Electronic Research Archive

ERA, 31(1): 401–420.
DOI: 10.3934/era.2023020
Received: 29 September 2022
Revised: 26 October 2022
Accepted: 28 October 2022
Published: 07 November 2022

http://www.aimspress.com/journal/ERA

Research article

A deep learning approach for vehicle velocity prediction considering the influence factors of multiple lanes

Mingxing Xu[1], Hongyi Lin[2,*] and Yang Liu[2]

[1] Ministry of Housing and Urban-Rural Development of the People's Republic of China
[2] State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University, Beijing, China

* **Correspondence:** Email: hy-lin22@mails.tsinghua.edu.cn.

Abstract: Predicting the future velocity of vehicles is essential for the safety of autonomous driving and the Intelligent Transport System. This study investigates how the surrounding vehicles influence a driving vehicle. Based on the HighD dataset, a scenario that considers the current lane and the neighboring lanes is selected while the drivers' visual angles and visual gap angles along with other parameters in the dataset are characterized as features. To predict the velocity of a driving vehicle and calibrate the influence of surrounding vehicles, a Transformer-based model integrating the features of multiple vehicles is proposed, and different features are added to the layers while constructing the model. Moreover, the information from previous timestamps of the vehicle state is integrated to estimate the duration of the influences, since the influence of an incident is not instantaneous. In our experiments, we find that the duration of the influence on the driving state perfectly fits the driver's reaction time when maneuvers occur in the surrounding vehicles. In addition, we further quantify the importance of the influence on the vehicle velocity prediction based on the Random Forest and obtain some practical conclusions, for instance, the velocity of a vehicle is more influenced by the front vehicle in the left lane than that in the right lane, but is still mainly influenced by the front vehicle in the current lane.

Keywords: vehicle velocity prediction; deep learning; feature engineering; time series analysis

1. Introduction

With the significant advances in machine learning over the past few decades, autonomous driving has attracted tremendous attention and achieved great progress [1–3]. Autonomous vehicles (AVs) will free the driver's hands, which may probably take the place of human-driven vehicles (HDVs), or at least capture a large market share in the future [4–6]. Moreover, AVs have the potential to alleviate traffic congestion, eliminate traffic accidents and reduce fuel consumption, which opens the door to highly efficient and sustainable transportation systems [7–9].

Nevertheless, every newly sprouted technology has its merits and defects. Most of the existing AVs make decisions by sensing the current environment information, whereas the predictive capability is often less mature especially when the surrounding environments are complex [10,11]. Since there is a slight time lag between the sensing and decision-making processes, AVs can barely avoid unexpected situations during the time gap subject to the lack of predictive capacity [12,13]. In some other cases, AVs usually behave very conservatively in the presence of other traffic participants, such as decelerating, which also increases the likelihood of traffic accidents [14]. Additionally, vehicles in the current lane, neighboring lanes, as well as pedestrians and other surroundings, have a combined influence on the driving state, making it more challenging for AVs to make decisions [15].

To solve these problems, we need to refer to how human drivers behave on certain occasions. It is known that experienced human drivers can predict the future maneuvers of the surrounding vehicles, and effectively handle complex road conditions. Since AVs and HDVs share the same road traffic system, AVs ought to understand the surrounding information and make predictions to behave the same as HDVs [16]. Meanwhile, velocity is the most fundamental property of a moving object, so if vehicle velocity prediction can be made precisely, road accidents will be greatly eliminated with the measures in advance. Therefore, predicting the velocity of vehicles and analyzing the importance of influence factors are essential for improving the safety of AVs and road traffic.

1.1. Literature review

The most traditional approaches to vehicle velocity prediction usually consider vehicle kinematic and dynamic constraints as well as environmental factors, such as yaw rate, acceleration rate, and friction coefficient of the road surface. Nonetheless, this approach can only achieve short-term motion prediction, because it cannot infer drivers' sudden behaviors or the influence of neighboring vehicles.

Since it is difficult for such conventional methods to predict the velocity accurately, there was a motivation to develop car-following models which describe the state of the vehicles over a longer period and reflect the relationship between neighboring vehicles in the traffic flow. There are many kinds of car-following models, which reproduce some traffic flow phenomena and explain some traffic flow characteristics [17–19]. For instance, the traffic flow can be simulated by the Safe Distance model represented by the research results of Gipps [20]. Azita Dabiri et al. modify the nominal Aw-Rascle (AR) traffic model with appropriate incident-related parameters and propose the Incident Traffic Flow (ITF) model [21]. However, there are still delays in current car-following models, which provides inaccuracy on some occasions [22,23]. Hence, more advanced methods are urgently needed to improve the accuracy of velocity prediction.

Over the past few decades, deep learning has been developing rapidly in the fields of computer vision, natural language processing, drug design, bioinformatics, and so on [24–27]. A large number

of researchers have proposed deep-learning-based approaches like Linear Regression (LR), Feedforward Neural Network (FNN), etc. to the transportation domain [28–33]. Specifically, Liu et al. provide a thorough review of machine-learning-based methodologies for on-demand ride-hailing services [34]. Wang et al. use four typical regression models in machine learning and a dataset from transport infrastructure workers to explore the appropriate sample size [35].

Predicting the velocity of vehicles is essentially a task of time series prediction as it varies with time, so many scholars adopt the Recurrent Neural Network (RNN) and its variant, the Long Short-Term Memory (LSTM) for the prediction. For instance, to characterize the driving environment, Kim et al. adopt the occupancy grid map and define a trajectory prediction method based on the LSTM [36]. Altché and de LaFortelle consider several surrounding vehicles and use LSTMs to predict the longitudinal velocity of the target vehicle [37]. What's more, some studies add attention mechanisms to this, such as Lin et al. proposing the STA-LSTM to explain the influence of the historical trajectories and neighboring vehicles on the target vehicle [14].

In addition, due to the complex nature of transport systems, vehicles and the surrounding environments interact with each other in a complicated way. To be specific, the staggered following behavior is common in the actual traffic flow, especially when the road condition is poor, driving discipline is not standardized, road signs are unclear, and lane widths are inconsistent [38]. There are some researchers building models to capture the basic rules of surrounding environments and demonstrate that the interdependencies of vehicles can have a slight influence on the driving state of a vehicle. For instance, Gindele et al. use factored states in prediction to model the mutual influence between vehicles [39]. Li et al. consider the influences of surrounding vehicles and use active fine lane management methods to solve the problem of congestion in intertwined areas [40]. The influences of neighboring lanes are also considered in the autonomous driving system, for instance, Yue et al. evaluate the influence of connected and autonomous vehicle merging algorithms on the driver behavior of human-driven vehicles [41].

To the best of our knowledge, some gaps in the existing studies on vehicle velocity prediction are observed and summarized as follows: (i) The majority of car-following models assume that the vehicle runs in the middle of a single lane, which is inconsistent with the actual situation. Some studies oversimplify or even ignore the influence of the vehicles in neighboring lanes, even though they have considered the staggered following phenomenon. (ii) Only the data from recent time intervals are used to predict whereas the duration of the influence is often neglected. (iii) The influence factors and the fundamental reasons for the accuracy in forecasting results should be further analyzed and quantified rather than just following the experience from other domains. These gaps are thus addressed in this paper. Moreover, with the challenges associated with the accelerating development of AVs and the Intelligent Transportation System (ITS), it is a timely topic to bridge these gaps.

1.2. Objectives and contributions

This paper aims to solve the vehicle velocity prediction problem by integrating the modeling skills of deep learning and domain knowledge in transportation. Apart from using the basic flow data of vehicles, we further characterize drivers' visual angles and visual gap angles to measure the influences of the front vehicles. Since the performance of the LSTM that most current tasks of time series forecasting apply degrades when characterizing long-term influences, we introduce the Transformer into the velocity prediction to take advantage of its powerful parallel computing

capabilities. Moreover, the duration of the influence and the importance of the influence factors are quantified in this paper to promote a better understanding the driving behaviors.

The contributions of this paper are three-fold. Firstly, a deep learning approach based on the Transformer is proposed. Herein, multiple features of the current vehicle, the front vehicle in the current lane, and the front vehicles in the neighboring lanes are characterized, some of which are based on the drivers' visual angles and the visual gap angles. Secondly, the influence of an incident on the driving state often lasts for a period, which affects driving safety to a large extent. To this end, we evaluate the duration of the influence on the driving state by integrating the features of the vehicle state at the previous timestamps. Thirdly, the ability to quantify the influence of each influence factor on vehicle velocity prediction is of great value for subsequent studies, therefore, we analyze the influence factors of the velocity prediction based on the Random Forest (RF) model.

The remainder of this paper is arranged as follows: Section 2 defines the context of the addressed problems. Modeling techniques are described in Section 3, including the Transformer, the LSTM, and the RF. In Section 4, the preprocessing of the adopted dataset is presented, and the characterized features are described. In Section 5, the experiments are carried out to verify the proposed deep learning approach along with the analysis of the results. The importance of each influence factor is also quantified in this section. Eventually, we make a conclusion in Section 6.

2. Problem statement

A common traffic dataset, the HighD dataset, is chosen for our study. The HighD dataset, published by the Institute of Automotive Engineering at the RWTH Aachen University in Germany, is a large collection of natural vehicle trajectory data on German highways. The dataset was collected from six different locations around Cologne, Germany, depending on the number of lanes and velocity limits, including vehicles and trucks. The dataset includes 11.5 hours of measurements from six locations and 110,000 vehicles, covering a total of 45,000 km of measured vehicle mileage. It also includes 5600 complete lane change records. By using advanced computer vision algorithms, the location error is typically less than 10 cm. It is suitable for driver model parameterization, automatic driving, traffic mode analysis, and other tasks.

The interval between each timestamp of the HighD dataset is 0.04 seconds. We use x_t to denote the integrated information of the current vehicle, the front vehicle in the current lane, and the front vehicles in the neighboring lanes at the tth timestamp, and use v_t to describe the velocity of the objective vehicle at the t th timestamp. Vehicle velocity prediction is evidently a time series forecasting problem as values of consecutive timestamps show temporal dependence. Thus, the problem addressed in this paper is to use the historical vehicle $\{x_{t-1}, x_{t-2}, x_{t-3}, ...\}$ to predict v_t. Since we need to use the data in previous k timestamps, we use a sliding window of size $k+1$ (i.e., previous k timestamps + label) while a step length of a 1-time slice is used to generate samples, as shown in Figure 1.

With these data, one major limitation in vehicle velocity prediction is that it is difficult to assess the influence of surrounding vehicles on the driving state. Moreover, it's challenging to accurately estimate the duration of influence and quantify the importance of the influence factors because the interdependency of the components in the transportation system is complicated. Thus, this paper aims to overcome the above-mentioned challenges through deep-learning-based approaches.

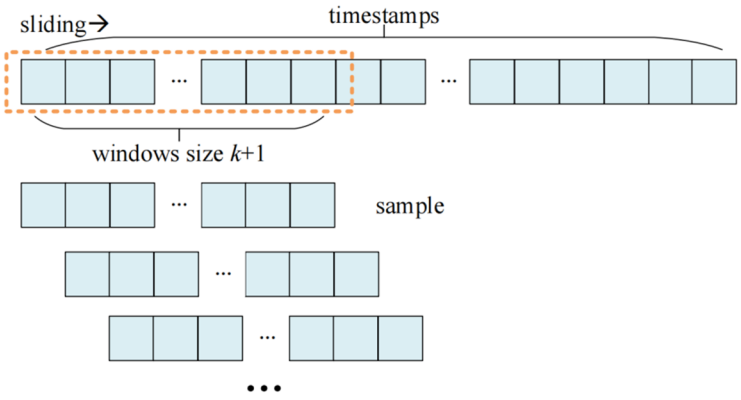


Figure 1. The generation process of samples.

3. Principles of deep learning models

In this section, the principle of the Transformer is first introduced. In contrast with the Transformer, the Long Short-Term Memory (LSTM) is also adopted in vehicle velocity prediction. Besides, to quantify the importance of the influence factors, the Random Forest (RF) is described in this section.

3.1. Transformer

The Transformer was brought to us in 2017 whose significant success in Natural Language Processing demonstrates its powerful modeling ability for temporal data [42]. In the Transformer, the entire network is composed of self-attention and Feedforward Neural Network (FNN) while traditional CNNs and RNNs are abandoned, and a trainable neural network can be built by stacking the Transformers [43]. The biggest merit of the Transformer is its powerful parallel computing capability, which avoids the degradation of performance when characterizing long-term effects [44]. The structure of the Transformer is essentially an Encoder-Decoder structure, as shown in Figure 2.

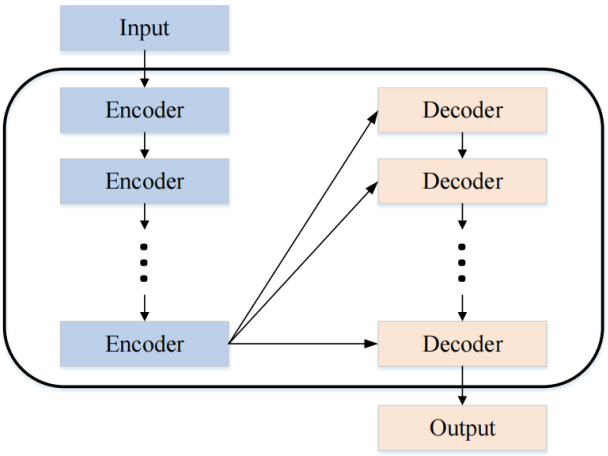


Figure 2. The structure of the Encoder-Decoder.

In the Encoder, the data will first pass through a module called "self-attention" to get a weighted feature vector Z, which can be described as

$$Z = \text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where three of the vectors Q, K, and V are the Query, Key, and Value, respectively. They are obtained by multiplying the embedding vector X by three different weight matrices W^Q, W^K, and W^V, all of which are 512 × 64 in size.

After Z is obtained, it is sent to the next module of the Encoder, which is the FNN. This full connection has two layers whose activation functions can be expressed as

$$FFN(Z) = max(0, ZW_1 + b_1)W_2 + b_2 \tag{2}$$

The difference between the Decoder and the Encoder is that the Decoder has an additional Encoder-Decoder Attention layer, which is used to calculate the weight of input and output, respectively. In this way, the structure of the Encoder and the Decoder in the Transformer can be expressed in Figure 3.

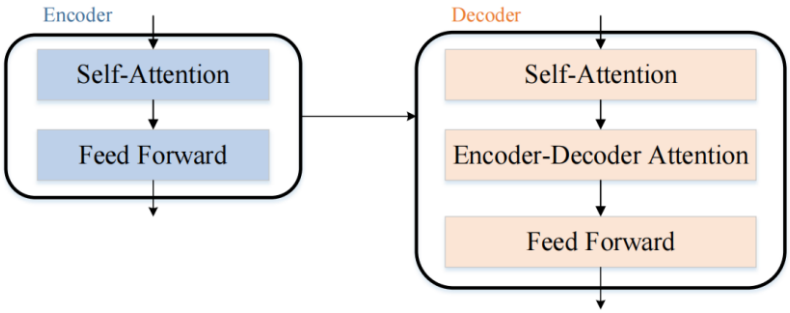
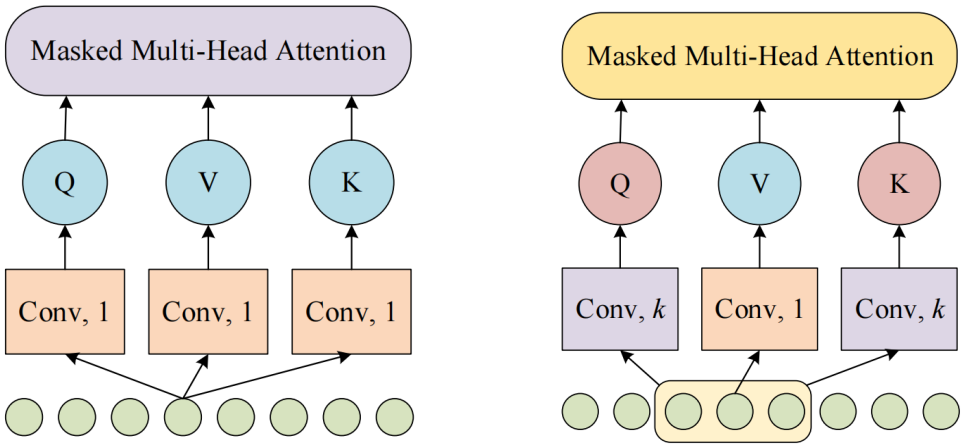


Figure 3. The structure of the Encoder-Decoder in the Transformer.

However, the calculation of Q, K, and V in the self-attention layer does not take into account its change trend, which may lead to abnormal data focus, as shown in Figure 4(a). Adopting the convolutional self-attention for improvement to calculate Q and K, the convolution kernel larger than 1 is used for convolutional operation, as shown in Figure 4(b). In this way, it enables the Attention to focus on its local change trend and allows more relevant features to be matched.

Building time series prediction based on the Transformer can break through many of the limitations of previous RNNs and LSTMs. The most obvious preponderance is that the Transformer can model both long-term and short-term characteristics based on a multi-head attention structure without much degradation when characterizing long-term effects. That is, the algorithm has a highly parallel computing capability, which is in line with the current hardware environment.

Therefore, considering the advantages above, we introduce the Transformer into our study for vehicle velocity prediction. We put the characters of different vehicles in different layers which can be called independently. To this end, it is easy to train our model with different features and compare the performance under different circumstances.



(a) Using self-attention (b) Using the convolutional self-attention

Figure 4. The calculation of Q, K, and V.

3.2. Long Short-Term Memory

Compared with the RNN model, the "GATE" structure incorporated in each neuron of the LSTM neural network is mainly used to control the information in and out of neurons to avoid the failure of the model. Figure 5 shows the structure expansion of the LSTM neural network.

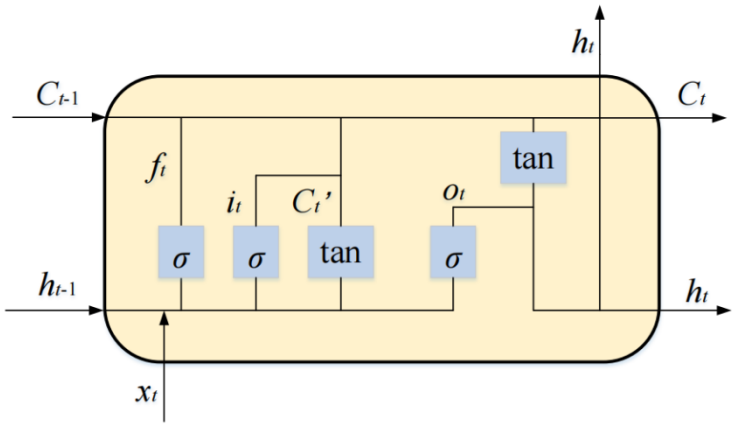


Figure 5. The structure of the LSTM.

The "GATE" structure contains three control gates that can selectively memorize and store internal data information to repeatedly modify the prediction parameters. In Figure 5, C_t, x_t, and h_t are the status, the input value, and the output value of the neuron at time t. f_t, i_t, and o_t are the forget gate, the input gate, and the output gate, respectively.

Their respective principles are as follows:

Firstly, the forget gate determines the amount of information remaining at the previous moment C_{t-1} in the current moment C_t. Its main function is to preserve the output information h_{t-1} and the

input information x_t, and to generate a value between 0 and 1 by using the function. The retention weight of the information is calculated as

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

where W_f is the weight matrix of the forget gate; $[h_{t-1}, x_t]$ represents the recombination of $h_{t-1}$and x_t as well as the connection into a large vector; and b_f is the bias term, a Sigmoid function with values between 0 and 1.

Secondly, the input gate determines the amount of input information x_t retained in C_t at the current moment t. Its main function is to save the output information h_{t-1} and the input information x_t. At the same time, it uses the function to generate a value between 0 and 1, and calculates the retention weight of information as

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

where W_i is the weight matrix of the input gate; $[h_{t-1}, x_t]$ represents the recombination of two vectors, h_{t-1} and x_t; b_i is the bias term, a Sigmoid function with values between 0 and 1. Simultaneously, the unit state at the current moment can be updated as

$$C_t' = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{5}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t' \tag{6}$$

where W_c is the parameter weight matrix representing the neuron state; $[h_{t-1}, x_t]$ represents the connection between the two vectors h_{t-1} and x_t; b_c is the bias term, $\tanh$ is the processing function, and the value of the generated result ranges from -1 to 1.

The neuron state calculated according to the current moment C_t is multiplied by the information of the input gate i_t. The neuron state at the previous moment C_{t-1} is multiplied by the forget gate information f_t, and the new unit state C_t at the current moment is obtained by adding them together. In this way, the useless information of the current moment is forgotten and some useful information from the previous moment is remembered.

Thirdly, the output gate determines the amount of neuron state C_t retained in the output value h_t at the current time t. Its main function is to save the output information h_{t-1} and input information x_t. At the same time, it uses the function to generate a value between 0 and 1, and calculates the retention weight of information as

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t \cdot tanh(C_t) \tag{8}$$

where W_o is the weight matrix of the forget gate; $[h_{t-1}, x_t]$ represents the recombination of h_{t-1} and x_t as well as the connection into a large vector; b_o is the bias term; $\tanh$ is the processing function, and the value of the generated result ranges from -1 to 1.

3.3. Random Forest

The Random Forest (RF) model is a machine learning model evolved and derived from the

Decision Tree algorithm. Its essential principle is the combinatorial idea in mathematics, that is, it integrates multiple independent decision trees and learns at the same time, and uses the voting mechanism of majority voting to vote on samples. Although the RF model is relatively new in the field of machine learning in recent years, it has been developing rapidly and widely used since its introduction.

The RF model mainly combines "Bootstrap Aggregating" and "Random Subspace Method" in the process of building decision tree sets. Figure 6 shows its modeling process.

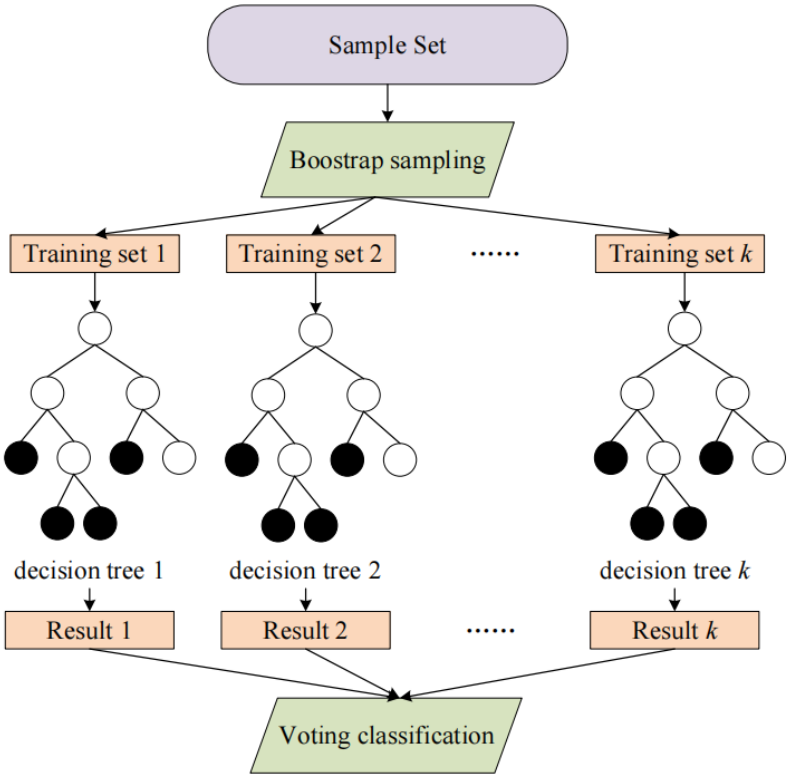


Figure 6. The modeling process of the RF model.

The specific principle of the training process of the RF model is as follows:

Firstly, set the size of the training set as N, and the RF model continuously samples the data set and extracts N samples.

Secondly, suppose that the sample has M feature dimensions, and the RF model selects m features (much smaller than M) completely randomly and without repetition, and selects the optimal m features according to the error rate during segmentation.

Thirdly, repeat the first two steps k times to get k decision trees.

Eventually, integrate the RF classifier formed by k decision trees, use the voting mechanism of the "Majority Voting" to vote on the prediction of each classification tree, and finally determine the prediction result of the Random Forest according to the principle of "Minority is Subordinate to Majority".

4. Data preprocessing and feature extraction

To complete this study, three periods of a two-way six-lane road segment numbered "track_12",

"track_13", and "track_14" are selected. The conditions of the selected road segment are shown in Figure 7, where the lanes are numbered 1–6 from top to bottom, and the three lanes on each side are the left lane (Lane 1 and Lane 6), the middle lane (Lane 2 and Lane 5), and the right lane (Lane 3 and Lane 4).

Figure 7. The conditions of the selected road segment.

Four constraints, as follows, are added to the data fragments in our experiments:

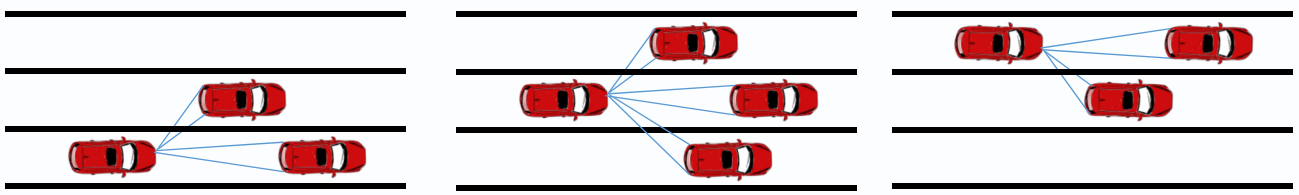
1) The vehicles selected for the study are all small cars with a velocity between 10–40 m/s.

2) The headway of the neighboring lanes (AHW), including the Left Headway (LHW) and the Right Headway (RHW), should be smaller than the headway of the current lane (HW), otherwise, the influence of the neighboring lanes will be excluded.

3) In 3 consecutive seconds, both HW and AHW should be less than 100 m, otherwise, it is deemed that there is no car-following phenomenon.

4) The influence of other factors other than vehicles is not considered.

To investigate the influences of the front vehicles, the selected scenarios of each lane are shown in Figure 8.



(a) Lane 1 and Lane 6 (b) Lane 2 and Lane 5 (c) Lane 3 and Lane 4

Figure 8. Scenarios selected in our study.

We then process our data according to the lane limit, the preceding vehicle limit, the distance limit, the following duration, and the velocity limit in turn, and the amounts of the samples obtained for each lane of each track are shown in Table 1.

While the car-following model is of great significance in traffic systems, we wish to design and introduce some features of traffic flow into our training process. We first group the rear vehicle in the current lane, the rear vehicle in the left neighboring lane, and the rear vehicle in the right neighboring lane into car-following pairs. To characterize the features of the surrounding vehicles, we refer to the Staggered Car-Following (SCF) model based on the visual angle proposed by Jin et al. [38]. The model considers the transverse separation effect and the dependency theory of the fitting field, which can perfectly describe some features of the front vehicle in the current lane and the front vehicles in the

neighboring lanes. The scenario considered in the SCF model is: the vehicle is not driving in the middle of the lane, and there is a lateral separation distance between the front and rear vehicles. Figure 9 shows the parameters of its scenario.

Table 1. The amounts of the samples.

Lane	The amounts of the samples		
	track_12	track_13	track_14
1	487	529	516
2	895	907	901
3	833	856	831
4	551	576	579
5	722	713	739
6	301	282	269

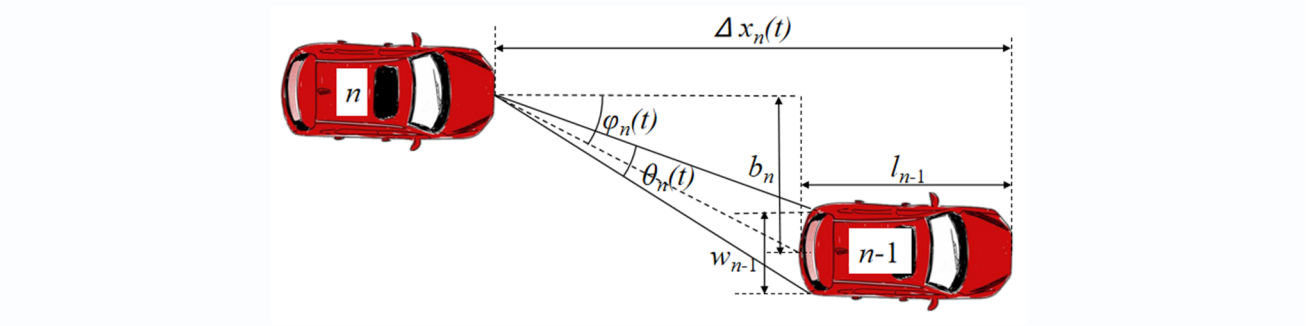


Figure 9. Parameters of researching scenarios.

In Figure 9, $\theta_n(t)$ is the visual angle which is observed by the driver of the n^{th} vehicle at time t, $\varphi_n(t)$ is the visual gap angle separating the $(n-1)^{\text{th}}$ vehicle from the moving direction of the n^{th} vehicle at time t. The expression of the SCF model is

$$a_n(t) = \alpha\{V[\theta_n(t)] - v_n(t)\} - \lambda\frac{1}{TTC_n(t)} \tag{9}$$

where $a_n(t)$ is the acceleration of the n^{th} vehicle at time t, $\{V[\theta_n(t)] - v_n(t)\}$ is a combination of the optimal velocity function and the velocity of the n^{th} vehicle at time t, α is the sensitivity coefficient of the driver to different velocities, and λ is the sensitivity coefficient of $1/TTC_n(t)$.

According to the SCF model, the influence factor of $a_n(t)$ consists of two components: $\alpha\{V[\theta_n(t)] - v_n(t)\}$ and $\lambda/TTC_n(t)$, and the former is much smaller than the latter when the acceleration change is small. Since the road segment selected for this study is a highway with almost no urgent stops and starts, in our experiments, only the leading vehicle in the current lane is considered. Therefore, the first part of the SCF model can be omitted, and the acceleration expression can be simplified as

$$a_n(t) = -\lambda\frac{1}{TTC_n(t)} \tag{10}$$

where λ is the sensitivity coefficient of $1/TTC_n(t)$ of the front vehicle.

For vehicles in neighboring lanes, this expression is still true. Since $1/TTC_n(t)$ is a parameter of the front vehicle in the current lane, $1/TTC_{ln}(t)$ and $1/TTC_{rn}(t)$ are the parameters of the front vehicles in the left neighboring lane and right neighboring lane, respectively. Based on this, we introduce $1/TTC_n(t)$, $1/TTC_{ln}(t)$, and $1/TTC_{rn}(t)$ as features characterized by the drivers' visual angles and the visual gap angles to our model. Then, we classify the features according to the relative position of vehicles. The four categories are the current vehicle, the front vehicle in the current lane, the front vehicle in the left neighboring lane, and the front vehicle in the right neighboring lane. All features utilized are listed in Table 2.

Table 2. Names of parameters corresponding to the characteristics of different lanes.

Vehicle	Feature	Lane 1 and Lane 6	Lane 2 and Lane 5	Lane 3 and Lane 4
The current vehicle	X-position	√	√	√
	Y-position	√	√	√
	acceleration	√	√	√
The front vehicle in the current lane	headway	√	√	√
	lateral distance	√	√	√
	velocity	√	√	√
	acceleration	√	√	√
	$-1/TTC_n(t)$	√	√	√
The front vehicle in the left neighboring lane	LHW	√	√	
	velocity	√	√	
	acceleration	√	√	
	$-1/TTC_{ln}(t)$	√	√	
The front vehicle in the right neighboring lane	RHW		√	√
	velocity		√	√
	acceleration		√	√
	$-1/TTC_{rn}(t)$		√	√

5. Experiments and results

As aforementioned, the data collected from the German highway are utilized in our experiments, where abnormal data were removed before constructing the final dataset, containing 11,487 samples in total. We choose 20% of them (2297 samples) as the testing set while the remaining samples as the training set. We integrate the previous 10 timestamps (i.e., $\{x_{t-1}, x_{t-2}, ..., x_{t-10}\}$) to predict the velocity of the next timestamp. It should be noted that if the number of integrated timestamps is too small, the prediction lacks accuracy as well as significance. Moreover, the performance metric used in the study is the Mean Square Error (MSE) which can be expressed as

$$MSE = \frac{1}{M}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{11}$$

where y_i represents the real traffic information; $\hat{y}_i$ is the predicted value of y_i; M is the number of vehicles [45].

Firstly, to investigate whether surrounding vehicles have influences on the driving state, vehicles in Lane 2 and Lane 5 are selected and different features are added to the layers while constructing a set of Transformer-based models, including the Transformer-C, Transformer-CF, and Transformer-CFN. As a comparative experiment, we utilize the same dataset to experiment with the model of LSTM-CFN.

Transformer-**C** indicates the model is Transformer-based and only the parameters of the **C**urrent vehicle are added. Transformer-C**F** denotes the extension of Transformer-C that further captures the **F**ront vehicle in the current lane. Transformer-CF**N** denotes the extension of Transformer-CF that further considers the front vehicles in the **N**eighboring lanes. LSTM-CFN means the model is LSTM based while features among the current vehicle, the front vehicles in the current lane, and the neighboring lanes are all added to the model.

Table 3 describes the hyperparameter settings, and Table 4 describes the adduced structure settings of our Transformer-based models.

Table 3. Hyperparameter settings of the models.

Hyperparameter	Loss	Optimizer	Batch Size	Epochs
Value	MSE	Adam	32	100

Table 4. The structure settings of the models.

Layer (type)	Output Shape	Param #
Input Layer	(None, 16)	0
Token_and_position_embedding	(None, 16, 32)	646,400
Transformer	(None, 16, 32)	10,656
Global_average_pooling_Id	(None, 32)	0
Dropout_2	(None, 32)	0
Dense_2	(None, 20)	660
Dropout_2	(None, 20)	0
Dense_2	(None, 2)	42

The performance of MSE in the validation set of each experiment is calculated in Table 5, while the obtained MSE curves are shown in Figure 10.

Table 5. The performance of MSE in the validation set.

Model	Transformer-C	Transformer-CF	Transformer-CFN	LSTM-CFN
MSE	0.0612	0.0539	0.0500	0.0611

The results indicate that the MSE of the four experiments tends to be relatively stable after 40 epochs, and there is no over-fitting or under-fitting. It can be observed that considering only the current vehicle leads to lower prediction as the MSE of the Transformer-C stables at 0.0612. Moreover, the MSE of the Transformer-CF and Transformer-CFN is 0.0539 and 0.0500, showing a decrease of 11.93% and 18.30% respectively, which demonstrates that the introduction of the features from the front vehicles in the current lane and the neighboring lanes will improve the performance. Thus, it is proved that the vehicle's driving state will be influenced by the front vehicles in the current lane as well as the

neighboring lanes.

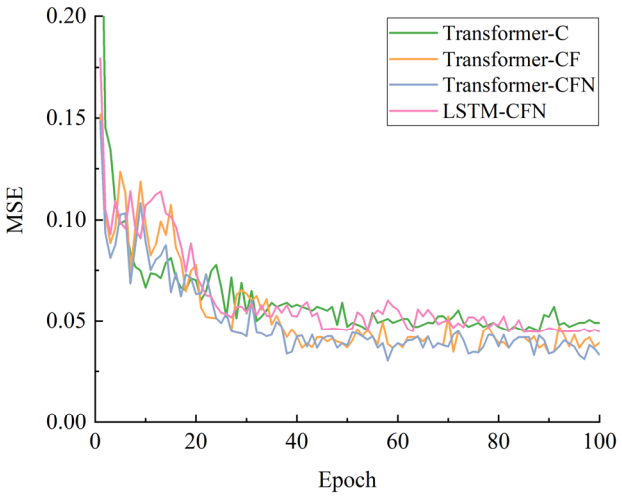


Figure 10. The MSE curves of the experiments.

The Transformer-CFN and LSTM-CFN constitute a set of comparative experiments. The MSE is 0.0500 for the Transformer-CFN and 0.0611 for the LSTM-CFN, showing a further improvement by the Transformer-based model. Therefore, it can be concluded that the Transformer is not only feasible to predict the velocity of vehicles but also better performed in our experiments compared with the LSTM.

Moreover, in the above experiments, the previous ten timestamps are used to predict the velocity of the next timestamp, which proves that the current vehicle, the front vehicle in the current lane, and the front vehicles in the neighboring lanes have a combined influence on the driving state. As is acknowledged, the influence on the vehicle state is not instantaneous but lasts a period, however, the duration of the influence is seldom calibrated in the previous study. To this end, we try to figure out the duration of the prominent influences on vehicle velocity using the aforementioned data and models.

Table 6. The MSE values by integrating multiple timestamps.

Timestamps	MSE	Timestamps	MSE	Timestamps	MSE	Timestamps	MSE
5	0.0517	30	0.0491	55	0.0503	80	0.0498
10	0.0500	35	0.0489	60	0.0499	85	0.0509
15	0.0503	40	0.0492	65	0.0505	90	0.0513
20	0.0499	45	0.0500	70	0.0503	95	0.0502
25	0.0496	50	0.0494	75	0.0508	100	0.0499

In the selected dataset, the interval for each timestamp is 0.04 seconds. We integrate the information of multiple previous timestamps in increments of five timestamps (0.2 s). Consistent with the previous experiments, we consider the current vehicle, the front vehicle in the current lane, the front vehicles in the neighboring lanes, and the mean value of MSE in each of the 100 epochs as the evaluation index. The values of MSE generated by integrating multiple timestamps are shown in Table 6 and Figure 11.

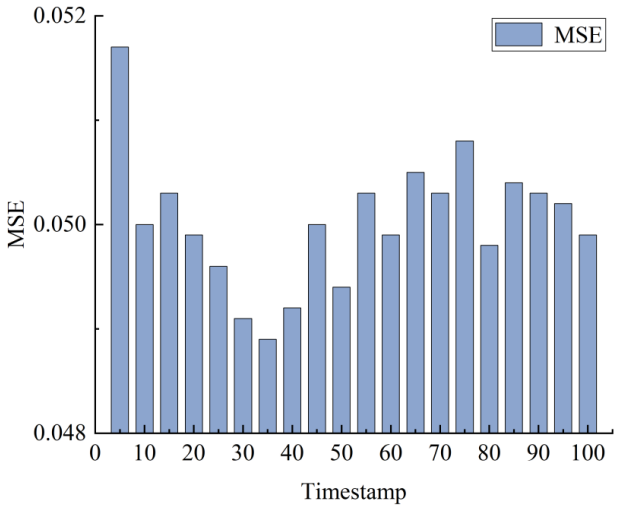


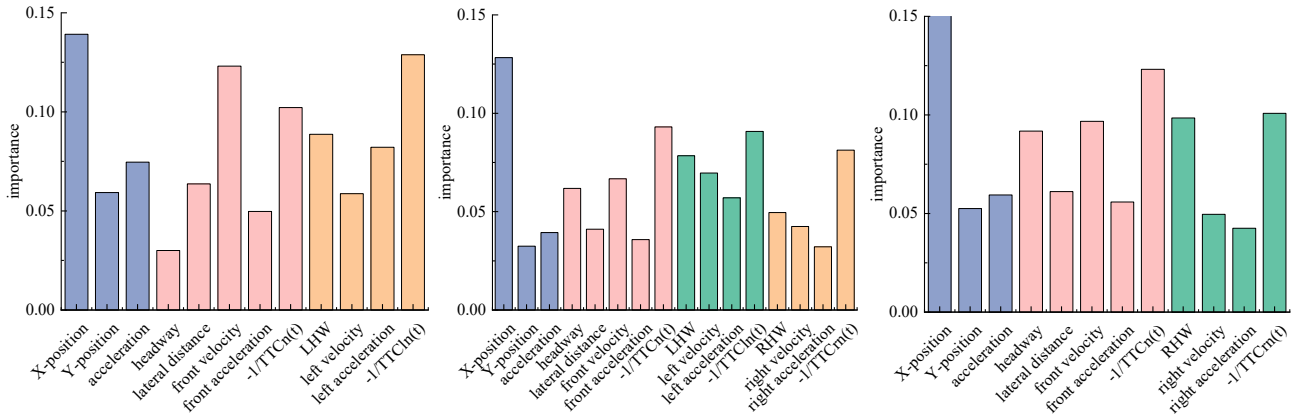
Figure 11. The MSE generated by integrating multiple timestamps.

As we can see in Figure 11, the MSE first decreases as the integration timestamps increase, and then reaches the lowest point at about 35 timestamps (1.4 s). Then, the MSE rises to about 0.051 and reaches a lower peak. Finally, it remains at a relatively stable level at 0.050 or so. Based on this, it is reasonable to conclude that the duration of the influence on a driving vehicle is about 1.4 seconds. According to Brunson et al., the reaction time of drivers in normal conditions approximates a normal distribution, and the fitting mean of the distribution is 1.387 seconds [46]. Thus, the result can be perfectly explained by the fact that the influence on a driving vehicle lasts about 1.4 seconds, which is the driver's reaction time when there are some maneuvers of the surrounding vehicles.

Furthermore, the ability to quantify the influence of each influence factor on the velocity prediction is of great value for the safety of the transportation systems, which is another critical focus of our study. When using machine-learning-based models to predict the velocity, models such as the LightGBM, Random Forest (RF), etc. can be used to quantify the influence of each parameter on the predicted value [47]. Among them, the RF model adopted here quantifies the importance of each parameter by calculating the number of times that each feature is used in the splitting process of all internal decision trees. Thus, we utilize the RF model to investigate the difference in the significance of features of different lanes when predicting the velocity.

We divide the dataset into three groups according to the lanes, that is, Lane 1 and Lane 6, Lane 2 and Lane 5, and Lane 3 and Lane 4, respectively. In each group, all the corresponding features in Table 2 are adopted in the training process, and the importance of each feature on the vehicle velocity prediction is obtained in Figure 12. Then, we calculate the proportions of each sort of feature in Table 7.

Table 7 indicates that no matter which lane the vehicle is driven in, the influence proportion of the front vehicle in the current lane is always bigger than that of the other lanes, which means the velocity of the vehicle is mainly influenced by the front vehicle in the current lane. Moreover, compared with the result of Lane 3 and Lane 4 (29.13%), the influence proportion of the neighboring lane in Lane 1 and Lane 6 is larger, which is 35.83%. Similarly, from the result of Lane 2 and Lane 5, the influence proportion of the left lane (29.58%) is larger than that of the right lane (20.55%), which indicates the velocity of vehicles is affected more by the left lane than the right lane.



(a) Lane 1 and Lane 6 (b) Lane 2 and Lane 5 (c) Lane 3 and Lane 4

Figure 12. the importance of each feature (blue-the current vehicle; pink-the front vehicle in the current lane; green-the front vehicle in the left neighboring lane; yellow-the front vehicle in the right neighboring lane).

Table 7. The proportions of each sort of feature.

Vehicle	Lane 1 and Lane 6	Lane 2 and Lane 5	Lane 3 and Lane 4
The current vehicle	27.31%	20.02%	28.02%
The front vehicle in the current lane	36.86%	29.85%	42.85%
The front vehicle in the left neighboring lane	35.83%	29.58%	--
The front vehicle in the right neighboring lane	--	20.55%	29.13%

To explain the results, on one hand, it is clear that the front vehicle in the current lane directly influences the speed of the rear vehicle, so it accounts for the largest proportion, for instance, when the front vehicle brakes, the rear vehicle must slow down to avoid the collision. On the other hand, the left lane is generally the fast lane, while the right lane is the slow lane. According to their habits, drivers usually overtake on the left, and the conditions in the right lane have little influence on relative behaviors. Plus, the traffic rules in some countries and regions advise drivers the overtake from the left, and some areas even prohibit drivers from passing on the right. Taking the above reasons into account, drivers will pay more attention to the left lane during driving intuitively. Therefore, we can safely conclude that the velocity of vehicles is influenced to a greater extent by the front vehicle in the left lane than that in the right lane, but is still primarily influenced by the front vehicle in the current lane.

6. Conclusions

This paper proposed a Transformer-based model integrating the features of previous timestamps to predict the velocity of vehicles. Herein, multiple features of the current vehicle, the front vehicle in

the current lane, and the front vehicles in the neighboring lanes are characterized, some of which are based on the drivers' visual angles and the visual gap angles. Firstly, this paper confirms that the introduction of the features from the front vehicles in the current lane and the neighboring lanes will improve the performance by 11.93 and 18.30% respectively, which means the surrounding vehicles do have influences on the vehicle's driving state. Secondly, a comparative experiment shows that the Transformer is feasible and better performed for the vehicle velocity prediction compared with the LSTM. Thirdly, this study indicates that the duration of the influence on the driving state calculated by integrating the features at the previous timestamps is about 1.4 seconds, which is exactly in line with the driver's reaction time when some movements occur in the surrounding vehicles. In addition, with the RF model, we analyze the importance of the influence factors and conclude that the velocity of vehicles is more influenced by the left lane than the right lane, but is still mainly influenced by the current lane.

The ability to quantify the influence of each feature on vehicle velocity prediction is of great value for subsequent studies. As future work, it is worthwhile to further investigate the essential causes for the effectiveness of a particular feature and to enable machine learning better comprehend the characteristics, instead of taking the algorithm entirely as a black box. Therein, we can better understand the human-attention-based driving behaviors to enhance the predictive ability of the AVs, which will greatly improve the safety of the Intelligent Transportation System.

Acknowledgments

This study was supported by the Key Project of National Natural Science Foundation of China under 52220105001 and sponsored by Tsinghua-Toyota Joint Research Fund. This study is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101025896.

Conflict of interest

The authors declare there are no conflict of interest.

References

1. J. V. Brummelen, M. O'Brien, D. Gruyer, H. Najjaran, Autonomous vehicle perception: The technology of today and tomorrow, *Transp. Res. Part C Emerging Technol.*, **89** (2018), 384–406. https://doi.org/10.1016/j.trc.2018.02.012
2. K. F. Yuen, L. Cai, G. Qi, X. Wang, Factors influencing autonomous vehicle adoption: An application of the technology acceptance model and innovation diffusion theory, *Technol. Anal. Strategic Manage.*, **33** (2021), 505–519. https://doi.org/10.1080/09537325.2020.1826423
3. A. Talebian, S. Mishra, Predicting the adoption of connected autonomous vehicles: A new approach based on the theory of diffusion of innovations, *Transp. Res. Part C Emerging Technol.*, **95** (2018), 363–380. https://doi.org/10.1016/j.trc.2018.06.005
4. T. Morita, S. Managi, Autonomous vehicles: Willingness to pay and the social dilemma, *Transp. Res. Part C Emerging Technol.*, **119** (2020), 102748. https://doi.org/10.1016/j.trc.2020.102748

5. X. Xu, C. K. Fan, Autonomous vehicles, risk perceptions and insurance demand: An individual survey in China, *Transp. Res. Part A Policy Pract.*, **124** (2019), 549–556. https://doi.org/10.1016/j.tra.2018.04.009
6. T. Stoiber, I. Schubert, R. Hoerler, P. Burger, Will consumers prefer shared and pooled-use autonomous vehicles? A stated choice experiment with Swiss households, *Transp. Res. Part D Transp. Environ.*, **71** (2019), 265–282. https://doi.org/10.1016/j.trd.2018.12.019
7. J. D. Ortúzar, Future transportation: Sustainability, complexity and individualization of choices, *Commun. Transp. Res.*, **1** (2021), 100010. https://doi.org/10.1016/j.commtr.2021.100010
8. A. Vahidi, A. Sciarretta, Energy saving potentials of connected and automated vehicles, *Transp. Res. Part C Emerging Technol.*, **95** (2018), 822–843. https://doi.org/10.1016/j.trc.2018.09.001
9. K. M. Gurumurthy, K. M. Kockelman, Analyzing the dynamic ride-sharing potential for shared autonomous vehicle fleets using cellphone data from Orlando, Florida, *Comput. Environ. Urban Syst.*, **71** (2018), 177–185. https://doi.org/10.1016/j.compenvurbsys.2018.05.008
10. M. A. Moore, P. S. Lavieri, F. F. Dias, C. R. Bhat, On investigating the potential effects of private autonomous vehicle use on home/work relocations and commute times, *Transp. Res. Part C Emerging Technol.*, **110** (2020), 166–185. https://doi.org/10.1016/j.trc.2019.11.013
11. Y. Xu, Z. Ye, C. Wang, Modeling commercial vehicle drivers' acceptance of advanced driving assistance system (ADAS), *J. Intell. Connected Veh.*, 2021. https://doi.org/10.1108/JICV-07-2021-0011
12. C. Lu, C. Liu, Ecological control strategy for cooperative autonomous vehicle in mixed traffic considering linear stability, *J. Intell. Connected Veh.*, 2021. https://doi.org/10.1108/JICV-08-2021-0012
13. W. Xue, R. Zheng, B. Yang, et al., An adaptive model predictive approach for automated vehicle control in fallback procedure based on virtual vehicle scheme, *J. Intell. Connected Veh.*, **2** (2019), 67–77. https://doi.org/10.1108/JICV-06-2019-0007
14. L. Lin, W. Li, H. Bi, L. Qin, Vehicle trajectory prediction using lstms with spatial-temporal attention mechanisms, *IEEE Intell. Transp. Syst. Mag.*, **14** (2021), 197–208. https://doi.org/10.1109/MITS.2021.3049404
15. S. Rezaei, R. Sengupta, H. Krishnan, X. Guan, R. Bhatia, Tracking the position of neighboring vehicles using wireless communications, *Transp. Res. Part C Emerging Technol.*, **18** (2010), 335–350. https://doi.org/10.1016/j.trc.2009.05.010
16. A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, A. Alahi, Social gan: Socially acceptable trajectories with generative adversarial networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 2255–2264. https://doi.org/10.48550/arXiv.1803.10892
17. Q. Cheng, Z. Liu, Y. Lin, X. S. Zhou, An s-shaped three-parameter (S3) traffic stream model with consistent car following relationship, *Transp. Res. Part B Methodol.*, **153** (2021), 246–271. https://doi.org/10.1016/j.trb.2021.09.004
18. Q. Cheng, Z. Liu, J. Guo, X. Wu, R. Pendyala, B. Belezamo, et al., Estimating key traffic state parameters through parsimonious spatial queue models, *Transp. Res. Part C Emerging Technol.*, **137** (2022), 103596. https://doi.org/10.1016/j.trc.2022.103596
19. X. S. Zhou, Q. Cheng, X. Wu, P. Li, B. Belezamo, J. Lu, et al., A meso-to-macro cross-resolution performance approach for connecting polynomial arrival queue model to volume-delay function with inflow demand-to-capacity ratio, *Multimodal Transp.*, **1** (2022), 100017. https://doi.org/10.1016/j.multra.2022.100017

20. P. G. Gipps, A behavioural car-following model for computer simulation, *Transp. Res. Part B Methodol.*, **15** (1981), 105–111. https://doi.org/10.1016/0191-2615(81)90037-0
21. A. Dabiri, B. Kulcsár, Incident indicators for freeway traffic flow models, *Commun. Transp. Res.*, **2** (2022), 100060. https://doi.org/10.1016/j.commtr.2022.100060
22. V. Punzo, B. Ciuffo, M. Montanino, Can results of car-following model calibration based on trajectory data be trusted?, *Transp. Res. Rec.*, **2315** (2012), 11–24. https://doi.org/10.3141/2315-02
23. V. Papathanasopoulou, C. Antoniou, Towards data-driven car-following models, *Transp. Res. Part C Emerging Technol.*, **55** (2015), 496–509. https://doi.org/10.1016/j.trc.2015.02.016
24. T. B. Shahi, C. Xu, A. Neupane, W. Guo, Machine learning methods for precision agriculture with UAV imagery: a review, *Electron. Res. Arch.*, **30** (2022), 4277–4317. https://doi.org/10.3934/era.2022218
25. I. Ahmed, S. Din, G. Jeon, F. Piccialli, G. Fortino, Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning, *IEEE/CAA J. Autom. Sin.*, **8** (2021), 1253–1270. https://doi.org/10.1109/JAS.2020.1003453
26. D. G. Hong, W. H. Han, C. H. Yim, Tapping stream tracking model using computer vision and deep learning to minimize slag carry-over in basic oxygen furnace, *Electron. Res. Arch.*, **30** (2022), 4015–4037. https://doi.org/10.3934/era.2022204
27. T. Lintonen, T. Räty, Self-learning of multivariate time series using perceptually important points, *IEEE/CAA J. Autom. Sin.*, **6** (2019), 1318–1331. https://doi.org/10.1109/JAS.2019.1911777
28. Y. Liu, C. Lyu, Y. Zhang, Z. Liu, W. Yu, X. Qu, DeepTSP: Deep traffic state prediction model based on large-scale empirical data, *Commun. Transp. Res.*, **1** (2021), 100012. https://doi.org/10.1016/j.commtr.2021.100012
29. P. M. Kebria, A. Khosravi, S. M. Salaken, S. Nahavandi, Deep imitation learning for autonomous vehicles based on convolutional neural networks, *IEEE/CAA J. Autom. Sin.*, **7** (2019), 82–95. https://doi.org/10.1109/JAS.2019.1911825
30. Y. Liu, F. Wu, C. Lyu, S. Li, J. Ye, X. Qu, Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform, *Transp. Res. Part E Logist. Transp. Rev.*, **161** (2022), 102694. https://doi.org/10.1016/j.tre.2022.102694
31. S. Li, Y. Liu, X. Qu, Model controlled prediction: A reciprocal alternative of model predictive control, *IEEE/CAA J. Autom. Sin.*, **9** (2022), 1107–1110. https://doi.org/10.1109/JAS.2022.105611
32. H. Wang, W. Yi, Y. Liu, An innovative approach of determining the sample data size for machine learning models: a case study on health and safety management for infrastructure workers, *Electron. Res. Arch.*, **30** (2022), 3452–3462. https://doi.org/10.3934/era.2022176
33. F. C. Pereira, F. Rodrigues, M. Ben-Akiva, Using data from the web to predict public transport arrivals under special events scenarios, *J. Intell. Transp. Syst.*, **19** (2015), 273–288. https://doi.org/10.1080/15472450.2013.868284
34. Y. Liu, R. Jia, J. Ye, X. Qu, How machine learning informs ride-hailing services: A survey, *Commun. Transp. Res.*, **2** (2022), 100075. https://doi.org/10.1016/j.commtr.2022.100075
35. H. Wang, W. Yi, Y. Liu, An innovative approach of determining the sample data size for machine learning models: a case study on health and safety management for infrastructure workers, *Electron. Res. Arch.*, **30** (2022), 3452–3462. https://doi.org/10.3934/era.2022176

36. B. D. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, J. W. Choi, Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network, in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, (2017), 399–404. https://doi.org/10.1109/ITSC.2017.8317943
37. F. Altché, A. L. Fortelle, An LSTM network for highway trajectory prediction, in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, IEEE, (2017), 353–359. https://doi.org/10.1109/ITSC.2017.8317913
38. S. Jin, D. Wang, C. Xu, Z. Huang, Staggered car-following induced by lateral separation effects in traffic flow, *Phys. Lett. A*, **376** (2012), 153–157. https://doi.org/10.1016/j.physleta.2011.11.005
39. T. Gindele, S. Brechtel, R. Dillmann, A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments, in *13th International IEEE Conference on Intelligent Transportation Systems*, IEEE, (2010), 1625–1631. https://doi.org/10.1109/ITSC.2010.5625262
40. H. Li, J. Zhang, Z. Zhang, Z Huang, Active lane management for intelligent connected vehicles in weaving areas of urban expressway, *J. Intell. Connected Veh.*, 2021. https://doi.org/10.1108/JICV-08-2020-0009
41. L. Yue, M. Abdel-Aty, Z. Wang, Effects of connected and autonomous vehicle merging behavior on mainline human-driven vehicle, *J. Intell. Connected Veh.*, 2021. https://doi.org/10.1108/JICV-08-2021-0013
42. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, *Adv. Neural Inf. Process. Syst.*, **30** (2017).
43. A. Zeyer, P. Bahar, K. Irie, R. Schlüter; H. Ney, A comparison of transformer and lstm encoder decoder models for asr, in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, (2019), 8–15. https://doi.org/10.1109/ASRU46091.2019.9004025
44. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, preprint, arXiv:1810.04805. https://doi.org/10.48550/arXiv.1810.04805
45. L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, et al., T-gcn: A temporal graph convolutional network for traffic prediction, *IEEE Trans. Intell. Transp. Syst.*, **21** (2019), 3848–3858. https://doi.org/10.1109/TITS.2019.2935152
46. S. J. Brunson, E. M. Kyle, N. C. Phamdo, G. R. Preziotti, *Alert algorithm development program: NHTSA rear-end collision alert algorithm*, (2002), No. HS-809526.
47. Y. Liu, F. Wu, C. Lyu, X. Liu, Z. Liu, Behavior2vector: Embedding users' personalized travel behavior to vector, *IEEE Trans. Intell. Transp. Syst.*, **23** (2021), 8346–8355. https://doi.org/10.1109/TITS.2021.3078229

AIMS Press