the corresponding adjusted speed  $V_{Fg}$  ( $V_{Sh}$ ) is within the adjusted speed range. The calculation results are shown in Table 1.

It can be seen from Table 1 that  $S_{V1} = \{V_{F1}, V_{F2}, V_{F4}, V_{F5}, V_{S6}, V_{S8}\}$ . Then, select the minimum value  $V_{F1}$  in  $S_{V1}$  as the adjusted speed  $V_{(2, 1)}$ . Then,  $V_{(2, 1)}$  is 12.11 m/s.

According to  $V_{(2,1)}$ , the end PTC line  $L_{E1}$  and the beginning PTC line  $L_{B1}$  of the outbound progression can be drawn. Determine the bottleneck intersections set of the end PTC line  $S_{E1} = \{I_1, I_7\}$  and the beginning PTC line bottleneck intersections set  $S_{B1} = \{I_3\}$ , according to  $L_{E1}$  and  $L_{B1}$ . Then,

the updating of the sets of bottleneck intersections is completed, as shown in Figure 13.

The sets  $S_{E1}$  and  $S_{B1}$  meet one of the conditions for stopping rotation. At this time,  $\exists I_1 \in S_{E1}, I_3 \in S_{B1}, I_7 \in S_{E1}$ , that is, there are two bottleneck intersections of the end PTC line located upstream and downstream of a bottleneck intersection of the beginning PTC line, which satisfies Condition 2 for stopping the rotation. Therefore, we can stop the further rotation transformation of the PTC line, and the second round of rotation transformation is finished.

**2.7.4. Scheme Generation.** The optimal adjusted speed  $V_B$  is 12.11 m/s, according to the final PTC line obtained after the second round of rotation transformation. Combined with the progression design speed  $V$ , it can be calculated that the optimal common signal cycle  $C_B$  is 88 s.

TABLE 1: The optional vehicle speed set during the first rotation transformation in the second round.

Bottleneck points		The start point/endpoint of the coordinated phase		Adjusted speed (m/s)	Optional vehicle speed $S_{V1}$
$P_{E7}$	(2010, 268)	$P_{F1}$	(0, 102)	$V_{F1} = 12.11$	$V_{F1} \in S_{V1}$
		$P_{F2}$	(350, 144)	$V_{F2} = 13.39$	$V_{F2} \in S_{V1}$
		$P_{F3}$	(750, 186)	$V_{F3} = 15.37$	$V_{F3} \notin S_{V1}$
		$P_{F4}$	(910, 186)	$V_{F4} = 13.41$	$V_{F4} \in S_{V1}$
		$P_{F5}$	(1450, 224)	$V_{F5} = 12.73$	$V_{F5} \in S_{V1}$
		$P_{F6}$	(1730, 266)	$V_{F6} = 140.00$	$V_{F6} \notin S_{V1}$
$P_{B3}$	(750, 134)	$P_{S4}$	(910, 134)	$V_{S4} = \infty$	$V_{S4} \notin S_{V1}$
		$P_{S5}$	(1450, 176)	$V_{S5} = 16.67$	$V_{S5} \notin S_{V1}$
		$P_{S6}$	(1730, 214)	$V_{S6} = 12.25$	$V_{S6} \in S_{V1}$
		$P_{S7}$	(2010, 212)	$V_{S7} = 16.15$	$V_{S7} \notin S_{V1}$
		$P_{S8}$	(2280, 260)	$V_{S8} = 12.14$	$V_{S8} \in S_{V1}$

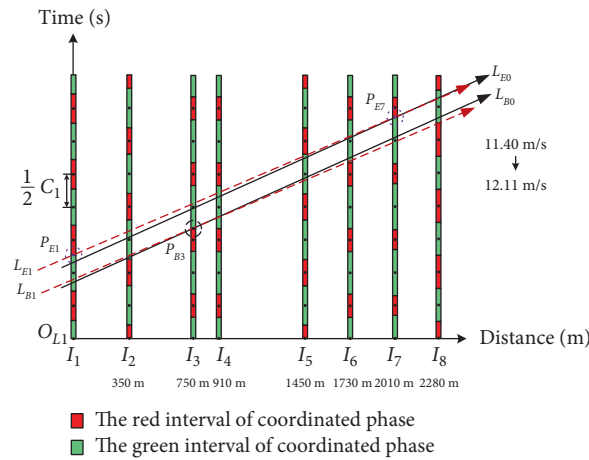


FIGURE 13: Updating the sets of bottleneck intersections.

According to the coordinated mode of each intersection and the optimal common signal cycle obtained after the two rounds of rotation transformation, combined with the green signal ratio distribution scheme, the green interval of the coordinated phase and absolute offset of each intersection can be calculated. The time-space diagram finally obtained is shown in Figure 14. The corresponding progression bandwidth is 32.85 s, and the bandwidth ratio reaches 37.3%.

**2.7.5. Programmatic Solution.** The proposed solution logic steps can be solved graphically and programmed to quickly obtain an optimal signal coordination control scheme. The above case can be solved by python programming to obtain the optimal signal coordination control scheme. The experimental environment of the model solver is the Windows10 64-bit operating system, and the CPU is intel i5-6600 3.30 GHz. The optimal signal coordination control scheme obtained by programming is the same as the graphical way, with a solution time of only  $2.3 \times 10^{-3}$  s.

**2.8. Model Comparison.** The MAXBAND model [7] and the improved algebraic method [20] can also be used to design the corridor progression schemes for the same case, and the calculation results are shown in Table 2. As for the

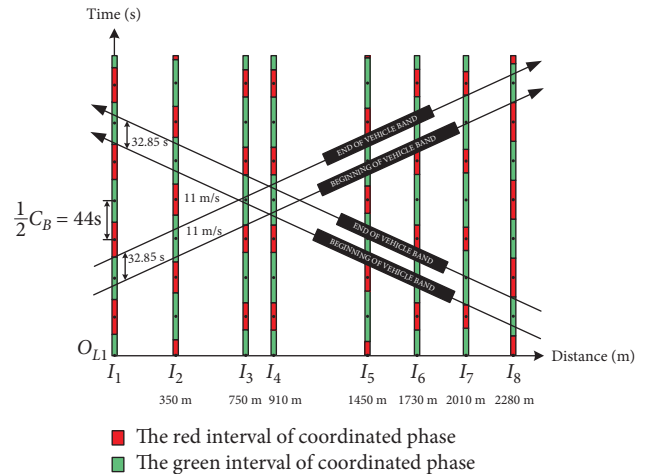


FIGURE 14: The finally obtained time-space diagram.

coordinated mode between intersection  $I_1$  and other intersections, the graphical method proposed in this study is consistent with the improved algebraic method and the MAXBAND model. As for the bandwidth ratio, the proposed graphical method can obtain 37.3% in both coordination directions, which is higher than the improved



TABLE 2: The comparison table of the coordination effects of various models.

Model	Intersection	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$	$I_8$
Proposed graphical method (MAXBAND model)	Coordinated mode	$S^*$	$B^*$	$S^*$	$S^*$	$B^*$	$S^*$	$S^*$	$B^*$
	Absolute offset (s)	63.8	17.6	59.4	59.4	17.6	59.4	57.2	22.0
	Common signal cycle (s)	88							
	Bandwidth ratio (%)	37.3							
Improved algebraic method	Coordinated mode	$S^*$	$B^*$	$S^*$	$S^*$	$B^*$	$S^*$	$S^*$	$B^*$
	Absolute offset (s)	66.0	18.2	61.4	61.4	18.2	61.4	59.2	22.8
	Common signal cycle (s)	91							
	Bandwidth ratio (%)	30.5							

$S^*$  = "synchronous coordinated mode" and  $B^*$  = "backstepping coordinated mode".

algebraic method. Moreover, it also achieves the global optimal coordination effect, the same as the MAXBAND model.

Using the LINGO programming solution, the MAXBAND model can be solved in 3 s. Meanwhile, the proposed graphical method obtains the global optimal scheme in  $2.3 \times 10^{-3}$  s, which means the solving efficiency is more than a thousand times higher than the MAXBAND model.

It can be seen from Table 2 that for the symmetrical corridor progression design, the method proposed in this study has the advantages of solid operability, solving simplicity, and a small amount of calculation. It can obtain an ideal coordination effect and be solved by a rule-guided graphical method with a limited number of steps.

### 3. Conclusions

In summary, a symmetrical bidirectional corridor progression method based on graphical optimization theory is established in this study. According to the internal relationship between the PTC lines rotation transformation and the progression bandwidth, a step-by-step optimization process of symmetrical corridor progression based on the rotation transformation of the PTC lines is proposed. After a two-round rotation transformation of the PTC line, the comprehensive optimization of the intersection coordinated mode, common signal cycle, and offsets are realized.

In the first round of PTC line rotation transformation, based on the existing graphical method designing flow, we have innovatively added a process for judging whether the PTC line passes through the red interval of the intersections with a determined coordinated mode or not, achieving a more optimized choice of coordinated mode at each intersection. Meanwhile, we have innovatively proposed the second round of PTC line rotation transformation. In the second round of PTC line rotation transformation, the regular bandwidth pattern during the rotation transformation of the PTC line is explored. Innovative rules for rotation transformation stop conditions and selection of the rotation points and direction are given to achieve the re-optimization of the common signal cycle.

The proposed graphical method can be used to obtain a signal coordination control scheme by a rule-guided graphical method and be programmed. It can be seen from the case study that the bandwidth ratio obtained by the

first round of rotation transformation of the proposed graphical method is 30.0%, with the optimization of the coordinated mode and the initial offsets. Then, through the second round of rotation transformation, the re-optimization of the common signal cycle and the offsets increases the bandwidth ratio to 37.3%. Finally, the effect of the corridor progression control scheme, which is better than the improved algebraic method, is ultimately the same as the global optimal result of the MAXBAND model with a more than a thousand times higher solving efficiency.

The graphical method proposed in this study has the advantages of intuitive readability, strong operability, and is particularly suitable for engineering applications. The horizontal and vertical scaling of the time-space diagram is one of the few attempts to standardize a graphic solution to corridor traffic coordination control. However, extending the graphical method to the corridor progression under the asymmetrical phasing and constructing a more applicable graphical method for corridor progression will be important for future follow-up research.

### Notations

$a_p$ :	The ideal distance between intersection $I_p$ and $I_1$
$b_m$ :	The bandwidth obtained after the $m$ th rotation transformation in the second round
C:	The common signal cycle
$C_1$ ( $C_B$ ):	The initial (optimal) common signal cycle
$C_{\max}$ ( $C_{\min}$ ):	The maximum (minimum) common signal cycle
$d_p$ :	The actual distance between intersection $I_p$ and $I_{p+1}$ ( $1 \leq p \leq n-1$ )
$d_p'$ :	The adjusted distance between intersection $I_p$ and $I_{p+1}$ ( $1 \leq p \leq n-1$ )
$D_p$ :	The adjusted distance between the nonbenchmark intersection $I_p$ and the benchmark intersection $I_1$
$\hat{D}_m$ :	The difference between the abscissa of the bottleneck rotation point of the beginning and end PTC line in the $m$ th rotation transformation in the second round
$f_p$ :	The primary judgment factor for the coordinated mode of intersection $I_p$

$f_{G(p,q)}$ ( $f_{R(p,q)}$ ):	The judgment factor of whether ray $l_{Gp}$ ( $l_{Rp}$ ) passes through the red interval of the intersection $I_q$ ( $1 < q < p$ ) or not	$V$ :	The corridor progression design speed
$F_p$ :	The coordinated mode judgment factor of intersection $I_p$	$V_{(i,j)}$ :	The adjusted speed obtained after the $j$ th rotation transformation in the $i$ th round
$F_A$ :	The termination decision parameter of the second round of rotation transformation	$V_B$ :	The optimal adjusted speed
$I_p$ :	The $p$ th signalized intersection along the corridor in the outbound direction	$V_{Fk}$ ( $V_{Sk}$ ):	The adjusted speed corresponding to the rotation line $L_{Fk}$ ( $L_{Sk}$ )
$K_{Dm}$ :	The rotation direction judgment factor in the $m$ th rotation transformation in the second round	$V_{Gp}$ ( $V_{Rp}$ ):	The adjusted speed corresponding to the line $l_{Gp}$ ( $l_{Rp}$ )
$K_E$ ( $m-1,p$ )	The judgment factors of the bottleneck intersection of the end and beginning PTC lines	$x_p$ :	The abscissa of intersection $I_p$ in the time-space diagram
$(K_{B(m-1,p)})$ :		$y_{Gp}$ ( $y_{Rp}$ ):	The ordinate of point $O_{Gp}$ ( $O_{Rp}$ ) in the time-space diagram
$l_{Gp}$ ( $l_{Rp}$ ):	A ray passes through point $O_{Gp}$ ( $O_{Rp}$ ) started at point $O_{L1}$	$y_{G(p,q)}$ ( $y_{R(p,q)}$ ):	The ordinate of the green center point of $I_q$ , which is closest to the intersection of ray $l_{Rp}$ ( $l_{Gp}$ ) and the timeline of $I_q$ ( $1 < q < p$ )
$L_p$ :	The PTC line obtained after the $p$ th rotation transformation in the first round	$y_{Lp}$ :	The ordinate of point $O_{Lp}$ in the time-space diagram
$L_{Bm}$ ( $L_{Em}$ ):	The beginning (end) PTC line obtained after the $m$ th rotation transformation in the second round	$\lambda_p$ :	The split of the coordinated phase of the intersection $I_p$ .
$L_{Fk}$ ( $L_{Sk}$ ):	The rotation line formed by connecting the bottleneck rotation point of the end and beginning PTC line and the coordinated phase endpoint $P_{Fk}$ (start point $P_{Sk}$ )		
$n$ :	The number of signalized intersections on the corridor		
$O_p$ :	The absolute offset of the intersection $I_p$ ( $1 \leq p \leq n$ )		
$O_{Gp}$ ( $O_{Rp}$ ):	The nearest crossing point of the coordinated phase horizontal green (red) center line at the intersection $I_1$ and the timeline of $I_p$		
$O_{Lp}$ :	The crossing point of PTC line $L_{p-1}$ and the timeline of intersection $I_p$		
$P_{Ej}$ ( $P_{Bi}$ ):	The bottleneck point of the end (beginning) PTC line at the bottleneck intersection $I_j$		
$P_{Fk}$ ( $P_{Sk}$ ):	The green interval end (start) point of the coordinated phase that is crossed by the PTC line at intersection $I_k$		
$R_m$ :	The ratio of the bandwidth obtained after the $m$ th rotation transformation in the second round		
$S_{Em}$ ( $S_{Bm}$ ):	The set of bottleneck intersections of the end (beginning) PTC line obtained after the $m$ th rotation transformation in the second round		
$S_{Vm}$ :	The set of optional vehicle speed obtained after the $m$ th rotation transformation in the second round		
$T_{Gp}$ ( $T_{Rp}$ ):	The time difference between points $O_{Lp}$ and $O_{Gp}$ ( $O_{Rp}$ )		
$T_{R(p,q)}$	The red interval crossing amount of $l_{Rp}$ ( $l_{Gp}$ ) in $I_q$ ( $1 < q < p$ )		
$(T_{G(p,q)})$ :			
$v_p$ :	The progression design speed between intersection $I_p$ and $I_{p+1}$ ( $1 \leq p \leq n-1$ )		

## Data Availability

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

## Conflicts of Interest

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

## Acknowledgments

This work was supported by the State Key Laboratory of Subtropical Building Science in China, and the PacTrans in University of Washington. This work was supported by the National Natural Science Foundation of China (61773168 and 52172326) and the Guangdong Basic and Applied Basic Research Foundation (2021A1515010727 and 2020B1515120095).

## References

- [1] D. Li, Y. Song, and Q. Chen, "Bilevel programming for traffic signal coordinated control considering pedestrian crossing," *Journal of Advanced Transportation*, vol. 2020, Article ID 3987591, 18 pages, 2020.
- [2] W. Ma, L. Zou, K. An, N. H. Gartner, and M. Wang, "A partition-enabled multi-mode band approach to arterial traffic signal optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 313–322, 2019.
- [3] K. Florek, "Arterial traffic signal coordination for general and public transport vehicles using dedicated lanes," *Journal of Transportation Engineering, Part A: Systems*, vol. 146, no. 7, Article ID 04020051, 2020.
- [4] R. Peñabaena-Niebles, V. Cantillo, J. L. Moura, and A. Ibeas, "Design and evaluation of a mathematical optimization model for traffic signal plan transition based on social cost function," *Journal of Advanced Transportation*, vol. 2017, Article ID 1943846, 12 pages, 2017.

- [5] J. T. Morgan and J. D. C. Little, "Synchronizing traffic signals for maximal bandwidth," *Operations Research*, vol. 12, no. 6, pp. 896–912, 1964.
- [6] N. H. Gartner, S. F. Assman, F. Lasaga, and D. L. Hou, "A multi-band approach to arterial traffic signal optimization," *Transportation Research Part B: Methodological*, vol. 25, no. 1, pp. 55–74, 1991.
- [7] J. D. C. Little, "The synchronization of traffic signals by mixed-integer linear programming," *Operations Research*, vol. 14, no. 4, pp. 568–594, 1966.
- [8] N. Papola, "Bandwidth maximization: split and unsplit solutions," *Transportation Research Part B: Methodological*, vol. 26, no. 5, pp. 341–356, 1992.
- [9] Z. Tian and T. Urbanik, "System partition technique to improve signal coordination and traffic progression," *Journal of Transportation Engineering*, vol. 133, no. 2, pp. 119–128, 2007.
- [10] G. Dai, H. Wang, and W. Wang, "Signal optimization and coordination for bus progression based on MAXBAND," *KSCE Journal of Civil Engineering*, vol. 20, no. 2, pp. 890–898, 2016.
- [11] E. C. P. Chang, S. L. Cohen, C. Liu, N. A. Chaudhary, and C. Messer, "MAXBAND-86: program for optimizing left-turn phase sequence in multi-arterial closed networks," *Transportation Research Record*, vol. 1181, no. 1, pp. 61–67, 1988.
- [12] K. Lu, X. Zeng, L. Li, and J. Xu, "Two-way bandwidth maximization model with prorated impact factor for unbalanced bandwidth demands," *Journal of Transportation Engineering*, vol. 138, no. 5, pp. 527–534, 2012.
- [13] X. Yang, Y. Cheng, and G.-L. Chang, "A multi-path progression model for synchronization of arterial traffic signals," *Transportation Research Part C: Emerging Technologies*, vol. 53, no. 1, pp. 93–111, 2015.
- [14] C. Zhang, Y. Xie, N. H. Gartner, C. Stamatidis, and T. Arsava, "AM-Band: an Asymmetrical Multi-Band model for arterial traffic signal coordination," *Transportation Research Part C: Emerging Technologies*, vol. 58, no. 1, pp. 515–531, 2015.
- [15] L. Zhang, Z. Song, X. Tang, and D. Wang, "Signal coordination models for long arterials and grid networks," *Transportation Research Part C: Emerging Technologies*, vol. 71, no. 1, pp. 215–230, 2016.
- [16] H. Zhou, H. G. Hawkins, and Y. Zhang, "Arterial signal coordination with uneven double cycling," *Transportation Research Part A: Policy and Practice*, vol. 103, no. 1, pp. 409–429, 2017.
- [17] T. Arsava, Y. Xie, and N. H. Gartner, "Arterial progression optimization using OD-BAND: case study and extensions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2558, no. 1, pp. 1–10, 2016.
- [18] T. Arsava, Y. Xie, and N. Gartner, "OD-NETBAND: an approach for origin-destination based network progression band optimization," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 18, pp. 58–70, 2018.
- [19] J. M. Xu, *Traffic Management and Control*, China Communications Press, Beijing, China, 2007.
- [20] K. Lu, J. M. Xu, and R. M. Ye, "Improvement of classical algebraic method of signal timing for arterial road coordinate control," *Journal of Highway and Transportation Research and Development*, vol. 1, pp. 120–124, 2009.
- [21] D. H. Wang, X. R. Yang, and X. M. Song, "Improvement of classical numerical method for arterial road signal coordinate control," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 41, no. 1, pp. 29–34, 2011.
- [22] K. Lu, J. M. Xu, and Y. S. Li, "Algebraic method of arterial road coordinate control for bidirectional green wave under signal design mode of one-phase-one-approach," *China Journal of Highway and Transport*, vol. 23, no. 3, pp. 95–101, 2010.
- [23] L. N. Ji and Q. H. Song, "Bidirectional green wave coordinate control for arterial road under asymmetric signal mode," *Journal of Highway and Transportation Research and Development*, vol. 28, no. 10, pp. 96–101, 2011.
- [24] K. Lu, Y. Y. Liu, H. Wu, and J. H. Huang, "Algebraic method of bidirectional green wave coordinated control under asymmetric traffic conditions," *China Journal of Highway and Transport*, vol. 28, no. 6, pp. 95–103, 2015.
- [25] China Highway Association, *Traffic Engineering Manual*, China Communications Press, Beijing, China, 1998.
- [26] S. D. Lu and L. Cheng, "Optimization of bidirectional green wave coordinated control graphical method under asymmetric phase sequence mode," *Journal of Highway and Transportation Research and Development*, vol. 32, no. 1, pp. 128–132, 2015.

## Research Article

# Mining Travel Time of Airport Ferry Network Based on Historical Trajectory Data

Cong Ding <sup>1</sup>, Jun Bi <sup>2</sup>, Dongfan Xie,<sup>1,2</sup> Xiaomei Zhao,<sup>1,2</sup> and Yi Liu<sup>1</sup>

<sup>1</sup>School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China

<sup>2</sup>Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Beijing Jiaotong University, Beijing, China

Correspondence should be addressed to Jun Bi; [jbi@bjtu.edu.cn](mailto:jbi@bjtu.edu.cn)

Received 5 August 2021; Accepted 8 October 2021; Published 31 October 2021

Academic Editor: Ming Xu

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

An airport ferry vehicle is a ground service vehicle used to transfer passengers between the far apron and the terminal. The travel time of ferry tasks in the airport ferry network is an important decision-making basis for ferry vehicle scheduling. This paper presents a graph-based method to mine the travel time between nodes in the airport ferry network. Firstly, combined with map and trajectory information, the method takes the terminal boarding gates, parking lots, and remote stands as road network nodes to build a complete airport ferry road network. Then, this paper uses big data processing technology to identify the travel time between regional connection nodes by data fusion through the temporal and spatial relationship between flight schedule and ferry vehicle GPS travel trajectory. Finally, the Floyd shortest path algorithm in graph theory is used to obtain the shortest path and travel time of all OD points. The experimental results show that all the ferry times calculated by the method proposed in this paper can better reflect the actual driving situation. This method saves the manpower, material resources, and time cost of on-site investigation and lays a foundation for the scheduling of ferry vehicles.

## 1. Introduction

When the aircraft is parked at a far apron, passengers need ferry vehicles to transfer between the aircraft and the terminal. After the airport operation center obtains the flight schedule and gate assignment [1, 2] results in a period of time, it will consider the travel time consumed by the ferry task in the road network and reasonably arrange the ferry vehicles for each flight parked in a remote stand. Only a reasonable ferry scheduling plan can ensure the normal implementation of the flight schedule. At present, the scheduling of ferry vehicles in many airports mainly depends on experienced staff. The efficiency of the scheduling results is difficult to be ensured, which is easy to lead to flight delays. Therefore, how to establish the optimal scheduling model of ferry vehicles by scientific means and improve the service level of airport ferry vehicles has attracted the attention of some scholars.

The premise of ferry vehicle scheduling modeling is to know the travel time information between OD points in the

ferry network. However, due to the closed environment in the airport flight area, nonstaff cannot enter the interior for field measurement. With the rapid development of smart airports, in order to realize the digitization and intelligence of airport management and meet the needs of airport safety management, some airports have established ground service vehicle GPS real-time monitoring and management systems to realize vehicle positioning, real-time tracking, track playback, cross-border alarm, and other functions. The historical GPS track data of ferry vehicles recorded by the system can reflect the time information and spatial information of vehicles in the process of performing flight support tasks. Therefore, using the trajectory data to mine and extract the travel time of the airport ferry network is a new idea.

This paper proposed a novel graph-based method to mine the travel time in the airport ferry network. The research work is mainly divided into three aspects. Firstly, combined with the map and trajectory information, this

paper takes the terminal boarding gates, parking lots, and remote stands as the road network nodes to build a complete airport ferry road network. Then, the nodes are divided into different regions, and the travel time between regional connecting nodes is identified by data fusion through the temporal and spatial relationship between flight schedule and ferry vehicle GPS travel trajectory. Finally, the Floyd shortest path algorithm in graph theory is used to obtain the shortest path and travel time of all OD points. The research results of this paper can save the manpower, material resources, and time cost of on-site investigation and provide a decision-making basis for the optimal scheduling model of ferry vehicles.

The rest of this paper is organized as follows. In Section 2, the literature on the optimal scheduling of airport ground support vehicles and GPS trajectory mining is reviewed. Section 3 takes Kunming Changshui International Airport in China as the research object and introduces the method of constructing the airport ferry network. In Section 4, the calculation method of OD point travel time in the road network is designed. The example verification and results are given in Section 5. Finally, conclusions and future research directions are proposed in Section 6.

## 2. Literature Review

With the rapid development of civil aviation, the disadvantages of airport management and operation relying only on manual experience are becoming more and more prominent. In order to ensure the normal operation of flights, how to improve the service level of airport ground support services through scientific means has attracted more and more attention of scholars. From the optimal operation of special vehicles providing various ground services to the overall optimization of joint scheduling of multiple vehicles, many achievements have been made. This paper summarizes the research literature of different ground service vehicles as a reference. Although there are differences in service content with ferry vehicles, there are many similarities in the optimization objectives and constraints in their scheduling.

Norin et al. [3] constructed an airport deicing vehicle optimal scheduling model aiming at minimizing the delay time and the minimum number of deicing vehicles and designed the corresponding solution algorithm based on a greedy randomized adaptive search algorithm. Du et al. [4] studied the trailer scheduling problem in flight transit service, described the problem as VRPTW, and constructed an integer programming model with minimizing the trailer operation cost as the optimization objective and the vehicle operation restriction as the constraint. Starting from the flexibility of vehicle service, Wang et al. [5] proposed a scheduling algorithm based on a greedy strategy to deal with the dynamic scheduling problem of airport refueling vehicles. With the diversification of airport ground service and considering the interaction between various vehicle services, many studies begin to pay attention to the joint scheduling of airport ground service vehicles. Padron et al. [6] constructed a double objective optimization model to minimize waiting time and turnaround time for the joint scheduling of

airport service equipment. Xu and Shao [7] evaluated the fluctuation of service equipment operation time by analyzing the airport historical data and proposed an optimization model of ground service support equipment with uncertain operation time. Fei and Shu'an [8] studied the optimization problem of airport service vehicle scheduling in peak hours and constructed a joint scheduling model of ground service vehicles to minimize the purchase cost and operation cost of service equipment. In terms of ferry vehicles, Zhao et al. [9] established an integer programming model with the goal of minimizing the number of ferry vehicles required in peak hours or a certain period and constructed a ferry vehicle sharing network, which transformed the model into the problem of maximum network flow. Han et al. [10] also proposed a ferry capacity network model, in which the directed edge indicates that the two associated nodes may be continuously served by the same ferry. The model aims to minimize the number of ferry vehicles required and is solved by the method of the graph. However, their research only roughly estimated the travel time between the terminal and the far apron area in the process of building the ferry network. For large airports, the terminal covers a wide area and is far away from different boarding gates. The unified estimation method will cause great errors in practical application.

With the development of positioning technology, many vehicles are equipped with GPS receivers. During the moving process, the vehicle continuously collects real-time information, including position, motion parameters, and positioning time, and transmits it to the data center. This type of data is called floating car data [11, 12]. The emergence of these data makes it possible to mine rich knowledge from GPS trajectory data by using big data analysis technology [13]. Palma et al. and Bhattacharya et al. [14, 15] proposed the method of analyzing motion characteristics (velocity, azimuth, acceleration, etc.) to mine important places related to people and objects. Zheng and Xie [16] proposed a trajectory mining algorithm to analyze the user's GPS trajectory data, so as to recommend personalized tourist attractions. In addition, Li et al. [17] analyzed the typical characteristics of the data of floating vehicles in the parking lot and used the DBSCAN algorithm to extract the parking lot location. Wang et al. [18] proposed the identification of key nodes and sections of urban road networks based on GPS trajectory data. In terms of path extraction, Schoredl et al. and Li et al. [19, 20] proposed two methods to extract a high-precision roadmap from GPS trajectory, which are applicable to high sampling frequency and low sampling frequency of positioning data respectively. Unlike them, Cao and Krumm [21] proposed a new gravity model to transform the original GPS track into a road network that can guide path selection. Tang et al. [22] proposed a lane-level road network information mining method based on lane number and turning rules. In addition to the region of interest and path recognition, scholars have also done a lot of research on spatiotemporal pattern extraction based on trajectory data, for example, the method of automatically extracting passenger train operation information from historical track data [23]. Dong et al. [24] proposed a study on the temporal and spatial change of

traffic accessibility under public health emergencies based on GPS trajectory. Lei et al. and Zhao et al. [25, 26] proposed a spatiotemporal analysis model to capture the motion mode of the object. Different from those, Lu et al. [27] proposed a visual analysis method to study the behavior of vehicles on a certain route. Some scholars have also studied periodic pattern recognition, such as the probabilistic periodic detection method of moving objects [28]. In general, most of the GPS data mining and analysis research by scholars is based on the road traffic network, and few studies use the GPS track of airport ferry vehicles to extract the road network structure and travel time of the flight area. Therefore, the research content of this paper is innovative.

### 3. Construction of Airport Ferry Network

This paper takes the real data of Kunming Changshui International Airport in China as the research object. The airport terminal covers an area of 548300 square meters, with 78 boarding gates, 104 remote stands, 2 entry ports, and 3 parking lots. Figure 1 shows the spatial layout of the airport flight area. The boarding gates and entry ports are distributed around the terminal, while the parking lots and remote stands are distributed at the far apron.

When a flight is assigned to an apron, the ferry vehicle needs to arrive at the corresponding stand in advance for the arriving flight. After the plane arrives, the ferry vehicle will drive along the planned route to deliver the passengers to the entrance port of the terminal. For departure flights, according to the flight schedule, the ferry vehicle needs to arrive at the boarding gate before ticket check-in. After the gate is opened, passengers will be sent to the designated parking stand. There is also a necessary transfer time between two consecutive services of the ferry vehicle, that is, the travel time required for the ferry vehicle from the end of the last service to the start of the next service. According to the arrival and departure attributes of two adjacent service flights, the transfer between two services can be divided into four categories, as shown in Table 1. After a ferry task, when the buffer time is sufficient, the ferry vehicle can go to the parking lot first and then go to the starting place of the next service when the next task is approaching. When the buffer time is insufficient, in order to avoid flight delays, the ferry vehicle can go directly to the starting point of the next service. The security efficiency of ferry vehicles is mainly affected by the transfer time. Flights with a remote stand at the airport usually change the boarding gate flexibly according to the situation. For large airports, the terminal covers a wide area and different boarding gates are far away from each other. Therefore, calculating the transfer path and time between all OD points in the whole airport is the prerequisite for the optimal scheduling of ferry vehicles.

Regarding the boarding gates and remote stands as road network nodes, directly calculating the travel time of any two points in the road network according to the coordinate information and the travel track of ferry vehicles will cause a large workload. In order to simplify the airport road network structure, firstly, the airport terminal boarding gate and apron are divided into several relatively independent areas

according to the adjacent relationship, and the spatial transfer of ferry vehicles between service points is regarded as the transfer between regions. According to the number of remote stands, apron 3 is divided into 1 to 4 parts, apron 5 is divided into 1 to 4 parts, and apron 7 is divided into 1 to 2 parts. The boarding gate area of the terminal is divided into 1 to 4 parts for the convenience of the road network structure. Figure 2 shows the historical track of ferry vehicles. Combined with the actual road network distribution of the airport, the connection relationship between regions can be obtained, as shown in Figure 3. The arrangement structure of service points in one region is relatively simple, including series and parallel. As shown in Figure 4, there are two rows of remote stands in area 5-2, which are distributed in parallel. Area 3-4 has only a single row of stands, which are distributed in series. Therefore, combined with the road network structure between regions, a complete road network structure of airport service points can be obtained, as shown in Figure 5. After obtaining the spatial connection relationship between all service points from the road network diagram of service points, any service point can communicate with other service points through the road network, and when combined with the distance information on the edge, the travel time between any two points can be calculated. Therefore, the next section will explore the travel time of each edge of the road network.

### 4. Calculation Method of Travel Time

According to the ferry road network established in the previous section, this section uses the historical GPS track data of ferry vehicles to obtain the weight of each edge. For the convenience of description, the following definitions are made in this paper: all nodes in the ferry network are represented by  $V = [D_1, D_2, \dots, D_i, \dots, D_n]$ , where  $D_i$  represents different regions. The nodes in one region are divided into internal nodes and connection nodes,  $D_i = [P_{i1}, P_{i2}, \dots, P_{im}, Q_{i1}, Q_{i2}, \dots, Q_{ik}]$ , the internal node  $P_{im}$  is only connected with the nodes in the region, and the connection node  $Q_{ik}$  can be connected with the nodes in other regions. The GPS tracks of all ferry vehicles are represented by  $R = [G_1, G_2, \dots, G_i, \dots, G_n]$ , where  $G_i$  represents the track set of ferry vehicle  $i$ , including the longitude and latitude coordinates  $[\text{lng}_i, \text{lat}_i, t]$  of the vehicle at each time. The flight set is represented as  $F = [f_1, f_2, \dots, f_i, \dots, f_n]$ , where the arrival and departure attributes, time, and OD point of the ferry task of flight  $f_i$  are known.

**4.1. Travel Time between Terminal and Far Apron.** During the flight support task, the ferry vehicle will travel between the boarding gate area and the far stand area. Therefore, the path and time of the connecting edge between the gate area and the far stand area can be mined according to the temporal and spatial correlation between the flight schedule and the GPS track of the ferry vehicle. For arrival flights, the service flow of ferry vehicles is shown in Figure 6.  $t_i^{\text{ETA}}$  indicates the estimated arrival time of the flight. The ferry service starts from the time of arriving at the remote stand in advance, and

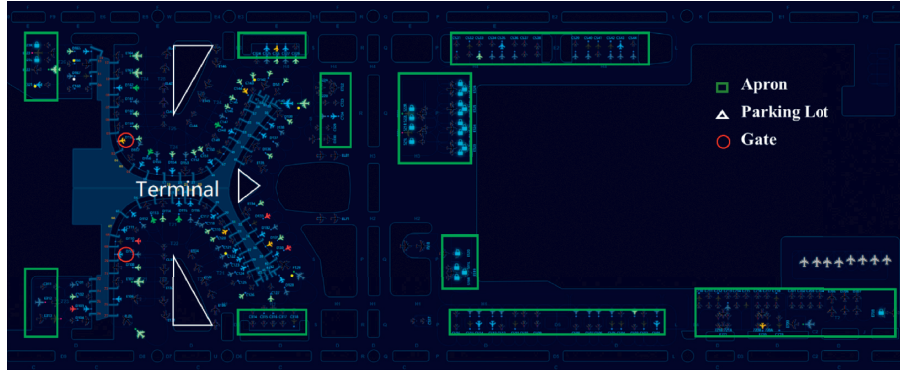


FIGURE 1: Spatial layout of the airport flight area.

TABLE 1: OD point conversion of continuous flights.

Flight 1	Flight 2	Origin	Destination
Arrive flight	Arrive flight	Arrive gate	Remote stand
Arrive flight	Departure flight	Arrive gate	Departure gate
Departure flight	Departure flight	Remote stand	Remote stand
Departure flight	Arrive flight	Remote stand	Departure gate

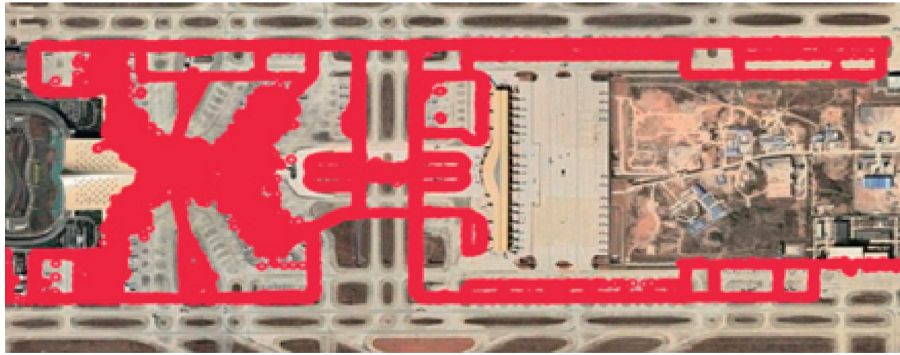


FIGURE 2: Historical GPS track of ferry vehicles.

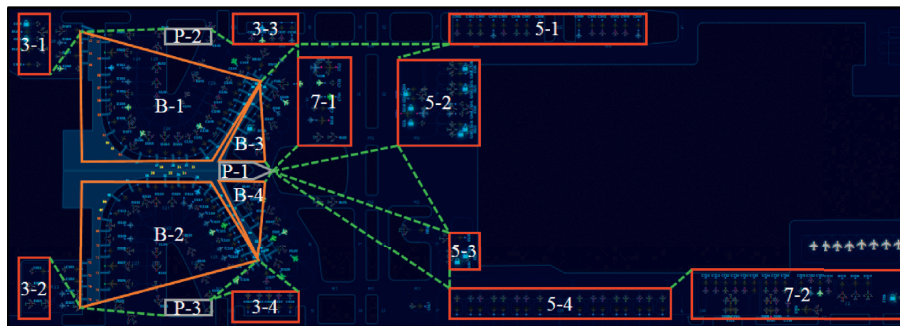


FIGURE 3: Road network connection between the regions.

the service start time  $TS_i^A$  can be calculated according to equation (1). After the flight arrives at the remote stand, it takes time  $T_{\text{wait}}$  to wait for passengers to board. The ferry service ends after the passengers get off at the port, and the service end time  $TE_i^A$  can be calculated according to equation (2).

$$TS_i^A = t_i^{\text{ETA}} - T_{\text{adv}}^A, \quad (1)$$

$$TE_i^A = t_i^{\text{ETA}} + 2 * T_{\text{wait}} + T_i^{\text{ferry}}. \quad (2)$$

Similarly, for departure flights, according to the flight schedule, the ferry vehicle arrives at the boarding gate before

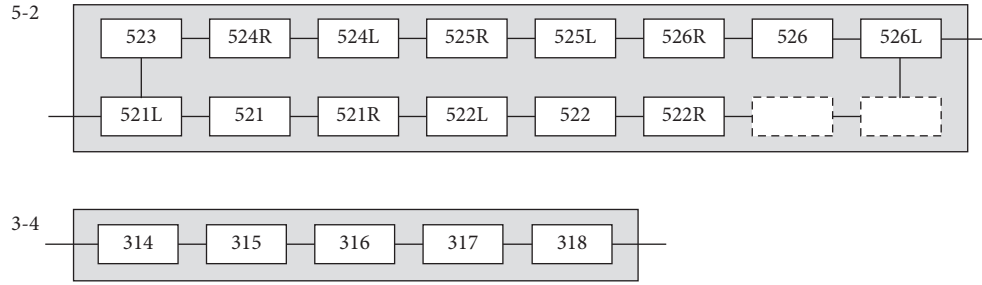


FIGURE 4: Nodes connection relationship in a region.

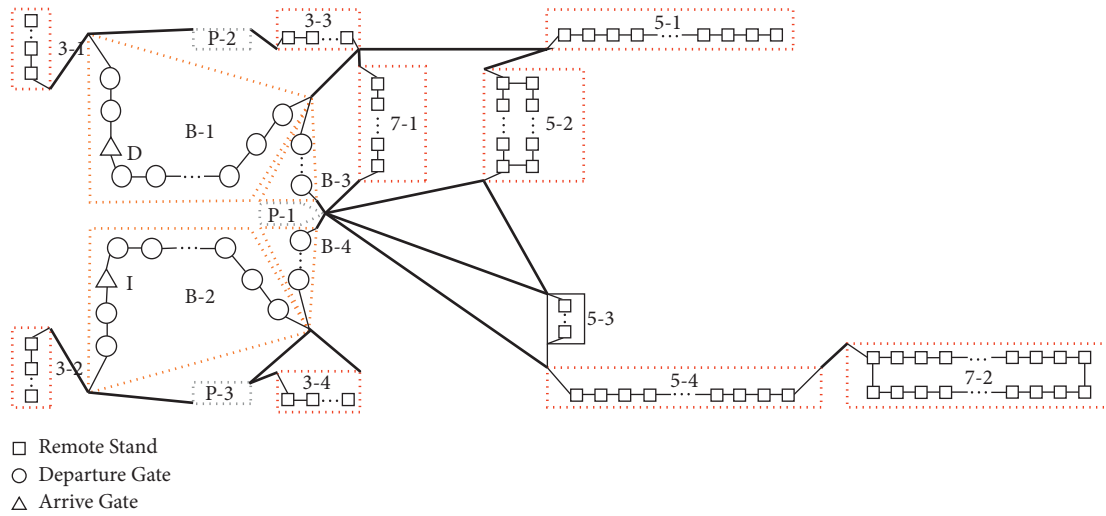


FIGURE 5: A complete structure diagram of airport ferry network.

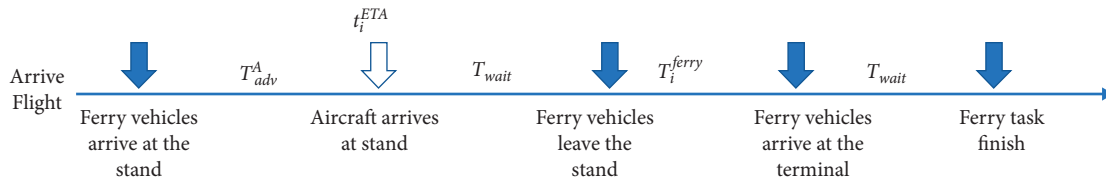


FIGURE 6: Ferry vehicle service process of arrival flight.

ticket check-in. After passengers get off at the parking stand, the service of the ferry vehicle ends. Figure 7 shows the ferry vehicle service process of departure flights. Assuming that the opening time of the boarding gate is 40 minutes (according to the service standards of China’s civil transport airport) before the departure time  $t_i^{ETD}$  of the flight, the start time of the ferry vehicle for the departure flight is calculated by equation (3) and the end time of the service is calculated by equation (4).

$$TS_i^D = t_i^{ETD} - 40 \text{ min} - T_{adv}^D, \quad (3)$$

$$TE_i^D = t_i^{ETD} - 40 \text{ min} + 2 * T_{wait} + T_i^{ferry}. \quad (4)$$

According to the ferry service process, the algorithm steps of mining the travel time  $T_i^{ferry}$  between the boarding

gate area and the apron by integrating GPS track data and flight data are as follows.

*Step 1.* According to the road network structure, find out the connecting nodes between the boarding gate area and the far apron area to form OD pairs  $[Q_{ik}, Q_{jl}]$ . Search the flight set  $F$  and find the flight  $i$  whose starting and ending point of the ferry task is  $[Q_{ik}, Q_{jl}]$ .

*Step 2.* If flight  $i$  found is a departure flight, obtain the departure time  $t_i^{ETD}$  from the flight information, take time  $t$  and distance  $m$  as the search range parameters, form a set  $Ferry_{start}$  of all ferry vehicles that appear at the boarding gate  $Q_{ik}$  in the period  $[t_i^{ETD} - 40 \text{ min} - t, t_i^{ETD} - 40 \text{ min} + t]$ , and record the time  $T_{start}$  when each ferry car leaves the search



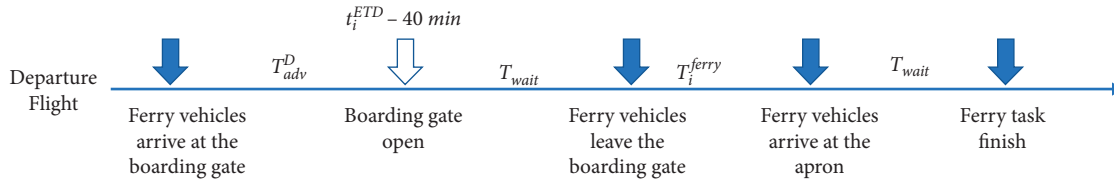


FIGURE 7: Ferry vehicle service process of departure flight.

range. Similarly, find out all the ferry vehicles that appear at the remote stand  $Q_{ji}$  in the period  $[t_i^{ETD} - 40 \text{ min} + t, t_i^{ETD}]$ , form a set  $\text{Ferry}_{\text{end}}$ , and record the time  $T_{\text{end}}$  when each ferry car enters the search range. If flight  $i$  found is an arrival flight, obtain the arrival time  $t_i^{ETA}$  from the flight information, and change the starting point to the remote stand  $Q_{ji}$ , the time range is  $[t_i^{ETA} - t, t_i^{ETA} + t]$ , the end point is the port  $Q_{ik}$ , and the time range is  $[t_i^{ETA}, t_i^{ETA} + t_{\text{max}}]$  where  $t_{\text{max}}$  represents the maximum ferry time range.

*Step 3.* Take the ferry car  $G_i$  existing in both  $\text{Ferry}_{\text{start}}$  and  $\text{Ferry}_{\text{end}}$  as the ferry car  $i$  serving the flight, and calculate the travel time  $T_{\text{end}}^i - T_{\text{start}}^i$  of the ferry car as the travel time from the boarding gate to the stand.

When judging whether the ferry car enters the search range, take the boarding gate or stand as the center of the circle, first calculate the longitude and latitude distance between the ferry car and the center of the circle, and then compare this distance with the reference distance  $m$ . If the calculated distance is less than the reference distance, judge whether the ferry car enters the search range.

**4.2. Travel Time between Different Far Aprons.** The starting and ending point of flight service is from the boarding gate to the stand, so it is impossible to obtain the time when OD points are both stands through flight information. There are two situations for the travel time of the ferry car between two target stands. The first situation is that the ferry car directly travels from the starting stand to the ending stand. At this time, the travel time represents the distance between the two stands. Another situation is that the ferry car drives from the starting stand to other places and then to the target stand. At this time, the travel time is obviously greater than the distance between the two stands. Therefore, without the mining time range, we can only intercept the GPS track of the ferry vehicle through the coordinate values of the connecting nodes in two far aprons, then count a large number of travel time values, and select the smaller value as the travel time between the aprons. In this way, the distance from the parking lot to other service points can also be obtained.

**4.3. Travel Time between Any Two Nodes.** After mining the travel time information of edges in the road network through the above method, the nodes, edges, and the weight of edges of the road network have been obtained. The travel time between any two nodes can be calculated by using the shortest path algorithm. Calculating the shortest path is a classical problem in graph theory. At present, there has been

a very mature research. The commonly used shortest path algorithms include Floyd [29], Dijkstra [30], and SPFA [31]. It is necessary to select the appropriate algorithm according to different use scenarios. From the characteristics of the algorithm, the Floyd algorithm is suitable for finding the shortest path from multiple sources, but the high time complexity makes the algorithm not suitable for road network maps with many nodes; the Dijkstra algorithm is the basis of all basic shortest path algorithms. It is the most stable algorithm, but it is suitable for finding the shortest path of a single source. It needs repeated operation to obtain the shortest path of all points; the SPFA algorithm is a queue optimization of the Bellman Ford algorithm. It is the shortest path algorithm based on BFS. It solves the problem of changing weight to negative value that the Dijkstra algorithm cannot solve. At the same time, the implementation of the algorithm is the most troublesome. The road network constructed in this paper has a total of 187 service points. Considering the scale of the road network, the positive right of way, and the need to obtain the path information between all points, the Floyd algorithm is selected to solve the shortest path.

Floyd algorithm is an algorithm that uses the idea of dynamic programming to find the shortest path between multiple source points in a given weighted graph. The algorithm has the characteristics of clear structure and fast implementation. Its core idea is as follows: for each pair of shortest vertices  $u$  and  $v$ , if there is a vertex  $w$  that makes the path from  $u$  to  $w$  and then to  $v$  shorter than the known path, update it. Considering that the nodes have been divided into different regions, this paper first uses the Floyd algorithm to calculate the shortest circuit between different regions and then calculates the shortest circuit between the nodes in the region and the connecting nodes. The algorithm flow of finding the shortest path of OD points of ferry task is as follows (Algorithm 1):

After the shortest path between regions is obtained by the above algorithm, combined with the distance from the internal node to the connecting node in the region, the shortest travel time between any two points can be obtained. The edge inside the area mainly represents the distance between two gates, and the fixed value  $C_{\text{gate}}$  can be adopted. If  $n$  stands are separated between the internal node and the connecting node, the distance is  $n * C_{\text{gate}}$ .

## 5. Experimental Results and Analysis

In order to verify the proposed travel time mining method of the ferry network, this paper uses the actual data of Changshui International Airport to verify the effectiveness

- (i) **Input:** region ferry network  $G = (Q, E)$ ,  $Q$  is the set of connection nodes representing each region, and  $E$  is the travel time between regions obtained by data mining as the weight of edges.
- (1) Initialization matrix  $M$ ,  $m_{ij}$  represents the travel time between vertices  $i$  and  $j$ . If  $i$  and  $j$  are not directly connected, the weight is set to  $\infty$ . If  $i$  and  $j$  come from the same region, the weight is set to 0.
- (2) Starting from the first node  $q_1$ , calculate that  $m_{ij}^{(1)} = \min\{m_{ij}^{(0)}, m_{i1}^{(0)} + m_{1j}^{(0)}\}$  is the shortest path from  $i$  to  $j$  that only allows the middle to pass through  $q_1$ , and update the matrix  $M$ .
- (3) Similarly, calculate that  $m_{ij}^{(k)} = \min\{m_{ij}^{(k-1)}, m_{ik}^{(k-1)} + m_{kj}^{(k-1)}\}$  is the shortest path from  $i$  to  $j$  that only allows the middle to pass through  $q_1, q_2, \dots, q_k$ , and update the matrix  $M$ .
- (4) When  $k = n$ ,  $m_{ij}$  in matrix  $M$  is the shortest path from node  $i$  to  $j$
- (ii) **Output:** matrix  $M$ .

ALGORITHM 1: Calculating the shortest path between ferry OD points by the Floyd algorithm.

TABLE 2: Flight information (partial departure flight).

Flight number	Stand	Boarding gate	Check-in time	Close time	Departure time
TV6031	153	27	2019/1/1 0:01	2019/1/1 0:13	2019/1/1 0:37
3U8837	155	17	2019/1/1 0:03	2019/1/1 0:24	2019/1/1 0:51
3U8107	326	4	2019/1/1 0:12	2019/1/1 0:36	2019/1/1 1:03
KY8335	521L	55	2019/1/1 0:15	2019/1/1 0:49	2019/1/1 0:59
CA5785	521L	55	2019/1/1 0:15	2019/1/1 0:49	2019/1/1 0:59
ZH5335	521L	55	2019/1/1 0:15	2019/1/1 0:49	2019/1/1 0:59
MU2569	109	70	2019/1/1 0:18	2019/1/1 0:32	2019/1/1 0:50
ZH3181	145	59	2019/1/1 0:23	2019/1/1 0:45	2019/1/1 0:56
KY8281	145	59	2019/1/1 0:23	2019/1/1 0:45	2019/1/1 0:56
CZ3416	141	60	2019/1/1 0:23	2019/1/1 0:34	2019/1/1 0:50

TABLE 3: GPS trajectory information of the ferry car (partial).

Car ID	GPS time	Longitude	Latitude
1	2019/1/1 12:01:03	102.932983	25.111266
1	2019/1/1 12:02:01	102.936	25.108516
1	2019/1/1 12:03:01	102.938266	25.10955
1	2019/1/1 12:04:01	102.939033	25.109433
1	2019/1/1 12:05:31	102.939066	25.109433
1	2019/1/1 12:06:29	102.939066	25.109416
1	2019/1/1 12:06:59	102.939066	25.109416
1	2019/1/1 12:08:59	102.939083	25.1094
1	2019/1/1 12:09:29	102.939066	25.1094
1	2019/1/1 12:10:59	102.93905	25.1094

and practicability of the method. Flight data from January to June 2019, historical GPS track data of ferry vehicles, and coordinate information of each parking stand and boarding gate of the airport are collected. The data samples are shown in Tables 2 to 4.

Since the airport does not record the daily scheduling historical data of ferry vehicles, we do not know which vehicle performs each task. Therefore, we need to identify them according to the data fusion method proposed in Section 4 to obtain the specific ferry vehicles corresponding to the task and the time nodes of their entry and departure starting and ending points. According to the method in Section 4, we first excavate the distance between the boarding gate area and the far apron area. Ferry vehicles will not stop too far away from the aircraft and stop at the task

site for too long, so we set the time scaling parameter  $t = 10$  min and the space distance scaling parameter  $m = 20$  m. Take boarding gate area B-1 and far apron 5-2 as examples, according to the identification results, the corresponding starting waiting time, travel time, and terminal waiting time can be calculated, and the GPS track of the vehicle can be checked back according to the identification time of the starting and ending points to obtain the support path, as shown in Figure 8. The yellow circle indicates the search range, and the time difference of these seven GPS data can be calculated as the travel time of connection nodes gate B48 and stand 521L. According to this, the results of each calculation are collected, and the mean value of all results is taken as the transfer time between the two. For the distance between different far apron areas, we count a lot of travel

TABLE 4: Latitude and longitude information of service point (partial).

Number	Boarding gates		Number	Remote stands	
	Longitude	Latitude		Longitude	Latitude
<i>b</i> 1	102.9274531	25.1011307	311	102.9290611	25.0955317
<i>b</i> 2	102.9265117	25.1012863	312	102.9295471	25.0951637
<i>b</i> 3	102.9271301	25.1010857	313	102.9300271	25.0946687
<i>b</i> 4	102.9268037	25.1012533	314	102.9359601	25.1012577
<i>b</i> 5	102.9257167	25.1015113	315	102.9362391	25.1016377
<i>b</i> 6	102.9252457	25.1018223	316	102.9364991	25.1019157
<i>b</i> 7	102.9248317	25.1021203	317	102.9367597	25.1021916
<i>b</i> 8	102.9243962	25.1024135	318	102.9370517	25.1025726
<i>b</i> 9	102.9236602	25.1022575	321	102.9222547	25.1004473
<i>b</i> 10	102.9238932	25.1027735	322	102.9216707	25.1006893

TABLE 5: Distances between areas in the road network.

Region 1		Region 2		Travel time (min)
Number	Connection node	Number	Connection node	
<i>B</i> -1	<i>b</i> 9	3-1	321	1.6
<i>B</i> -1	<i>b</i> 56	3-3	328	1.7
<i>B</i> -2	<i>b</i> 72	3-2	311	1.5
<i>B</i> -2	<i>b</i> 40	3-4	318	2.3
<i>B</i> -3	<i>b</i> 48	<i>P</i> -1	<i>p</i> 5  <i>p</i> 6	0.5
<i>B</i> -4	<i>b</i> 32	<i>P</i> -1	<i>p</i> 5  <i>p</i> 6	0.5
<i>P</i> -1	<i>p</i> 5  <i>p</i> 6	7-1	330	3.1
<i>P</i> -1	<i>p</i> 5  <i>p</i> 6	5-2	521L	6.5
<i>P</i> -1	<i>p</i> 5  <i>p</i> 6	5-3	520	8.5
<i>P</i> -1	<i>p</i> 5  <i>p</i> 6	5-3	516	9.5
<i>P</i> -2	<i>p</i> 1	<i>B</i> -1	<i>b</i> 13	2.2
<i>P</i> -2	<i>p</i> 1	3-3	324	1.4
<i>P</i> -3	<i>p</i> 3	<i>B</i> -2	<i>b</i> 78	2.1
<i>P</i> -3	<i>p</i> 3	<i>B</i> -2	<i>b</i> 40	2.8
<i>P</i> -3	<i>p</i> 3	3-4	314	1.3
3-3	328	7-1	722a	1.8
3-3	328	5-1	531	3.6
5-2	526L	5-1	531	1.0
5-3	519R	5-4	516	1.4
5-4	501	7-2	710	1.3

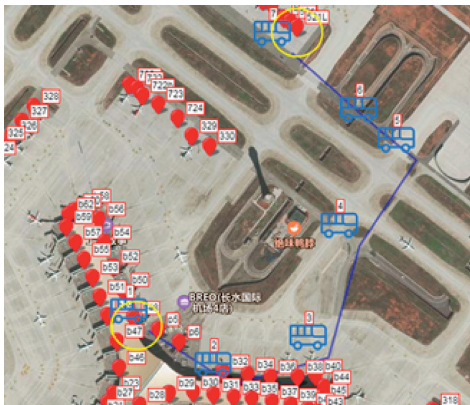


FIGURE 8: Example of ferry car path calculation.

time between two connection nodes. Figure 9 shows the distance distribution characteristics from gate 48 to *P*-1 parking lot connection point *p*5|*p*6 and from stand 526L to stand 531. The corresponding distance is represented by

selecting a smaller value, which can be taken as 0.5 and 1.0 minutes. Through the above two methods, the transfer time between regional connection nodes in the outlet network can be calculated, as shown in Table 5.

With the travel time between connection nodes, the shortest travel time between any regions can be calculated according to the Floyd shortest path algorithm. The distance between aircraft stands in the area is  $C_{gate} = 40$  m according to the actual situation of Changshui airport. When the maximum allowable speed of the ferry vehicle in the flight area is 25 km/h, the travel time between the inner edges of the area can be calculated as about 0.1 minutes. Therefore, the distance between any two nodes in the airport ferry network is equal to the distance between regions plus the distance between these two points and the connecting nodes in the region. The calculation results are shown in Table 6.

In order to check the accuracy of the distance information mined in this paper, we directly calculate the distance between the internal nodes of the two randomly selected areas as the real value through the above distance calculation method and

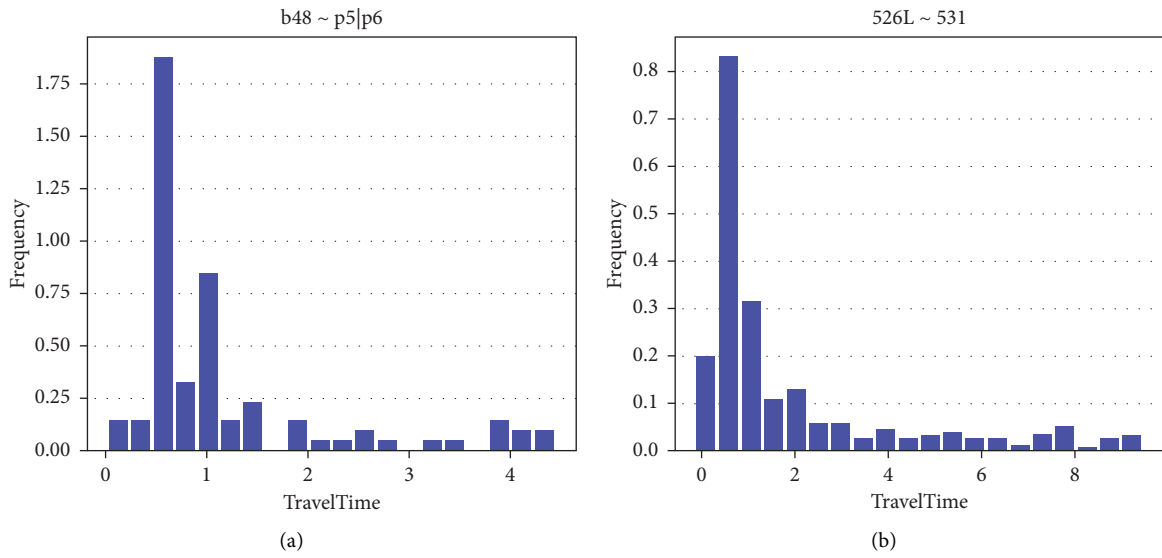


FIGURE 9: Histogram of OD distance distribution from b48 to p5|p6 (a) and 526L to 531 (b).

TABLE 6: OD distances calculated by the Floyd algorithm (partial).

	328	329	330	521	521L	521R	522	522L	522R	523
328	0.0	2.3	2.4	4.8	4.9	4.7	4.5	4.6	4.4	5.0
329	2.3	0.0	0.1	7.1	7.2	7.0	6.8	6.9	6.7	7.3
330	2.4	0.1	0.0	7.2	7.3	7.1	6.9	7.0	6.8	7.4
521	4.8	7.1	7.2	0.0	0.1	0.1	0.3	0.2	0.4	0.2
521L	4.9	7.2	7.3	0.1	0.0	0.2	0.4	0.3	0.5	0.1
521R	4.7	7.0	7.1	0.1	0.2	0.0	0.2	0.1	0.3	0.3
522	4.5	6.8	6.9	0.3	0.4	0.2	0.0	0.1	0.1	0.5
522L	4.6	6.9	7.0	0.2	0.3	0.1	0.1	0.0	0.2	0.4
522R	4.4	6.7	6.8	0.4	0.5	0.3	0.1	0.2	0.0	0.6
523	5.0	7.3	7.4	0.2	0.1	0.3	0.5	0.4	0.6	0.0

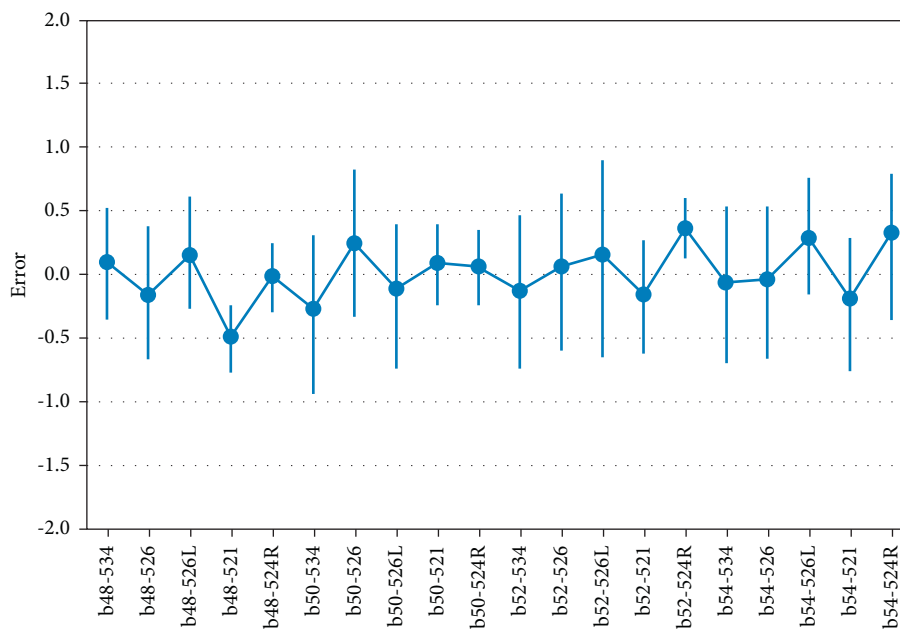


FIGURE 10: Error distribution of OD distance.

TABLE 7: Error index of OD distance.

Gate	MAE				RSME			
	b48	b50	b52	b54	b48	b50	b52	b54
521	0.6611	0.8489	0.7278	0.8766	1.0447	1.1118	1.1822	1.2811
524R	0.5761	0.707	0.7507	0.3767	0.7919	0.8516	0.9228	0.8409
526	0.7936	0.6212	0.8163	0.9594	0.8462	0.8449	1.1006	0.8715
526L	0.7636	0.7184	1.0703	0.8291	0.9939	0.8487	0.9494	1.1001
534	1.0132	1.0117	1.0679	1.2102	1.1882	1.3363	1.2691	1.3307

compare it with the corresponding distance in the distance matrix obtained by Floyd algorithm. In order to better measure this difference, this paper introduces absolute error (Error), mean absolute error (MAE), and root mean square error (RMSE) as evaluation indexes to test the accuracy of the distance matrix, as shown in equations (5) to (7).

$$\text{Error} = Y - \hat{Y}, \quad (5)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|, \quad (6)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}. \quad (7)$$

The error is shown in Figure 10, and the MAE and RMSE calculation results are shown in Table 7. The results show that the mean value of the distance error between the distance calculated by the road network and the GPS track is kept at about zero, and the deviation is kept at about 1 minute. Considering that there are many random factors in the actual driving of the ferry car, the current results can better reflect the actual driving situation and meet the needs of data accuracy for modeling.

## 6. Conclusion

This paper studies the ferry vehicles in the airport flight area. Based on the analysis and mining of ferry vehicle trajectory data, the ferry road network map is established, and a new method is proposed to extract the travel time between OD points in the ferry network. The experimental results show that the shape of the ferry road network constructed in this paper and the calculation results of travel time can reflect the real operation to a great extent. The purpose of this research paper is exactly to solve the problem of ferry vehicles' dynamic scheduling. With the travel time between OD points of the ferry network obtained in this paper, combined with the real-time GPS data of ferry vehicles, we can study how to predict the task state of vehicles and schedule vehicles dynamically and reasonably in the future.

## Data Availability

The flight data and ferry vehicle GPS track data used in this paper are provided by Kunming Changshui Airport, China. The coordinates of each boarding gate and remote stand of the airport are obtained by Google satellite map.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (71621001) and an open fund of Key Laboratory of Airports Cluster Intelligent Operation under Grant no. KLAGIO20180801.

## References

- [1] J. Bi, Z. Wu, L. Wang, D. Xie, and X. Zhao, "A tabu search-based algorithm for airport gate assignment: a case study in Kunming, China," *Journal of Advanced Transportation*, vol. 2020, pp. 1–13, 2020.
- [2] B. Liang, Y. Li, J. Bi, C. Ding, and X. Zhao, "An improved adaptive parallel genetic algorithm for the airport gate assignment problem," *Journal of Advanced Transportation*, vol. 2020, no. 4, pp. 1–17, 2020.
- [3] A. Norin, D. Yuan, T. A. Granberg, and P. Värbrand, "Scheduling de-icing vehicles within airport logistics: a heuristic algorithm and performance evaluation," *Journal of the Operational Research Society*, vol. 63, no. 8, pp. 1116–1125, 2012.
- [4] J. Y. Du, J. O. Brunner, and R. Kolisch, "Planning towing processes at airports more efficiently," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, pp. 293–304, 2014.
- [5] Z. Wang, Y. Li, X. Hei, and H. Meng, "Research on airport refueling vehicle scheduling problem based on greedy algorithm," in *Proceedings of the Intelligent Computing Theories and Application*, pp. 717–728, Wuhan, China, 2018.
- [6] S. Padrón, D. Guimarans, J. J. Ramos, and S. Fitouri-Trabelsi, "A bi-objective approach for scheduling ground-handling vehicles in airports," *Computers & Operations Research*, vol. 71, pp. 34–53, 2016.
- [7] C. Xu and Q. Shao, "Optimization of airport ground service support equipment scheduling under uncertain operation time," *Science Technology and Engineering*, vol. 018, no. 3, pp. 372–378, 2018.
- [8] T. Fei and L. Shu'An, "Research of aircraft ground service scheduling problems," in *Proceedings of the Control & Decision Conference*, May 2016.
- [9] P. Zhao, X. Han, and D. Wan, "Evaluation of the airport ferry vehicle scheduling based on network maximum flow model," *Omega*, vol. 3, pp. 35–47, 2019.
- [10] X. Han, P. Zhao, Q. Meng, S. Yin, and D. Wan, "Optimal scheduling of airport ferry vehicles based on capacity network," *Annals of Operations Research*, vol. 295, no. 1, pp. 163–182, 2020.

- [11] M. Rahmani and H. N. Koutsopoulos, "Path inference from sparse floating car data for urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 30, pp. 41–54, 2013.
- [12] B. Y. Chen, H. Yuan, Q. Li, W. H. K. Lam, S.-L. Shaw, and and K. Yan, "Map-matching algorithm for large-scale low-frequency floating car data," *International Journal of Geographical*, vol. 28, no. 1, pp. 22–38, 2014.
- [13] G. Draijer, N. Kalfs, and J. Perdok, "Global positioning system as data collection method for travel research," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1719, no. 1, pp. 147–153, 2000.
- [14] A. T. Palma, V. Bogorny, Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," in *Proceedings of the 23rd Annual ACM-Symposium on Applied Computing*, pp. 863–868, Fortaleza Ceara, Brazil, 2008.
- [15] T. Bhattacharya, L. Kulik, and J. Bailey, "Extracting significant places from mobile user GPS trajectories: a bearing change based approach," in *Proceedings of the ACMSIGSPATIALGIS'12*, ACM, Redondo Beach, Calif, USA, 2012.
- [16] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated GPS traces," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 1, 2011.
- [17] J. Li, Q. M. Qin, L. You, and C. Xie, "Parking lot extraction method based on floating car data," *Journal of Wuhan University (Information Science Edition)*, vol. 38, no. 5, pp. 599–603, 2013.
- [18] J. Wang, C. Dong, C. Shao, S. Huang, and S. Wang, "Identifying the key nodes and sections of urban roadway network based on GPS trajectory data," *Journal of Advanced Transportation*, vol. 2021, no. 1, pp. 1–14, 2021.
- [19] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, "Mining GPS traces for map refinement," *Data Mining and Knowledge Discovery*, vol. 9, no. 1, pp. 59–87, 2004.
- [20] J. Li, Q. Qin, J. Han, L.-A. Tang, and K. H. Lei, "Mining trajectory data and geotagged data in social media for road map inference," *Transactions in GIS*, vol. 19, no. 1, pp. 1–18, 2015.
- [21] L. L. Cao and J. Krumm, "From GPS traces to a routable road map," in *Proceedings of the Workshop on Advances in Geographic Information Systems*, pp. 3–12, New York, NY, USA, 2009.
- [22] L. Tang, X. Yang, Z. Kan, and Q. Li, "Lane-level road information mining from vehicle GPS trajectories based on naïve bayesian classification," *ISPRS International Journal of Geo-Information*, vol. 4, no. 4, pp. 2660–2680, 2015.
- [23] J. Li, Q. Li, Y. Zhu, Y. Ma, Y. Xu, and C. Xie, "An automatic extraction method of coach operation information from historical trajectory data," *Journal of Advanced Transportation*, vol. 2019, Article ID 3634942, 15 pages, 2019.
- [24] L. Dong, Y. Lv, H. Sun, D. Zhi, and T. Chen, "GPS trajectory-based spatio-temporal variations of traffic accessibility under public health emergency consideration," *Journal of Advanced Transportation*, vol. 2021, no. 7, 22 pages, Article ID 8854451, 2021.
- [25] P.-R. Lei, T.-J. Shen, W.-C. Peng, and I.-J. Su, "Exploring spatial-temporal trajectory model for location prediction," in *Proceedings of the 12th IEEE International Conference on Mobile Data Management (MDM '11)*, pp. 58–67, Lulea, Sweden, June 2011.
- [26] P. Zhao, X. Liu, W. Shi, T. Jia, W. Li, and M. Chen, "An empirical study on the intra-urban goods movement patterns using logistics big data," *International Journal of Geographical Information Science*, vol. 34, no. 10, pp. 1–28, 2018.
- [27] M. Lu, Z. C. Wang, and X. R. Yuan, "TrajRank: exploring travel behaviour on a route by trajectory ranking," in *Proceedings of the IEEE Pacific Visualization Symposium*, pp. 14–17, China, 2015.
- [28] J. Li, J. Wang, J. Zhang, Q. Qin, T. Jindal, and J. Han, "A probabilistic approach to detect mixed periodic patterns from moving object data," *GeoInformatica*, vol. 20, no. 4, pp. 715–739, 2016.
- [29] R. W. Floyd, "Algorithm 97: shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.
- [30] E. W. Dijkstra, "A note on two problems in connexion with graphs[J]," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1958.
- [31] F. D. Duan, "A faster algorithm for shortest-Path-SPFA," *Journal of Southwest Jiaotong University*, vol. 29, no. 2, pp. 207–211, 1994.

## Research Article

# An Influence Analytical Model of Dedicated Bus Lane on Network Traffic by Macroscopic Fundamental Diagram

Yingying Ma , Yuanqi Xie , and Yongjie Lin 

*Department of Transportation Engineering, South China University of Technology, Guangzhou 510640, China*

Correspondence should be addressed to Yongjie Lin; [linyjscut@scut.edu.cn](mailto:linyjscut@scut.edu.cn)

Received 30 May 2021; Revised 26 July 2021; Accepted 10 August 2021; Published 24 August 2021

Academic Editor: Ming Xu

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

To study the influence mechanism of dedicated bus lanes on the urban road network, this paper proposes a novel analytical model of macroscopic fundamental diagram (MFD) and passenger macroscopic fundamental diagram (p-MFD) and the corresponding indicators based on MFD and p-MFD to evaluate the operation of the network. Taking the grid network as an example, this paper collects traffic flow to calibrate the developed MFD and p-MFD and evaluates the network performance under different proportions of dedicated bus lanes. The simulation results show that the larger the proportion of dedicated bus lanes, the greater the impact on the rising section and the stable section of MFD and the descending section and post-stable section of p-MFD. Further analysis for the sensitivity of simulation experiments found that the strategy of setting dedicated bus lanes will improve the efficiency of vehicle and passenger transport when the road network is in a smooth state and ensure the continuous output of passengers when the network is in a congested state.

## 1. Introduction

The increasing number of vehicles causes the traffic demand for the road network in cities to increase rapidly, and corresponding traffic congestion in many cities is serious. The transit priority strategy of setting dedicated bus lanes plays an important role in improving bus operating efficiency and the level of service. The dedicated bus lane is an economical implementation method to improve the level of public transport service and ease the pressure of urban traffic [1]. A large number of cities in the world have implemented dedicated bus lanes to promote public transport. By the end of 2019, the total length of dedicated bus lanes in China was 14,951.7 km [2]. The planning and implementation of dedicated bus lanes aim to optimize the allocation of road resources in urban cities and improve traffic conditions for bus priority [3, 4].

The implementation of dedicated bus lanes has a certain degree of impact on road network traffic. A comprehensive evaluation of dedicated bus lanes is a necessary step to analyse the feasibility and benefits of its implementation that concerns the trade-off of the benefits of public and private

transport in the urban road network. Urban transport systems should meet the travel needs of urban residents. Therefore, when setting up dedicated bus lanes, the impact on the vehicle transport and passenger transport of urban road networks should be comprehensively considered. Moreover, the benefits of dedicated bus lanes are related to the traffic state of the urban road network. In order to determine the effect of setting up dedicated bus lanes, it is important to propose a group of indicators to analyse the overall performance of the road network, which are helpful capture the impact of dedicated bus lanes on the urban road network and understand how it will change with the varying traffic status.

In recent years, the latest trend in the study of urban road network transport is to use the macroscopic fundamental diagram (MFD) to model and analyse urban-scale road network traffic. Daganzo proposed the concept and basic description of the MFD of the urban network in 2007 and verified the existence of MFD using traffic simulation [5] and empirical data [6]. Through the analysis of MFD, Daganzo [7] found that MFD is an inherent attribute of the road network itself, and it remains unchanged when traffic

demand changes. The basic curve of MFD is a parabola. However, in the actual analysis, it is often simplified to a triangle or trapezoid [8]. MFD not only reflects the general relationship between network traffic flow and network operation level but also describes the relationship between the number of vehicles in the road network and the output flow of the road network [9]. However, the MFD can only express the performance of vehicles in the transport system but not consider the average number of passengers of each transport mode.

In order to improve the concept of MFD, Zheng [10] and Nicolas [11] proposed the concept of passenger macroscopic fundamental diagram (p-MFD) to evaluate the passenger transport capacity of the multimodal urban network with cars and buses. The existence of MFD [9, 12–15] and p-MFD [16] has been verified by the empirical data of urban road networks around the world. As the inherent attributes of the road network, MFD and p-MFD, respectively, indicate the relationship between the density of vehicles in the road network and the output flow of vehicles of the road network, and also the relationship between the density of vehicles in the road network and the output flow of passengers of the road network. They help to evaluate the performance of the urban multimodal network. In recent years, MFD has been extensively studied and applied, including model formulation [17, 18], optimization [19–22], traffic control [23, 24], and perimeter control [25, 26].

Most previous studies of the impact of dedicated bus lanes on MFD and p-MFD mainly focus on data sources, modelling methods, analysis factors, and the impact on road network transport. In terms of data sources, MFD can be obtained through traffic software simulation [27–30], theoretical model [3, 10, 31], and empirical data [12, 16], and p-MFD can be derived by fixed passenger occupancy rate [30], estimating passenger occupancy rate [3, 11, 12, 16, 28] or using theoretical models [32] on the passenger occupancy rate of each transport mode in the multimodal urban road network. Most previous studies of the impact of dedicated bus lanes on MFD and p-MFD focus on theoretical road networks [3, 10, 11, 31] or actual road networks [12, 16, 27–30, 32]. In terms of modelling methods, the existing modelling methods of MFD and p-MFD include scattering plot method [10, 12, 16, 28–30], contour map (3D-MFD) [12, 16, 19, 20], measured or simulated data fitting method [16, 27, 28], and analytical model method [3, 11, 32]. The literature review on the impact of bus lanes on MFD and p-MFD is summarized in Table 1.

Although existing literature has found that dedicated bus lanes have a certain impact on the road network MFD and p-MFD, they pay more attention to the changes in shape and the maximum value of MFD and p-MFD under different conditions. Limited attention is paid to propose comprehensively a group of indicators for urban road networks based on MFD and p-MFD and use them as the basis to study the influence of dedicated bus lanes on the operation of the road network under different conditions of the road network, regarding vehicles and passengers, respectively.

Microcosmic traffic evaluation indicators only focus on traffic operations of a certain road section or under a certain network state. Through in-depth study of MFD and p-MFD, the road network can be viewed as a whole, and the traffic state under different conditions can be evaluated. Comparing microcosmic traffic evaluation indicators, MFD-based indicators can more fully reflect the transport efficiency, capacity, and reliability of the road network under different conditions at the network level.

Therefore, this paper proposes a novel description and modelling methods of MFD and p-MFD based on the Gaussian mixture model (GMM). The indicators based on MFD and p-MFD are proposed to analyse comprehensively the operation of the road network. This methodology is suitable for various urban networks, which can be extended to multiple modes (cars, buses, etc.) to evaluate the efficiency and other aspects of the urban network from vehicles and passengers' perspectives. This paper takes the grid network as an example and uses traffic simulation to obtain the traffic data when the proportion of dedicated bus lanes is changed. The characteristic parameters of MFD and p-MFD are analysed in detail to study and summarize the impact mechanism of dedicated bus lanes on the urban network and to support and expand existing research on the impact of dedicated bus lanes on MFD and p-MFD.

## 2. Analytical Framework Based on MFD and p-MFD

*2.1. Description of Macroscopic Fundamental Diagram.* MFD (macroscopic fundamental diagram) is composed of a series of scattered points, which are from the empirical or experimental data at a certain time interval  $\Delta t$  during the operation process of the urban road network. In this paper, MFD takes the vehicle density in the road network  $K$  (veh/km/ln) as the abscissa and the vehicle flow leaving the road network  $Q$  (veh/ln/h) as the ordinate. According to the definition of MFD, the relevant parameter calculation equation is as follows:

$$q(t) = q(t - \Delta t) + q^i(t) - q^o(t), \quad (1)$$

$$Q(t) = \frac{q^o(t) \cdot 3600}{\Delta t \cdot N}, \quad (2)$$

$$K(t) = \frac{q(t)}{L}, \quad (3)$$

where  $\Delta t$  represents the time interval for data collection (s),  $q(t)$  denotes the number of vehicles running in the road network at time  $t$  (veh),  $q^i(t)$  is the number of vehicles entering the road network at time  $t$  (veh),  $q^o(t)$  represents the number of vehicles leaving the road network at time  $t$  (veh),  $Q(t)$  means the flow of vehicles leaving the road network at time  $t$  (veh/ln/h),  $N$  represents the total number of exit lanes of the road network (ln),  $K(t)$  represents the density of running vehicles in the road network at time  $t$  (veh/km/ln), and  $L$  denotes the total length of lanes in the road network (km).



TABLE 1: Literature review on implementation of dedicated bus lanes on MFD and p-MFD.

	MFD	p-MFD
Data sources	Traffic software simulation [27–30] Theoretical model [3, 10, 31] Empirical data [12, 16]	Fixed passenger occupancy rate [30] Estimated passenger occupancy rate [3, 11, 12, 16, 28] Theoretical models [32]
Road network	Theoretical road network [3, 10, 31] Actual road network [12, 16, 27–30]	Theoretical road network [3, 11] Actual road network [12, 16, 28, 30, 32]
Modelling method	Scatter plot [10, 12, 29] and contour map (3D-MFD) [12, 29] Empirical or simulated data fitting: least square method and exponential function fitting (3D-MFD) [16, 27, 28] Analytical model: variational method [3]	Scatter plot [12, 28, 30] and contour map (3D-MFD) [12, 16, 30] Empirical or simulated data fitting: least square method and exponential function fitting (3D-MFD) [16] Analytical model: variational method [3, 32], the analytical model considering the proportion of passengers choosing car transport and public transport [11]
Analysis factors	The shape of MFD (maximum) [3, 10, 16, 27–29]	The shape of p-MFD (maximum) [3, 11, 12, 16, 28, 30, 32]
Impact on road network transport	(i) The shape of the MFD of the dedicated bus lane is similar to the shape of the MFD of the mixed traffic [27, 28], and its shape is related to the average bus dwell time [28] and headway [3]. (ii) The number of vehicles in the road network when setting up dedicated bus lanes will not reach the maximum value of the MFD of the previous road network [10, 11, 29]. (iii) With a very low accumulation of buses, the road network achieves the best operating conditions (maximum capacity value). When the proportion of dedicated bus lanes increases, the transport performance of cars gets worse and the transport performance of buses improves [29]. (iv) When dedicated bus lanes are available, buses have significantly improved speed and flow compared with mixed flow conditions [16]. (v) In the case of car traffic congestion, increasing the space of dedicated bus lanes can reduce the total cost of society and increase the share rate of buses [31].	(i) The number of passengers in the road network when setting up dedicated bus lanes will not reach the maximum value of the p-MFD of the previous road network [3]. (ii) To obtain more passenger throughput, when the road network has implemented dedicated bus lanes, more buses can be deployed in the network [28]. A certain number of buses can improve the transport performance of the road network [30]. (iii) The increase in the proportion of bus lanes will reduce the marginal effect of the speed of the bus fleet [12]. (iv) Under noncongested traffic conditions, mixed lanes can always produce higher passenger throughput. However, as the degree of congestion increases, the passenger throughput of dedicated bus lanes is greater than that of mixed lanes [32]. (v) The dedicated bus lane ensures that the bus system can still operate even when the road network is very crowded [11].

Using network traffic empirical or experimental data, according to equations (1)–(3), a scatter diagram of MFD can be obtained as shown in Figure 1. In-depth research found that in the complete road network MFD, as the density of running vehicles in the road network  $K$  increases from 0, the flow of vehicles leaving the road network  $Q$  has a significant upward, stable, and downward trend. Then,  $Q$  drops to the minimum value  $Q_{\min}$  since the downtrend occurred. As  $K$  continues to increase,  $Q$  maintains fluctuations within a certain range above and below  $Q_{\min}$ . If  $Q_{\min}$  is obviously greater than 0 at this time, it means that there are still a small number of vehicles in the road network that can leave the road network, such as the vehicles near the exit of the road network. If  $Q_{\min}$  is close to 0 at this time, it means that the road network has a deadlock and the majority of vehicles on the road network cannot leave the road network.

*2.2. Description of Passenger Macroscopic Fundamental Diagram.* The concept of the p-MFD is proposed to evaluate the passenger transport capacity of the multimodal road network in this study. Cars can travel according to the demand of car users when the road network condition permits. As the public transport mode, the bus must travel

along the preset bus lanes, and the departure interval has certain restrictions. The bus is an independent mode of transport, which is different from the car in terms of vehicle structure, passenger capacity, and driving characteristics. Therefore, there are much difference in operation characteristics between cars and buses. As the road network traffic facilities or traffic management measures (bus lanes, bus priority, etc.) change, the differences between the two are significant. Car p-MFD and bus p-MFD will also change accordingly, and there may be differences in the changing patterns. Therefore, this paper studies the bus p-MFD and car p-MFD of urban road networks separately to study better the changes in the capacity of these two modes to transport passengers when the states of the road network change.

p-MFD is composed of a series of scattered points. The scattered points are the empirical or experimental data at a certain time interval  $\Delta t$  during the operation process of the road network. In this paper, p-MFD takes the vehicle density in the road network of  $K$  (veh/km/ln) as the abscissa and the passenger flow leaving the road network of PAX (person/ln/h) as the ordinate. According to the definition of p-MFD, the relevant parameter calculation equation is as follows:

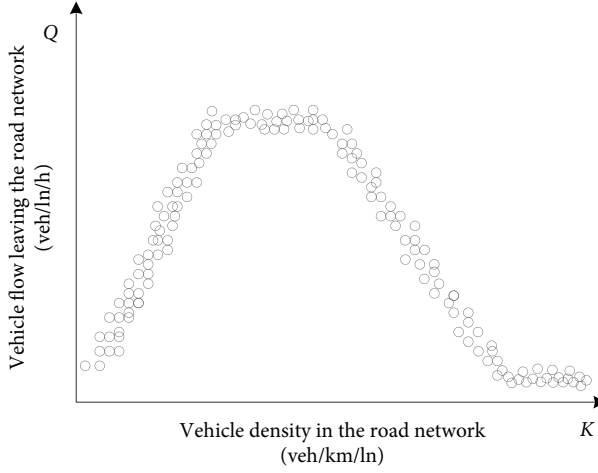


FIGURE 1: MFD.

$$P_c(t) = \frac{n_c \cdot [q_c(t - \Delta t) + q_c^i(t) - q_c^o(t)] \cdot 3600}{\Delta t \cdot N}, \quad (4)$$

$$P_b(t) = \frac{n_b \cdot [q_b(t - \Delta t) + q_b^i(t) - q_b^o(t)] \cdot 3600}{\Delta t \cdot N}, \quad (5)$$

$$P(t) = P_c(t) + P_b(t), \quad (6)$$

$$K(t) = \frac{q(t)}{L}, \quad (7)$$

where  $\Delta t$  represents the time interval for data collection (s),  $q(t)$  means the number of vehicles running in the road network at time  $t$  (veh),  $q_c^i(t)$  represents the number of cars entering the road network at time  $t$  (veh),  $q_b^i(t)$  represents the number of buses entering the road network at time  $t$  (veh),  $q_c^o(t)$  represents the number of cars leaving the road network at time  $t$  (veh),  $q_b^o(t)$  denotes the number of buses leaving the road network at time  $t$  (veh),  $n_c$  represents the average passenger occupancy of cars (person),  $n_b$  represents the average passenger occupancy of buses (person),  $P_c(t)$  means the flow of car passengers leaving the road network at time  $t$  (person/ln/h),  $P_b(t)$  represents the flow of bus passengers leaving the road network at time  $t$  (person/ln/h),  $N$  represents the total number of exit lanes of the road network (ln),  $K(t)$  represents the density of running vehicles in the road network at time  $t$  (veh/km/ln), and  $L$  represents the total length of lanes in the road network (km).

**2.3. Segmented Linear Model of MFD and p-MFD.** The MFD scatter plots of different road networks, car p-MFD, and bus p-MFD have similar changing trends, namely the rising, stable, descending, and post-stable trends. The post-stable trend with consistently low value means that a very small number of vehicles still can leave the road network even when the road network is seriously congested, such as the vehicles near the exit. However, there are still some differences between MFD and p-MFD. Therefore, the scattered points of MFD and p-MFD can be clustered into several

categories according to the changing trend. Based on research [33], this paper uses Gaussian mixture clustering to classify the scattered points of MFD and p-MFD. The Gaussian mixture model (GMM) is a linear combination of multiple Gaussian models and is a common clustering method. This method uses the Gaussian probability density function to calculate the probability that the scatter points belong to each category. The category with the highest probability of the scatter points is regarded as the category to which the scatter points belong, and the scatter points are aggregated into several categories with obvious differences. GMM can divide the data with any shape of an ellipse and has a better clustering effect on curved data points. The `sklearn.mixture` library is used in the open-source software of python to implement GMM in this paper.

The probability distribution of the Gaussian mixture model is shown in the following equation:

$$P(x|\theta) = \sum_{m=1}^M \alpha_m \varphi(x|\theta_m), \quad (8)$$

where  $M$  represents the number of sub-Gaussian models in the mixture model,  $m = 1, 2, \dots, M$ ,  $M \geq 3$ ;  $\alpha_m$  means the probability that the scatter points belong to the  $m$ th sub-model,  $\alpha_m \geq 0$ ,  $\sum_{m=1}^M \alpha_m = 1$ ; and  $\varphi(x|\theta_m)$  represents the Gaussian distribution density function of the  $m$ th sub-model,  $\theta_m = (\mu_m, \sigma_m^2)$ .

According to equations (1)–(3) and (7)–(10), MFD and p-MFD scatter plots of various modes can be obtained. The GMM model is used to cluster the MFD and the p-MFD scattered points of cars and buses separately and divide them into  $M$  categories. The  $M$ -segment scatter points of the road network MFD, the car p-MFD, and the bus p-MFD are linearly fitted and represented by a segmented linear model as follows:

$$F_w(k) = \begin{cases} a_w^1 k + b_w^1, & 0 \leq k \leq k_w^1, \\ a_w^2 k + b_w^2, & k_w^1 < k \leq k_w^2, \\ \dots & \\ a_w^{M-1} k + b_w^{M-1}, & k_w^{M-2} < k \leq k_w^{M-1}, \\ a_w^M k + b_w^M, & k_w^{M-1} < k, \end{cases} \quad (9)$$

$$k_w^j = \frac{b_w^j - b_w^{j-1}}{a_w^{j-1} - a_w^j}, \quad j = 2, \dots, M, \quad (10)$$

where  $F = \{Q, PAX\}$ . When  $F = Q$ ,  $Q_w(k)$  represents the flow of vehicles leaving the road network when the density is  $k$  (veh/ln/h).  $w = \{\text{net}\}$ , net represents the road network. When  $F = PAX$ ,  $PAX_w(k)$  represents the passenger flow of cars or buses leaving the road network when the density is  $k$  (person/ln/h).  $w = \{c, b, \text{net}\}$ , where  $c$  represents cars,  $b$  is buses, and net means the road network.  $F_w^j(k) = a_w^j k + b_w^j$ , where  $j = 1, 2, \dots, M$ ,  $j$  represents the corresponding number of each segment,  $a_w^j$  denotes the slope of the function,  $b_w^j$  means the constant of the function, and  $k_w^j$

represents the corresponding vehicle density  $k$  in the road network at the beginning of the  $j$  segment of MFD or p-MFD. In this paper,  $k_w^1$  is generally considered to be 0.

To avoid state mutation in MFD and p-MFD, if  $a_w^j a_w^{j+2} < 0, j = 1, 2, \dots, M - 2$ , then  $a_w^j = 0$ . When  $a_w^j > 0$ , the segment is in a rising state. When  $a_w^j = 0$ , the segment is in a steady state. When  $a_w^j < 0$ , the segment is in a descending segment. The rising section consists of multiple consecutive rising segments, and the descending section consists of multiple consecutive descending segments.

## 2.4. MFD and p-MFD Modelling

### 2.4.1. MFD Modelling

**Step1:** Use empirical or experimental data to calculate the vehicle density  $K(t)$  in the road network and the flow of vehicles leaving the road network  $Q(t)$  according to equations (1)–(3)

**Step2:** Use  $K(t)$  as the abscissa and  $Q(t)$  as the ordinate to obtain the scatter plot of road network MFD

**Step3:** Employ GMM to cluster road network MFD scatters and divide scatters into  $M$  categories

**Step4:** Linear fitting is performed on the  $M$ -segment scatters of the road network MFD, and the piecewise linear function is used to represent the road network MFD model

According to the changing trend of the scatter plot of road network MFD, the number of clusters of MFD  $M$  is set to 4 in this paper. The 4-segment scattered points of MFD classified by GMM are linearly fitted, and MFD is represented by a piecewise linear function. MFD obtained by the above modelling method is shown in Figure 2.

According to Figure 2, each section of MFD can be described as follows:

(1) *Rising Section.* The traffic in the road network is in a smooth state, and the throughput of cars and buses in the road network is smooth. At this section,  $Q(t)$  increases with the increase of  $K(t)$ .

(2) *Stable Section.* The traffic in the road network is still in a saturated state, and the road network continues to transport cars and buses. As  $K(t)$  increases,  $Q(t)$  stabilizes within a certain range, and  $Q(t)$  reaches the maximum value  $Q_{\max}$ . If  $K(t)$  continues to increase,  $Q(t)$  fluctuates around  $Q_{\max}$ .

(3) *Descending Section.* The traffic in the road network is in a front-congested state. Some intersections and road sections in the road network are blocked, and part of vehicles cannot leave the road network due to traffic congestion. At this time, the road network has reached the critical point where it can effectively bear no more vehicles. As  $K(t)$  continues to increase,  $Q(t)$  continues to decrease.

(4) *Post-Stable Section.* The traffic in the road network is in a post-congested state, and most intersections and road sections in the road network remain seriously blocked, and the corresponding most vehicles cannot leave the road network.  $Q(t)$  is stable within a certain range, and  $Q(t)$

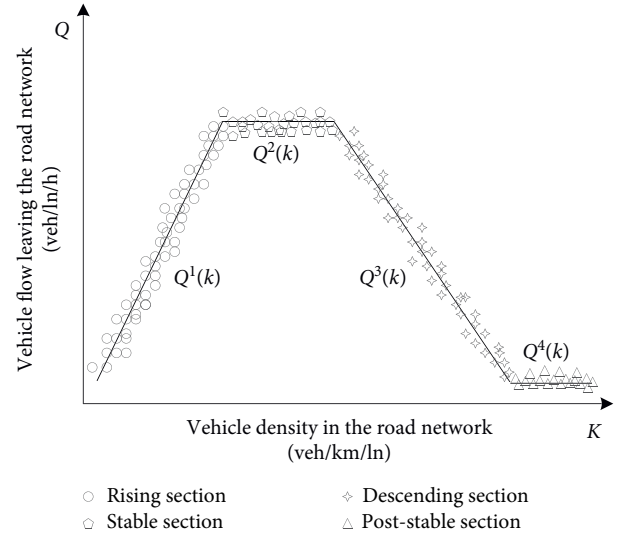


FIGURE 2: MFD obtained by the proposed method.

reaches the minimum value  $Q_{\min}$  since MFD enters the descending stage. If  $K(t)$  continues to increase,  $Q(t)$  will fluctuate around  $Q_{\min}$ .

### 2.4.2. p-MFD Modelling

**Step1:** Use empirical or experimental data to calculate the vehicle density  $K(t)$  in the road network, the number of passenger-car users leaving the road network  $P_c(t)$  and bus passengers leaving the road network  $P_b(t)$  according to equations (4)–(7)

**Step2:** Use  $K(t)$  as the abscissa and  $P_c(t)$  and  $P_b(t)$  as the ordinate separately to obtain the scatter plot of car p-MFD and bus p-MFD, respectively

**Step3:** Use GMM to cluster the car p-MFD and the bus p-MFD scatters and divide scatters into  $M$  categories separately

**Step4:** Linear fitting is performed on the  $M$ -segment scatters of car p-MFD and bus p-MFD, and the piecewise linear function is used to represent car p-MFD and bus p-MFD model

**Step5:** Add the piecewise linear function of car p-MFD and bus p-MFD to get road network p-MFD model, namely

$$\begin{aligned} PAX(k) &= PAX_c(k) + PAX_b(k), \\ g_p(k) &= g_{pc}(k) + g_{pb}(k), \end{aligned} \quad (11)$$

where  $PAX(k)$  denotes the passenger volumes leaving the road network when the density is  $k$  (persons/ln/h),  $PAX_c(k)$  denotes the flow of car passengers leaving the road network when the density is  $k$  (persons/ln/h),  $PAX_b(k)$  represents the flow of bus passengers leaving the road network when the density is  $k$  (persons/ln/h),  $g_{pb}(k)$  denotes the slope of road network p-MFD when the density is  $k$ ,  $g_{pc}(k)$  means the slope of car p-MFD when the density is

$k$ , and  $g_{pb}(k)$  denotes the slope of bus p-MFD when the density is  $k$ .

According to the changing trend of p-MFD, the number of p-MFD scatter clusters for cars and buses in this paper is set to 4, namely,  $M = 4$ . Therefore, there are 3 inflection points for the p-MFD of the car and the bus,  $k_w^i$ ,  $w = \{c, b\}$ ,  $i = 1, 2, 3$ . Because the operating characteristics of cars and buses are different, the inflection points of car p-MFD and bus p-MFD may also be different. Therefore, the road network p-MFD obtained by equation (12) is a linear function with  $s$  inflection points ( $3 \leq s \leq 6$ ), and  $s = M - 1$ . The p-MFD obtained using the above method is shown in Figure 3.

In Figure 3, the sections of car p-MFD, bus p-MFD, and road network p-MFD are described as follows:

(1) *Rising Section*. The traffic in the road network is in a smooth state, and the throughput of car users and bus passengers in the road network is smooth.  $P_c(t)$ ,  $P_b(t)$ , and  $P(t)$  increase with the increase of  $K(t)$ , respectively. The rising section of  $P(t)$  may have multiple segments depending on the slope.

(2) *Stable Section*. The traffic in the road network is in a saturated state, and the road network continues to carry car and bus passengers. As  $K(t)$  increases,  $P_c(t)$  and  $P_b(t)$  stabilize within a certain range and reach the maximum  $P_{cmax}$  and  $P_{bmax}$ , respectively. As  $K(t)$  continues to increase,  $P_c(t)$  fluctuates around  $P_{cmax}$ , and  $P_b(t)$  fluctuates around  $P_{bmax}$ . Notably,  $P_c(t)$  and  $P_b(t)$  do not reach the maximum value at the same time, that is,  $k_c^1$  and  $k_b^1$  are not necessarily the same, respectively. When  $P_c(t)$  and  $P_b(t)$  enter the stable section,  $P(t)$  enters the stable section, too.

(3) *Descending Section*. The traffic in the road network is in a front-congested state. Some intersections and road sections in the road network are blocked, and some bus and car passengers cannot leave the road network. As  $K(t)$  continues to increase,  $P_c(t)$ ,  $P_b(t)$ , and  $P(t)$  continues to decrease. The descending section of  $P(t)$  may have multiple segments depending on the slope.

(4) *Post-Stable Section*. The traffic in the road network is in a post-congested state, and most of the intersections and road sections are seriously blocked. Without public bus priority measures (dedicated bus lanes, bus signal priority, etc.), most of the car and bus passengers cannot leave the road network. As  $K(t)$  increases,  $P_c(t)$  and  $P_b(t)$  are stable within a certain range. Then,  $P_c(t)$  and  $P_b(t)$  reach the minimum value  $P_{cmin}$  and  $P_{bmin}$  since car p-MFD and bus p-MFD entered the descending stage separately. As  $K(t)$  continues to increase,  $P_c(t)$  fluctuates around  $P_{cmin}$ , and  $P_b(t)$  fluctuates around  $P_{bmin}$ . It is worth noting that  $P_c(t)$  and  $P_b(t)$  do not reach the minimum value at the same time, namely  $k_c^3$  and  $k_b^3$  are not necessarily the same, respectively. When  $P_c(t)$  and  $P_b(t)$  enter the post-stable section,  $P(t)$  also enters the post-stable section.

### 3. Network Evaluation Based on MFD and p-MFD

This study proposes several indicators to evaluate the urban road network based on MFD and p-MFD from different

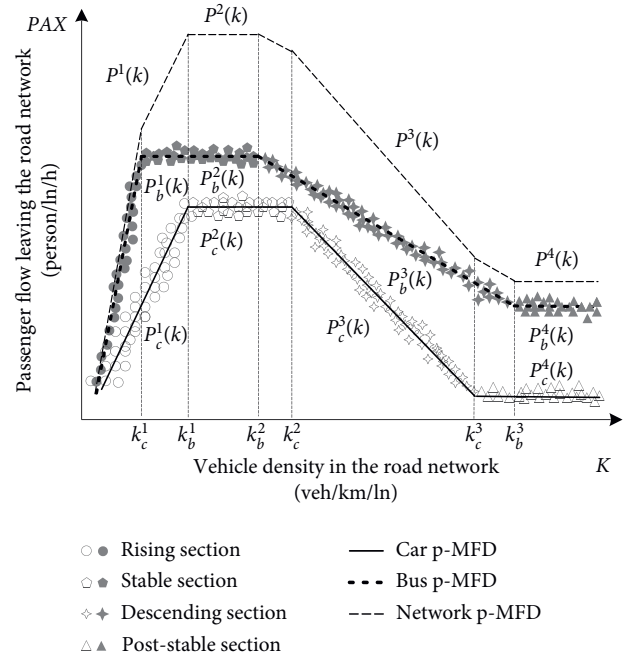


FIGURE 3: p-MFD obtained by the proposed modelling method.

perspectives. Through the analysis of these indicators, it is helpful analyse more comprehensively the operation of the urban road network under different scenarios.

**3.1. Road Network Transport Efficiency.** The indicators in this section are proposed to evaluate the operational efficiency of the urban road network based on MFD and p-MFD proposed in the previous section.

The slope of the MFD  $g_c$  (p-MFD  $g_p$ ) is defined as the ratio between the increment of traffic flow  $\Delta Q$  (passengers  $\Delta PAX$ ) leaving the road network and the increment of vehicle density in the road network  $\Delta K$ :

$$g_c = \frac{\Delta Q}{\Delta K} \quad (12)$$

$$g_p = \frac{\Delta PAX}{\Delta K} \quad (13)$$

In this study, the slopes of the rising section and the descending section are mainly studied. When the slope of the rising section is larger, the increment in the flow of passengers (vehicles) leaving the road network is larger when the increment in vehicle density is the same. Thus, the efficiency of transporting passengers (vehicles) improves when traffic flow on the road network is in a smooth state; otherwise, the efficiency of transporting passengers (vehicles) deteriorates when the road network is in a smooth state. When the absolute value of the slope of the descending section is larger, the increment of passengers (vehicles) flow leaving the road network is greater when the increment in vehicle density is the same. It reveals that the efficiency of transporting passengers (vehicles) improves when the road network is in a congested state; otherwise, it reveals that the

efficiency of transporting passengers (vehicles) deteriorates when the road network is in the congested state.

The value of the stable section  $F_w^2(k)$  is defined as the corresponding flow of vehicles leaving the road network in the stable section. In the same way, the value of the post-stable section  $F_w^4(k)$  is defined as the corresponding flow of vehicles leaving the road network in the post-stable section as follows:

$$\begin{aligned} F_w^2(k) &= a_w^2 k + b_w^2, \\ F_w^4(k) &= a_w^4 k + b_w^4. \end{aligned} \quad (14)$$

In this study, when the stable or post-stable section is close to horizontal,  $a_w^2 = 0$  or  $a_w^4 = 0$ , and  $F_w^2(k) = b_w^2$ ,  $F_w^4(k) = b_w^4$ . Within the density range of the stable section, when the value of the stable section is larger, the flow of passengers (vehicles) leaving the road network is larger, which indicates that the road network is more efficient in transporting passengers (vehicles) in a saturated state. The above indication also applies to the post-stable section.

**3.2. Road Network Capacity and Reliability.** The indicators proposed in this section are used to evaluate the capacity and reliability of the urban road network based on MFD and p-MFD.

The front-inflection point ( $k_1$  and  $q_{\max}$ ) is defined as the connecting point between the rising and the stable sections, and the back-inflection point ( $k_2$ ,  $q_{\max}$ ) is defined as the connecting point between the stable and the descending sections. The schematic diagram is shown in Figure 4.

When the inflection point moves toward the zero point, the abscissa and ordinate values of the inflection point decrease. If the front-inflection point is closer to the zero point, it means that the road network quickly enters the stable state from the smooth one; if the back-inflection point is closer to zero, it shows the road network quickly changes from the stable state to the congested one. Therefore, the faster the road network state changes and the lower the road network throughput is, the less effective the resource utilization of road network is.

The density range of  $x$  section  $\Delta k_{\text{range}}^x$  is defined as the difference of the vehicle density corresponding to the end of the  $x$  section minus the vehicle density corresponding to the beginning of the  $x$  section. The expression is as follows:

$$\Delta k_{\text{range}}^x = k_e^x - k_b^x, \quad (15)$$

where  $k_2^x$  denotes the vehicle density corresponding to the end of the  $x$  section,  $k_1^x$  denotes the vehicle density corresponding to the beginning of the  $x$  section,  $\Delta k_{\text{range}}^x$  denotes the density range of  $x$  section.  $x = \{r, s, d, p\}$ , where  $r$  represents the rising section,  $s$  represents the stable section,  $d$  represents the descending section, and  $p$  represents the post-stable section. When the density range is larger, it means that the road network has a stronger ability to output traffic continuously in this state, and the road network has a stronger continuity in this state.

The goodness of fit,  $R^2$ , is defined as the degree of the regression line fitting the observed values:

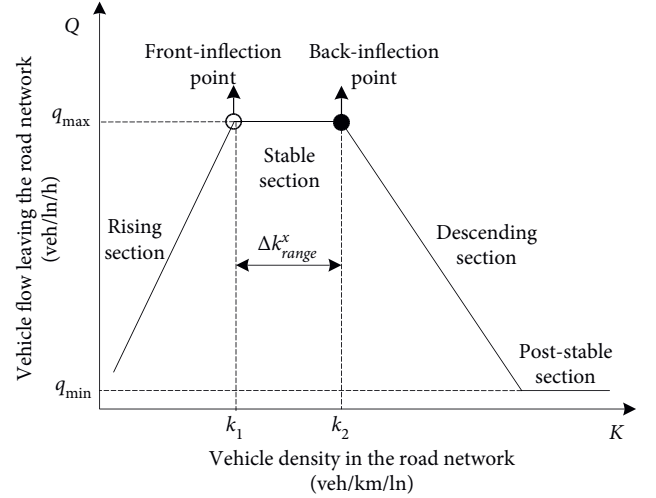


FIGURE 4: Schematic diagram of indicators.

$$R^2 = \frac{\sum_{g=1}^G (\hat{y}_g - \bar{y})^2}{\sum_{g=1}^G (y_g - \bar{y})^2}, \quad (16)$$

where  $R^2$  denotes the goodness of linear fitting;  $\hat{y}_g$  denotes the value of the fitted data, which is the data value obtained by linear fitting in this paper;  $\bar{y}$  denotes the mean value of the empirical data or experimental data, which is the mean value of the simulated data in this paper;  $y_g$  denotes the empirical data or experimental data, which is the simulated data in this paper;  $G$  denotes the number of data samples; and  $g$  denotes the  $g^{\text{th}}$  data sample. When the goodness of fit,  $R^2$ , is larger, the scattered points of this section are more concentrated, which means that the stability of the road network transport in this state is better. This study mainly studied the goodness of fit of the rising section and the descending section, respectively.

To sum up, the aforementioned indicators and the symbolic meanings corresponding to the changes in indicators for road network evaluation are summarized in Table 2.

## 4. Experimental Design and Results

**4.1. Introduction of the Experiment.** Actually, the MFD and p-MFD method and the corresponding indicators proposed in this paper can be extended to multiple modes (cars, buses, etc.) to evaluate the operation of various urban road networks when the traffic improvement is implemented. For example, the implementation of a dedicated bus lane is a very effective measure to improve bus operation. However, it may cause traffic congestion for private cars due to the reduction of road resources. Next, MFD and p-MFD are used to evaluate the influences of dedicated bus lanes on the road network operation in this experiment. A grid road network with signalized intersections is taken for experiments, which is similar to the urban network studied in reference [10]. The roads are divided into main roads, secondary roads, and branch roads. The main roads are 3,000 meters long with six lanes, and each lane is 3.5

TABLE 2: Road network evaluation indicators based on MFD and p-MFD.

Category	Indicator	Trend	Symbolic meaning
Road network transport efficiency	The slope of the rising section of MFD (p-MFD)	Increase	The efficiency of transporting vehicles (passengers) improves when the road network is in the smooth state
		Decrease	The efficiency of transporting vehicles (passengers) deteriorates when the road network is in the smooth state
	The value of the stable section of MFD (p-MFD)	Increase	The efficiency of transporting vehicles (passengers) and the ability to endure the access of vehicles improve when the road network is in the saturated state
		Decrease	The efficiency of transporting vehicles (passengers) and the ability to endure the access of vehicles deteriorate when the road network is in the saturated state
	The absolute value of the slope of the descending section of MFD (p-MFD)	Increase	The efficiency of transporting vehicles (passengers) improves when the road network is in the congested state
		Decrease	The efficiency of transporting vehicles (passengers) deteriorates when the road network is in the congested state
	The absolute value of the post-stable section of MFD (p-MFD)	Increase	The efficiency of transporting vehicles (passengers) improves when the road network is in the post-congested state
		Decrease	The efficiency of transporting vehicles (passengers) deteriorates when the road network is in the post-congested state
Road network capacity and reliability	The inflection point of MFD (p-MFD)	Move toward (0, 0)	The utilization of road network deteriorates resources from vehicles (passengers) perspective
		Move away from (0, 0)	The utilization of road network resources improves from vehicles (passengers) perspective
	The density range of MFD (p-MFD)	Increase	The continuity of the road network at this state improves from vehicles (passengers) perspective
		Decrease	The continuity of the road network at this state deteriorates from vehicles (passengers) perspective
	The goodness of fit of MFD (p-MFD)	Increase	The stability of the road network improves from vehicles (passengers) perspective
		Decrease	The stability of the road network deteriorates from vehicles (passengers) perspective

meters wide. The secondary roads are 2,100 meters long with four lanes, and each lane is 3.5 meters wide. The main roads and the secondary roads are both two-way roads. The branch roads are 100 meters long. Two pairs of harbour-style bus stops are set up on each main road. There are two types of vehicles in the road network, cars and buses. In this experiment, the dedicated bus lanes are set on the roadside. The schematic diagram of the dedicated bus lanes and road network are shown in Figures 5 and 6, respectively. In order to facilitate simulation, bus lanes are only set up on the main roads, and the average bus flow on the dedicated bus lanes is set as 75 veh/ln/h based on the field investigation of the average interval on dedicated bus lanes.

The network traffic data under the different scenarios are obtained by a microscopic traffic simulation software of PTV Vissim. This study has simulated five scenarios with different proportions of dedicated bus lanes, which are 0, 0.05, 0.11, 0.16, and 0.21.

In the simulation network, data detectors are installed at the road network boundary, internal entrances, and exits. The data collection time interval is 60 s, and the total duration of one simulation is 10,500 s. To study the efficiency and capacity of the network comprehensively, cars are continuously input with the time increment of  $\Delta_t$  in each scenario. By continuously increasing the input of vehicles in the road network, the number of vehicles running on the

road network increases continuously. As the density of vehicles in the road network increases, the network traffic state ranges from a smooth state to a saturated state, then congested state, and finally severe congestion state. Therefore, one can collect the traffic parameters from the smooth state to the severely congested state for MFD and p-MFD modelling. In this paper,  $\Delta_t$  is set to 200 s, and the car arrival patterns on each grade of the road are shown in Table 3. The flow of cars entering the road network increases according to a certain pattern, which makes the road network transit to a severely congested state from a smooth state. The number of vehicles entering and leaving the road network at each interval during the simulation period is obtained. Five simulations are run for each scenario by changing the random seed such as to reduce the random errors.

In this experiment,  $n_c = 1.3$ , and  $n_b = 72$ . Notably, for a realistic purpose, the value of  $n_c$  in this paper is set as the observed average occupancy of cars, similar to reference [28]. The values of  $n_c$  and  $n_b$  can be determined through field investigation in practice. Every set of data is processed by equations (1)–(3) and (4)–(7). The four-segment linear function model of MFD, car p-MFD, and bus p-MFD under different scenarios are obtained by the modelling methods proposed, and then road network p-MFD is obtained by equation (12). Finally, the characteristic parameters of the road network MFD and p-MFD are analysed.

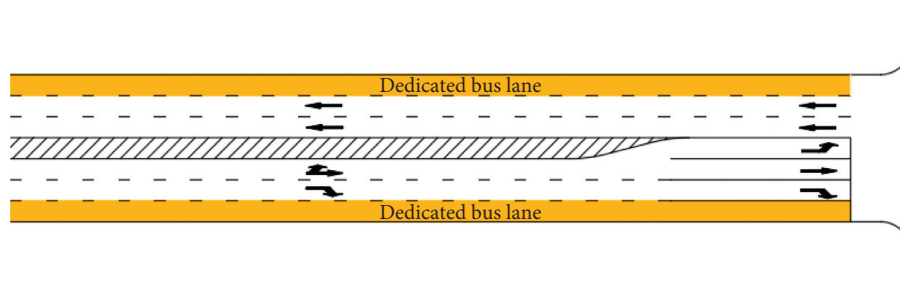


FIGURE 5: Schematic diagram of dedicated bus lanes.

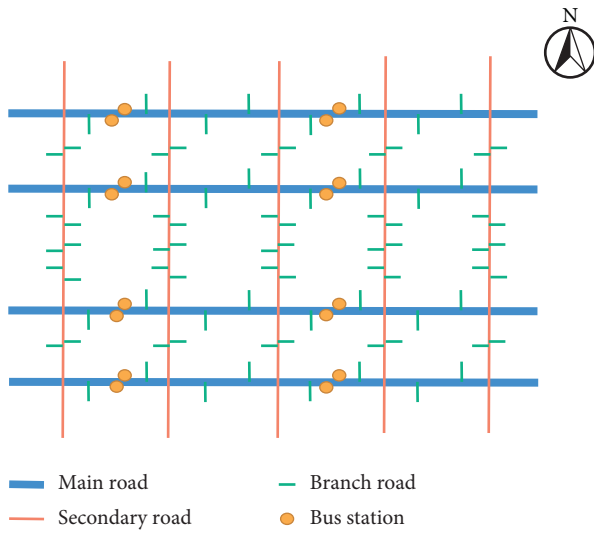


FIGURE 6: Experimental grid road network.

TABLE 3: Traffic arrival pattern in the experimental simulation network.

Road grade	Increase of car flow (veh)	Maximum value of car flow (veh/ln/h)
Main road	100	1,500
Secondary road	50	800
Branch road	10	150 ~ 300

In addition, the values of  $n_b$  is related to the average number of passengers carried by buses. It mainly affects bus p-MFD, which reflects the network performance of carrying bus passengers. Compared with car passenger occupancy, the value and fluctuation of bus passenger occupancy are greater, which has a more significant impact on p-MFD. Therefore, taking the proportion of dedicated bus lanes as 0.11 as an example, the bus p-MFD and network p-MFD with the values of  $n_b$  changing from 10 to 90 are analysed.

#### 4.2. Experimental Results

4.2.1. MFD. When the dedicated bus lane proportion is 0.11, the MFD is taken as an example to show the modelling result of MFD, as shown in Figure 7(a). Figures 7(b) and 7(c) show

the comparisons of the MFD linear functions under the different proportions of dedicated bus lanes. Table 4 shows the segmented expressions of MFD with different proportions of dedicated bus lanes obtained by GMM clustering and linear fitting.

In Figure 7, it can be seen that the road network has not reached the final paralysis point at the end of the simulation, which means the flow of vehicles leaving the road network is not 0. Although the entire road network has serious congestion, and most vehicles cannot leave the road network, there are still a small number of vehicles near the exits that can still leave the road network. Figures 7(b) and 7(c) show the trend of MFD as the proportion of dedicated bus lanes increases. With the increase of the proportion of dedicated bus lanes, the slope of the rising section gradually increases, and the front- and the back-inflection points gradually move toward zero. The difference in the value of the post-stable section is not obvious. The indicators proposed in Section 3 on MFD are analysed in detail in Section 5.

4.2.2. p-MFD. The car p-MFD, bus p-MFD, and road network p-MFD under the proportion of 0.11 of dedicated bus lanes are taken as the example to show the modelling result of p-MFD, as shown in Figures 8(a)–8(c). It can be seen that both car p-MFD and bus p-MFD have consistent curve shape of rising, stable, descending, and finally stable trends in Figures 8(a) and 8(b). However, there are many differences in the inflection points, peak value, and slopes. Tables 5 and 6 show the segmented expressions of car p-MFD and bus p-MFD with different proportions of dedicated bus lanes obtained by GMM clustering and linear fitting.

According to Figure 8(c), there are three different inflection points for the car p-MFD and the bus p-MFD. Therefore, the road network p-MFD by equation (12) has six inflection points with a seven-segment linear function expression. The rising section ( $g_p > 0$ ) has two segments; the descending section ( $g_p < 0$ ) has three; and the stable and the post-stable sections ( $g_p = 0$ ) both have one segment. This shows that the number of network p-MFD segments is determined by the inflection points of the car p-MFD and the bus p-MFD, which has a certain degree of uncertainty. However, the general trend of the road network p-MFD is still rising and stable, descending until it reaches a plateau near the minimum value. The rising and descending sections may have multiple segments, which is determined by the

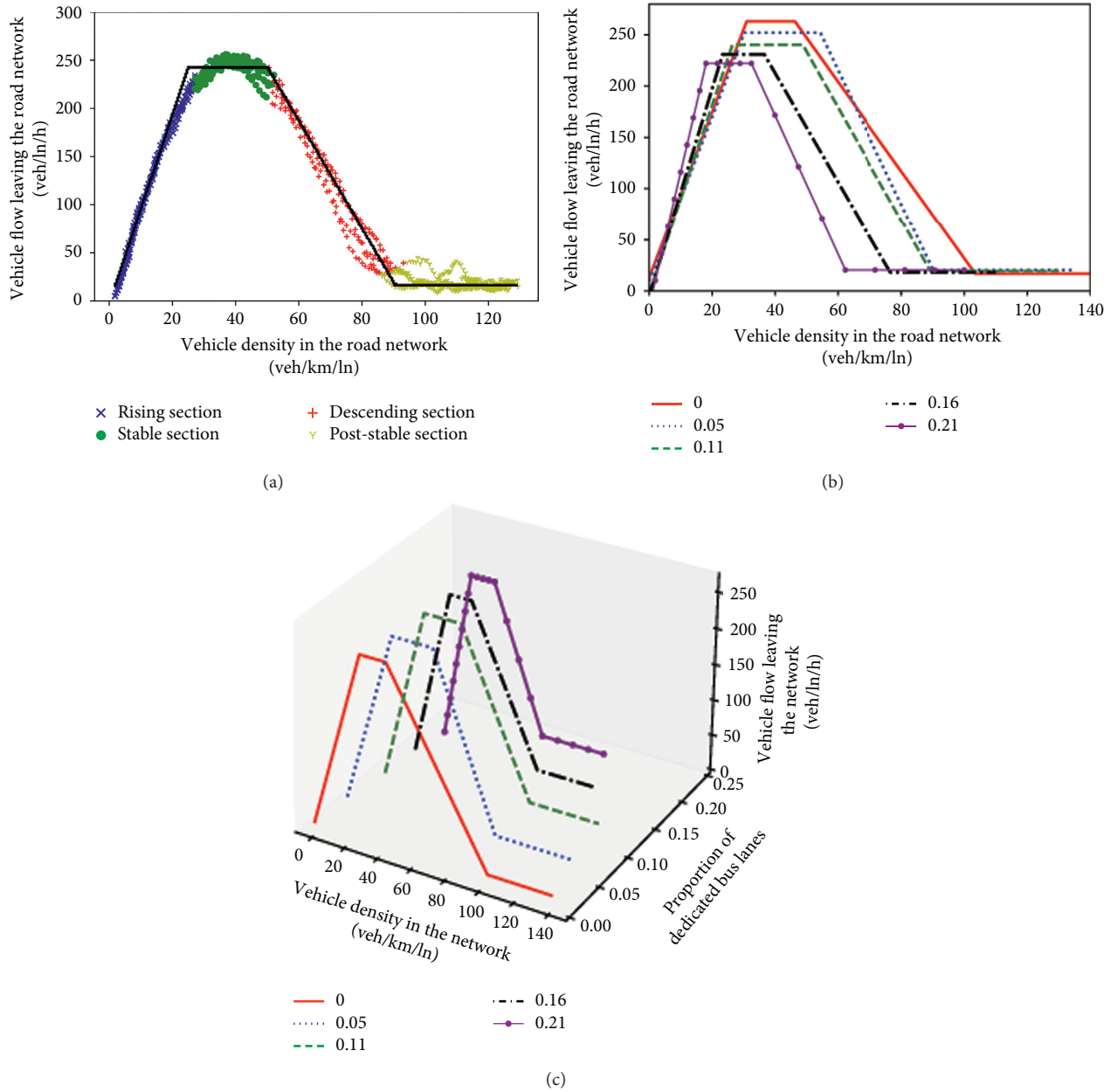


FIGURE 7: (a) MFD cluster fitting results with dedicated bus lane proportion of 0.11, (b) MFD with different proportions of dedicated bus lanes, and (c) MFD with different proportions of dedicated bus lanes.

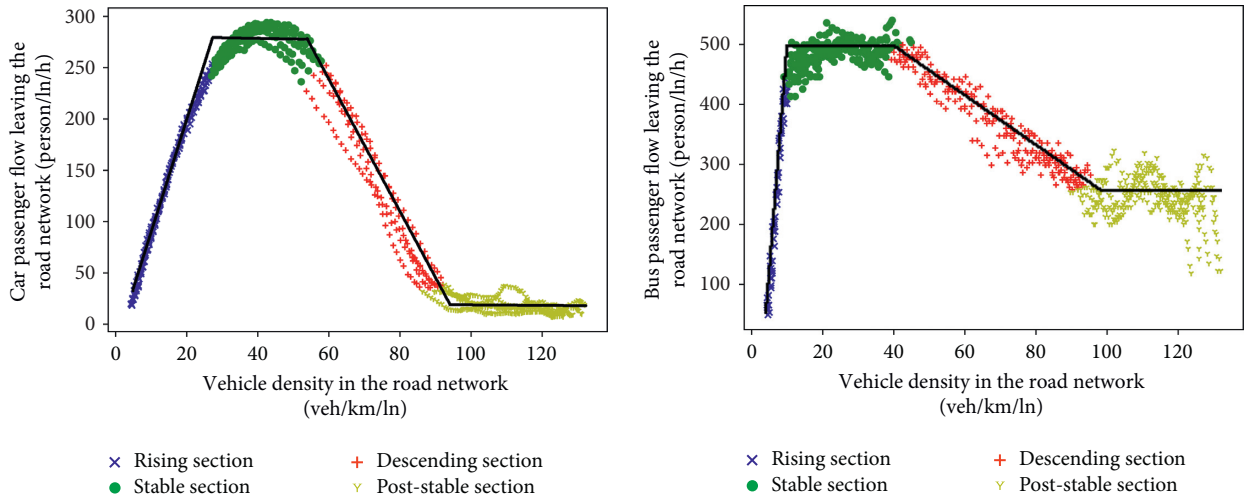
TABLE 4: MFD expressions with different proportions of dedicated bus lanes.

Proportion of dedicated bus lanes	0	0.05	0.11	0.16	0.21
Rising section	8.094k + 12.650	8.032k + 9.511	9.169k - 2.224	10.133k - 5.309	13.201k - 16.332
Stable section	262.915	252.181	240.344	230.968	222.027
Descending section	-4.301k + 461.961	-6.470k + 603.528	-5.451k + 505.999	-5.326k + 425.497	-6.755k + 441.243
Post-stable section	16.955	20.477	19.398	18.283	20.423

speed of the car p-MFD and the bus p-MFD entering and leaving the rising and descending sections, respectively. Notably, when the proportion of dedicated bus lanes is 0.21, all buses in the road network can freely enter and exit the road network through dedicated bus lanes. Therefore, the bus p-MFD has only the rising and the stable sections.

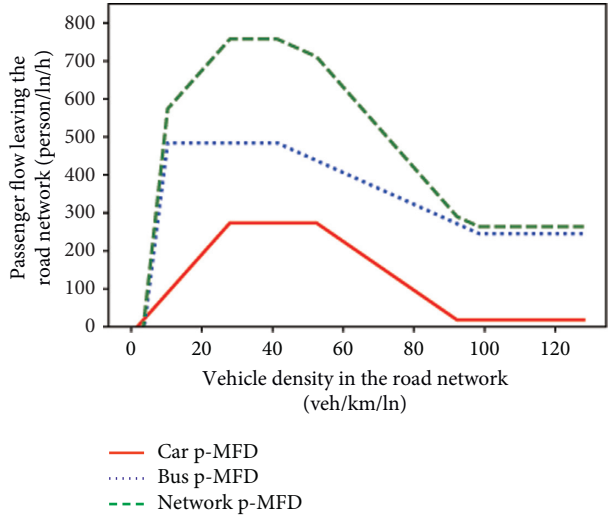
The road network p-MFD with different proportions of dedicated bus lanes is obtained by using equation (12) and the segmented expression of p-MFD for cars and buses. Figures 8(d) and 8(e) show the comparison of the linear function models of p-MFD with different proportions of dedicated bus lanes. From the p-MFD obtained, when the



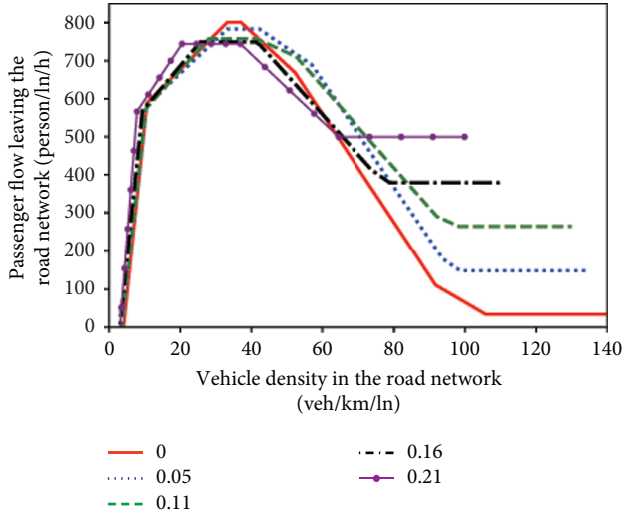


(a)

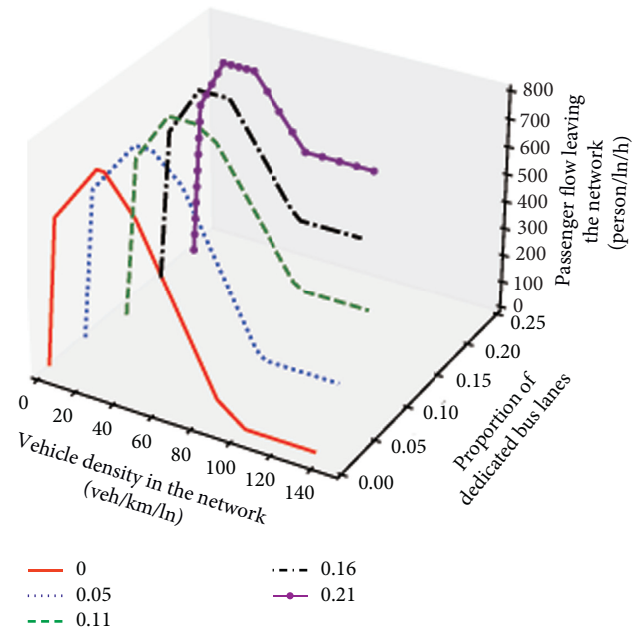
(b)



(c)



(d)



(e)

FIGURE 8: (a) p-MFD cluster fitting results of cars with dedicated bus lane proportion of 0.11, (b) p-MFD cluster fitting results of buses with bus lane proportion of 0.11, (c) p-MFD of road network with dedicated bus lane proportion of 0.11, (d) p-MFD with different proportions of dedicated bus lanes, and (e) p-MFD with different proportions of dedicated bus lanes.

TABLE 5: Car p-MFD expression with different proportions of dedicated bus lanes.

Proportion of dedicated bus lanes	0	0.05	0.11	0.16	0.21
Rising section	9.436k - 0.132	8.819k + 3.010	10.428k - 17.927	11.140k - 17.496	14.101k - 27.883
Stable section	311.952	295.718	273.427	267.126	260.573
Descending section	-5.421k + 594.830	-7.473k + 721.232	-6.419k + 609.990	-6.673k + 541.792	-8.861k + 586.643
Post-stable section	20.890	21.754	18.479	14.609	15.780

TABLE 6: Bus p-MFD expression with different proportions of dedicated bus lanes.

Proportion of dedicated bus lanes	0	0.05	0.11	0.16	0.21
Rising section	80.497k - 365.479	76.460k - 319.657	73.042k - 270.203	78.423k - 252.100	104.309k - 333.356
Stable section	488.711	487.780	484.173	481.680	483.791
Descending section	-8.715k + 811.437	-6.389k + 756.586	-4.182k + 657.039	-3.765k + 641.574	—
Post-stable section	12.843	126.845	245.322	364.262	—

proportion of dedicated bus lanes is no greater than 0.16, the road network p-MFD has a seven-segment linear function; the rising section ( $g_p > 0$ ) has two segments; and the stable and the post-stable sections have one segment, and the descending section ( $g_p < 0$ ) has three segments. When the proportion of dedicated bus lanes is 0.21, the road network p-MFD is a five-segment linear function; the rising section ( $g_p > 0$ ) has two segments; the stable and the post-stable sections have one segment, and the descending section ( $g_p < 0$ ) has also one section. The indicators proposed in Section 3 on p-MFD are analysed in detail in Section 6.

**4.2.3. Impact Analysis of  $n_b$  on p-MFD.** The bus p-MFD and network p-MFD with the values of  $n_b$  changing from 10 to 90 when the proportion of dedicated bus lanes is 0.11 are taken as the example to demonstrate the impact of  $n_b$  on p-MFD, as shown in Figures 9(a) and 9(b). When the average passenger occupancy of buses in the road network increases, the basic shapes of the bus p-MFD and the network p-MFD remain unchanged. It shows that with the increase of the vehicle density, the passenger flow first increases, then stabilizes, and then drops to a certain small value.

For bus p-MFD and network p-MFD, when the values of  $n_b$  increases, changing from 10 to 90, the slopes of the rising section, the values of the stable section, and the post-stable section become larger. According to the indicators based on the p-MFD, it means that when the number of passengers carried by buses in the network becomes larger, the efficiency of transporting passengers improves.

This shows that when the average passenger occupancy of cars in the network is certain, appropriately increasing the average passenger occupancy of buses can improve the transport efficiency of the road network. The optimization of the network transport efficiency can be realized from two aspects. On the one hand, the passenger capacity of buses should be improved, and the carriages should be rationally optimized to increase the number of passengers that can be accommodated. On the other hand, the level of public transportation services should be improved to increase the attractiveness of public transportation.

## 5. Impact Analysis of Dedicated Bus Lanes on MFD

**5.1. Geometric Features.** The proportion of dedicated bus lanes affects the geometric characteristics of MFD, mainly including the rising section, the descending section, the stable section, the points of inflection, and the density range of each section.

**5.1.1. The Rising and Descending Sections.** The absolute values of the slopes of the rising and the descending sections of the MFD under different proportions of dedicated bus lanes are compared as shown in Figure 10. In general, as the proportion of dedicated bus lanes in the road network increases from 0 to 0.21, the slope of the rising section increases. However, the absolute value of the slope of the descending section is less affected. It indicates that the implementation of dedicated bus lanes is conducive to the transport of vehicles in the smooth state of the road network, and as the proportion of dedicated bus lanes increases, this advantage becomes more obvious. When traffic flow in the road network is small, the vehicles in the road network are in a noncongested state. Cars and buses driving in separate lanes can ensure that their speeds are maintained within the ideal range, thereby improving the overall transport efficiency of the road network. Meanwhile, the implementation of dedicated bus lanes has little effect on network efficiency when the road network is congested.

**5.1.2. The Stable Section.** The values of the stable section  $q_{\max}$  with different proportions of dedicated bus lanes are compared as shown in Figure 11. As the proportion of dedicated bus lanes increases from 0 to 0.21, the value of the stable section decreases. It indicates that the implementation of dedicated bus lanes reduces the capacity of transporting vehicles when the road network is in a saturated state. The impact is more significant with the increase in the proportion of dedicated bus lanes. Due to the limited road resources, the increase in the number of dedicated bus lanes will cause the reduction of the number of passenger-car

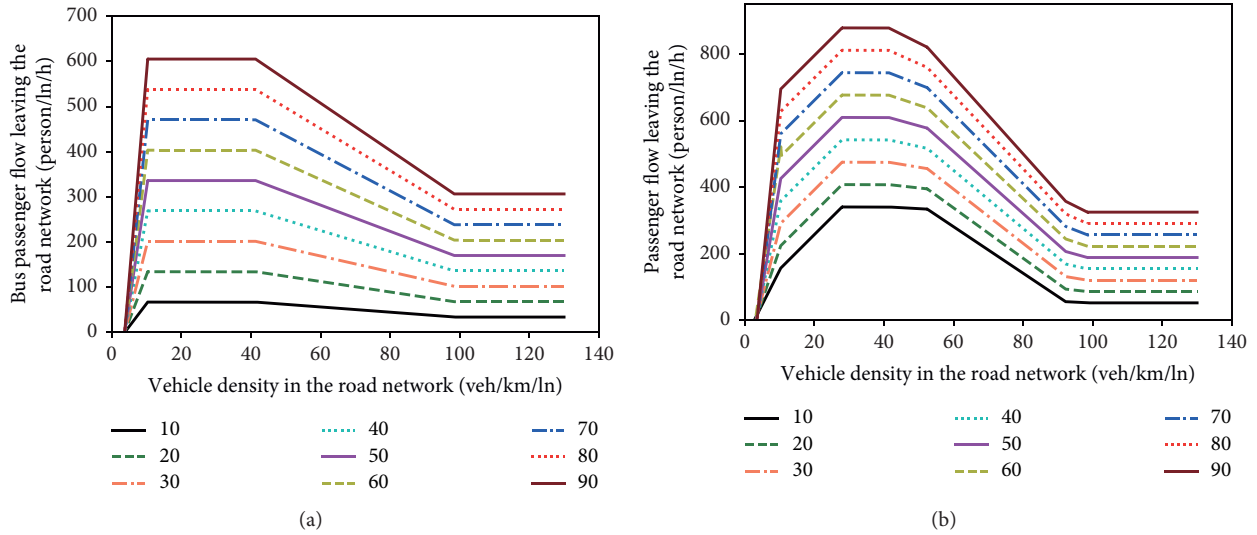


FIGURE 9: (a) Bus p-MFD with different  $n_b$  (person/veh) and (b) network p-MFD with different  $n_b$  (person/veh).

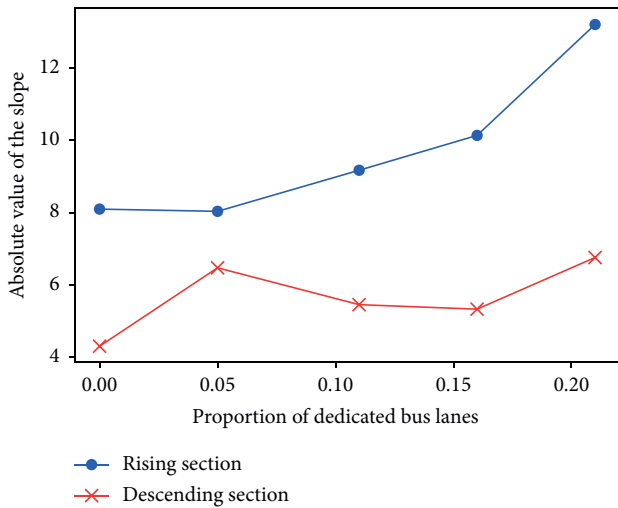


FIGURE 10: The slope of MFD under the different proportions of dedicated bus lanes.

lanes. The number of buses running in the dedicated bus lane is related to the setting of the bus lanes, the bus departure interval, and so on. Due to its transport characteristics and public service attributes, the output flow of the dedicated bus lanes is less than the output flow of the passenger-car lanes. Therefore, the implementation of dedicated bus lanes reduces the number of vehicles that can be served by the road network.

5.1.3. *The Points of Inflection.* The horizontal and vertical coordinates of the inflection points of the MFD with different proportions of dedicated bus lanes are compared as shown in Figure 12. As the proportion of dedicated bus lanes increases from 0 to 0.21, the values of  $k_b^s$  and  $q_{max}$  both show a decreasing trend, and the value of  $k_b^d$  first increases and then decreases. The position of the front-inflection point moves roughly toward the zero point. It shows that the

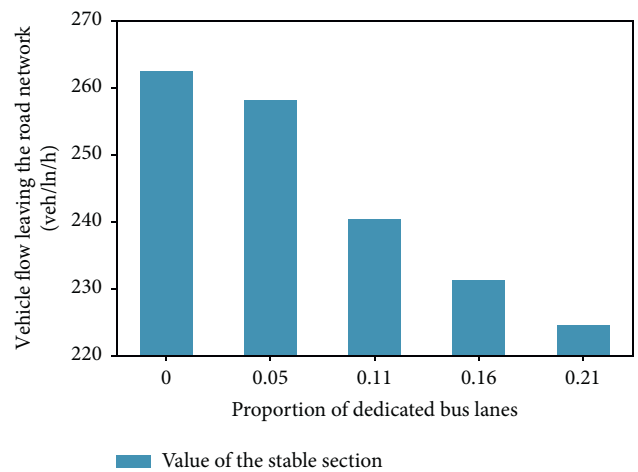


FIGURE 11: Value of the stable section of MFD with different proportions of bus lanes.

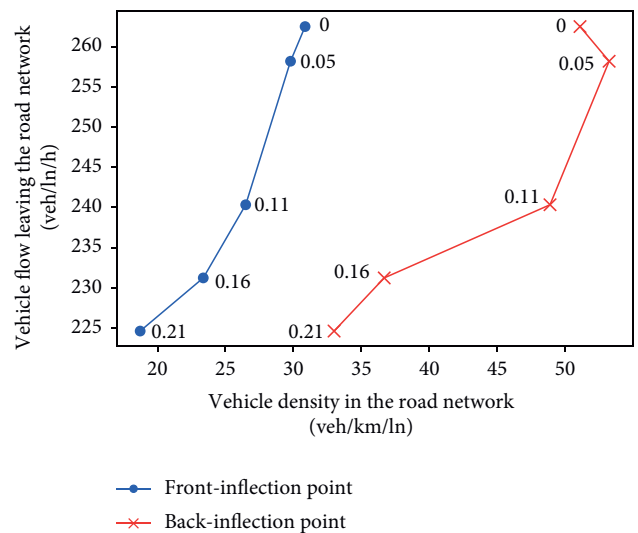


FIGURE 12: Inflection point of MFD with different proportions of dedicated bus lanes.

implementation of dedicated bus lanes makes the road network transit from the smooth state to the saturated state faster, and as the proportion of dedicated bus lanes increases, the changing speed of the transport state increases. The abscissa of the back-inflection point first increases and then decreases, indicating that when the proportion of dedicated bus lanes is set in the range of 0 ~ 0.11, the speed at which the road network enters congestion will slow down. As the proportion of dedicated bus lanes continues to increase, the road network will quickly change from a saturated state to a congested state.

The implementation of dedicated bus lanes in the road network causes the reduction of the number of passenger-car lanes and also the access of vehicles that the road network can withstand. When the proportion of dedicated bus lanes is in the range of 0 ~ 0.11, buses and cars running in different lanes effectively reduce the influence of each other. It is beneficial to increase the number of vehicles effectively borne by the road network. However, as the proportion of dedicated bus lanes continues to increase, passenger-car lanes are reduced more, and the number of vehicles effectively borne by the road network decreases sharply.

*5.1.4. The Density Range of Each Section.* The density range of each section  $\Delta k_{\text{range}}^x$  with different proportions of dedicated bus lanes is compared as shown in Figure 13. When the proportion of dedicated bus lanes in the road network is within the range of 0 ~ 0.11,  $\Delta k_{\text{range}}^s$  and  $k_b^d$  are relatively large, respectively. It means the road network can effectively bear more vehicles and maintain high-efficiency operation in a long density range. In addition, it can be seen that the implementation of dedicated bus lanes has led to a decrease in traffic density on the road network, which indicates that the traffic density on dedicated bus lanes is less than that on the normal lanes. The reason is that dedicated bus lanes can maintain a certain driving speed. However, the provision of dedicated bus lanes reduces the utilization of road resources by vehicles to a certain extent.

*5.2. Goodness of Fit.* The proportion of dedicated bus lanes affects the clustering of MFD scattered points, and the goodness of fit is studied.

The goodness of fit of the rising section and that of the descending section,  $R^2$ , with different proportions of dedicated bus lanes are compared as shown in Figure 14. As the proportion of dedicated bus lanes increases from 0 to 0.21, the  $R^2$  of the rising and the descending sections shows an increasing trend. The scatter points of the MFD with a certain proportion of dedicated bus lanes are more concentrated than the ones with no dedicated bus lanes. The implementation of dedicated bus lanes is conducive to maintain the stability of road network operation. The main reason is that the dedicated bus lanes allow buses and cars to drive in different lanes, reducing the mutual influence caused by the difference in operating characteristics between them, thereby improving the stability of road network operation.

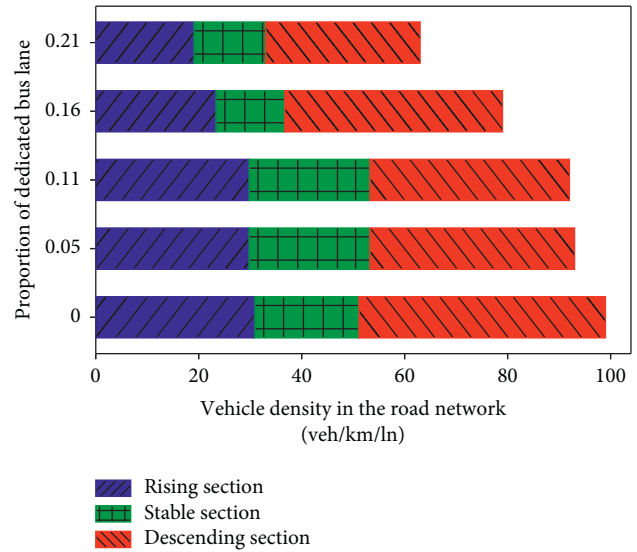


FIGURE 13: Density range of MFD with different proportions of bus lanes.

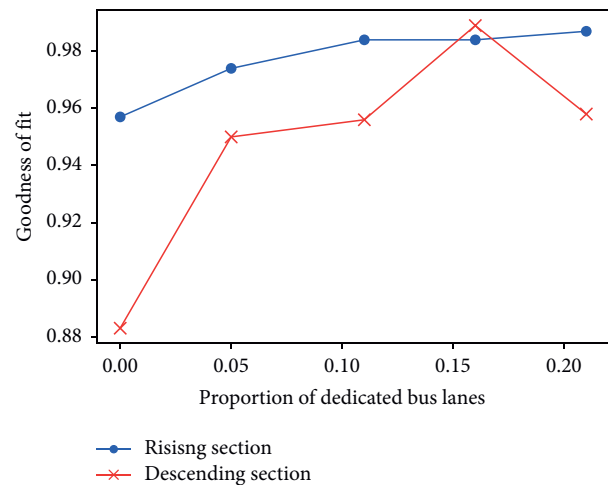


FIGURE 14:  $R^2$  of MFD with different proportions of dedicated bus lanes.

In summary, the proportion of dedicated bus lanes has certain influences on MFD, especially from the following perspective: the slopes of the rising section, the value of the stable section, the inflection points, density range, and goodness of fit. The impacts of the proportion of dedicated bus lanes on the slopes of the descending section and the value of the post-stable section are not obvious.

## 6. Impact Analysis of Dedicated Bus Lanes on p-MFD

The proportion of dedicated bus lanes affects the geometric characteristics of p-MFD, mainly including the rising, the descending, the stable and the post-stable sections.

*6.1. The Rising and the Descending Sections.* The slopes of the rising section of the p-MFD with different dedicated bus

lanes proportions are compared as shown in Figure 15. The slope of the first segment belonging to the rising section is greater than the second, indicating that there are two stages in the process of transporting passengers when the road network is in a smooth state. According to equation (13), the slope of the p-MFD rising section is the sum of the slopes of the rising section of the corresponding car p-MFD and the bus p-MFD. When  $0 < k < \min\{k_b^1, k_c^1\}$ , the bus p-MFD enters the stable section first since  $k_c^1 > k_b^1$ . At this time,  $g_{pb}(k) = 0$ , so the slope of the second segment that belongs to the rising section of p-MFD is equal to the slope of car p-MFD. This shows that in the first stage, with the continuous input of vehicles, passenger transport of cars and buses is in a smooth state, and the passenger transport efficiency reaches the maximum. In the second stage, as the density of vehicles in the road network continues to increase, bus passenger transport reaches saturation first. At this time, the flow of the road network to transport passengers continues to increase. It is worth noting that the speed at which the passenger transport of buses reaches saturation is related to the interval between bus departures and the number of bus routes on dedicated bus lanes.

The absolute slopes of the p-MFD descending section with different dedicated bus lanes proportions are compared as shown in Figure 16. The slope value of the descending section changes twice, indicating that there are three stages in the process of transporting passengers when the road network is in a congested state.

When the proportion of bus lanes is 0, the slope of the first stage corresponds to that of the bus p-MFD. The slope of the second stage is corresponding to the sum of the slope of the descending section of the car p-MFD and the bus p-MFD. The slope of the third stage is corresponding to that of the car p-MFD. This shows that when the road network is not implemented with dedicated bus lanes, in the first stage, with the continuous input of vehicles, some intersections and road sections on the road network will be blocked, and the passenger transport of the bus first reaches the congested state. At this time, the flow of the road network to transport passengers continues to decline, and the efficiency of bus passenger transport has slowed down. In the second stage, as the traffic density continues to increase, both bus and car passenger transports have reached the congested state. At this time, the passenger flow of the road network continues to drop, and the efficiency of the passenger transport is significantly reduced. In the third stage, the passenger transport capacity of buses is reduced to the minimum, and most bus passengers cannot leave the road network. At this time, the flow of the road network to transport passengers continues to decline.

If the proportion of dedicated bus lanes is greater than 0, the slope of the first and third stages of the descending section corresponds to that of the bus, and the slope of the second stage of the p-MFD descending section is the sum of the slopes of the descending sections of the corresponding car p-MFD and the bus p-MFD. It indicates that the passenger transport in the first and second stages of the road network is the same as the transport without dedicated bus lanes. The difference is that in the third stage, the passenger transport capacity of cars is reduced to the minimum, and most car passengers cannot leave the road network. At this

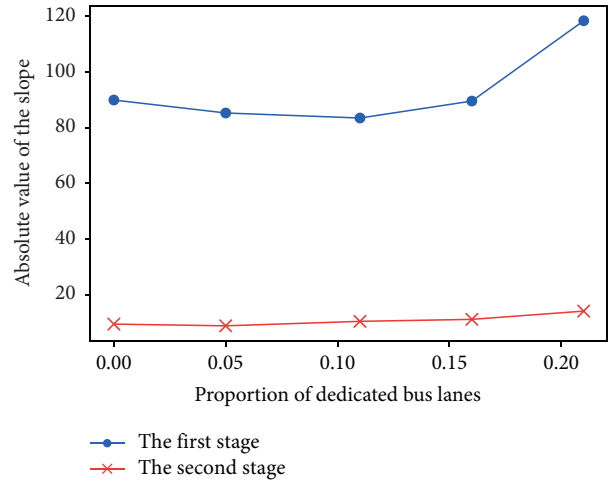


FIGURE 15: Slope of the rising section of p-MFD with different proportions of bus lanes.

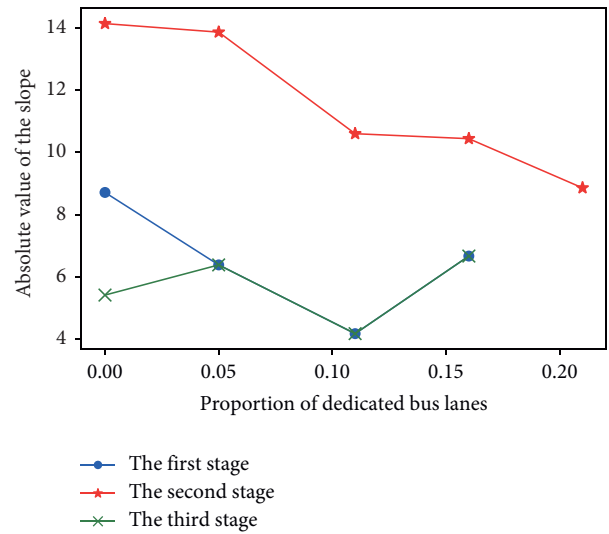


FIGURE 16: Slope of the descending section of p-MFD with different proportions of bus lanes.

time, the flow of the road network to transport passengers continues to decline. This indicates that when the road network is congested, bus passengers can use the dedicated bus lanes to enter quickly and exit the road network, while the serious blockage of passenger-car lanes makes it difficult for car passengers to do so.

It can be seen from Figure 16 that as the proportion of dedicated bus lanes increases from 0 to 0.21, the absolute value of the slope of the descending section in the second stage has a decreasing trend. It indicates that the dedicated bus lanes improve the transport efficiency of passengers when a road network is in a congested state. As the proportion increases, the impact becomes more significant.

**6.2. The Stable Section.** The values of the stable section of the p-MFD with different proportions of dedicated bus lanes are compared as shown in Figure 17(a). As the proportion of

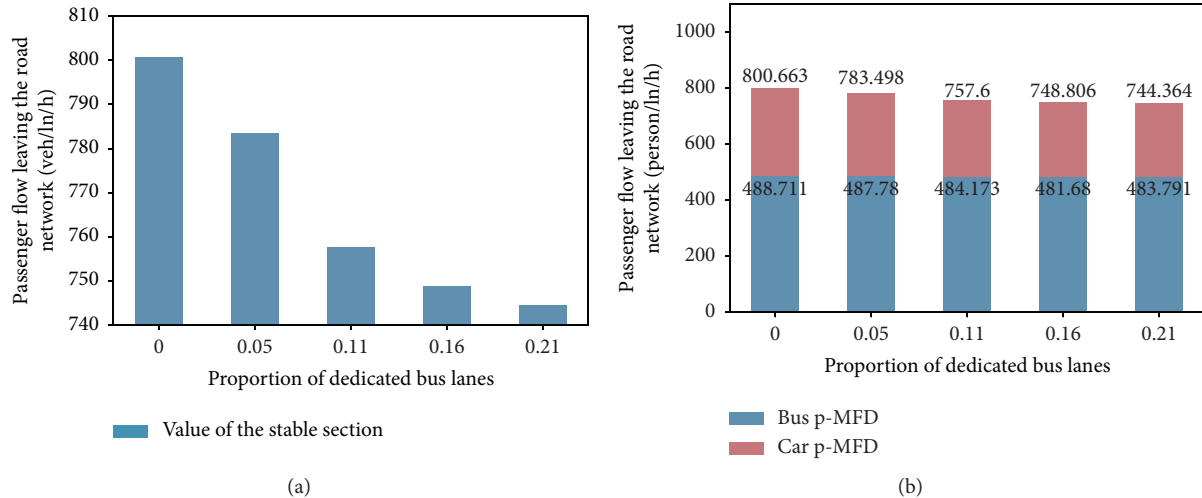


FIGURE 17: (a) Value of stable section of p-MFD with different proportions of bus lanes and (b) value of car p-MFD and bus p-MFD stable section with different proportions of bus lanes.

dedicated bus lanes increases from 0 to 0.21, the value of the stable section continues to decrease. It indicates that dedicated bus lanes reduce the capacity of the road network to transport passengers in a saturated state. The proportion increases, and the impact is more significant.

The values of the stable section of the bus p-MFD and the car p-MFD with different proportions of dedicated bus lanes are compared, as shown in Figure 17(b). It can be seen from Figure 17(b) that as the proportion of dedicated bus lanes increases, the value of the stable section of bus p-MFD fluctuates around 484 persons/lane/h, while the value of the stable section of car p-MFD continues to decrease. This indicates that the dedicated bus lanes reduce the ability of the road network to withstand the entry and exit of car passengers. At this time, dedicated bus lanes do not bring obvious benefits to the road network to transport bus passengers. The main reason is that in this state, the road network transport is able to maintain efficient operation. Most of the roads have not been congested, and at the same time, the passenger transport of buses is saturated according to the bus service level. Therefore, even if there is no dedicated bus lane, buses can still efficiently transport passengers in and out of the road network.

**6.3. The Post-Stable Section.** The values of the post-stable section of the p-MFD with different proportions of dedicated bus lanes are compared as shown in Figure 18. As the proportion of dedicated bus lanes increases from 0 to 0.21, the value of the post-stable section continues to increase. It indicates that the dedicated bus lanes improve the ability of the road network to transport passengers in the post-congested state. The impact is more significant when the proportion increases. The main reason is that when the road network is in a post-congested state, most of the intersections and road sections in the road network are severely congested. Without dedicated bus lanes, most car passengers and bus passengers cannot leave the road network. However, the dedicated bus lanes can ensure the passage of buses;

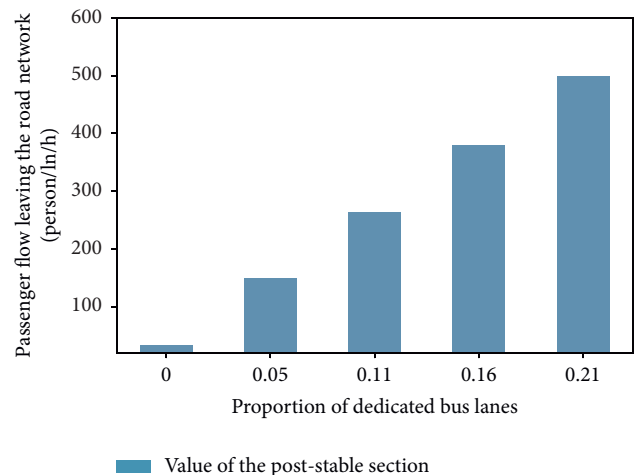


FIGURE 18: Value of p-MFD post-stable section with different proportions of dedicated bus lanes.

therefore, the road network can maintain efficient transport of bus passengers when most roads are blocked.

In summary, the proportion of dedicated bus lanes has certain influences on p-MFD, specifically from the following perspective: the slopes of the rising and the descending sections and the values of the stable and the post-stable sections. Since the number of segments of network p-MFD is flexible, the patterns of the other indicators caused by the impacts of the proportion of dedicated bus lanes are not obvious.

## 7. Conclusions

This paper proposes the description and modelling methods of MFD and p-MFD based on GMM and linear fitting and the corresponding performance indicators of urban road networks. The methodology proposed in this study can be applicable to various scenarios of urban road networks and

even can be extended to multiple modes (cars, buses, etc.). A grid network is taken as a case study such as to analyse the impact of different dedicated bus lane proportions on network performance via microscopic traffic simulation for MFD and p-MFD modelling. There are some interesting findings as follows:

- (i) The larger the proportion of dedicated bus lanes, the more the rising and the stable sections of MFD, and the descending and post-stable sections of p-MFD are affected.
- (ii) The slope of the MFD rising section tends to increase, and the value of the MFD stable section tends to decrease as the proportion of dedicated bus lanes increases, which indicates that dedicated bus lane improves the efficiency of vehicle transport under the smooth state of the road network. However, it might reduce the vehicle throughput of the road network.
- (iii) The slope of the p-MFD descending section and the value of the post-stable section tend to increase when the proportion of dedicated bus lanes is increased. It indicates that dedicated bus lanes improve the efficiency of passenger transport under the saturated state of the road network. The impact is more significant when the proportion grows.
- (iv) When the proportion of dedicated bus lanes on the road network is within the range of 0 to 0.11, the road network can effectively bear more vehicles and maintain a longer density range with efficient operation.
- (v) As the proportion of dedicated bus lanes increases, the goodness of fit of the MFD rising and descending sections tend to increase, which indicates that dedicated bus lanes are conducive to improve the transport stability of the road network.
- (vi) As the bus passenger occupancy increases, the slopes of the rising section and the values of the stable and the post-stable sections of p-MFD become larger.

To sum up, for the simulated road network, dedicated bus lanes can improve the efficiency of vehicle and passenger transport in the smooth state of the road network, also ensure that the road network continues to carry passengers in the congested state, and improve the stability of the road network transport. However, in the meantime, dedicated bus lanes also affect the ability of the road network to withstand the entry and exit of car passengers. Therefore, when implementing dedicated bus lanes, reasonable consideration should be given to the proportion and the operating period. The operation state of the road network is recommended to be monitored in real-time, and dedicated bus lanes are suggested to be implemented when the road network is in the smooth and the post-congested states. In addition, the proportion of dedicated bus lanes should be reasonably allocated based on bus demand and operation conditions, which provides a reference for better implementation of

dedicated bus lanes. Besides, the bus passenger capacity and bus service should be improved to increase the attractiveness of public transportation.

This paper proposes a group of indicators based on MFD and p-MFD and summarizes the influence mechanism of dedicated bus lanes on road network transport in different states, aiming to support and expand existing research. In the future, the influence of different forms of bus lanes, operating periods of bus lanes, as well as the influence of bus signal priority on MFD and p-MFD should be studied to assist traffic planning and management departments to optimize the public transport system further. Meanwhile, it will help increase the attractiveness of public transport and improve the comprehensive transport utility of the road network.

### Data Availability

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

### Conflicts of Interest

The authors declare that there is no conflict of interest.

### Acknowledgments

The authors acknowledge Xiaoran Qin from the Hong Kong University of Science and Technology for her valuable comments on this research. This paper was supported by the General Project of the National Natural Science Foundation of China (52072129).

### References



- [1] M. Garcia and K. Yamamoto, "Busways and bus lanes in Brazil and Japan," in *Proceedings of the VNIS'94-1994 Vehicle Navigation and Information Systems Conference*, Yokohama, Japan, August 1994.
- [2] Ministry of Transport of the People's Republic of China, *Statistical Bulletin on the Development of the Transportation Industry in 2019*, 2019.
- [3] N. Chiabaut, X. Xie, and L. Leclercq, "Performance analysis for different designs of a multimodal urban arterial," *Transportation Business: Transport Dynamics*, vol. 2, no. 3, pp. 229–245, 2014.
- [4] J. Zhao, J. Yu, X. Xia, J. Ye, and Y. Yuan, "Exclusive bus lane network design: a perspective from intersection operational dynamics," *Networks and Spatial Economics*, vol. 19, no. 4, pp. 1143–1171, 2019.
- [5] C. F. Daganzo, "Urban gridlock: macroscopic modeling and mitigation approaches," *Transportation Research Part B: Methodological*, vol. 41, no. 1, pp. 49–62, 2007.
- [6] N. Geroliminis and C. F. Daganzo, "Existence of urban-scale macroscopic fundamental diagrams: some experimental findings," *Transportation Research Part B: Methodological*, vol. 42, no. 9, pp. 759–770, 2008.
- [7] C. F. Daganzo and N. Geroliminis, "An analytical approximation for the macroscopic fundamental diagram of urban traffic," *Transportation Research Part B: Methodological*, vol. 42, no. 9, pp. 771–781, 2008.

- [8] M. J. Cassidy, K. Jang, and C. F. Daganzo, "Macroscopic fundamental diagrams for freeway networks: theory and observation," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2260, no. 1, pp. 8–15, 2011.
- [9] V. L. Knoop and S. P. Hoogendoorn, "Empirics of a generalized macroscopic fundamental diagram for urban freeways," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2391, no. 1, pp. 133–141, 2013.
- [10] N. Zheng and N. Geroliminis, "On the distribution of urban road space for multimodal congested networks," *Procedia-Social and Behavioral Sciences*, vol. 80, pp. 119–138, 2013.
- [11] N. Chiabaut, "Evaluation of a multimodal urban arterial: the passenger macroscopic fundamental diagram," *Transportation Research Part B: Methodological*, vol. 81, pp. 410–420, 2015.
- [12] A. Loder, L. Ambühl, M. Menendez, and K. W. Axhausen, "Empirics of multi-modal traffic networks-using the 3D macroscopic fundamental diagram," *Transportation Research Part C: Emerging Technologies*, vol. 82, pp. 88–101, 2017.
- [13] C. Buisson and C. Ladier, "Exploring the impact of homogeneity of traffic measurements on the existence of macroscopic fundamental diagrams," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2124, no. 1, pp. 127–136, 2009.
- [14] J. Haddad and N. Geroliminis, "On the stability of traffic perimeter control in two-region urban cities," *Transportation Research Part B: Methodological*, vol. 46, no. 9, pp. 1159–1176, 2012.
- [15] T. Tsubota, A. Bhaskar, and E. Chung, "Macroscopic fundamental diagram for Brisbane, Australia," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2421, no. 1, pp. 12–21, 2014.
- [16] C. Huang, N. Zheng, and J. Zhang, "Investigation of bimodal macroscopic fundamental diagrams in large-scale urban networks: empirical study with GPS data for Shenzhen city," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 6, pp. 114–128, 2019.
- [17] E. Saffari, M. Yildirimoglu, and M. Hickman, "A methodology for identifying critical links and estimating macroscopic fundamental diagram in large-scale urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 119, 2020.
- [18] G. Tilg, S. Amini, and F. Busch, "Evaluation of analytical approximation methods for the macroscopic fundamental diagram," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 1–19, 2020.
- [19] Z. Y. Gu and M. Saberi, "Simulation-based optimization of toll pricing in large-scale urban networks using the network fundamental diagram: a cross-comparison of methods," *Transportation Research Part C: Emerging Technologies*, vol. 122, 2021.
- [20] Z. Gu, F. Safarighouzhdi, M. Saberi, and T. H. Rashidi, "A macro-micro approach to modeling parking," *Transportation Research Part B: Methodological*, vol. 147, pp. 220–244, 2021.
- [21] Z. Y. Gu, A. Najmi, M. Saberi et al., "Macroscopic parking dynamics modeling and optimal real-time pricing considering cruising-for-parking," *Transportation Research Part C: Emerging Technologies*, vol. 118, 2020.
- [22] I. Dakic, K. Yang, M. Menendez, and J. Y. J. Chow, "On the design of an optimal flexible bus dispatching system with modular bus units: using the three-dimensional macroscopic fundamental diagram," *Transportation Research Part B: Methodological*, vol. 148, pp. 38–59, 2021.
- [23] S. Sirmatel, D. Tsitsokas, A. Kouvelas et al., "Modeling, estimation, and control in large-scale urban road networks with remaining travel distance dynamics," *Transportation Research Part C: Emerging Technologies*, vol. 128, 2021.
- [24] W. Lu, J. Liu, J. Mao, G. Hu, C. Gao, and L. Liu, "Macroscopic fundamental diagram approach to evaluating the performance of regional traffic controls," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 7, pp. 420–430, 2020.
- [25] L. Ambühl, A. Loder, M. C. J. Bliemer et al., "A functional form with a physical meaning for the macroscopic fundamental diagram," *Transportation Research Part B-Methodology*, vol. 137, 2020.
- [26] S. Sirmatel and N. Geroliminis, "Stabilization of city-scale road traffic networks via macroscopic fundamental diagram-based model predictive perimeter control," *Control Engineering Practice*, vol. 109, 2021.
- [27] N. Zheng, K. Aboudolas, N. Geroliminis et al., "Investigation of a city-scale three-dimensional macroscopic fundamental diagram for Bi-modal urban traffic," in *Proceedings of the 2013 16th International IEEE Conference on Intelligent Transportation Systems*, pp. 1029–1034, Hague, The Netherlands, October 2013.
- [28] N. Geroliminis, N. Zheng, and K. Ampountolas, "A three-dimensional macroscopic fundamental diagram for mixed bi-modal urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 42, pp. 168–181, 2014.
- [29] J. Ortigosa, M. Menendez, N. Zheng et al., "Analysis of the 3D-vMFDs of the urban networks of Zurich and San Francisco," in *Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Spain, September 2015.
- [30] T. Dantsuji, D. Fukuda, N. Zheng et al., "A macroscopic approach for optimizing road space allocation of bus lanes in multimodal urban networks through simulation analysis: an application to the Tokyo CBD network," in *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems*, Yokohama, Japan, October 2017.
- [31] F. Zhang, N. Zheng, H. Yang, and N. Geroliminis, "A systematic analysis of multimodal transport systems with road space distribution and responsive bus service," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 208–230, 2018.
- [32] I. Dakic, L. Ambühl, O. Schümperlin, and M. Menendez, "On the modeling of passenger mobility for stochastic bi-modal urban corridors," *Transportation Research Part C: Emerging Technologies*, vol. 113, pp. 146–163, 2020.
- [33] M. Yingying, W. Chen, and J. Zehao, "Sensitivity analysis of MFD of ring radial road network on signal cycle," *Journal of Transportation Systems Engineering and Information Technology*, vol. 19, pp. 78–85+93, 2019.



## Research Article

# Highway Traffic Speed Prediction in Rainy Environment Based on APSO-GRU

Dongqing Han,<sup>1</sup> Xin Yang,<sup>1</sup> Guang Li,<sup>2</sup> Shuangyin Wang,<sup>1</sup> Zhen Wang ,<sup>3</sup>  
and Jiandong Zhao ,<sup>3,4</sup>

<sup>1</sup>Zhong Dian Jian Ji Jiao Highway Investment Development Company Limited, Shijiazhuang, Hebei 050090, China

<sup>2</sup>Hebei Intelligent Transportation Technology Co., Ltd of HEBTIG, Shijiazhuang, Hebei 050090, China

<sup>3</sup>School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

<sup>4</sup>Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Ministry of Transport, Beijing 100044, China

Correspondence should be addressed to Jiandong Zhao; zhaojd@bjtu.edu.cn

Received 12 July 2021; Accepted 4 August 2021; Published 18 August 2021

Academic Editor: Ming Xu

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

In order to accurately analyse the impact of the rainy environment on the characteristics of highway traffic flow, a short-term traffic flow speed prediction model based on gate recurrent unit (GRU) and adaptive nonlinear inertia weight particle swarm optimization (APSO) was proposed. Firstly, the rainfall and highway traffic flow data were cleaned, and then they are matched according to the spatiotemporal relationship. Secondly, through the method of multivariate analysis of variance, the significance of the impact of potential factors on traffic flow speed was explored. Then, a GRU-based traffic flow speed prediction model in rainy environment is proposed, and the actual road sections under different rainfall scenarios were verified. After that, in view of the problem that the prediction accuracy of the GRU model was low in the continuous rainfall scenario, the APSO algorithm was used to optimize the parameters of the GRU network, and the APSO-GRU prediction model was constructed and verifications under the same road section and rain scene were carried out. The results show that the APSO-GRU model has significantly improved prediction stability than the GRU model and can better extract rainfall features during continuous rainfall, with an average prediction accuracy rate of 96.74%.

## 1. Introduction

Rainfall is the most frequently occurring severe weather, which brings serious impact to highway traffic safety. It is important to study the traffic flow characteristics of highways under rainy environment and grasp the regularities of rainfall on traffic flow, making stable prediction and analysis of traffic flow to implement effective traffic control [1–4].

In terms of the impact of adverse weather on traffic flow characteristics, with the improvement of the highway system and the continuous development of information observation and collection technology, scholars at home and abroad have conducted continuous research [5–8]. At present, the data analysis and modelling system for the

impact of weather factors on highway traffic flow is well established. In 1994, Ibrahim and Hall [9] studied the impact of adverse weather on the flow-occupancy and speed-flow relationships through regression analysis and showed that heavy rain and snow caused a 10–20% and 30%–48% reduction in maximum highway flow, respectively. In 2005, Agarwal et al. [10] used years of traffic flow data and contemporaneous weather data of the Northern United States to quantify the effects of adverse weather conditions and roadway conditions on highway traffic flow. The results showed that heavy rain, snow, and low visibility resulted in a 10%–17%, 19%–27%, and 12% of reduction in capacity, respectively, and a reduction of vehicle speed by 4%–7%, 11%–15%, and 10%–12%, respectively. In 2015, Li [11] derived the mean values of vehicle speeds of different

rainfall intensities on highways based on statistical analysis of data, used standard deviations to measure the dispersion of vehicle speeds, and investigated the variability of vehicle speed on different lane locations, vehicle types, and time periods during rainfall.

In terms of traffic flow prediction models, they are mainly divided into prediction models based on statistical theory analysis, nonlinear theory prediction models, machine learning prediction models, and combined prediction models [12]. However, nonlinear theoretical model related theories are very complex, especially in terms of mathematical processing. The model has a high degree of complexity and a large amount of calculation, which is suitable for more complicated emergency transportation systems. With the rise of artificial intelligence, machine learning has been more often applied to the field of traffic flow prediction, and related prediction algorithms have emerged. They are mainly divided into support vector machines, artificial neural networks, and deep learning [13]. In 2009, Castro-Neto et al. [14] proposed a supervised online SVR statistical learning model, which optimized the problem of limited applicability of general models in atypical cases. The developed model outperformed models such as Gaussian maximum likelihood, Holt exponential smoothing, and artificial neural networks in typical and atypical traffic flow prediction. In 2013, Jeong et al. [15] proposed an online learning weighted SVR model (OLWSVR) for short-term traffic flow prediction, which outperformed prediction models such as locally weighted regression, conventional SVR, and online learning SVR. Smith and Demetsky [16] analysed short-term traffic flow prediction models based on neural network and non-parametric regression. Cai et al. [17] proposed a neural network based on improved cuckoo algorithm with optimized radial basis function (CS-RBF) for highway traffic flow prediction under heavy rainfall, and the study showed that the algorithm has better prediction accuracy and convergence speed. In 2015, Lv et al. [18] proposed traffic flow feature learning using a stacked autoencoder model and trained it with greedy hierarchical unsupervised learning deep learning model. Zhang and Wang [19] built an urban trunk road travel time prediction model based on GRU network and simulated it using real road network data. In 2020, Wang et al. [20] proposed an LSTM travel time prediction model considering rainfall data, and the results showed that the prediction results with the inclusion of rainfall features were more accurate than when the rainfall features were not included. Meng proposed the LSTM-GRU combined model to predict the short-term traffic flow speed of highways in rainy days. This model is well adapted to the uncertainty and sudden change of traffic flow speed in rainy days [21].

Reviewing the above literature, we can find the following research trends regarding the influence of rainy weather environment on traffic flow characteristics and traffic flow prediction. (1) For the research on the influence of rainy weather environment on traffic flow characteristics, most domestic and foreign scholars divide the rainfall intensity into levels [22]. Using the traffic flow data and rainfall data of

the actual road section, the changes of the macro traffic flow parameter values of the road section under different levels of unfavorable weather are given. However, there is a lack of comprehensive consideration of the impact of multiple factors on traffic flow. (2) In terms of traffic flow prediction, various prediction models have different principles. Currently, machine learning and deep learning models have become the mainstream of research in the field of traffic flow prediction. Throughout the many previous studies on traffic flow prediction, there are fewer studies on traffic flow prediction under rainy environment, and more related studies only add the verification of rainy weather scenarios on the traditional prediction.

Therefore, in this paper, we consider the influence of various factors to carry out the research on traffic flow characteristics of highways under rainy environment. Also, we add rainfall features to the deep learning model to carry out the prediction of highway traffic flow speed under rainy environment. In view of the fact that the PSO algorithm can adjust the hyperparameters of the deep learning model and bring better prediction performance, this article will build the APSO-GRU model.

## 2. Data Preprocessing

*2.1. Preprocessing of Traffic Flow Data.* The data in this article come from the floating car data of Beijing-Harbin Highway (JingHa Highway), Beijing-Tianjin Highway (JingJin Highway), Beijing-Taipai Highway (JingTai Highway), and Beijing-Kaifeng Highway (JingKai Highway). By fusing multisource floating car data and combining the original data with relevant geographic information through the MapInfo interface, the space-time matching of traffic flow data is completed. After data preprocessing, the proportion of abnormal data accounts for 5% of the total original data.

The raw traffic flow data are recorded in 5-minute intervals, spanning a total of six months from June 1 to August 31, 2018, and June 1 to August 31, 2019. The raw data include information such as highway section ID, section direction, average vehicle speed, and traffic flow. The data format is shown in Table 1.

In the table, the first 13 digits of Section\_id indicate the number of a section of the highway, and the last digit indicates the direction of vehicle travel on the section, with 1 representing upward and 0 representing downward; Speed\_avg indicates the average speed of all vehicles passing the section during the collection time; volume indicates the traffic volume of the section during the collection time.

The data cleaning is divided into two parts: rejection of erroneous data and repair of missing data. For the rejection of erroneous data, a “rule rejection method” is used, which integrates the threshold method and the basic theory of traffic flow [23]. For the missing original data [24] of traffic flow, a simple nearest neighbor mean fill method is used to fill the data, which combines the mean filling method of replacing the missing data with the mean of the existing data and the nearest neighbor interpolation method using the observation values near the missing value to replace the missing value. The nearest neighbor interpolation method of

missing values is combined. Then, we take the average value of the valid data adjacent to the missing data as filling, as in the following equation:

$$H_t = \frac{(H_{t-1} + H_{t+1})}{2}, \quad (1)$$

where  $H_t$  represents the missing traffic flow data of the  $t^{\text{th}}$  cycle, including  $V$  and  $Q$ , and  $H_{t-1}, H_{t+1}$  are traffic flow data of the two adjacent cycles of the  $t^{\text{th}}$  cycle.

**2.2. Preprocessing of Rainfall Data.** The rainfall data were obtained from Beijing Nanjiao Observatory Station (No. 54511), Tongzhou Station (No. 54431), and Daxing Station (No. 54594). In this paper, only hourly rainfall data and their corresponding dates and times are extracted. A total of 4397 meteorological data from Daxing District and 4401 meteorological data from Tongzhou District were extracted, and the format of rainfall data is shown in Table 2. The time item indicates the end moment of the data collection time, and the rainfall amount is the accumulated rainfall amount within the data collection time.

Only a small amount of rainfall data was found to be missing through inspection. According to the method of traffic flow data filling, the average value of rainfall in the adjacent hours of the missing data was used to fill in the missing data.

**2.3. Spatial and Temporal Matching.** The recording period of traffic flow data is 5 minutes, while the recording period of rainfall data is 1 hour. It is necessary to match two data from time granularity. High-precision rainfall data are currently not available, and it is difficult to decompose long-period data into short-period data. In addition, the time accuracy of the floating car data acquisition system needs to be improved. Even if high-precision rainfall data are obtained, the time-space error of the two data matching is difficult to evaluate. So, it is more reasonable to combine the traffic flow data from 5-minute recording period to 1-hour recording period, and the combination rule is expressed as follows:

$$\begin{cases} \bar{C} = \sum_{i=1}^{12} C_i, \\ \bar{V} = \frac{\sum_{i=1}^{12} V_i C_i}{C}, \end{cases} \quad (2)$$

where  $\bar{C}$  represents traffic flow on the section in one hour (veh/h);  $C_i$  represents traffic flow on the section in 5 minutes (veh/5 min);  $\bar{V}$  represents the average speed of vehicles on the section in one hour (km/h); and  $V_i$  represents the average vehicle speed on the section in 5 minutes (km/h).

Based on the weighted average method, the average vehicle speed time series and traffic flow time series of each highway are calculated. The weight of the road section is the proportion of its length in the whole highway. The calculation method is shown in the following equation:

TABLE 1: Format of raw traffic flow data.

Section_id	Speed_avg (km/h)	Volume (veh/5 min)
59565500000081	86.52	267
59565500000090	85.57	238
59565500000101	86.34	314
59565500000221	85.61	284
59565500000231	85.82	329
59565600000010	85.66	283

TABLE 2: Rainfall data format.

Station ID	Date	Time	Rainfall
54594	20190615	22:00	0
54594	20190615	23:00	0.2
54594	20190615	24:00	1.0
54594	20190616	1:00	1.7

$$\begin{cases} V = \sum_{n=1}^K \frac{\bar{V}_n \cdot L_n}{L}, \\ C = \sum_{n=1}^K \frac{\bar{C}_n \cdot L_n}{L}, \end{cases} \quad (3)$$

where  $V$  represents the average speed of vehicles on the highway (km/h);  $\bar{V}_n$  represents the average vehicle speed of the road section (km/h);  $L_n$  represents the length of the road section (m);  $L$  represents the length of highway (m);  $K$  represents the total number of sections of the highway;  $C$  represents the overall traffic volume of the highway (veh/h); and  $C_n$  represents the traffic volume of the road section (veh/h).

The traffic flow time series and the rainfall time series in the region are integrated according to the corresponding time to complete the spatiotemporal matching, and the format of the matched data is shown in Table 3. In the table, Date\_hour indicates the date and time; precipitation indicates the rainfall amount in mm/h; Volume\_sum indicates the traffic flow in veh/h; and Speed\_avg indicates the average vehicle speed in km/h.

### 3. Analysis of the Influence of Rainy Weather on Traffic Flow Speed of Highway

**3.1. Analysis of the Factors Influencing the Traffic Flow Speed.** The traffic flow speed of highway is affected by many factors. Four potential factors, such as rainfall intensity, date category, time period, and number of lanes, are selected to explore whether these factors affect the traffic flow speed of highway by multivariate analysis of variance. According to the statistical analysis of the characteristics of highway traffic flow, it can be seen that the "morning peak hour" of highway is relatively lagging behind that of urban roads. Before the arrival of the peaking hour, it can be clearly seen that both traffic flow speed and traffic flow have experienced two processes of first decreasing and then increasing. Through observation, it is found that it is more reasonable to divide every four hours as a time period, as shown in Table 4.

TABLE 3: Data format after spatiotemporal matching.

Highway	Date_hour	Precipitation	Volume_sum	Speed_avg
Beijing-Harbin Highway	20190706_09	0	1106	89.61
Beijing-Harbin Highway	20190706_10	9	1051	85.34
Beijing-Harbin Highway	20190706_11	2.8	1137	87.74
Beijing-Harbin Highway	20190706_12	2.1	963	87.93
Beijing-Harbin Highway	20190706_13	1.5	897	88.57

TABLE 4: The division of different periods of the day.

Period number	Time span	Characteristic
1	0:00–4:00	During the low period in the morning, the speed and flow decrease slowly
2	4:00–8:00	During the rising period, the speed and flow increase rapidly
3	8:00–12:00	In the morning peak, the speed and flow are high
4	12:00–16:00	In the afternoon, the peak was flat, and the speed and flow decreased smoothly and slowly
5	16:00–20:00	In the evening peak, the flow reaches the peak again, and the speed decreases gradually
6	20:00–24:00	In the low period of night, the speed and flow decrease rapidly

SPSS software was used for multivariate analysis of variance, and the output of SPSS is shown in Figure 1. The results show that the four factors have significant influence on the traffic flow speed. In addition, the interaction of date category, time period, and number of lanes has a significant impact on traffic flow speed. The combination of other factors has no significant effect on traffic flow speed.

### 3.2. Influence of Rainfall on Traffic Flow Speed of Highway

**3.2.1. Distribution Characteristics of Traffic Flow Speed under Different Rainfall Intensities.** Considering the different levels of date categories and number of lanes, the distribution statistics of standard traffic flow speed under different rainfall intensities are carried out, which are divided into Tongzhou District and Daxing District, as shown in Figure 2.

It can be seen that the standard speed of highway vehicles in rainy days decreases with the increase of rainfall intensity. In terms of date category, weekend is more vulnerable to rainfall than working day. The slope of “rainfall intensity standard speed” of the four-lane highway in the two areas is greater than that of the three-lane highway, which indicates that the four-lane highway is more vulnerable to rainfall.

**3.2.2. Speed Distribution Characteristics of Traffic Flow in Different Speed Periods.** Considering the different levels of rainfall intensity, date category, and number of lanes, the distribution statistics of the standard speed of vehicles in different periods of each highway are carried out, as shown in Figure 3.

It can be seen from Figure 3 that the standard speed of four-lane highway is generally lower than that of three-lane highway when other factors are the same, which indicates that its traffic flow speed is more easily affected. The same rainfall intensity has different influence on the traffic flow speed in different periods of the day, and the morning peak and evening peak are more easily affected by rainfall. Similarly, the speed of traffic flow in the first period, the second period, and the sixth period is relatively less

susceptible to rainfall. With the increase of rainfall intensity, the above differences will be more obvious.

## 4. The Establishment of the Prediction Model of the Traffic Flow Speed of APSO-GRU

**4.1. Design of Traffic Flow Speed Prediction Model Based on GRU.** The proposed GRU model is composed of three sections, i.e., input layer, hidden layer, and output layer. The output layer is a fully connected dense layer. Adam algorithm is selected as the weight optimizer to optimize the internal weight of the model. The structure of the prediction model is shown in Figure 4 [25].

The input of the model is a time series matrix composed of traffic flow speed, traffic flow, and rainfall, the output is traffic flow speed, and the loss function is MAE. MSE is more affected by outliers, while MAE is more stable. After actual data validation, the parameters of the GRU prediction model are set as follows: the number of hidden layer nodes is 15, dropout parameter is 0.3, batch size is 200, epoch is 180, and learning rate is 0.004. And the training set and test set are divided in a ratio of 4:1.

**4.2. PSO Algorithm.** Part of the parameters of the GRU model is automatically adjusted by the model, and the other part of the parameters needs to be set artificially, which are called superparameters, including the number of hidden layers, the number of hidden layer nodes, the number of iterations, etc., and the rationality of superparameter setting directly affects the convergence speed of model calculation and the accuracy of prediction. Therefore, this section uses PSO algorithm to optimize the GRU model.

Particle swarm optimization algorithm is described in the D-dimensional search space, and N different particles form a search population. The current position of the  $i^{\text{th}}$  particle is  $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , current speed is  $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ , and the best location currently searched by the individual is  $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ ; it is called individual extremum [26]. The optimal position of the

Intersubjective effect test					
Dependent variable: traffic flow speed					
category	Type III sum of squares	Degree of freedom	Mean square	F	Significance
Revised model	34335.592 <sup>a</sup>	95	361.427	434.476	.000
Bias	4594866.533	1	4594866.535	5523544.621	.000
rainfall intensity	11347.833	3	3782.611	4547.122	.000
date category	28.831	1	28.831	34.658	.000
number of lanes	2532.155	1	2532.155	3043.934	.000
time period	13278.496	5	2655.699	3192.448	.000
rainfall intensity* date category	50.508	3	16.836	20.239	.000
rainfall intensity* number of lanes	6.983	3	2.328	2.798	.039
rainfall intensity* time period	31.436	15	2.096	2.519	.001
date category* number of lanes	22.878	1	22.878	27.502	.000
date category* time period	138.678	5	27.736	33.341	.000
number of lanes* time period	181.608	5	36.322	43.663	.000
rainfall intensity* date category* number of lanes	1.339	3	.446	.537	.657
rainfall intensity* date category* time period	1.971	15	.131	.158	1.000
rainfall intensity* number of lanes* time period	2.142	15	.143	.172	1.000
date category* number of lanes* time period	35.738	5	7.148	8.592	.000
rainfall intensity* date category* number of lanes* time period	2.787	15	.186	.223	.999
deviation	625.566	752			
total	6023624.627	848			
Revised total	34961.157	847			
a. R square = .982 (revisedR square = .980)					

FIGURE 1: SPSS output.

whole population is called global extremum, and it is  $g = (g_{i1}, g_{i2}, \dots, g_{iD})$ .

The current position of the particle corresponds to a candidate solution of the optimization problem, and the flight process is the search process of the individual. Each particle iterates continuously to update its speed and position, which are determined by equations (4) and (5), respectively:

$$v_i(t+1) = v_i(t) + c_p \cdot r_1 \cdot (p_i(t) - x_i(t)) + c_g \cdot r_2 \cdot (g(t) - x_i(t)), \quad (4)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (5)$$

where  $v_i(t)$  represents the velocity of the  $i^{\text{th}}$  particle at time  $t$ ;  $x_i(t)$  represents the position of the  $i^{\text{th}}$  particle at time  $t$ ;  $c_p, c_g$  represent the acceleration coefficients, where  $c_p$  is the cognitive learning factor and  $c_g$  is the social learning factor, respectively, representing the self-learning ability of particles

and the ability to learn from the optimal individual of the group,  $c_p, c_g > 0$ ;  $r_1, r_2$  represent random numbers with (0, 1) interval uniform distribution;  $p_i(t)$  represents the historical optimal position of the  $i^{\text{th}}$  particle at time  $t$ ; and  $g(t)$  represents global optimal position of particle swarm optimization at time  $t$ .

In order to further optimize the performance of the PSO algorithm, Shi introduced a new parameter inertia weight [27] into the particle velocity update formula of the original PSO algorithm, and equation (4) becomes

$$v_i(t+1) = w \cdot v_i(t) + c_p \cdot r_1 \cdot (p_i(t) - x_i(t)) + c_g \cdot r_2 \cdot (g(t) - x_i(t)). \quad (6)$$

Inertia weight determines the influence of particle velocity at the previous time on the current velocity, which can effectively balance the role of global search and local search. Equation (5) consists of three parts. The first part is inertial motion, which indicates the degree to which the particle

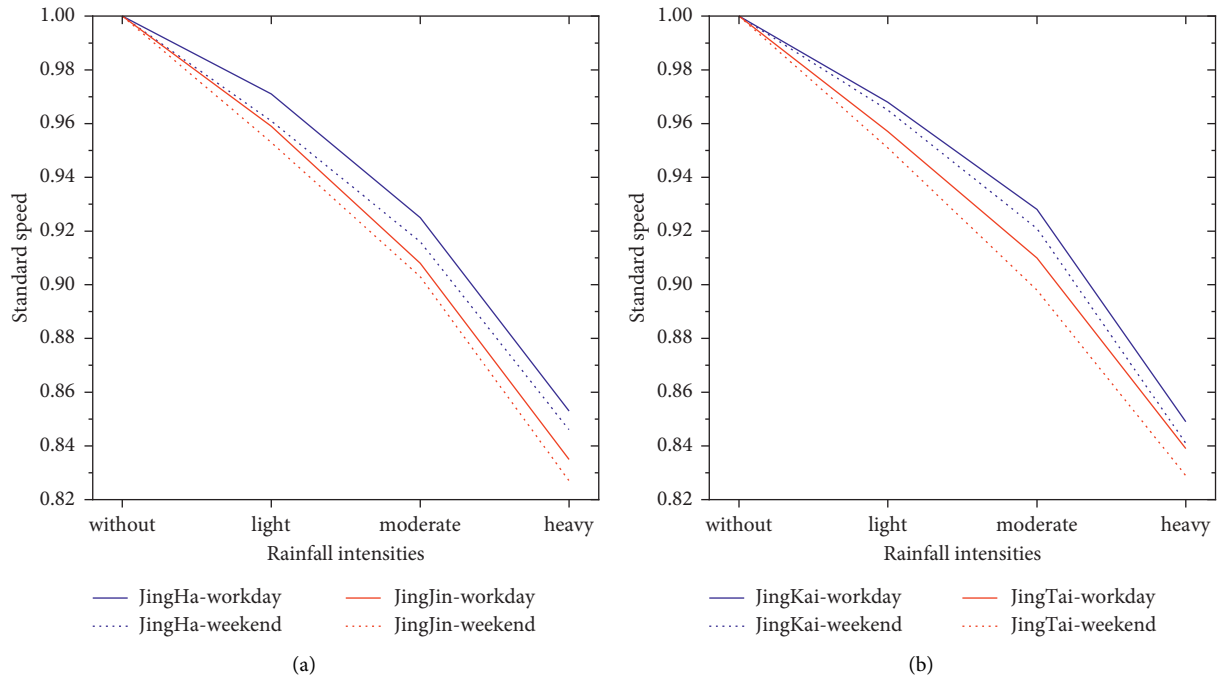


FIGURE 2: Standard speed distribution of highway under different rainfall intensities: (a) highway in Tongzhou District; (b) highway in Daxing District.

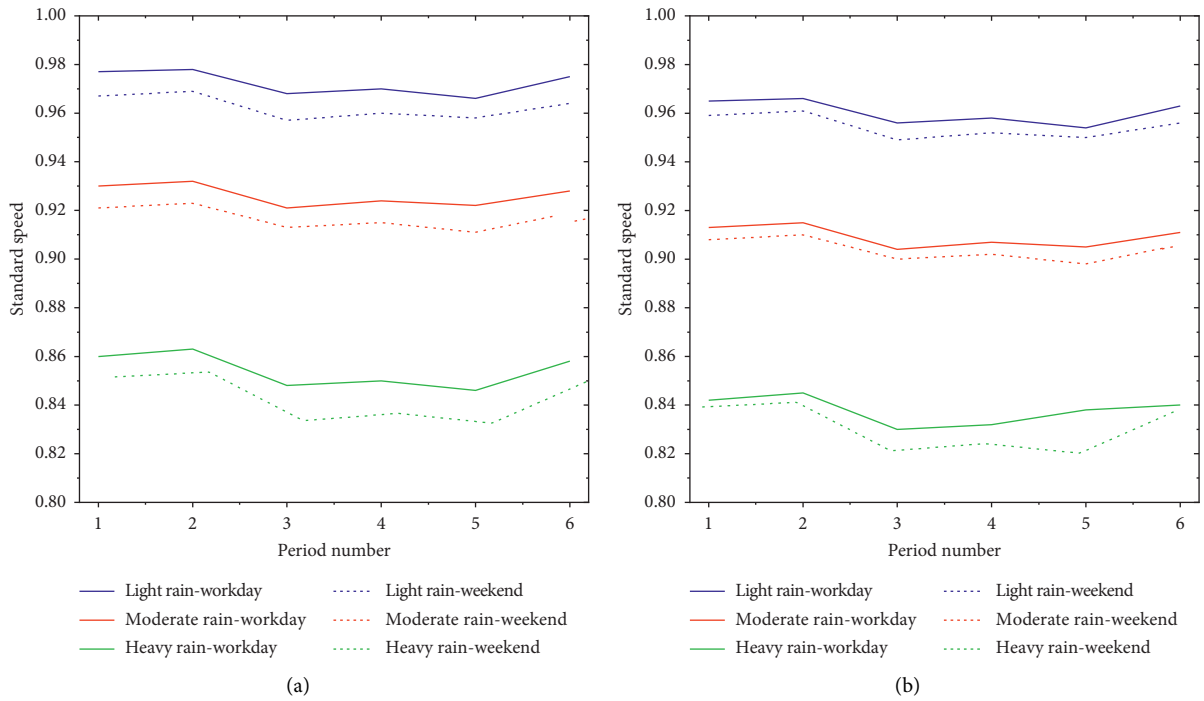


FIGURE 3: Continued.

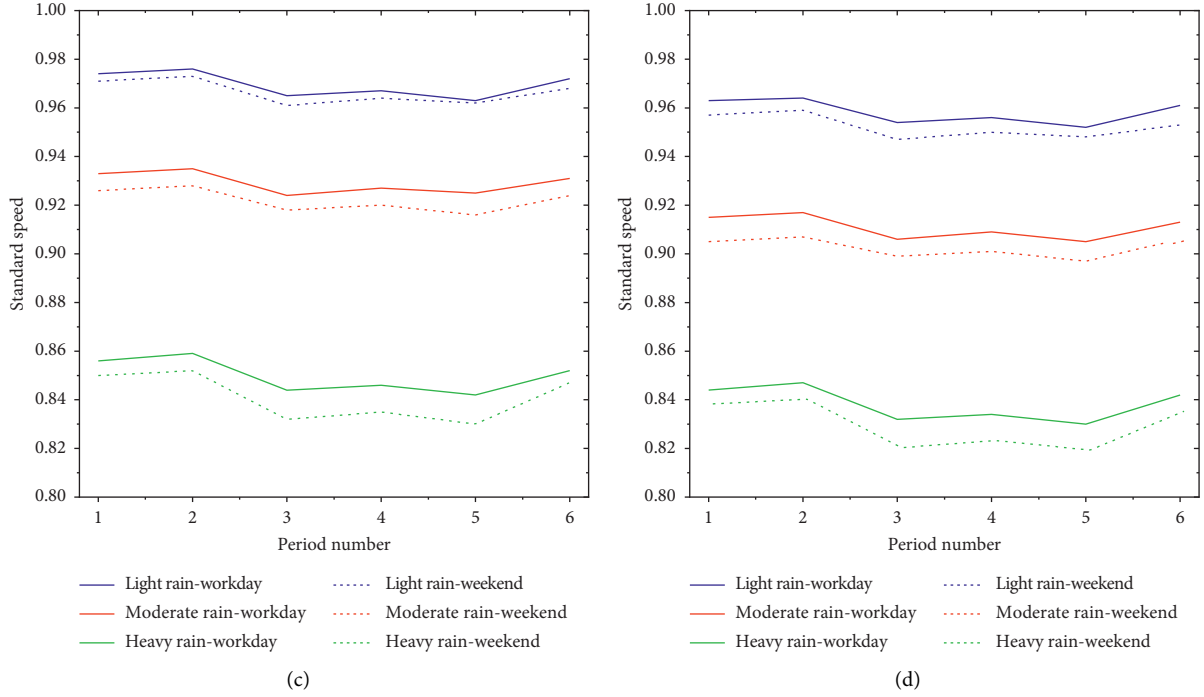


FIGURE 3: Time distribution characteristics of standard speed of highway in rainy environment: (a) Beijing-Harbin Highway; (b) Beijing-Tianjin Highway; (c) Beijing-Kaifeng Highway; (d) Beijing-Taipei Highway.

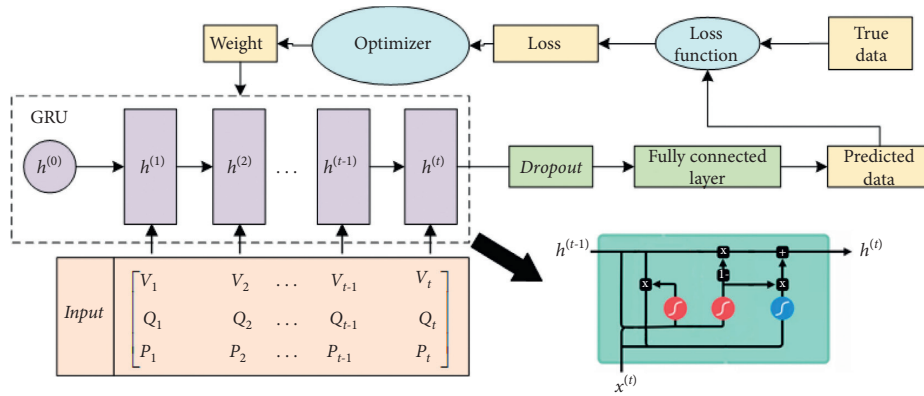


FIGURE 4: Structure of the GRU prediction model.

maintains its velocity at the previous moment; the second is cognitive learning, which means that the particles memorize their historical optimal position and make them close to the historical optimal position. Finally, social learning, which means the information exchange between particles, makes particles close to the historical optimal position of the population [28]. It can be seen from equation (4) that particles have memory, and they move towards the direction of the optimal particle combined with their own and group experience. Equations (5) and (6) constitute a new PSO algorithm called standard PSO algorithm (SPSO).

4.3. Adaptive Nonlinear Inertia Weight PSO. In order to further improve the problem that PSO algorithm is easy to

fall into local optimal solution and reasonably balance the ability of local search and global search of PSO algorithm, an adaptive nonlinear inertia weight method is used to adjust, as shown in the following equation:

$$w = \begin{cases} w_{\min} + \frac{(w_{\max} - w_{\min}) * (f_i - f_{\min})}{f_{\arg} - f_{\min}}, & f_i \leq f_{\arg}, \\ w_{\max}, & f_i > f_{\arg}, \end{cases} \quad (7)$$

where  $w_{\max}$ ,  $w_{\min}$  are the maximum and minimum values of inertia weight;  $f_i$  represents the adaptation value of particle  $i$ ;  $f_{\min}$  represents the minimum fitness of all current

particles;  $f_{\max}$  represents the maximum fitness of all current particles; and  $f_{\arg}$  represents the average fitness value of all current particles.

**4.4. Establishment of APSO-GRU Prediction Model.** Based on the adaptive nonlinear inertia weight PSO algorithm proposed in the previous section, an APSO-GRU prediction model is proposed, and the parameters of GRU network are optimized by PSO algorithm. It is finally determined that the number of hidden layers of the model is 2, the number of nodes in the first layer is 20, and the number of nodes in the second layer is 15 through cross combination test. And the relevant parameters of the APSO are set as follows: the population number is 20, the number of iterations is 30, the acceleration coefficients are both 2, and the maximum and minimum inertia weights are 0.9 and 0.1, respectively,  $[h_{1\min}, h_{1\max}]$  and  $[h_{2\min}, h_{2\max}]$  both are  $[1, 50]$ ,  $[\eta_{\min}, \eta_{\max}]$  is  $[0.001, 0.01]$ , and  $[n_{\min}, n_{\max}]$  is  $[1, 500]$ . The model flow is shown in Figure 5.

## 5. Instance Verification

**5.1. Experimental Verification Scenario and Evaluation Indexes.** In the field of traffic flow prediction, the most common loss functions include mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) [29]. Their calculations are shown in equations (8)–(10). RMSE and MAPE are used as error functions to evaluate the performance of the prediction model.

$$\text{MAE} = \frac{1}{L} \sum_{t=1}^L |Y(t) - Y_p(t)|, \quad (8)$$

$$\text{RMSE} = \sqrt{\frac{1}{L} \sum_{t=1}^L (Y_p(t) - Y(t))^2}, \quad (9)$$

$$\text{MAPE} = \frac{1}{L} \sum_{t=1}^L \frac{|Y_p(t) - Y(t)|}{Y(t)} \times 100\%, \quad (10)$$

where  $L$  represents the total length of time series;  $Y_p(t)$  represents prediction value at time  $t$ ; and  $Y(t)$  represents true value at time  $t$ .

Based on the established model of the traffic flow speed prediction of GRU and APSO-GRU, the traffic flow speed of Beijing-Harbin Highway, Beijing-Tianjin Highway, Beijing-Taiwan Highway, and Beijing-Kaifeng Highway in Beijing is predicted, respectively. At the same time, the support vector regression (SVR) model is compared with the APSO-LSTM model on prediction performance.

In order to more comprehensively and accurately evaluate the performance of the model under different rainfall scenarios, the rainy environment is divided into two categories: noncontinuous rainfall and continuous rainfall. Noncontinuous rainfall is the situation that the rainfall process is relatively short and sparse in a specific period of

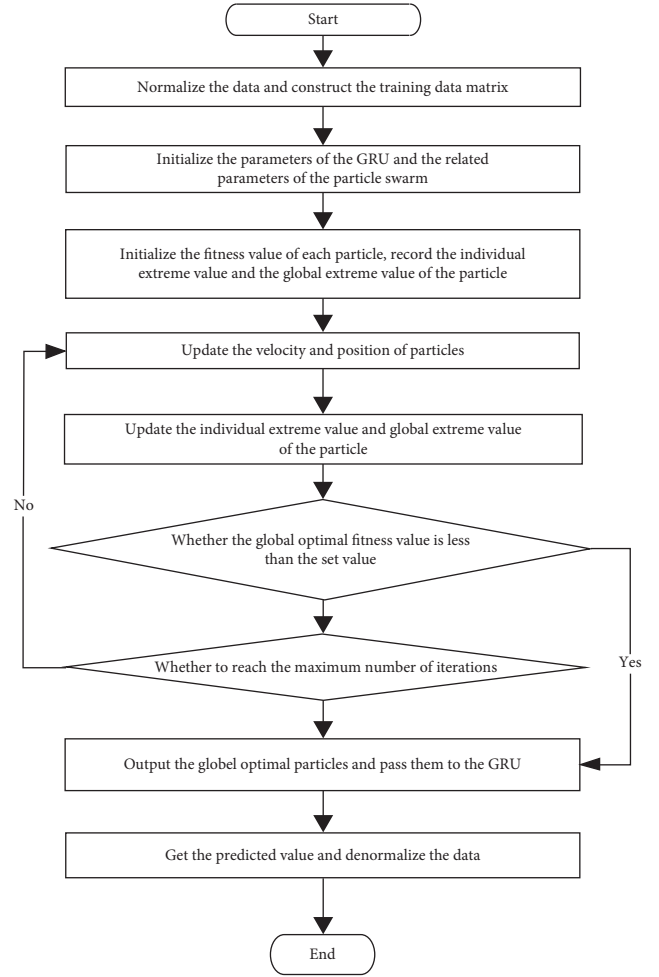


FIGURE 5: Process of the APSO-GRU prediction model.

time, while continuous rainfall is the situation that the rainfall process is relatively long and dense in a specific period of time.

### 5.2. Analysis of Prediction Results of APSO-GRU

**5.2.1. Prediction Results of Noncontinuous Rainfall.** The time span of noncontinuous rainfall in Tongzhou District is the interval between 0:00 on 9th August (working day) and 24:00 on 10th August (weekend) in 2019, and the time span of noncontinuous rainfall in Daxing District is from 0:00 on 11th August (weekend) to 24:00 on 12th August (working day) in 2019.

Due to article content limitation, this paper only selects to visualize prediction results of Beijing-Harbin Highway, as shown in Figure 6. The horizontal axis represents the time, and the left vertical axis represents the speed, corresponding to the broken line chart. The right vertical axis represents precipitation, corresponding to the histogram. This setting is used in the following.

Under the noncontinuous rainfall scenario, the traffic flow speed of highway is obviously disturbed during the rainfall (moderate rain and heavy rain), and the change



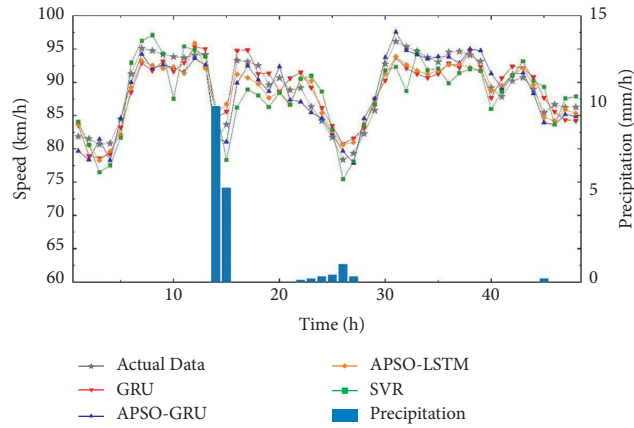


FIGURE 6: Prediction results of traffic flow speed during unsustained rainfall on Beijing-Harbin Highway.

TABLE 5: Errors in traffic flow speed prediction during unsustained rainfall.

Model	Beijing-Harbin Highway		Beijing-Tianjin Highway		Beijing-Kaifeng Highway		Beijing-Taiwan Highway	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
GRU	3.78	2.91	4.15	3.22	4.04	3.07	4.20	3.13
APSO-GRU	2.84	2.15	3.27	2.76	2.53	1.80	2.71	2.09
APSO-LSTM	3.17	2.68	3.77	2.93	4.23	3.26	3.13	2.75
SVR	6.02	4.63	5.89	4.39	6.21	4.78	4.87	3.89

trend of the predicted value of each model is basically consistent with the real value. The prediction error is shown in Table 5.

5.2.2. *Prediction Results of Continuous Rainfall.* The time span of continuous rainfall in Tongzhou District and Daxing District is from 0:00 on 28th July 2019 (weekend) to 24:00 on 29th July 2019 (working day).

The prediction results of traffic flow speed of Beijing-Harbin Highway during continuous rainfall are shown in Figure 7.

It can be seen from the figure that under the continuous rainfall scenario, the traffic flow speed of highway is greatly affected, and the operation state continuously fluctuates. In this unstable situation, the trend of the predicted values of each model is basically consistent with the real values, but the prediction accuracy is significantly lower than that of noncontinuous rainfall. The prediction error is shown in Table 6.

5.2.3. *Comparative Evaluation and Analysis of Prediction Results.* The statistics of the average prediction error of the traffic flow speed of each model under different rainfall scenarios on each highway can be seen in Table 7.

In the aspect of traffic flow speed prediction, the prediction accuracy of the APSO-GRU model is better than that of the GRU model and APSO-LSTM model under the two rainfall scenarios, and the accuracy of the SVR model is the lowest, which verifies the performance improvement of the

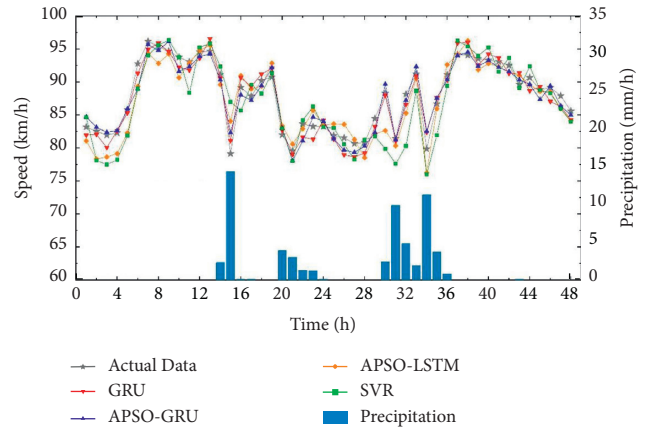


FIGURE 7: Prediction results of traffic flow speed during sustained rainfall on Beijing-Harbin Highway.

built deep learning model. The average prediction accuracy of the GRU model, APSO-GRU model, APSO-LSTM model, and SVR model is 95.96%, 97.16%, 96.42%, and 94.25%, respectively. The average prediction accuracy of the APSO-GRU model is 1.20% higher than that of the GRU model and 0.74% higher than that of the APSO-LSTM model.

The average prediction accuracy of the APSO-GRU model is 96.75% under the continuous rainfall scenario, which is 2.38% and 2.22% higher than that of the GRU model and APSO-LSTM model, respectively. The prediction accuracy of each model has declined, but the decline of the APSO-GRU model is not obvious, followed by the GRU

TABLE 6: Errors in traffic flow speed prediction during sustained rainfall.

Model	Beijing-Harbin Highway		Beijing-Tianjin Highway		Beijing-Kaifeng Highway		Beijing-Taiwan Highway	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
GRU	5.24	3.77	6.19	4.40	5.77	4.12	5.31	3.86
APSO-GRU	3.52	2.86	3.47	2.83	2.88	2.17	3.14	2.31
APSO-LSTM	5.63	4.56	4.69	4.16	6.30	5.10	5.24	4.43
SVR	10.37	7.29	11.40	7.71	12.63	8.02	10.71	7.55

TABLE 7: Error comparison of traffic flow speed prediction results.

Model	Noncontinuous		Continuous	
	MAPE (%)	RMSE	MAPE (%)	RMSE
GRU	4.04	3.08	5.63	4.04
APSO-GRU	2.84	2.20	3.25	2.54
APSO-LSTM	3.58	2.91	5.47	4.58
SVR	5.75	4.42	11.28	7.64

model. The prediction accuracy of the APSO-LSTM model and SVR model is lower than that of the former two models.

## 6. Conclusions

The main conclusions obtained in this paper are as follows:

- (1) Based on the results of the multivariate analysis of variance, rainfall intensity, date category, time of day, and number of lanes have significant effects on traffic flow speed. The higher the intensity of rainfall is, the more the traffic flow is affected. Traffic flow is more likely to be affected by rainfall on weekends than weekdays, and it is more likely to be affected by rainfall during daytime (especially AM peak and PM peak) than at night.
- (2) An APSO-GRU traffic flow speed prediction model was built for the rainy environment. Under the noncontinuous rainfall scenario, the average prediction accuracy of the APSO-GRU model reaches 97.33%, which is 1.19% and 0.71% higher than that of the GRU model and the APSO-LSTM model, respectively. Under the continuous rainfall scenario, the average prediction accuracy of the APSO-GRU model reaches 96.74%, which is 2.69% and 2.39% higher than that of the GRU model and the APSO-LSTM model, respectively. The results show that the prediction accuracy and stability of the APSO-GRU model are significantly improved compared with the APSO-LSTM model under different rainfall scenarios.
- (3) Comparison of the traffic flow speed prediction results between the machine learning model SVR and the APSO-LSTM model in deep learning shows that the prediction accuracy of APSO-LSTM is higher than that of the SVR model by 2.18% and 5.55% under noncontinuous rainfall and continuous rainfall scenarios, respectively. It indicates that the prediction accuracy and stability of the model based on LSTM are better than those of the SVR model,

which fully proves that the prediction performance of the deep learning model is better than the traditional SVR model.

## Data Availability

The data used to support the findings of this study have not been made available because of data ownership issues.

## Conflicts of Interest

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

## Acknowledgments

This research was funded by the National Natural Science Foundation of China (71871011) and the Science and Technology Project of Hebei Provincial Department of Transportation, China (TH-201921).

## References

- [1] K. M. Saberi and R. Bertini, "Empirical analysis of the effects of rain on measured freeway traffic parameters," in *Proceedings of the 89th Annual Meeting of the Transportation Research Board*, Washington DC, USA, January 2010.
- [2] D. Akin, V. P. Sisiopiku, and A. Skabardonis, "Impacts of weather on traffic flow characteristics of urban freeways in Istanbul," *Procedia—Social and Behavioral Sciences*, vol. 16, pp. 89–99, 2011.
- [3] Y. J. Stephanedes, P. G. Michalopoulos, and R. A. Plum, "Improved estimation of traffic flow for real-time control (discussion and closure)," in *Proceedings of the 60th Annual Meeting of the Transportation Research Board*, pp. 28–39, Washington, DC, USA, January 1981.
- [4] L. Peng, Z. Li, C. Wang, and T. Sarkodie-Gyan, "Evaluation of roadway spatial-temporal travel speed estimation using mapped low-frequency AVL probe data," *Measurement*, vol. 165, Article ID 108150, 2020.
- [5] Y. Tian, "Research on short-term traffic status prediction of freeway network," Shanghai Jiaotong University, Shanghai, China, 2016.

- [6] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2340–2350, 2017.
- [7] J. Tang and Y. Zeng, "Spatiotemporal gated graph attention network for urban traffic flow prediction based on license plate recognition data," *Computer-Aided Civil and Infrastructure Engineering*, pp. 1–21, 2021.
- [8] X. Meng, H. Fu, L. Peng et al., "D-LSTM: short-term road traffic speed prediction model based on GPS positioning data," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.
- [9] A. T. Ibrahim and F. L. Hall, "Effect of adverse weather conditions on speed-flow-occupancy relationships," *Transportation Research Record Journal of the Transportation Research Board*, vol. 1457, pp. 184–191, 1994.
- [10] M. Agarwal, T. H. Maze, and R. R. Souleyrette, "Impacts of weather on urban freeway traffic flow characteristics and facility capacity," in *Proceedings of the 2005 Mid-Continent Transportation Research Symposium*, pp. 1121–1134, Ames, IA, USA, August 2005.
- [11] C. Li, "Research on traffic flow characteristics and guidance control of highway in bad weather," Beijing University of Technology, Beijing, China, 2015.
- [12] F. Xu, "Calculation and analysis of Urban road capacity under rainfall condition," Chang'an University, Xi'an, China, 2017.
- [13] C. Dong, C. Shao, Z. Xiong et al., "Short-term traffic flow forecasting of road network based on elman neural network," *Journal of Transportation Systems Engineering and Information Technology*, no. 1, pp. 149–155, 2010.
- [14] M. Castro-Neto, Y. Jeong, M. Jeong, and L. 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.
- [15] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1700–1707, 2013.
- [16] B. L. Smith and M. Demetsky, "Short-term traffic flow prediction models—a comparison of neural network and nonparametric regression approaches," in *Proceedings of the Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA, August 2002.
- [17] Y. Cai, B. Yue, E. Cai et al., "Short-term traffic flow forecast of highway under heavy rain," *Computer Engineering*, vol. 46, no. 6, pp. 34–39, 2020.
- [18] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
- [19] M. Zhang and X. Wang, "Traffic time prediction of Urban main road based on GRU-RNN model," *Journal of Beijing Information Science & Technology University*, vol. 34, no. 4, pp. 30–35, 2019.
- [20] Z. Wang, D. Li, and X. Cui, "Travel time prediction based on LSTM neural network in precipitation," *Journal of Transportation Systems Engineering and Information Technology*, vol. 20, no. 1, pp. 137–144, 2020.
- [21] L. Meng, "Research on traffic safety assessment and early warning management of highway on rainy days based on speed prediction," Beijing Jiaotong University, Beijing, China, 2019.
- [22] E. Chung, "Does weather affect highway capacity," in *Proceedings of the 5th International Symposium on Highway Capacity & Quality of Service*, pp. 139–146, Yokohama, Japan, July 2006.
- [23] Z. Wang, "Highway traffic flow characteristic analysis and prediction in rainy environment," Beijing Jiaotong University, Beijing, China, 2020.
- [24] L. Zheng, H. Huang, C. Zhu, and K. Zhang, "A tensor-based K-nearest neighbors method for traffic speed prediction under data missing," *Transportation Business: Transport Dynamics*, vol. 8, no. 1, pp. 182–199, 2020.
- [25] J. Zhao, Y. Gao, Y. Guo, and Z. Bai, "Travel time prediction of expressway based on multi-dimensional data and PSO-ARMAX model," *Advances in Mechanical Engineering*, vol. 10, no. 2, pp. 1–16, 2018.
- [26] Y. Gao, J. Zhao, Z. Qin, Y. Feng, Z. Yang, and B. Jia, "Traffic speed forecast in adjacent region between highway and urban expressway: based on MFD and GRU model," *Journal of Advanced Transportation*, vol. 2020, Article ID 8897325, 18 pages, 2020.
- [27] Y. Shi, "A modified particle swarm optimizer," in *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, pp. 67–73, IEEE, Anchorage, AK, USA, May 1998.
- [28] L. Kang, "Several improvements to particle swarm optimization and their theoretical fundamentals," Wuhan University, Wuhan, China, 2017.
- [29] G. Qi, A. Ceder, A. Huang, and W. Guan, "A methodology to attain public transit origin-destination mobility patterns using multi-layered mesoscopic analysis," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–19, 2020.

## Research Article

# Correlation Analysis of External Environment Risk Factors for High-Speed Railway Derailment Based on Unstructured Data

Haixing Wang , Yuanlanduo Tian , and Hong Yin 

*School of Traffic & Transportation, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Haixing Wang; [hwxwang@bjtu.edu.cn](mailto:hwxwang@bjtu.edu.cn)

Received 26 May 2021; Accepted 10 July 2021; Published 20 July 2021

Academic Editor: Ling Huang

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

In railway operation, unsafe events such as faults may occur, and a large number of unsafe event records are generated in the process of unsafe events' recording and reporting. Unsafe events have been described in unstructured natural language, which often has inconsistent structure and complex sources, involving multiple railway specialties, with multisource, heterogeneous, and unstructured characteristics. In practical application, the efficiency of processing is extremely low, leading to potentially unsafe management utilization. Based on the data on unsafe events, this paper utilizes big data processing technology, conducts association rules mining and association degree analysis, extracts the word segmentation, and obtains the feature vector of unsafe fault event data. At the same time, the unsafe event data analysis model is constructed in combination with regular expression and pattern matching technology. This paper establishes the matching model of high-speed railway derailment-based external environment risk factors and applies it to the occurrence of unsafe events. This model could be utilized to analyze and excavate the link between external environment risk factors and the occurrence of unsafe events and carry out the automatic extraction of characteristic information such as risk possibility and consequence severity; hence, it has potential for identifying, with enhanced accuracy, high-risk factors that may lead to high-speed railway derailment. Based on this study, we could make full use of the unsafe event data, forecast the risk trend, and discover the law of high-speed railway derailment. This study introduces a viable approach to analyzing the unsafe event data, forecasting risk trend, and conceptualizing high-speed railway derailment. It could also enforce the accurate quantification of high-speed railway safety situation, refine the risk index and conduct in-depth analysis combined with the model, and effectively support the digitalization and intellectualization of high-speed railway operation safety.

## 1. Introduction

The occurrence of high-speed railway derailment accident may result in severe financial and human losses, which have significant disaster characteristics and strong nonlinear characteristics and pose a challenge to high-speed railway safety management. Technological advances have helped to mitigate the internal factors behind railway derailment, but the external factors remain an underexplored area of research.

Advances in computer technology have benefited large-scale numerical calculation by enhancing operation speed, storage capacity, and operation scale. Big data analysis, furthermore, has been building on progress in accuracy, quality, and reliability and has become a major area of academic interest in the context of high-speed railway safety.

To minimize risks that may lead to unsafe events, the railway corporation has accordingly built the reporting system to keep the records, and it has generated a large amount of data thereof. Figure 1 exemplifies a typical record of an unsafe event.

However, the records on unsafe events, which are now described in natural language as shown above, could be incorporated into a digital database, and it could not be established without a consistent description standard. At the same time, the railway infrastructure, rolling stock, and other equipment are diverse and complicated, and the data sources are complex, involving many specialties, such as track maintenance, power supply equipment maintenance, signal and communication equipment maintenance, EMU maintenance, passenger transportation, and external environment. The database covers a wide range of faults' fixed

10:17 on April 18, 2021, at 191 km 188 m on XX line of XX railway corporation the mountain watcher inspected and found that the mountain on the right side of 191 km 201 m was cracked, and about 3 cubic meters rockfall intruded into the clearance (the largest one was 1.1 \* 0.9 \* 1.2m). The mountain watcher immediately informed the relevant personnel to block the line. At 10:21, the line was blocked. After being handled by the track maintenance staff, the line was opened at 11:17. The incident affected the train traffic and caused the train no. XXX to change its parking place. The rockfall at 191 km 201 m, and this site is the straight line section of the line, half dike and half cutting section, and the right side is 4 meters away from the track center of the line.

FIGURE 1: Example of unstructured unsafe event data of railway.

equipment faults, mobile equipment faults, perimeter intrusion events, and others. A sizable amount of data have been gathered. In practice, the information needs to be retrieved, read, and updated manually, which costs a huge amount human and material resources, and the processing efficiency is considerably low, resulting in the low utilization rate of railway unsafe event data.

For the requirements of railway safety management and risk analysis for real time and accuracy to be met, it is urgent to carry out railway risk effect factors association analysis based on unstructured data, while taking into account different data structures, different sources, and scattered and independent records of railway unsafe events. Through this research, the heterogeneous data sources could be integrated, the heterogeneity of data can be eliminated, and the accurate association between data and railway risk could be achieved. Therefore, this paper will take the external environment risk of high-speed railway derailment as the research focus and carry out a correlation analysis of external environment risk factors of high-speed railway derailment based on unstructured data. The study seeks to identify the risk factors of high-speed railway derailment and select appropriate models to process unstructured data, including data collection, data cleaning, data dictionary construction, data extraction, data storage, and other steps. Accordingly, the association between data and risk occurrence possibility and consequence severity will be realized. The paper then moves on analyzing the high-speed railway derailment risk factors and effective extraction of unstructured data.

In recent years, with the breakthrough of big data and artificial intelligence technology, a number of investigations have been carried out regarding the railway safety index, unstructured data analysis, and multisource heterogeneous railway safety data identification and extraction. In terms of railway safety index, since 2015, the International Union of Railways (UIC) has started to build the global safety index (GSI) [1]. Based on safety data and accident information, it evaluates the safety level of railway in Europe, some Asian, and Middle Eastern countries and regions and analyzes the statistical data of safety accidents, the impact of accidents, and the safety level and development trend. Zhao et al. [2] established a railway accident index by measuring the occurrence frequency and consequence severity of railway accidents, which is used to evaluate the overall situation of China's railway safety. In the aspect of unstructured data analysis and mining, Zhang [3] built an unstructured data analysis platform based on report documents with Chinese

word segmentation technology, unstructured data extraction method, pattern matching, and other methods. Zhu et al. [4] put forward a new HGD tree index technology and a new partition method, in order to use probability density function to partition data and improve the speed of data access, and gave a solution based on the optimization operation method. Wang et al. [5, 6] analyzed and studied the safety data of dangerous goods transportation based on the data mining method. In the aspect of multisource heterogeneous railway safety data extraction and data analysis, Wang et al. [7] conducted quantitative analysis on railway derailment and the change of accident rate based on American railway safety data. Lin. et al. [8] analyzed the data of American trunk line passenger trains and quantitatively analyzed the causes of passenger train accidents. Liu et al. [9] analyzed the causes of major train derailment and their effect on accident rates. Turla et al. [10] analyzed the freight train collision risk in the United States. Li [11] recognized and extracted the fault features of high-speed railway equipment by establishing the +bilstm and +CRF method for character representation and the +transformer method for word segmentation representation. Zhou and Li [12] established a method of fault data feature recognition and extraction for railway signal equipment based on MCNN.

To sum up, previous studies on high-speed railway risks mostly employ the expert evaluation method, which is arguably based on subjective deliberation and may undermine the research validity. Besides, past investigations mainly focus on feature recognition, extraction for specific structure of safety data, and the processing of unstructured data. As such, there still lacks research on the correlation analysis between safety data and risk. This study proposes a data-driven risk judgment method for analysis to facilitate accurate association between data and railway risk, with implications that the proposed model can contribute to improving the feasibility and accuracy of risk judgment.

## 2. Analysis of External Environment Risk Associated Factors

To explore the derailment mechanism of high-speed railway, this paper establishes a dynamic derailment-related element model of high-speed railway. As shown in Figure 2, the derailment of high-speed railway is mainly related to EMU subsystem and line subsystem, and the coupling relationship between wheel and rail has an important impact on the derailment of high-speed railway. In addition, the external

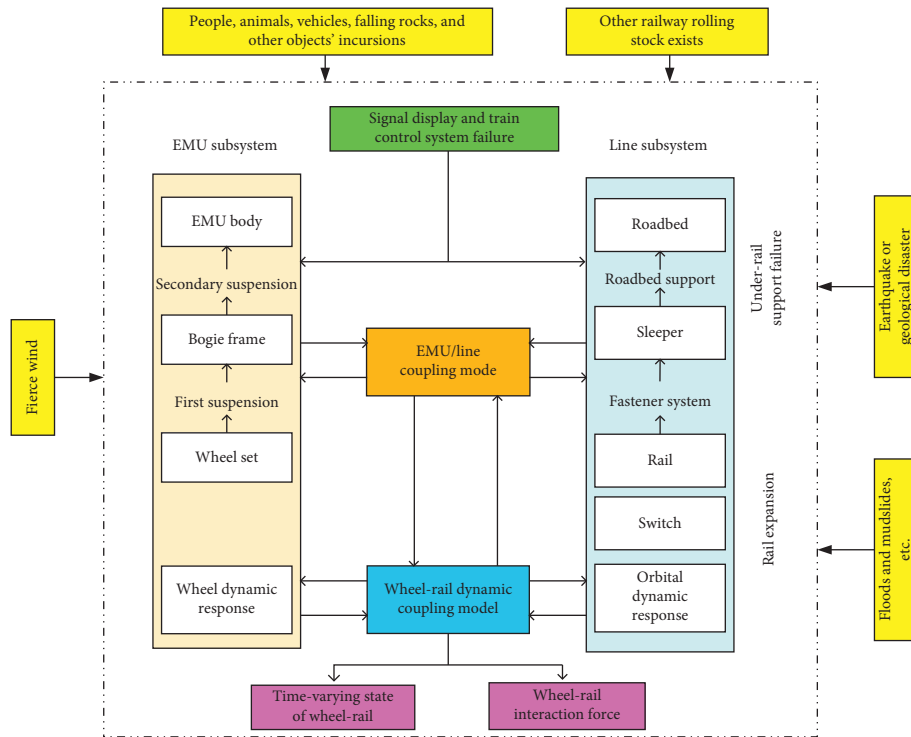


FIGURE 2: Dynamic derailment related element model of high-speed railway.

environmental factors such as natural geology and perimeter intrusion factors also play an important role in the derailment of high-speed railway. The external environmental safety for high-speed railway derailment has been widely investigated and is the focus of the current study [13].

Based on the above analysis, the paper puts forward the framework model of risk associated external environmental factors for high-speed railway derailment, which fall into two categories: natural and geological factors (natural hazard factors and geological hazard factors) and perimeter intrusion factors (perimeter intrusion of animals, perimeter intrusion of objects and plants, and perimeter intrusion of people), as shown in Figure 3, and the classification is detailed in Table 1 [14–17].

### 3. Unstructured Data Mining Method

The raw data stem from a database of a railway company that has kept its records of, as many as, 15,000 past unsafe events is obtained.

**3.1. General Analysis.** At the same time, based on the scientific analysis for the unsafe event data, the study combines regular expression and pattern matching technology and establishes the matching model of external environmental factors for high-speed railway derailment risk associated unsafe events. This paper analyzes and mines the relationship between the external environmental factors of high-speed railway derailment and the unsafe events, automatically, quickly, and accurately extracts the key characteristic information such as the possibility of risk occurrence and the

severity of the consequences, so as to transform unstructured data into structured information. The main data analysis and mining process is shown in Figure 4. The process includes (1) unstructured railway safety data, (2) split and match keywords, and (3) association rules' mining and association degree analysis. Through the process, the accurate association between data and railway risk could be acquired [18–20].

**3.2. Keyword Extraction and Matching.** If you want to extract keywords in the text, it is relatively simple for English and other languages. Keywords' extraction can be achieved in a number of languages, including English. In the case of English, for example, there are spaces between words as segmentation. In Chinese, however, such expressions are unavailable, so it is necessary to break coherent sentences into keywords. Expressions in Chinese may vary widely, leading to potential ambiguity in the word segmentation. In keyword extraction and matching, the railway safety dictionary is designed, and the algorithms and models such as tire tree, DAG, Viterbi, HMM (hidden Markov model), and keyword matching are comprehensively used. The main processing of keyword extraction and matching is shown in Figure 5 [21–23].

**3.3. Association Analysis-Based Apriori Algorithm.** This study utilizes the association rules to explore the correlations between data generated by different mode methods, so as to build rules that may inform the decision-making.

The data mining of association rules mainly includes two processes. First, identify the frequent item sets whose

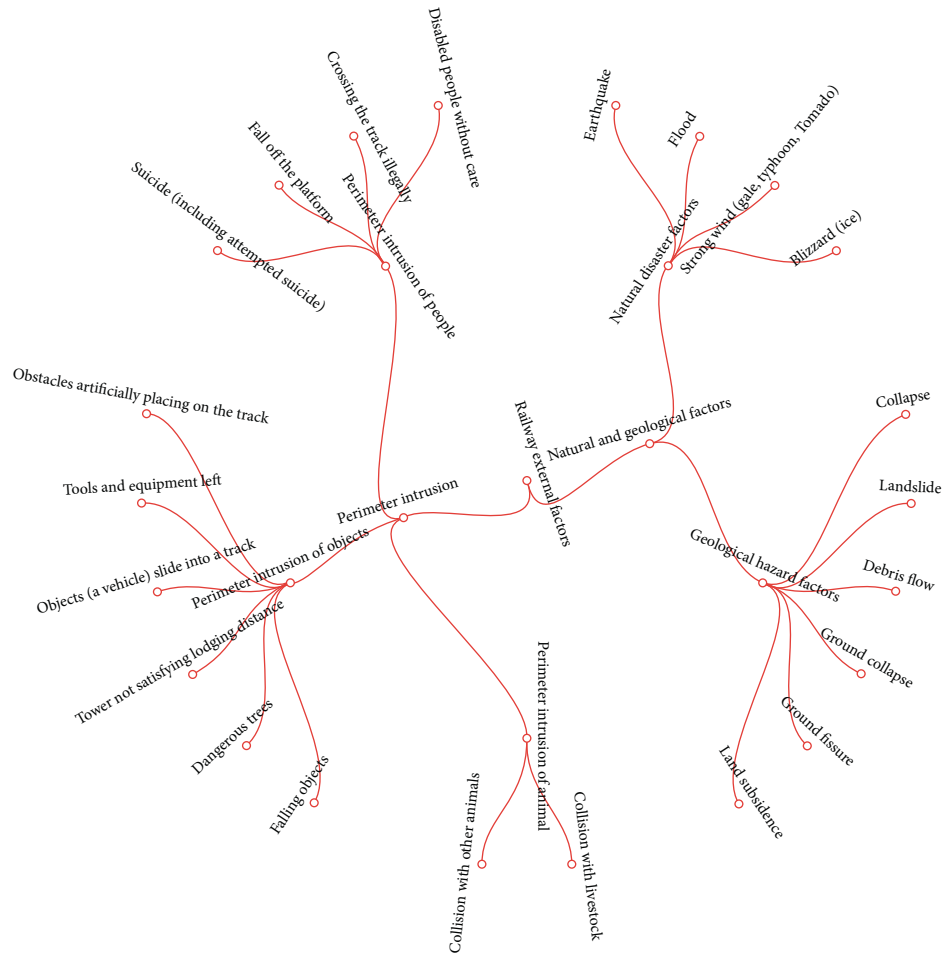


FIGURE 3: Framework model of external environmental factors associated with high-speed railway derailment.

TABLE 1: Main external environmental factors associated with derailment of high-speed railway.

No.	Serial number	Class I factors	Class II factors	Name of related factors		
1	Z11	Natural and geological factors	Natural hazard factors	Earthquake		
2	Z12			Flood		
3	Z13			Strong wind (gale, typhoon, and tornado)		
4	Z14			Blizzard (ice)		
5	Z21			Geological hazard factors		
6	Z22		Collapse			
7	Z23		Landslide			
8	Z24		Debris flow			
9	Z25		Ground collapse			
10	Z26		Ground fissure			
11	Y11	Perimeter intrusion factors	Perimeter intrusion of animals	Land subsidence		
12	Y12			Collision with livestock		
13	Y21	Perimeter intrusion of objects and plants	Perimeter intrusion of objects and plants	Collision with other animals		
14	Y22			Falling objects		
15	Y23			Dangerous trees		
16	Y24			Tower not satisfying lodging distance		
17	Y25			Objects (a vehicle) slide into a track		
18	Y26			Tools and equipment left behind		
19	Y31			Perimeter intrusion of people	Perimeter intrusion of people	Obstacles artificially placing on the track
20	Y32					Suicide (including attempted suicide)
21	Y33	Fall off the platform				
22	Y34	Crossing the track illegally				
		Disabled people without care				

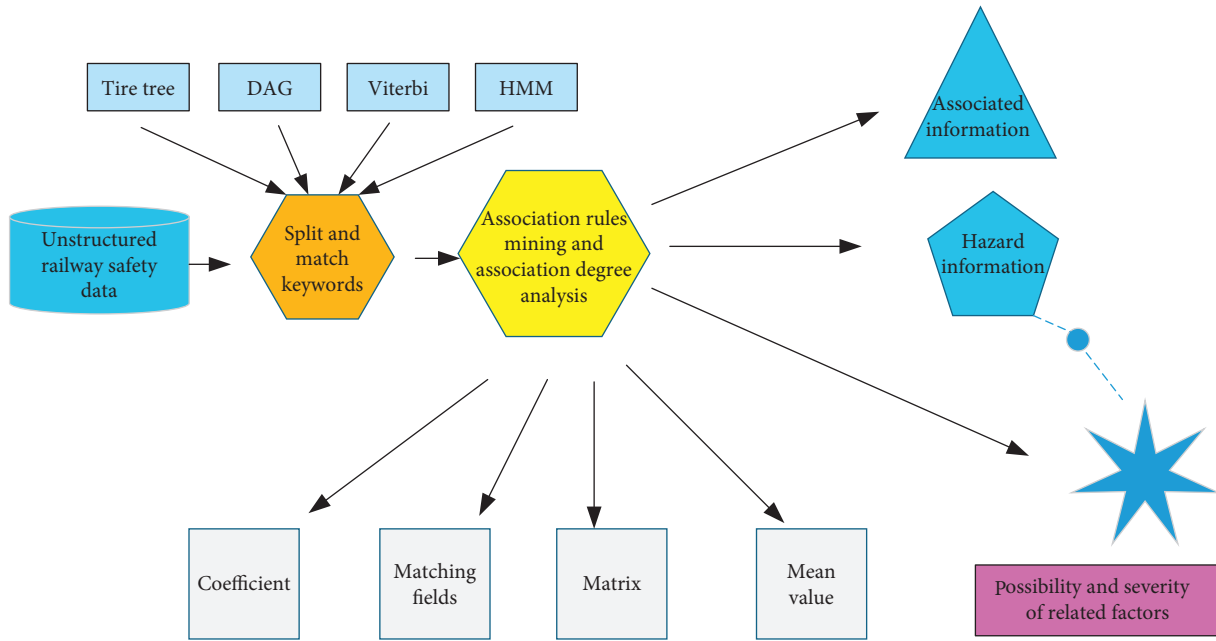


FIGURE 4: Data analysis and mining process.

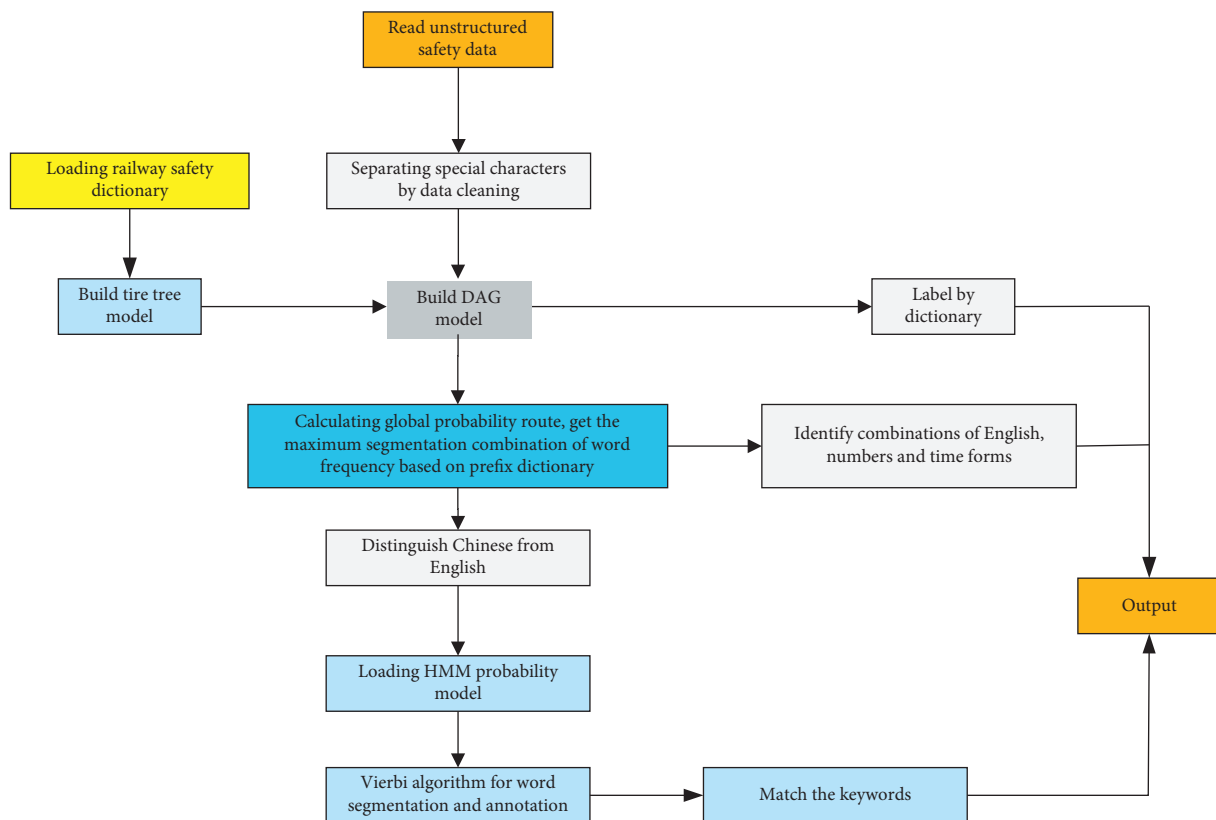


FIGURE 5: Keyword extraction and matching method.

frequency is not less than the minimum support degree of all item sets. Second, conduct mining strong association rules that satisfy the minimum confidence based on the frequent item sets obtained. The overall performance of association

rule data mining is determined by the operation of the previous process [24–28].

Finding frequent item sets is not easy because the data explosion involved in the calculation process may



TABLE 2: Sample data set of Apriori algorithm.

No.	Key words				
1	Error	Violation	Conflict	Class D accident	
2	Violation	Discipline	Derailment	Class D accident	
3	Violation	Collision	Class D accident		
4	Error	Violation	Train delay	Class D accident	
5	Violation	Dispatch command	Train delay	Class D accident	
6	Equipment	Control	Error	Train delay	Class D accident
7	Vehicle	Construction machinery	Train delay	Class D accident	
8	Error	Loading and unloading	Train delay	Class D accident	
9	Equipment	Fault	Train delay	Class D accident	

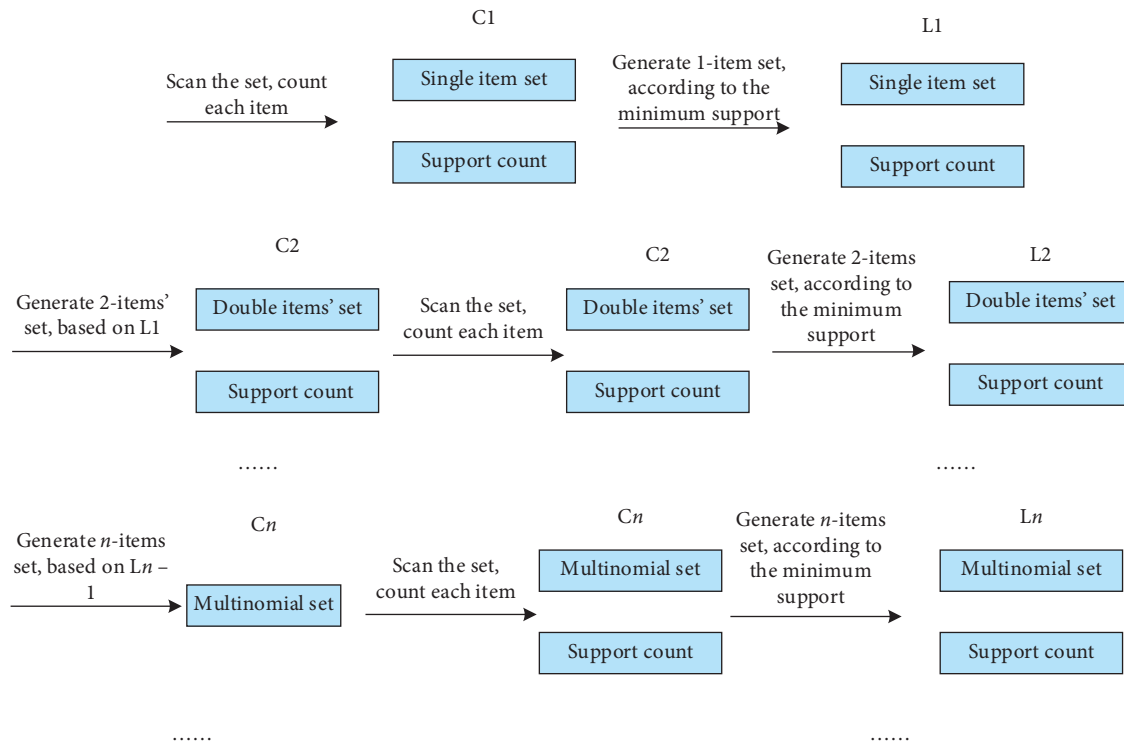


FIGURE 6: Frequent item sets' mining method.

lead to unacceptable computational complexity. However, as long as frequent item sets are obtained, association rules whose confidence is not less than minimum confidence could be explored. The association rules' mining algorithm used in this paper is Apriori algorithm. The following data in Table 2 is an example to illustrate the implementation process of Apriori algorithm [29–32].

Apriori algorithm could be used to mine association rules. The process of mining frequent item sets is shown in Figure 6.

$L_k$ : frequent item sets of length  $k$

$C_k$ : candidate item set of length  $k$

Support\_count ( $k$ ): the support count of  $k$ -item sets

It is concluded that all item sets of  $L_1$ ,  $L_2$ , and  $L_n$  are frequent item sets, and then, the confidence of each frequent item set is calculated. When the support threshold is set to

TABLE 3: Calculation results of Apriori algorithm example.

Relationship	Support (%)	Confidence (%)
Train delay-class D accident	66.67	100
Error-class D accident	44.44	100
Class D accident-train delay	66.67	66.67

40% and the confidence threshold is set to 50%, the results shown in Table 3 can be obtained.

**3.4. Grey Relation Analysis.** Grey relation analysis (GRA) is a multifactor statistical analysis method. The basic method of calculating the correlation degree is to initialize the original data sequence, then calculate the correlation coefficient, get the correlation degree and the correlation matrix through the combination of the correlation coefficient, and finally sort them according to the correlation degree calculation results of each correlation factor sequence [33].

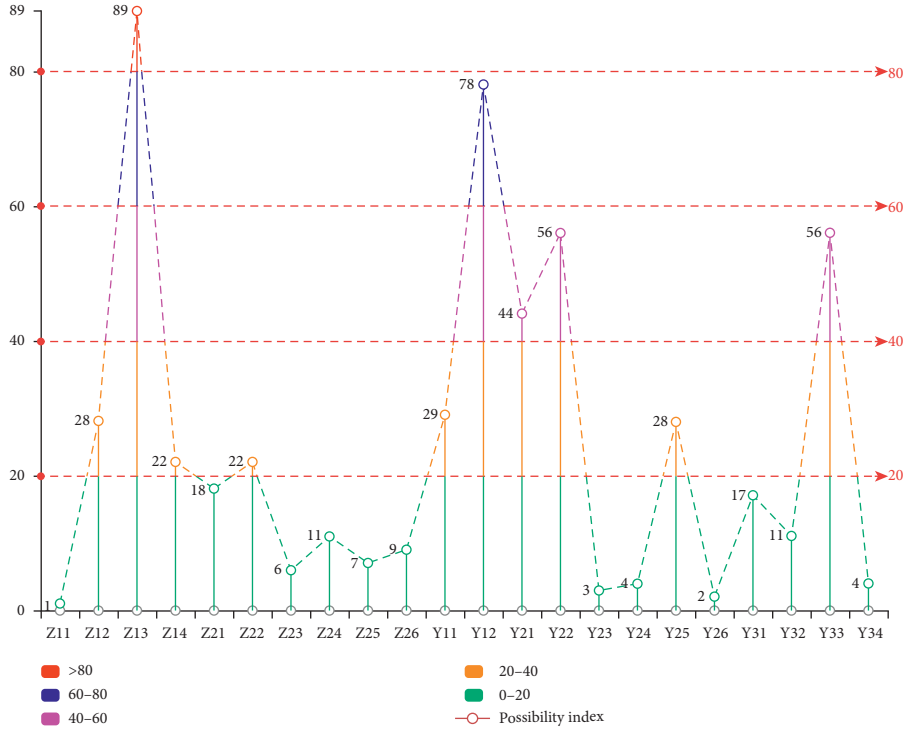


FIGURE 7: Possibility of external environment factors associated with high-speed railway derailment.

The calculation method is as follows.

Let the characteristic behavior sequence of the system be

$$X_i = (x_i(1), x_i(2), \dots, x_i(n)). \quad (1)$$

Because the units or initial values of each data sequence are different, in order to make them comparable, it is necessary to implement dimensionless processing on the original data so that the data of different dimensions (or

magnitudes) could be compared, and the initial value method could be used for calculation:

$$X'_i = \frac{X_i}{x_i(1)} = (x'_i(1), x'_i(2), \dots, x'_i(n)), \quad i = 0, 1, 2, \dots, m. \quad (2)$$

Correlation coefficient refers to the degree of correlation based on the geometric shape and development trend of each factor sequence. The expression is as follows:

$$\gamma_{0i}(k) = \frac{\min_i \min_k |x'_0(k) - x'_i(k)| + \rho \max_i \max_k |x'_0(k) - x'_i(k)|}{|x'_0(k) - x'_i(k)| + \rho \max_i \max_k |x'_0(k) - x'_i(k)|}, \quad \rho \in (0, 1), k = 1, 2, \dots, n, i = 1, 2, \dots, m. \quad (3)$$

Among them,  $\rho$  is called resolution coefficient. The smaller the  $\rho$  value is, the greater the resolution is.

The characteristic of association sequence is that it has a huge amount of data. When the information is processed in a centralized way, it is necessary to summarize the

association coefficients at different positions of different times into a specific value and calculate their average value. The average value obtained is the correlation degree. The expression is as follows:

$$\gamma_{0i}(k) = \frac{1}{n} \sum_{k=1}^n \frac{\min_i \min_k |x'_0(k) - x'_i(k)| + \rho \max_i \max_k |x'_0(k) - x'_i(k)|}{|x'_0(k) - x'_i(k)| + \rho \max_i \max_k |x'_0(k) - x'_i(k)|}, \quad \rho \in (0, 1), k = 1, 2, \dots, n, i = 1, 2, \dots, m. \quad (4)$$

#### 4. Risk Analysis Based on Unstructured Data

This paper attempts to analyze the risk of external environment associated with high-speed railway derailment from the possibility and the severity of the consequences and

realizes the scientific measurement of the risk by mining the possibility and the severity of the consequences of the unsafe events related to the external environment risk factors associated with high-speed railway derailment. The determination of the occurrence possibility is mainly based on the

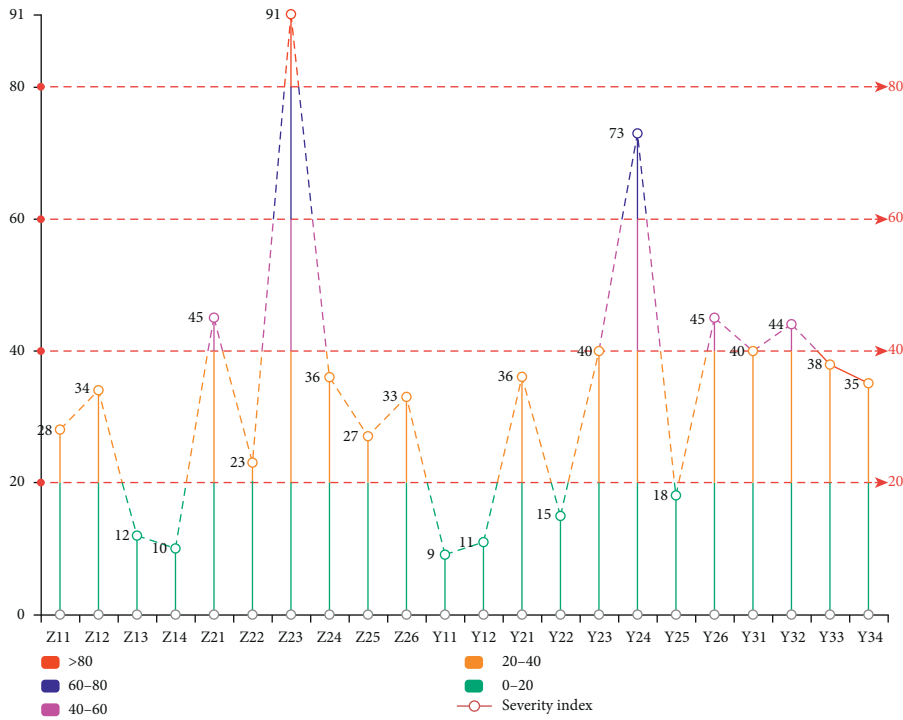


FIGURE 8: Severity of external environment factors associated with high-speed railway derailment.

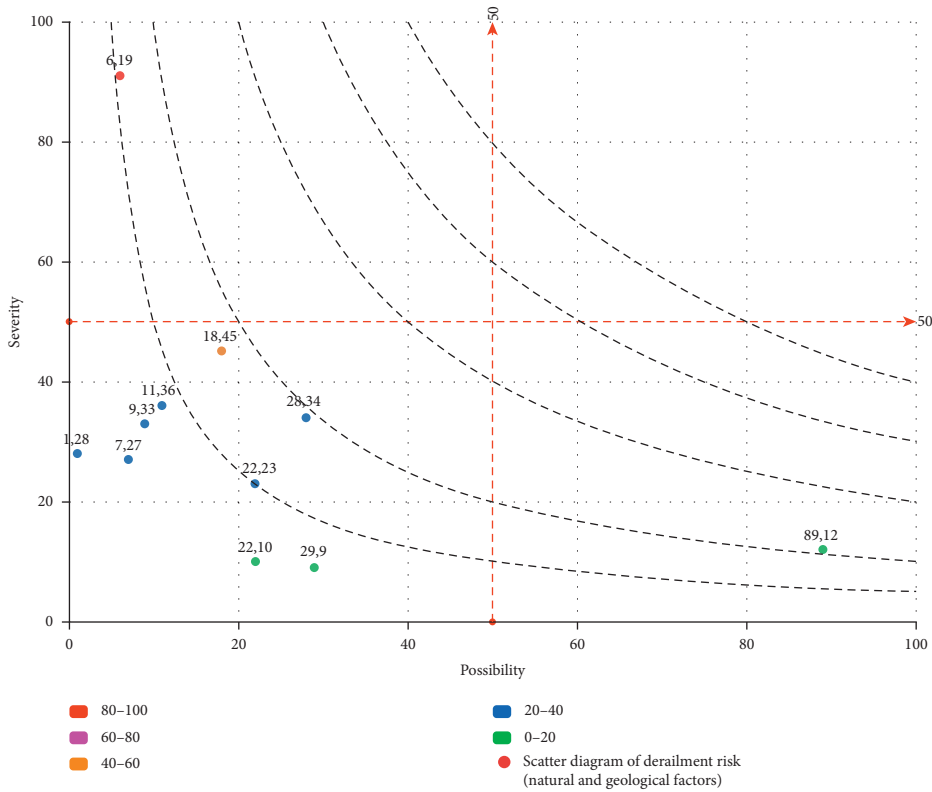


FIGURE 9: Risk distribution of natural and geological factors associated with high-speed railway derailment.

unstructured safety data mining and then accurately associates the unsafe events with the corresponding risk factors. It is realized by accumulating the occurrence frequency of the unsafe events associated with the risk factors outside the

high-speed railway derailment. Because the basic data involved in the study is mainly unsafe event data, the consequence is mainly the interruption time, so the severity of the consequence is mainly considered to mine the

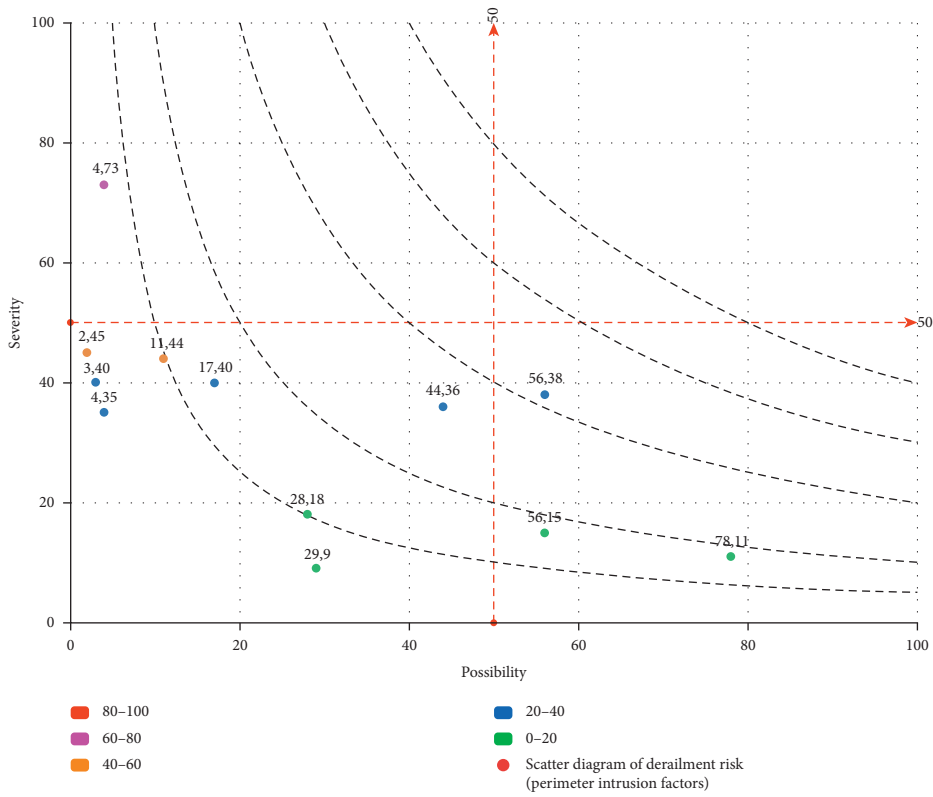


FIGURE 10: Risk distribution of perimeter intrusion factors associated with high-speed railway derailment.

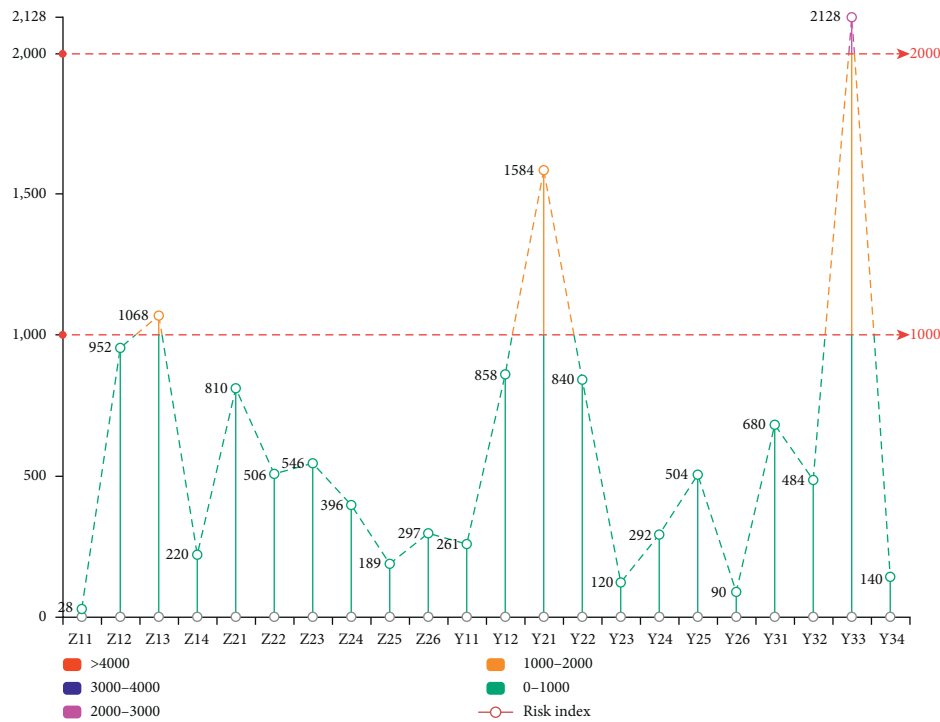


FIGURE 11: Distribution of comprehensive risk index of external environment factors associated with high-speed railway derailment.

interruption time caused by the unsafe events associated with the risk factors of high-speed railway derailment [34–37].

For a small number of factors with low probability of occurrence, they may not be associated with events in the data. In this case, we could consider using some evaluation

methods based on expert experience as a supplement, such as analytical hierarchy process. Based on the above method, after analysis, mining, and unifying the dimensions, the probability of external environment factors associated with high-speed railway derailment is shown in Figure 7, the consequence severity of external environment factors associated with high-speed railway derailment are shown in Figure 8, the risk distribution scatter diagram of natural and geological factors is shown in Figure 9, and the risk distribution scatter diagram of perimeter intrusion factors is shown in Figure 10.

The distribution of comprehensive risk index of external environment factors related to high-speed railway derailment is shown in Figure 11. It could be seen that Z13, Y21, and Y33 are high-risk factors, especially Y33. Risk management procedures should be implemented, and targeted measures should be taken to control them. When implementing control measures, we should pay attention to the actual effect and offer feedback to the implementation to ensure the full implementation of control measures [38–41].

## 5. Conclusion

Utilizing data on railway fault unsafe events, this paper establishes a matching model that builds correlations between unsafe events and external environment factors in the context of high-speed railway derailment. Operating in an automatic fashion, the model may be employed to analyze and mine the relationship between external environment factors of high-speed railway derailment and unsafe events. The model may also be used, with an enhanced accuracy for identifying high-risk elements, to extract the key feature information such as risk possibility and consequence severity.

The current investigation contributes to the field of research by introducing a statistical method for analyzing unsafe events' data. It seeks to identify high-risk elements of high-speed railway derailment, refines external environment risk index for high-speed railway derailment, and analyzes data in combination with the proposed model. As such, the study achieves a dynamic display of the results arising from external environment risk analysis in the context of high-speed railway derailment. The study is significant in which it seeks to rationalize the methods for analyzing external environment risk and to better visualize the safety laws of high-speed railway derailment. Therefore, the study helps to advance the operation of high-speed railway and its safety arrangements towards a more digitalized and smarter system.

Previous studies on high-speed railway risks mostly employ the expert evaluation method, which is arguably based on subjective deliberation and may undermine the research validity. The current project proposes a data-driven method of risk judgment, which may help to advance the feasibility and accuracy of the analysis. This study contributes to improving the safety level of railway operation by putting forward a method for integrating heterogeneous data sources, minimizing data heterogeneity, and thus, with enhanced accuracy, and building the association between the data and railway risks.

According to the needs of high-speed railway operation safety management, with the continuous accumulation of railway unsafe event data, the external environment risk model related to high-speed railway derailment could be continuously modified and improved, and the correlation matching between risk and unsafe event could be more accurate, which could ensure the continuous improvement of high-speed railway operation safety.

## Data Availability

Some or all data, models, or code generated or used during the study are proprietary or confidential in nature and may only be provided with restrictions.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was supported by the Fundamental Research Funds for the Central Universities (2021QY008) and Science & Technology Development Plan of China Railway Beijing Bureau Group Co., Ltd. (2020AQ01).

## References

- [1] UIC, *UIC Safety Report: Significant Accidents 2017 Public Report*, UIC, Paris, France, 2018.
- [2] L. Zhao, W. D. Wang, H. M. Xu, and Z. Guo, "Research on construction method of railway accident index in China," *China Railway*, vol. 2020, no. 5, pp. 29–32, 2020, in Chinese.
- [3] Y. L. Zhang, *Information Extraction Method of Unstructured Pathological Report Based on Pattern Matching*, Donghua University, Shanghai, China, 2017.
- [4] R. Zhu, B. Wang, X. C. Yang, and G. R. Wang, "Research on supporting probabilistic data range query index in big data environment," *Acta Computa Sinica*, vol. 39, no. 10, pp. 1929–1946, 2016.
- [5] H. X. Wang and Q. Q. Liang, "Risk analysis and route optimization of dangerous goods transportation based on the empirical path set," *Journal of Advanced Transportation*, vol. 2020, Article ID 8838692, 13 pages, 2020.
- [6] H. X. Wang, X. Y. Wang, Z. X. Wang, and X. D. Li, "Clustering analysis of dangerous goods transportation risk driving behavior based on data mining," *Transportation System Engineering and Information*, vol. 20, no. 1, pp. 183–189, 2020.
- [7] B. Z. Wang, C. P. L. Barkan, and M. R. Saat, "Quantitative analysis of changes in freight train derailment causes and rates," *Journal of Transportation Engineering, Part A: Systems*, vol. 146, no. 11, pp. 1–15, 2020.
- [8] C. Y. Lin, M. R. Saat, and C. P. L. Barkan, "Quantitative causal analysis of mainline passenger train accidents in the United States," *Journal of Rail and Rapid Transit*, vol. 234, pp. 1–16, 2019.
- [9] X. Liu, M. R. Saat, and C. P. L. Barkan, "Analysis of causes of major train derailment and their effect on accident rates," *Transportation Research Record*, vol. 2289, no. 1, pp. 154–163, 2012.
- [10] T. Turla, X. Liu, and Z. P. Zhang, "Analysis of freight train collision risk in the United States," *Journal of Rail and Rapid Transit*, vol. 233, no. 9, pp. 817–830, 2019.

- [11] X. Q. Li, *Research on Big Data Analysis Method of High Speed Railway Safety Text*, China Academy of Railway Sciences, Beijing, China, 2020.
- [12] Q. H. Zhou and X. L. Li, "Research on short text classification method of railway signal equipment fault based on MCNN," *Journal of Railway Science and Engineering*, vol. 16, no. 11, pp. 2859–2865, 2019.
- [13] W. G. Xie, J. Liu, T. X. Chen, G. Yang, and Y. J. Liu, "Research on railway safety big data analysis platform," *Railway Computer Application*, vol. 29, no. 9, pp. 21–25, 2020.
- [14] W. Dong, "Analysis of factors affecting railway safety and preventive measures," *Science and Technology Innovation Guide*, vol. 16, no. 14, pp. 165–166, 2019.
- [15] X. Li, "Analysis of railway safety accidents and countermeasures," *Henan Science and Technology*, vol. 2020, no. 7, pp. 96–98, 2020.
- [16] R. X. Wu, *Research on Acoustic Fault Diagnosis Method of Train Bearing Track Based on Two-Stage Learning Model*, Anhui University, Hefei, China, 2020.
- [17] Y. H. Ren, "Research on fault diagnosis of railway signal equipment," *Management and Technology of Small and Medium Sized Enterprises*, vol. 2020, no. 8, pp. 162–163, 2020.
- [18] H. Yang, Y. J. Yu, G. W. Chen, Y. B. Si, and D. F. Xing, "Research on fault diagnosis algorithm of turnout based on CNN-GRU model," *Acta Sinica Sinica*, vol. 42, no. 7, pp. 102–109, 2020.
- [19] A. P. Han, J. Y. Kan, X. Y. Jiang, W. D. Wang, D. J. Fu, and P. Liu, "Construction of China high speed railway electrical safety index system," *China Railway*, vol. 2020, no. 5, pp. 33–38, 2020.
- [20] J. F. Guo, Z. T. Ke, W. D. Wang et al., "Calculation method of power supply failure index of high speed railway based on data modeling," *China Railway*, vol. 2020, no. 5, pp. 39–45, 2020.
- [21] Y. J. Liu, T. C. Yang, W. Wu, Q. M. Liu, and W. D. Wang, "Design and application of high speed railway operation safety law analysis system based on big data technology," *China Railway*, no. 9, pp. 28–33, 2020.
- [22] X. W. Mi and S. Zhao, "Wind speed prediction based on singular spectrum analysis and neural network structural learning," *Energy Conversion and Management*, vol. 216, Article ID 112956, 2020.
- [23] X. W. Mi, H. Liu, and Y. Li, "Wind speed prediction model using singular spectrum analysis, empirical mode decomposition and convolutional support vector machine," *Energy Conversion and Management*, vol. 180, pp. 196–205, 2019.
- [24] X. Liu, "Analysis of collision risk for freight trains in the United States," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2546, no. 1, pp. 121–128, 2016.
- [25] R. T. Anderson and C. P. L. Barkan, "Railroad accident rates for use in transportation risk analysis," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1863, no. 1, pp. 88–98, 2004.
- [26] H. X. Wang, G. P. Xiao, and Z. Wei, "Optimizing route for hazardous materials logistics based on hybrid ant colony algorithm," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 752830, 6 pages, 2013.
- [27] T. X. Chen, W. G. Xie, G. Yang, E. C. Su, and Y. J. Liu, "Design scheme of railway safety integrated management system based on safety big data application platform," *Railway Computer Application*, vol. 30, no. 1, pp. 39–43, 2021.
- [28] C. F. Lu and D. Wang, "Technical system and safeguard measures for safe operation of high speed railway in China," *Chinese Journal of Safety Sciences*, vol. 29, no. 8, pp. 1–9, 2019.
- [29] G. L. Kang, "Research and practice of China's high speed railway security system," *Journal of Railways*, vol. 39, no. 11, pp. 1–7, 2017.
- [30] Y. M. Jia, L. Nie, H. X. Wang, and H. F. Liu, "Research on CRSMS architecture," *China Railway*, vol. 2020, no. 1, pp. 43–47, 2020.
- [31] Y. X. Fan, G. Fu, Y. W. Zhu, and X. Wang, "Review on the emergence and development of safety management system," *Chinese Journal of Safety Science*, vol. 25, no. 8, pp. 3–9, 2015.
- [32] P. Zhang, J. H. Liu, L. V. Feng, and X. Y. Li, "Analysis and enlightenment of European and American railway safety management mode," *Railway Transportation and Economy*, vol. 36, no. 11, pp. 40–44, 2014.
- [33] K. L. Wen and C. Chang, "Data preprocessing for grey relational analysis," *The Journal of Grey System*, vol. 11, no. 2, pp. 139–141, 1999.
- [34] J. Sun and H. J. Gao, "Analysis on construction and implementation of civil aviation safety management system in China," *China Railway*, vol. 2012, no. 4, pp. 10–14, 2012.
- [35] J. Li, "Research on acceptable safety level of civil aviation industry in China," *Science and Technology of Work Safety in China*, vol. 6, no. 5, pp. 137–142, 2010.
- [36] H. Liu, X. Mi, and Y. Li, "An experimental investigation of three new hybrid wind speed forecasting models using multi-decomposing strategy and ELM algorithm," *Renewable Energy*, vol. 123, pp. 694–705, 2018.
- [37] X. W. Mi, H. Liu, and Y. F. Li, "Wind speed forecasting method using wavelet, extreme learning machine and outlier correction algorithm," *Energy Conversion and Management*, vol. 151, pp. 709–722, 2017.
- [38] M. R. Saat and J. Aguilar Serrano, "Multicriteria high-speed rail route selection: application to Malaysia's high-speed rail corridor prioritization," *Transportation Planning and Technology*, vol. 38, no. 2, pp. 200–213, 2015.
- [39] M. R. Saat, C. J. Werth, D. Schaeffer, H. Yoon, and C. P. L. Barkan, "Environmental risk analysis of hazardous material rail transportation," *Journal of Hazardous Materials*, vol. 264, pp. 560–569, 2014.
- [40] M. R. Saat and C. P. L. Barkan, "Generalized railway tank car safety design optimization for hazardous materials transport: addressing the trade-off between transportation efficiency and safety," *Journal of Hazardous Materials*, vol. 189, no. 1–2, pp. 62–68, 2011.
- [41] X. Liu, M. Rapik Saat, and C. P. L. Barkan, "Freight-train derailment rates for railroad safety and risk analysis," *Accident Analysis & Prevention*, vol. 98, no. 1, pp. 1–9, 2017.