

Research Article

DeepCF: A Deep Feature Learning-Based Car-Following Model Using Online Ride-Hailing Trajectory Data

Yizhen Xie,¹ Qichao Ni,² Osama Alfarraj ,³ Haoran Gao,² Guojiang Shen,⁴ Xiangjie Kong ,⁴ and Amr Tolba^{3,5}

¹International School, Beijing University of Posts and Telecommunications, Beijing 100876, China

²School of Software, Dalian University of Technology, Dalian 116620, China

³Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia

⁴College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

⁵Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin-El-Kom 32511, Egypt

Correspondence should be addressed to Osama Alfarraj; oalfarraj@ksu.edu.sa

Received 14 August 2020; Revised 26 September 2020; Accepted 20 November 2020; Published 7 December 2020

Academic Editor: Nathalie Mitton

Copyright © 2020 Yizhen Xie 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 car-following model describes the microscopic behavior of the vehicle. However, the existing car-following models set the drivers' reaction time to a fixed value without considering its dynamics. In order to improve the accuracy of car-following model, this paper proposes Deep Feature Learning-based Car-Following Model (DeepCF), a car-following model based on fatigue driving and Generative Adversarial Networks (GAN). The model is composed of the drivers' reaction time model and the car-following decision algorithm. First, we regard driving fatigue as the starting point to study the influence of driving time and the acceleration of the preceding vehicle on the drivers' reaction time, and develop a coarse-grained drivers' reaction time model. Secondly, considering the impact of fatigue driving on car-following decisions, we utilize GAN to generate a driving decision database based on reaction time and use Euclidean distance as a decision search indicator. Finally, we conduct experiments on a real data set, and the results indicate that our DeepCF model is superior to baseline models.

1. Introduction

Vehicle following, the most common drivers' behavior in traffic, exerts more important influence on many factors including traffic flow characteristics, traffic safety, and traffic simulation results. The car-following model serves as a basic algorithm of traffic simulation tools (such as SUMO and VIS-SIM) and an indispensable control algorithm for automated vehicles [1, 2]. The model is aimed at replicating drivers' car-following behavior. The kinematics-based car-following model attempts to describe the kinematic mechanism of vehicle-following maneuver [3–12]. Most of the parameters have obvious physical meaning. The output of the model can be easily controlled by adjusting the model parameters, so as long as the appropriate parameters, it can perform better in car safety. As the car-following model based on machine learning attempts to learn the human drivers'

vehicle-following motion from a large number of human drivers' vehicle-tracking data [13–17], this category of model has a high accuracy in simulating the human drivers' vehicle following.

However, as the existing car-following models have become increasingly accurate in predicting driving decisions, they overlook the dynamic time for the drivers to execute the decision [18]. They set the drivers' reaction time to a fixed value [19]. This setting will greatly affect the car-following simulation performance. For example, in the car-following case, the front car brakes suddenly, causing the rear drivers to decide to slow down [20]. If this decision is 1.3 seconds late, it is likely to collide with the car ahead. Khodayari et al. use the performance characteristics of the drivers' stimulus and reaction while driving to calculate the drivers' reaction time in NGSIM data [21]. And they add the reaction time as known information to the existing car-following

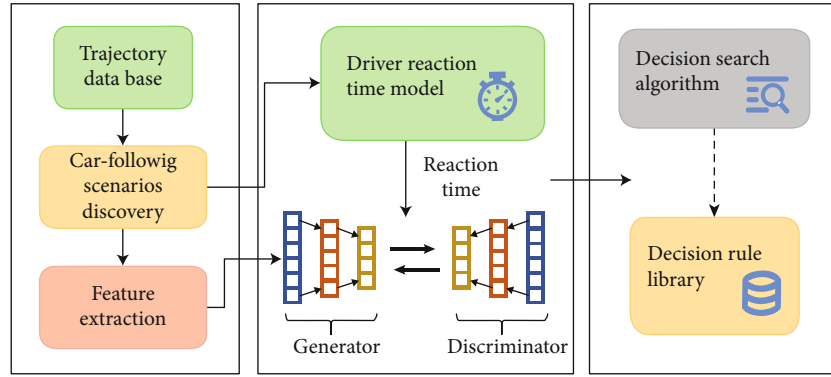


FIGURE 1: The framework of DeepCF model.

model for simulation experiments [22]. Finally, they confirmed that the drivers' reaction time existing in the car-following model will greatly improve the accuracy of the model simulation. However, they failed to build a model capable of calculating the drivers' reaction time. In addition, in predicting the driving behavior of the drivers, the existing car-following model does not involve two crucial factors. One is the impact of driving fatigue on driving decisions; the other is that driving fatigue causes the drivers' sensitivity and judgment to decline [23]. For example, driving for 1 hour continuously, a driver who faces the sudden acceleration of the vehicle ahead may be inclined to follow the same behaviour. However, when driving continuously for 3 h, in the face of the same scenario, the drivers' decision may be conservative acceleration, that is, the throttle will be much lighter than that before 2 h.

In this paper, we propose Deep Feature Learning-based Car-Following Model (The model frame diagram is shown in Figure 1). First, we study the influence of driving time and the acceleration of the preceding vehicle on the drivers' reaction time and establish a coarse-grained drivers' reaction time model. Secondly, grounded in the impact of driving time on car-following decisions, we generate a driving decision database based on GAN and use Euclidean distance as a decision indicator. Then, the decision search algorithm is proposed. Finally, we conduct contrast experiments on a real trajectory data, and the performance of DeepCF is evaluated.

The main contributions of this paper are listed as follows:

- (1) A car-following decision algorithm based on Generative Adversarial Networks is proposed, and a method for establishing a driving decision database based on Euclidean distance as a decision index is proposed
- (2) We design the framework based on the evidence that driving for a longer time will lead to a longer reaction time of the drivers
- (3) Analyze the model using China's online car-hailing trajectory data

The paper is organized as follows: Section 2 introduces the related work of the car-following model. Section 3 designs the car-following model. Section 4 provides experimental

results and compares them with traditional regression models. Section 5 summarizes the whole article.

2. Related Work

In this section, we first introduce the existing car-following model based on kinematics and the car-following model using machine learning algorithms. Then, we illustrate that the fatigue state will affect the drivers' reaction time.

2.1. Car-Following Model. Car-following models can be divided into kinematics-based models and models with machine learning algorithms. In the kinematics-based car-following models, Chandler et al. [3] first proposed the General Motors (GM) model. This model puts forward the relative speed of the front and rear vehicles to calculate the acceleration of the rear vehicle [4, 5]. The Gipps model takes the safety distance into account [6]. The optimal speed model obtains the expected rear vehicle speed based on the distance between the front and rear car heads [7, 8]. The action point model divides car following into different stages and sets space or speed thresholds separately [11, 12]. The car-following models based on machine learning algorithms are data-driven models. Wewerinke [13] uses neural networks to model the car-following behavior and achieves high performance. Khodayari et al. [14] propose an improved neural network car-following model with response time as input and verified it using NGSIM data [21]. The results show that the error is significantly less than other neural network models. Wei et al. [16] established a car-following model based on least squares support vector machine (LS-SVR) and used the microscopic traffic simulation system dataset [17] to verify the model. The experimental results show that LS-SVR car-following model is more accurate than the Gipps car-following model [6] and neural network car-following model.

2.2. Drivers' Reaction Time. In driving tasks, the drivers' brain nerve activity caused by exogenous stimuli is correlated with the drivers' reaction time [24], and there are many factors that can significantly affect the drivers' reaction time. Weng [23] divide the drivers' reaction behavior into three stages of perception, determination, and action. And this

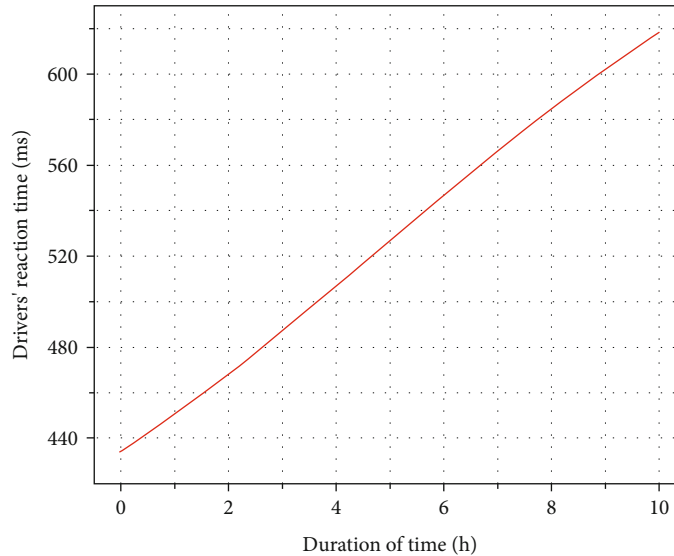


FIGURE 2: Drivers' reaction time function. Reaction time is positively correlated with driving duration.

paper verified that the drivers' sensitivity and judgment ability would weaken on a long road under fatigue state, and it would take longer time to make driving decisions. Visual distraction can further cause drivers to lose their attention on the road and affect their reaction time [25, 26]. Ru et al. [27] confirm that the subjective conditions that interfere with the drivers' response include driving experience, mental and physical conditions, and adaptability. Petermeijer et al. [28] explore the interaction between nondriving task types and take-over request methods. The test experiment of 101 volunteers confirm that nondriving tasks will increase the response time of the corresponding take-over request, and the initial response time of tactile and auditory take-over requests is lower than that of visual. In a traffic accident scenario, the drivers' reaction is related to the speed of obstacles before the accident [29], and the drivers' reaction time is linearly related to the collision time [30]. Xue et al. [31] analyze the simulation data of 47 volunteers in simulated car-following scenarios and found that in the case of high traffic flow density, the response time of drivers is usually shorter than that of low to medium traffic density [32, 33], while the response time of male and nonprofessional drivers tends to be slightly longer [34].

3. Methodology

This section introduces the DeepCF model in detail, including driver reaction time model and car-following decision algorithm based on GAN.

3.1. Drivers' Reaction Time Model Based on Fatigue Driving Phase Combination Model. It is intuitively obvious that the drivers' reaction time of fatigue driving is longer than that of normal driving. And in the assumption of this paper, the car-following decision is closely related to the drivers' reaction time. We need to reveal the relationship between the drivers' reaction time and driving time to build drivers' reaction time model. The reaction time model can quantitatively

TABLE 1: Training set data structure for GAN.

Attribute name	Speed difference between front and rear cars	Rear vehicle speed	Front and rear distance	Acceleration of rear car after reaction time
Unit	Km/h	Km/h	m	m/s ²

represent the relationship between driving time and reaction time. They investigate 294 drivers and test their reaction time [35]. The final statistical results showed a strong correlation between duration of driving and reaction time. On the basis of their experimental results, we use the cubic function to fit the relationship between duration of driving and the slowest reaction time. Our experiments show that cubic polynomials are the best choice to fit this relation. Due to the lack of original experimental data, the accuracy of the fitted function remains to be verified. The formulas are shown in Equation (1) and function visualization is shown in Figure 2.

According to the results of our experiment, in Equation (1), α_3 is -0.067, α_2 is 0.9769, α_1 is 15.27 and, α_0 is 434.4. These parameters can fit the duration of time and drivers' reaction time well. And there is no overfitting in this model.

$$\tau(x) = \alpha_3 x^3 + \alpha_2 x^2 + \alpha_1 x + \alpha_0. \quad (1)$$

3.2. Car-following Decision Algorithm Model Based on Generative Adversarial Network. We need to design the drivers' car-following decision algorithm. First, we need to establish a decision library. We will use GAN (Generative Adversarial Network) to generate the decision library rules. GAN, an unsupervised deep learning model, is composed of two parts: generator and discriminator. The generator generates data close to the characteristics of the training set as much as possible, and the discriminator should attempt to determine the authenticity of the generator. It can find out the internal statistical law of the given observation data and

Input:

Training set data distribution $P_{\text{data}}(x)$; random noise distribution $P_g(z)$; Total training times epochs; the number of iterations of the discriminator k ; the learning rate of the discriminator s_1 ; the learning rate of the generator s_2 ; the amount of training data per batch n .

Output:

The network parameters of the discriminator θ_d ; network parameters of the generator θ_g .

Begin

1. Initialize θ_d ,
2. For epochs do
3. For k do
4. Sample n samples $\{z^{(j)}\}_{j=1}^n$ from the random noise distribution $P_g(z)$
5. Sample n samples $\{x^{(j)}\}_{j=1}^n$ from the real data distribution $P_{\text{data}}(x)$
6. Update θ_d by boosting the stochastic gradient:
7. $\nabla_{\theta_d}(1/n)\sum_{j=1}^n[\log D(x^{(j)}) + \log(1 - D(G(z^{(j)})))]$
8. End for
9. Sample n samples $\{z^{(j)}\}_{j=1}^n$ from the random noise distribution $P_g(z)$
10. Update ∇_{θ_g} with gradient by decreasing:
11. $\nabla_{\theta_g}(1/n)\sum_{j=1}^n[\log D(x^{(j)}) + \log(1 - D(G(z^{(j)})))]$
12. End

ALGORITHM 1. Minibatch stochastic gradient descent training algorithm for generating adversarial networks.

can generate brand new data similar to the observation data based on the obtained probability distribution model. The target formula is shown in Equation (2).

$$\min_G \max_D V(G, D) = E_{x \sim P_{\text{data}}(x)}[\log(D(x))] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))], \quad (2)$$

where G is the differentiable function of the generator, D is the differentiable function of the discriminator, P_{data} is a real sample, x is the sample taken from P_{data} , and $D(x)$ means to distinguish x ; we hope that the result of this discrimination is closer to 1, as possible the bigger the loss function $\log(D(x))$. P_z is the sample generated by G , z is the sample taken from P_G , and $D(G(z))$ means to distinguish $G(z)$; we hope that the smaller the result, as possible the bigger the loss function $\log(1 - D(G(z)))$. The structure of the training set is shown in Table 1.

The network is divided into two parts: generator and discriminator. The generator is composed of input layer, hidden layer, activation layer, and output layer. The dimension of the input and output layers is $1 * 4$. The hidden layer contains $2 * 2 * 11$ neural network units. The activation layer is composed of the Maxout activation function and sets the k value to 2. Maxout has a strong fitting ability; it can fit any convex function. Maxout has the advantages of ReLU, such as no linear saturation, but also does not have the disadvantage that the ReLU unit is fragile and may die.

Let the distribution generated by the generator (G) be $P_G(x; \theta)$, where θ is the parameter of the distribution. And let x^i be derived from the true distribution in the generator to calculate the likelihood $P_G(x^i; \theta)$ as shown in Formula (3).

$$L = \prod_{m=1}^1 P_G(x^i; \theta). \quad (3)$$

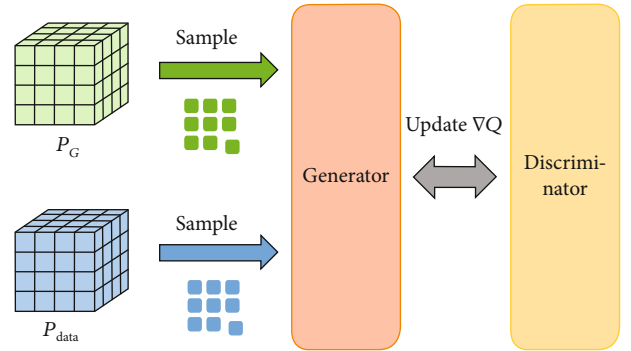


FIGURE 3: Structure diagram of Algorithm 1.

Then, you need to find a θ^* to maximize the likelihood, as shown in Formula (4).

$$\theta^* = \arg \min_{\theta} KL(P_{\text{data}}(x) || P_G(x; \theta)). \quad (4)$$

The discriminator (D) is composed of an input layer, a hidden layer, an activation layer, and an output layer. The activation function is the sigmoid function. The discriminator's data consists of two parts, the first part is the real data set P_{data} , and the second part is the fake data P_G generated by the generator. If x comes from P_{data} , $D(x)$ should be as close to 1 as possible. If x comes from P_G , $D(x)$ should be as close to 0 as possible. The pseudocode of GAN algorithm is shown in Algorithm 1. The structure diagram of Algorithm 1 is shown in Figure 3.

The decision index is used to evaluate the similarity of the input features and the data of the decision database, that is, to calculate the similarity of the two feature vectors. There are many ways to calculate vector similarity, such as Pearson correlation coefficient, cosine similarity, and Manhattan distance. The decision index is mainly used to calculate the

Input:
Decision library D ; Searched feature vector x

Output:
The fourth feature of the most similar variable y_4

Begin

1. Standardized D and x
Set the threshold
2. Brush the database according to the feature vector x
3. Pick out the vector set V within the threshold
4. for i in V
5. use the decision index algorithm to calculate the distance between x and i , $c_1 = s(x, i)$
6. Sort the vector in V according to c_1
7. Output the fourth feature of the closest vector

End

ALGORITHM 2. Decision database retrieval algorithm s_{su} .

TABLE 2: Data attributes.

Field	Type of data	Example	Remarks
Driver ID	String	glox.jrrlltBMvCh8nxqktdr2dtopmlH	Data encryption processing
Order ID	String	jkkt8kxniovIFuns9qrrlvst@iqnpkwz	Data encryption processing
Timestamp	String	1501584540	Unix timestamp, in seconds
Longitude	String	104.04392	GCJ-02 coordinate system
Latitude	String	30.6863	GCJ-02 coordinate system

distance between two feature vectors, so this paper uses Euclidean distance. The index formula is shown in Formula (5). The purpose of the retrieval algorithm of this machine is to search the decision database for the decision plan (the fourth feature of Table 1) that is closest to the current driving state (the first three features of Table 1) to reach the goal of driving decision-making.

$$s(x, y) = \sqrt{\sum_{j=1}^3 (x_j - y_j)^2}, \quad (5)$$

where y represents a feature vector in the decision library, x represents the input feature vector, and the first three features of x and y correspond to the first three features of the data structure shown in Table 1.

Due to the relatively large amount of rule data in the decision database, traversing the entire decision database to find the most similar vectors may affect the algorithm performance, so the retrieval of the decision database should be optimized. The statistics show that, aside from extreme cases, the value range of y_1 is approximately $[-5, 10]$, the value range of y_2 is approximately $[0, 70]$, and the value range of y_3 is $[1, 80]$. Then, there are about 80,000 combinations of these three feature vectors. The following defines the retrieval algorithm s_{su} :

3.3. Deep Feature Learning-Based Car-Following Model. The driver reaction time model and the car-following decision algorithm have been obtained above, then we propose Deep

Feature Learning-based Car-Following Model (DeepCF) here. Equations (6) and (7) describe the model details.

$$\tau_n = \tau(x), \quad (6)$$

$$a_x(t + \tau) = S_{su}(D, x), \quad (7)$$

where x is the feature vector.

4. Experiments

This chapter mainly introduces the experimental part, first introduces the comparative experiment, then defines the modulus evaluation index, and finally, analyzes the experimental results.

4.1. Experiment Data. The data set used in this article is the trajectory data of Didi drivers in Xi'an on October 26 and 27, 2016, provided by Didi Travel (please visit: <https://outreach.didichuxing.com/research/opendata/>).

The data set tracks approximately 18,000 vehicles, including the vehicle number, location, and time. The trajectory tracking time interval is 1 s. We screened the vehicles driving continuously for more than 3 h between 4:00-22:00, and finally extracted 5748 following scenes. The car-following duration is 9 s. We further add attributes such as speed, acceleration, driving duration, and distance to each piece of data. We divided the data set into two groups, the first group is the following vehicles in the scene where the driving time of the rear vehicle is less than two hours, and the second group is the vehicle that exceeds two hours.

Besides that, we select 100 follow-up scenes in each group as the test set. Data attribute list refers to Table 2.

4.2. Comparative Experiment. T. Cover and P. Hart proposed k-nearest neighbor in 1967. The working principle is there is a sample data set, and each data in the sample set has a label, that is, we know the correspondence between each data in the sample set and its classification. After inputting new data without labels, we compare each feature of the new data with the features corresponding to several types of data in the sample, and then, the algorithm extracts the classification label of the most similar data (nearest neighbor) of the sample. It uses the following Formula (8) to calculate the distance D . x and y are the two features that need to be calculated.

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (8)$$

Random forest is a model composed of many decision trees. In the training process, each tree in the random forest will learn from randomly sampled data points. The idea is to train each tree on different samples. Although for a specific training data set, the variance of each tree may be high, but in general, the variance of the entire forest will be very low without increasing the offset. In the test, predictions are made by averaging the predictions of each decision tree. This process of training a single learner on different self-sampled data subsets and averaging predictions is called bagging. The cart tree is used in the random forest algorithm in Sklearn.

The Gini index reflects the probability that two samples are randomly selected from the data set, and their category labels are inconsistent. Therefore, the smaller the Gini index, the higher the purity of the data set. The Gini index (Formula (9)) can be used to measure any uneven distribution. It is a number between 0 and 1.0 that is completely equal, and 1 is completely unequal.

$$\text{Gini}(D|A) = \sum_{k=1}^K \frac{|C_k|}{|D|} \left(1 - \frac{|C_k|}{|D|}\right), \quad (9)$$

where k represents the category.

The CART classification tree uses the size of the Gini coefficient to measure the division points of features. In the regression model, we adopt the common sum variance measurement method. For any partition feature A , the corresponding arbitrary partition point s is divided into data sets D_1 and D_2 on both sides, and the mean square error of each set of D_1 and D_2 is minimized. The feature and feature value division point corresponding to the minimum sum of mean square error of D_1 and D_2 . The expression is Formula (10):

$$(y_i - c_1) \min_{a,s} \left[\min_{c_1} \sum_{x_i \in D_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in D_2} (y_i - c_2)^2 \right]. \quad (10)$$

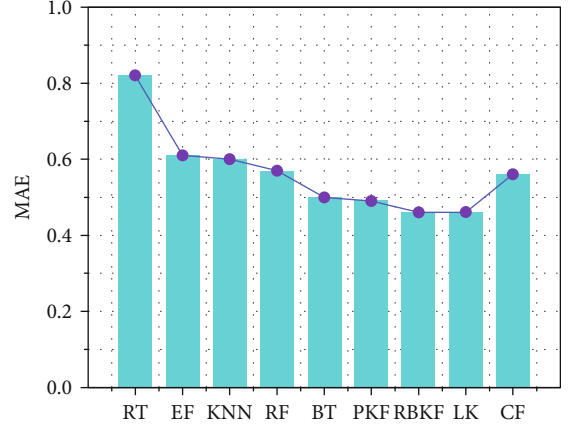


FIGURE 4: Comparison of CF and regression model with MAE as an evaluation index.

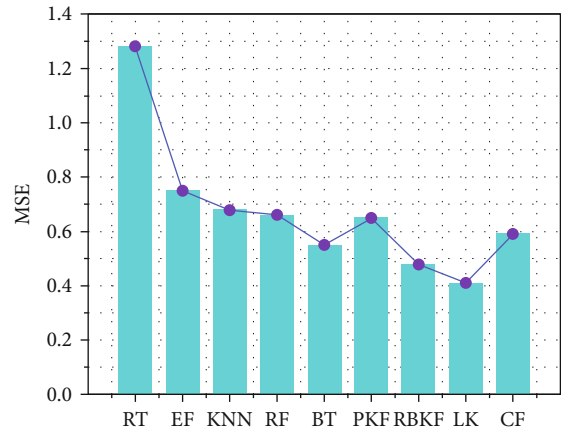


FIGURE 5: Comparison of CF and regression model with MSE as an evaluation index.

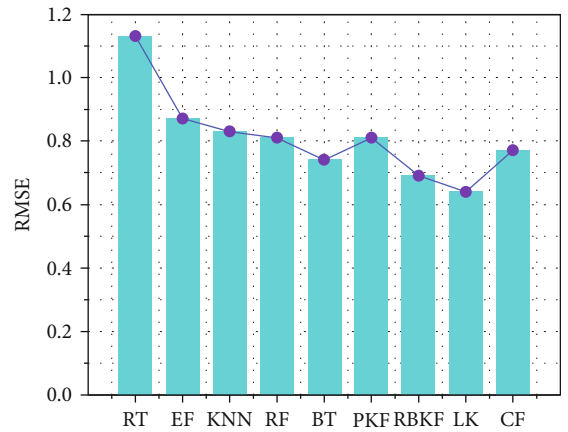


FIGURE 6: Comparison of CF and regression model with RMSE as an evaluation index.

Among them, c_1 is the sample output average of the D_1 data set, and c_2 is the sample output average of the D_2 data set.

4.3. Model Evaluation. In this paper, mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) are used for measurement and evaluation [36]. The indicator formulas are shown in Equations (11), (12), and (13).

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |x_i - \hat{x}_i|, \quad (11)$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2, \quad (12)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2}, \quad (13)$$

where N is the total number of samples, x_i represents real data value of car i , and \hat{x}_i means the predicted value of the model.

5. Results and Discussion

This paper proposes a car-following model based on driving fatigue and generative confrontation network, namely, dynamic car-following model (DeepCF). We use 474 real car-following scenes for experiments, and the experiments combine the DeepCF model proposed in this article with the regression model regression tree (RT), polynomial kernel function (PKF), radial basis kernel function (RBKF), boost tree (BT), extreme forest (EF), linear kernel (LK), k-nearest neighbor (KNN), and random forest (RF) for comparison. We use MAE, MSE, and RMSE as evaluation indicators. We use 474 real car-following scenes as the test set.

First, we use all real 8000 car-following scenes as a decision library. Besides, we adopt the decision database retrieval algorithm $s-su$; the acceleration of the following vehicle after the reaction time is matched according to the speed of the following vehicle, the distance between the front and rear vehicles, and the speed difference between the front and rear vehicles in the test set. And we calculate the evaluation index of the comparison between the real car-following scene and the matching result (for the convenience of subsequent experiments, we will name the experiment CF here). Then, we use the above regression model to obtain the acceleration of the following vehicle after the reaction time. It also calculates the evaluation index comparing the real car-following scene with the regression result. Finally, we compare the evaluation index of the matching result with the evaluation index of the regression result. The results are shown in Figures 4, 5, and 6.

According to Figures 3, 4, and 5, we can find that when using a limited number of real car-following scenes as the decision-making database, using the decision-making database retrieval algorithms, the matching results are better than RT, EF, KNN, and other regression models. However, compared with RBKF, LK, and other regression models, there is still a certain gap.

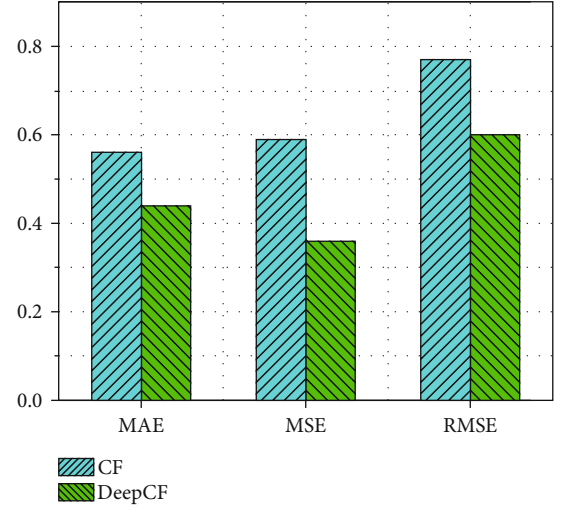


FIGURE 7: DeepCF compared with CF.

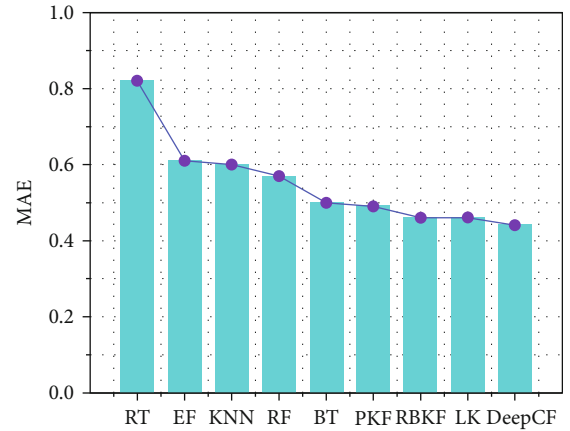


FIGURE 8: Comparison of DeepCF and regression model with MAE as an evaluation index.

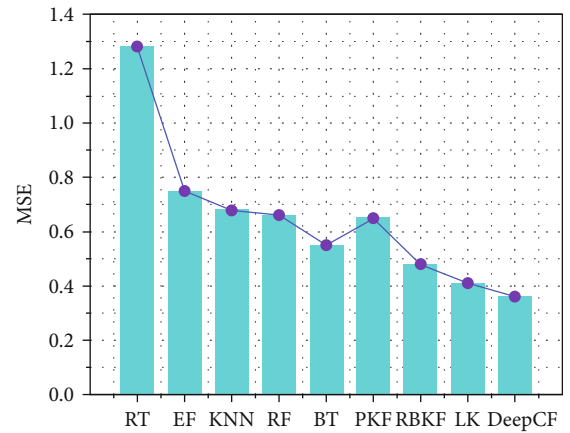


FIGURE 9: Comparison of DeepCF and regression model with MSE as an evaluation index.

We use 8000 real car-following scenes as raw data to use the GAN to generate 80,000 car-following scenes, and combine the real and generated car-following scenes as a decision

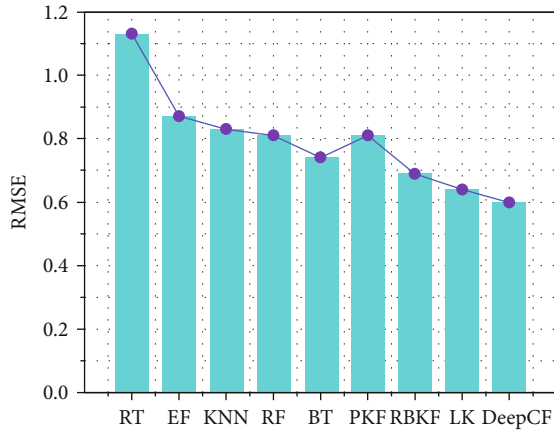


FIGURE 10: Comparison of DeepCF and regression model with RMSE as an evaluation index.

library. Using the decision database retrieval algorithm s^{-su} , the acceleration of the following vehicle after the reaction time is matched according to the speed of the following vehicle, the distance between the front and rear vehicles, and the speed difference between the front and rear vehicles in the test set. And we calculate the evaluation index of the comparison between the real car-following scene and the matching result (our model DeepCF). We compare the evaluation indicators obtained after the DeepCF model experiment with the above CF model Figure 7.

We find that after using the GAN to expand the decision-making database, the matching results are significantly better than before data generation. This proves the necessity of using GAN to generate car-following scenes.

We compare the evaluation indicators obtained after the DeepCF model experiment with the regression model, and the results are shown in Figures 8, 9, and 10.

In the premise of using the confrontation generation network to generate 80,000 car-following scenes with 8000 real car-following scenes as raw data, combining the real and generated car-following scenes as a decision-making database and adopting the decision-making database retrieval algorithms, the matching data performs better than most data in the regression model. Thus, the effectiveness of our DeepCF model is proved.

6. Conclusions

In view of the fact that existing car-following models fail to consider the impact of driving fatigue on driver reaction time and decision-making, this paper proposes a car-following model (DeepCF) based on driving fatigue and generating a confrontation network. A car-following decision algorithm based on a generative confrontation network is proposed, and we build a driving decision database with GAN. Besides a comparative experiment, we further conduct a comparative evaluation of the model. The results demonstrate that our car-following model (DeepCF) is closer to the real scene than the regression model.

Data Availability

The data used to support the findings of this study were supplied by Didi Travel under license and so cannot be made freely available. Requests for access to these data should be made to the corresponding author.

Conflicts of Interest

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

Acknowledgments

This work was funded by the Researchers Supporting Project No. (RSP-2020/102) King Saud University, Riyadh, Saudi Arabia, and by the National Natural Science Foundation of China (Grant nos. 62072409 and 62073295). The author would like to thank Mengyuan Wang and Zhen Ren from the School of Software, Dalian University of Technology, for their help.

References

- [1] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2018.
- [2] L. Li and W. Ma, "A collision-free car-following model for connected automated vehicles," *Transportation Research Board 96th Annual Meeting*, pp. 1–19, 2017.
- [3] G. Shen, C. Chen, Q. Pan, S. Shen, and Z. Liu, "Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels (May, 2018)," *IEEE Access*, vol. 6, pp. 51756–51765, 2018.
- [4] X. Kong, X. Liu, B. Jedari, and M. Li, "Mobile crowdsourcing in smart cities: technologies, applications and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019.
- [5] Z. Ning, P. Dong, X. Wang et al., "Mobile edge computing enabled 5G health monitoring for internet of medical things: a decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, pp. 1–16, 2020.
- [6] R. E. Chandler, R. Herman, and E. W. Montroll, "Traffic dynamics: studies in car following," *Operations Research*, vol. 6, no. 2, pp. 165–184, 1958.
- [7] R. Herman, E. W. Montroll, R. B. Potts, and R. W. Rothery, "Traffic dynamics: analysis of stability in car following," *Operations Research*, vol. 7, no. 1, pp. 86–106, 1959.
- [8] D. C. Gazis, R. Herman, and R. B. Potts, "Car-following theory of steady-state traffic flow," *Operations Research*, vol. 7, no. 4, pp. 499–505, 1959.
- [9] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B: Methodological*, vol. 15, no. 2, pp. 105–111, 1981.
- [10] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Physical review E*, vol. 51, no. 2, pp. 1035–1042, 1995.

- [11] R. Jiang, Q. Wu, and Z. Zhu, "Full velocity difference model for a car-following theory," *Physical Review E*, vol. 64, no. 1, article 017101, 2001.
- [12] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.
- [13] L. Evans and R. Rothery, "Perceptual thresholds in car-following—a comparison of recent measurements with earlier results," *Transportation Science*, vol. 11, no. 1, pp. 60–72, 1977.
- [14] P. Wewerinke, "Modeling human learning involved in car driving," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1968–1973, San Antonio, TX, USA, 1994.
- [15] A. Khodayari, A. Ghaffari, R. Kazemi, and R. Brauningl, "A modified car-following model based on a neural network model of the human driver effects," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 6, pp. 1440–1449, 2012.
- [16] D. Wei, F. Chen, and T. Zhang, "Least square-support vector regression based car-following model with sparse sample selection," in *2010 8th World Congress on Intelligent Control and Automation*, pp. 1701–1707, Jinan, China, 2010.
- [17] D. Wei, F. Chen, and X. Sun, "An improved road network partition algorithm for parallel microscopic traffic simulation," in *2010 International Conference on Mechanic Automation and Control Engineering*, pp. 2777–2782, Wuhan, China, 2010.
- [18] X. Kong, F. Xia, J. Li, M. Hou, M. Li, and Y. Xiang, "A shared bus profiling scheme for smart cities based on heterogeneous mobile crowdsourced data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1436–1444, 2020.
- [19] Z. Ning, P. Dong, X. Wang et al., "When deep reinforcement learning meets 5G-enabled vehicular networks: a distributed offloading framework for traffic big data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1352–1361, 2019.
- [20] A. Khodayari, R. Kazemi, A. Ghaffari, and R. Brauningl, "Design of an improved fuzzy logic based model for prediction of car following behavior," in *2011 IEEE International Conference on Mechatronics*, pp. 200–205, Istanbul, Turkey, 2011.
- [21] M. Weng, "A study on evaluation of automobile driver fatigue performance," in *2010 International Conference on Machine Vision and Human-machine Interface*, pp. 92–94, Kaifeng, China, 2010.
- [22] Z. Ning, K. Zhang, X. Wang et al., "Joint computing and caching in 5G-envisioned internet of vehicles: a deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.
- [23] X. Kong, J. Cao, H. Wu, and C. Hsu, "Mobile crowdsourcing and pervasive computing for smart cities," *Pervasive and Mobile Computing*, vol. 61, p. 101114, 2020.
- [24] Z. Ning, K. Zhang, X. Wang et al., "Intelligent edge computing in internet of vehicles: a joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.
- [25] H. Zhang, R. Chavarriaga, L. Gheorghe, and J. D. R. Millán, "Brain correlates of lane changing reaction time in simulated driving," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3158–3163, Kowloon, China, 2015.
- [26] Y. Liang and J. D. Lee, "Combining cognitive and visual distraction: less than the sum of its parts," *Accident Analysis & Prevention*, vol. 42, no. 3, pp. 881–890, 2010.
- [27] G. K. Kountouriotis, P. Spyridakos, O. M. Carsten, and N. Merat, "Identifying cognitive distraction using steering wheel reversal rates," *Accident Analysis & Prevention*, vol. 96, pp. 39–45, 2016.
- [28] G. Ruhai, Z. Weiwei, and W. Zhong, "Research on the driver reaction time of safety distance model on highway based on fuzzy mathematics," in *2010 International Conference on Optoelectronics and Image Processing*, vol. 2, pp. 293–296, Hainan, China, 2010.
- [29] S. Petermeijer, F. Doubek, and J. de Winter, "Driver response times to auditory, visual, and tactile take-over requests: a simulator study with 101 participants," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1505–1510, Banff, AB, Canada, 2017.
- [30] M. Sieber and B. Färber, "Driver perception and reaction in collision avoidance: implications for adas development and testing," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 239–245, Gothenburg, Sweden, 2016.
- [31] R. S. Jurecki and T. L. Stańczyk, "Analyzing driver response times for pedestrian intrusions in crash-imminent situations," in *2018 XI International Science-Technical Conference Automotive Safety*, pp. 1–7, Casta, Slovakia, 2018.
- [32] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, p. 1, 2020.
- [33] X. Kong, S. Tong, H. Gao et al., "Mobile edge cooperation optimization for wearable Internet of things: a network representation-based framework," *IEEE Transactions on Industrial Informatics*, 2020.
- [34] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: a decentralized computation offloading algorithm," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, pp. 411–425, 2021.
- [35] P. Philip, J. Taillard, M. Quera-Salva, B. Bioulac, and T. Åkerstedt, "Simple reaction time, duration of driving and sleep deprivation in young versus old automobile drivers," *Journal of Sleep Research*, vol. 8, no. 1, pp. 9–14, 2002.
- [36] A. Khodayari, A. Ghaffari, R. Kazemi, and N. Manavizadeh, "Anfis based modeling and prediction car following behavior in real traffic flow based on instantaneous reaction delay," in *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 599–604, Funchal, Portugal, 2010.