sustainability

Article

Car-Following Modeling Incorporating Driving Memory Based on Autoencoder and Long Short-Term Memory Neural Networks

Pengcheng Fan [1], Jingqiu Guo [1,*], Haifeng Zhao [2], Jasper S. Wijnands [2] and Yibing Wang [3,*]

[1] Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University, Shanghai 201804, China; fpcfw2018@tongji.edu.cn

[2] Transport, Health and Urban Design Research Hub, Melbourne School of Design, The University of Melbourne, Melbourne 3052, Australia; haifeng.zhao@unimelb.edu.au (H.Z.); jasper.wijnands@unimelb.edu.au (J.S.W.)

[3] Institute of Intelligent Transportation Systems, Zhejiang University, Hangzhou 310058, China

* Correspondence: guojingqiu@hotmail.com (J.G.); wangyibing@zju.edu.cn (Y.W.)

Received: 12 September 2019; Accepted: 29 October 2019; Published: 28 November 2019

Abstract: Although a lot of work has been conducted on car-following modeling, model calibration and validation are still a great challenge, especially in the era of autonomous driving. Most challengingly, besides the immediate benefit incurred with a car-following action, a smart vehicle needs to learn to evaluate the long-term benefits and become foresighted in conducting car-following behaviors. Driving memory, which plays a significant role in car-following, has seldom been considered in current models. This paper focuses on the impact of driving memory on car-following behavior, particularly, historical driving memory represents certain types of driving regimes and drivers' maneuver in coordination with the variety of driving regimes. An autoencoder was used to extract the main features underlying the time-series data in historical driving memory. Long short-term memory (LSTM) neural network model has been employed to investigate the relationship between driving memory and car-following behavior. The results show that velocity, relative velocity, instant perception time (IPT), and time gap are the most relevant parameters, while distance gap is insignificant. Furthermore, we compared the accuracy and robustness of three patterns including various driving memory information and span levels. This study contributes to bridging the gap between historical driving memory and car-following behavior modeling. The developed LSTM methodology has the potential to provide personalized warnings of dangerous car-following distance over the next second.

Keywords: car-following; road safety; LSTM; autoencoder; IPT; driving memory

1. Introduction

Traffic safety is facing more challenges with the fast development of our society [1]. The heterogeneity of driving behavior happening in the adjacent spatio-temporal space makes traffic much more uncertain and variable. Car-following (CF) and lane-changing behaviors can partly account for observed traffic phenomena, such as traffic oscillation and capacity drop, and they are highly associated with the traffic safety risks [2,3]. In the past few years, many studies focused on modeling car-following and lane-changing behaviors to investigate the underlying mechanism of traffic phenomena in detail, e.g., the Gazis-Herman-Rothery (GHR) model, full velocity difference model [4] and intelligent driver model (IDM) [5]. Most models capture traffic characteristics and drivers' car-following behaviors well if correctly calibrated, but limitations remain. On the other hand, machine learning techniques, especially deep learning (DL) models using artificial neural networks, have made great achievements when solving classification, regression, and forecasting problems [6]. Many data-driven car-following models use machine learning techniques such as random forests, gated recurrent unit (GRU) neural networks, and reinforcement learning (RL), to predict driver actions given a specific traffic environment.

Driving memory, defined as the historical vehicle-pair information between perceiving stimuli and predicting time, is closely associated with drivers' subsequent behaviors, but has been neglected in most models. Most models treat the drivers as machines reacting to stimuli from surrounding vehicles and scenarios; they assume that drivers keep receiving and memorizing traffic information but maneuver based on instantaneous stimuli without considering such historical traffic information. In reality, the historical traffic information (driving memory) does affect drivers' decision making but has seldom been considered in the past decades [7].

Driving memory contains more information than instantaneous stimuli (from the leading vehicle) for drivers' decision making. This paper aims at bridging the gap between short-term driving memory and car-following behaviors using DL methods. We extracted the main features underlying the time-series data in historical driving memory. A long short-term memory (LSTM) recurrent neural network was employed to model car-following behaviors and investigate the relationship with driving memory. The results show that velocity, relative velocity, instant perception time (IPT) and time gap are the most relevant parameters, while the distance gap is insignificant. Furthermore, we compared three modeling approaches (i.e., patterns) including various driving memory information and span levels in accuracy and robustness. The contributions of this paper are as follows:

1. We developed a car-following model highlighting driving memory and its impact on driving behavior;
2. It identified the significance of different parameters underlying the time-series data in historical driving memory for car-following modeling;
3. This paper also investigated three prediction patterns including various driving memory information and span levels in car-following models to predict car-following behaviors.

The remaining parts of this paper are organized as follows. Section 2 provides a review of car-following models. Section 3 describes the Next Generation Simulation (NGSIM) I-80 datasets used in this study and its pre-processing procedures. In Section 4, a multi-layer sparse autoencoder has been used to identify the most relevant parameters underlying historical driving memory. An LSTM-based car-following model has been developed in Section 5, and Section 6 concludes this study.

2. Literature Review

2.1. Mathematical CF Models

Mathematical models generally consider CF behaviors from either an engineering or a human factors perspective [5]. Some engineering models (i.e., classical stimulus-based models) assume that the acceleration of a following vehicle is associated with its relative distance and relative speed to the leading vehicle. Such models require calibration and include the GHR model [5], full velocity difference [4], Wiedemann's 99 [5,8], IDM, and Gipps models [9].

The Gipps model is a safety distance gap model built on the assumption that following vehicles can stop in case the leading vehicle brakes suddenly. Calibration of the Gipps model is simple as the number of parameters is small and their ranges are known. Newell [10] assumed that on homogeneous highways the space between vehicles should remain nearly constant given the velocity and the drivers. This is consistent with Lighthill-Whitham-Richards theory, although reaction time has not been considered.

2.2. Data-Driven CF Models

Data-driven methods using techniques such as artificial neural networks (ANNs) and fuzzy logic are outperforming the mathematical models described above, and are more accurate in describing behavior characteristics and reproducing traffic phenomena.

Khodayari et al. [11] used an ANN to develop a modified CF model with unfixed instantaneous reaction delay determined by the relative distance to and acceleration rate of the leading vehicle, which showed high accuracy. Chong et al. [12] proposed a model using an agent-based back-propagation neural network, which outperformed the GHR model significantly and can accurately capture 95% of the CF behaviors from driving trajectories. Wei and Liu [13] proposed a self-learning support vector regression to study the asymmetric characteristics in car-following behaviors and reproduced microscopic hysteresis in stop-and-go traffic effectively. Wang et al. [2] developed a car-following model that can better describe freeway driving behaviors based on an adaptive neuro-fuzzy inference system and wavelet analysis for denoising. Zhu et al. [14] proposed a framework for a human-like autonomous car-following model based on the deep deterministic policy gradient algorithm of RL, which showed good capability of generalization under various driving situations. It can be driver-adapted by successive learning and is outperforming most of existing data-driven models. A limitation of these models is that only instantaneous variables are considered.

2.3. CF Models Considering Historical Driving Memory

The models above require no historical time-sequence information, contradicting that drivers' maneuvers are based on preceding actions. Few studies have incorporated the driving memory.

Lee [7] first introduced the driving memory named 'weighting memory' function into the GHR model. He analyzed the stability of the leading vehicle's velocity, but the output acceleration showed unrealistic peaks. Recurrent neural networks (RNNs) incorporating historical information was used to model the CF behaviors to predict different types of traffic oscillation, but the driving memory has not been explicitly expressed [3]. Asymmetric CF behavior characteristics were effectively captured based on LSTM which outputs velocity and distance gap, and driving memory was emphasized and incorporated in different time scales [15]. Wang et al. [16] applied GRU neural networks to study the relationship between long memory effect and hysteresis phenomena in congested freeway traffic, inferring that long memory is important and should be embedded in CF models. The importance of historical driving memory in CF modeling has also been acknowledged in some other studies [3]. One limitation shared by all these studies is that they paid much attention to the choice of time scale of driving memory but ignored the timeliness and redundancy of the memory information from the inputs.

3. Data Description and Pre-Processing

The Next Generation Simulation (NGSIM) is a set of high-quality traffic datasets with real-world vehicle trajectory data for microsimulation. The total of 45-min data include the periods of the transition between uncongested and congested conditions, and full congestion during the peak period. This vehicle trajectory dataset provided the precise location of each vehicle every 0.1 s, resulting in detailed lane positions and locations relative to other vehicles. This study used the data collected from eastbound I–80 freeway in the San Francisco Bay area in Emeryville, CA, on 13 April 2005. The study area is approximately 500 m (1640 feet) in length and consists of six freeway lanes, including a high-occupancy vehicle lane on the far left side [3,17]. The data collection frequency was 10 Hz.

The original dataset cannot be used without denoising, otherwise it may lead to biased or even wrong results as indicated by previous research [13,15]. Using unsmoothed data, both the autoencoder model and LSTM model will yield poor performance, and it will take more time for the training process to converge. Symmetric exponential moving average filter (sEMA) was proposed by Thiemann et al. [18] to denoise the data including longitudinal location, instantaneous velocity, and acceleration of the vehicles. Then, the velocity and acceleration data were smoothed using first-order and second-order finite difference of the original location, respectively (see Figure 1 for an example).

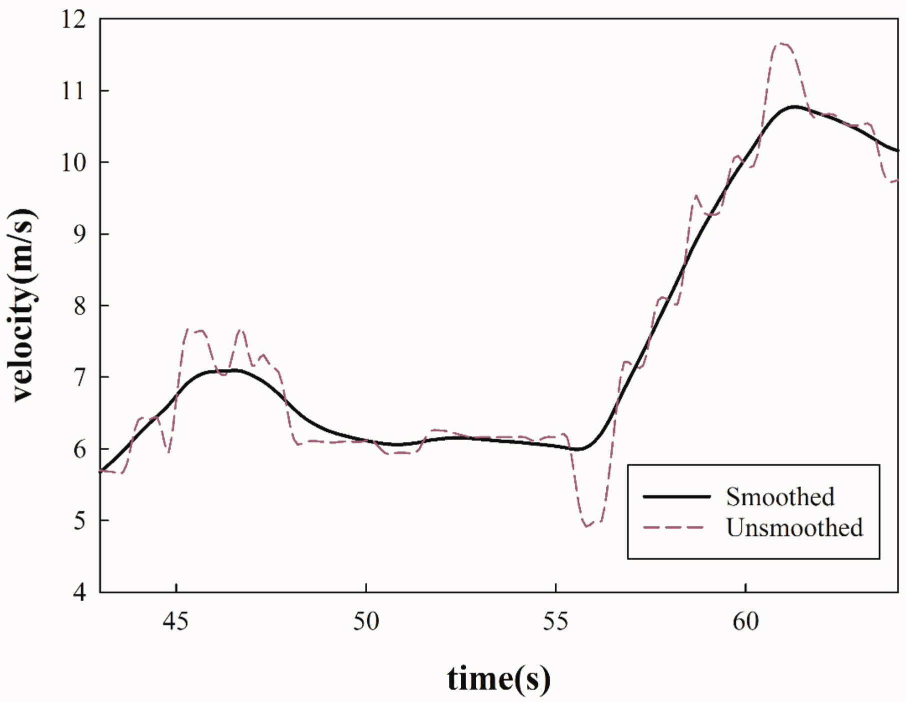


Figure 1. Comparison of smoothed and unsmoothed data for velocity.

We removed records from the dataset where any of the following applied:

- A fake collision (e.g., if the relative distance was too small) or incorrect location.
- Lane changing of the following vehicle during the first or last 1.5 s of an identified trajectory (i.e., this does not reflect CF regimes).
- Identified trajectory lasted less than 1.0 s.
- The leading or following vehicle was a truck or a motorcycle.

4. Method

4.1. Extracting Features Using Autoencoder

In most CF models, velocity, distance headway and relative velocity are the primary variables for explaining the following vehicle's acceleration. Time-to-collision has also been incorporated in some CF models [19,20]. However, few studies have investigated these variables' significance to drivers' decision making, especially in a specific time window. We propose that driving regimes were determined by the short-term (e.g., 2.0 s) driving memory. The first task was to identify the main features in the historical driving memory using time-series data. The regimes in CF behaviors were identified based on these instantaneous regime variables [21,22] without taking historical information into consideration. We defined the driving memory as the historical information contributing to decision-making, which contains multidimensional features and is more than just psychological activities.

An autoencoder is a type of deep learning architecture which can be used for data compression and feature extraction [23]. It learns to reconstruct the original data following Equation (1):

$$\hat{x} = f_{w,b}(x) \approx x \tag{1}$$

where x and $\hat{x}$ are the input and output vectors, while w and b represent the weights and biases in autoencoder f. In general, an autoencoder is a symmetric neural network which consists of an encoder and a decoder (see Figure 2). The encoder creates a compact representation of the input data while the decoder expands this representation to its original dimensions. A calibration process modifies parameters w and b to minimize the differences between the autoencoder's final layer and the input data. In case that massive high-dimensional data needs to be processed, the autoencoder can generally outperform the principle component analysis approach remarkably due to its nonlinear activation functions and multilayered structure [20–22].

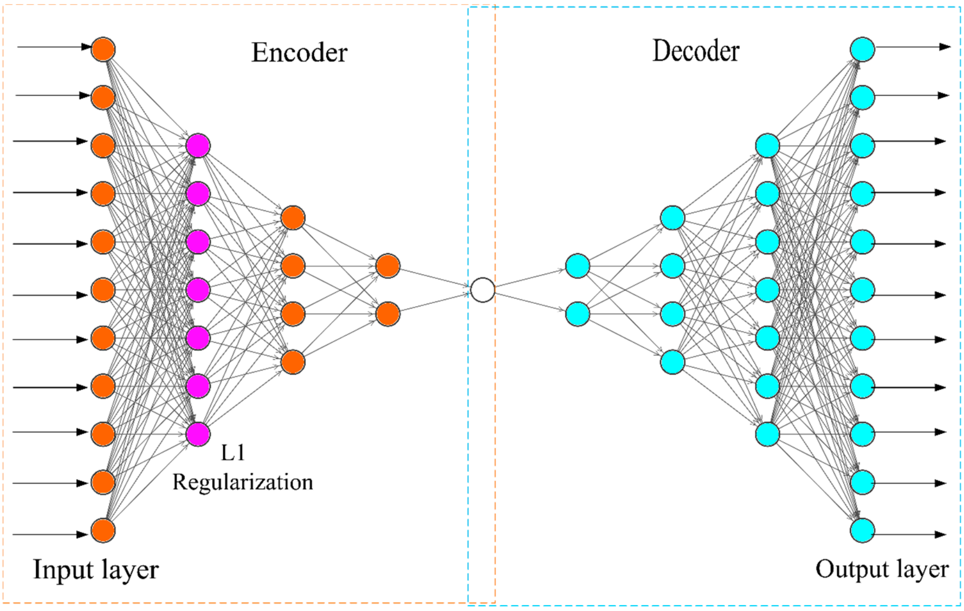


Figure 2. The sparse autoencoder structure used in this study.

Sparse autoencoders are autoencoders that add a sparsity penalty to the loss function L. Both L1 regularization term (Equation (4)) and Kullback-Leibler divergence (Equation (3)) can perform the sparseness constraint on the hidden neural units to enforce the model's sparsity. Optional loss functions are as follows:

$$L_{MAE} = \frac{1}{m}\sum\nolimits_{i=1}^{m}\left|\hat{y}_i - y_i\right| \tag{2}$$

$$L = \frac{1}{m}\sum\nolimits_{i=1}^{m}\left(\hat{y}_i - y_i\right)^2 + \beta\sum\nolimits_{j=1}^{L_2}\left[\rho\bullet\log\frac{\rho}{\overline{\rho}_{\mathrm{j}}} + (1-\rho)\log\frac{1-\rho}{1-\overline{\rho}_j}\right] \tag{3}$$

$$L = \frac{1}{m}\sum\nolimits_{i=1}^{m}\left|\hat{y}_i - y_i\right| + \alpha\sum\nolimits_{j=1}^{S_1}\sum\nolimits_{k=1}^{S_2}\left|w_{j\mathrm{k}}\right| \tag{4}$$

where $\hat{y}_i$ and y_i are the output value and the original value of the input vector i; m is the number of training samples; ρ denotes the sparsity target, namely, the average activation degree of neurons in the first hidden layer whose value is usually close to zero, and $\overline{\rho}_j$ is the actual activation level value of the jth neuron; α and β are hyperparameters to scale the sparsity penalty terms. S_1 and S_2 are the number of neurons in the input layer and the first hidden layer.

4.2. Testing on NGSIM Datasets

The autoencoder models have been employed to study the driving behaviors, which require massive volume and multidimensional data [24–27]. Table 1 shows the parameters extracted from the trajectories as the representative features of the driving memory for car-following behavior modeling. The range of parameters were constrained for model calibration purposes. Some extremum parameters derived from the statistics of inputs were extracted, as drivers' perception of different information has been proven to be within a certain range (e.g., small changes of driving information can hardly be perceived) [8,27]. 38,375 samples were selected from the original trajectories, where 80% of the samples were used for training the sparse autoencoder model (SAE) and the rest were used for validation.

Table 1. Parameters related to driving memory as inputs of the autoencoder model.

Symbols	Meaning	Upper Bound	Lower Bound	Unit
IPT	Instantaneous perception time	18	0.3	s
$\overline{v}$	The average velocity during the time window	22	0	m/s
$\triangle v_{min}$	The minimum velocity difference during the time window (negative)	0	−6	m/s
$\triangle v_{max}$	The maximal velocity difference during the time window (positive)	6	0	m/s
$\triangle v_{abs}$	The maximal absolute value of all velocity difference values	6	0	m/s
$\triangle x_{min}$	The minimum distance gap during the time window	45	0.5	m
$\triangle x_{max}$	The maximal distance gap during the time window	45	0.5	m
$\triangle \overline{x}$	The average distance gap during the time window	45	0.5	m
t_{min}	The minimum time gap during the time window	5	0.2	s
t_{max}	The maximal time gap during the time window	5	0.2	s
$\overline{t}$	The average time gap during the time window	5	0.2	s

$\triangle v = v_{n-1} - v_n$, where v_n denotes the following vehicle.

We incorporated a new parameter named instant perception time (IPT) [28], which reflects the heterogeneous safety risk and the aggressive intensity of one special state (one driver's regime) in car-following behaviors. IPT can be interpreted as the driver's value to driving safety in the regimes in CF behaviors, and has been used as safety measures in Indian intercity expressways. When a following vehicle is getting closer to a leading vehicle, it tends to decelerate and maintain a sufficient time gap to avoid a rear-end collision. IPT is the minimum time gap maintained in a certain time window (i.e., 2.0 s in this paper). As Figure 3 illustrates, IPT is generally computed as the slope of the line joining the origin and the point having the maximum relative velocity [28]. When the following vehicle is getting farther away from the leading vehicle, IPT is set to the upper bound as shown in Table 1.

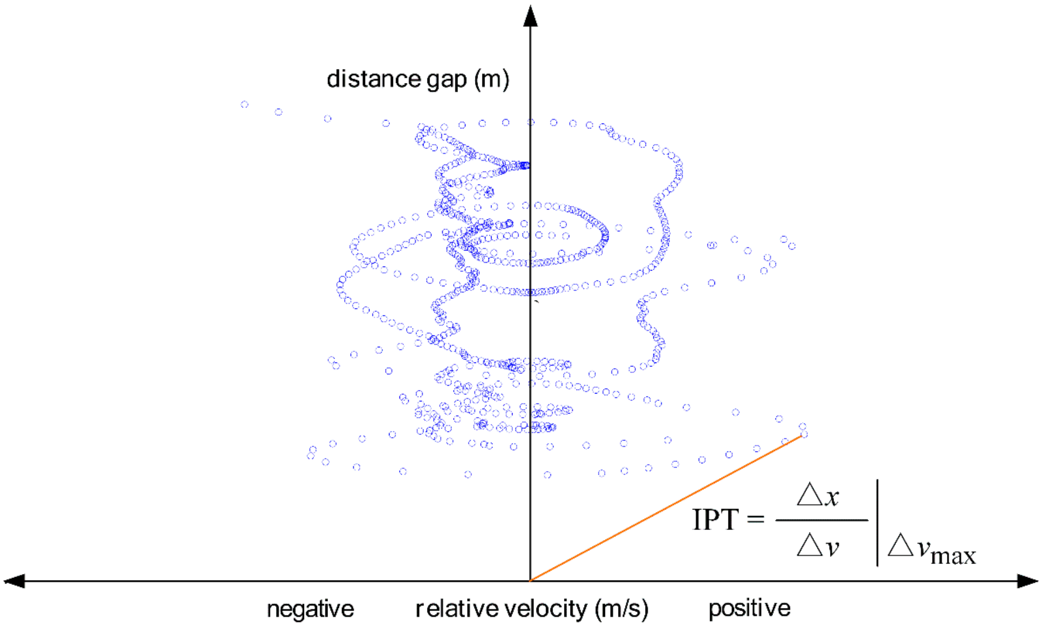


Figure 3. Schematic illustration of instant perception time (IPT).

In order to force the model to learn compact characteristic representations of the input data, an SAE was applied with a penalty term added to the final loss function. The hyperbolic tangent function was chosen as the activation function in Equation (5) with x (normalized into the scale [0,1]) as the inputs. We have chosen L1 regularization and constrained the weights of the first hidden layer by adding penalty terms to the final loss function when training the SAE model (see Equation (4)). These penalty terms also prevent overfitting.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{5}$$

Drivers get short-term driving memory represented by the parameters in Table 1, which affects driving behaviors in an unknown way. We assume not all these features are equally affecting drivers' decision making. Some of these features (e.g., IPT reflecting the rear-end crash risk in car-following behaviors) may be more important than others when drivers make car-following related decisions. We identified the main features of driving memory using an SAE.

The SAE we implemented has a symmetrical structure with seven hidden layers (see Figure 2). Table 2 shows the parameters for training the SAE model.

Table 2. Parameters of the sparse autoencoder.

Parameters	Value	Parameters	Value
Learning rate	0.009	Loss function	See Equation (4)
Epochs	25	Optimizer	Adam
Batch size	40	Number of hidden layers	7
Activation function	tanh	Sparsity penalty (α)	0.002

After training, we used an 11-dimensional elementary matrix as the input for the SAE model and obtained the following coefficients (see Table 3) extracted from the last layer of the encoder, showing the relevance of each feature in driving memory. For robustness, the SAE was calibrated three times, each resulting in a slightly different activation value. The value of these coefficients represents the compact proportion weight of the final compacted characters in the encoder [25], and reflects the significance of each parameter in driving memory for decision making. In practice, this reflects the fact that drivers choose to remember and use only partial information from the past for decision making. We also trained the SAE using different time windows (e.g., 6.0 s) and obtained a similar ranking of the selected features (results not presented).

Table 3. Ranking of parameters.

Parameter	Model Run 1	Model Run 2	Model Run 3	Average Activation
$\overline{v}$	1.743	1.182	1.334	1.420
$\triangle v_{abs}$	1.491	1.147	1.087	1.242
$\triangle v_{min}$	1.434	1.149	1.115	1.233
$\triangle v_{max}$	1.365	1.162	1.051	1.193
$\overline{t}$	1.361	1.065	1.088	1.171
IPT	1.295	1.134	1.032	1.154
t_{max}	1.151	1.14	0.858	1.050
t_{min}	0.979	1.152	0.759	0.963
$\triangle x_{max}$	0.998	1.055	0.79	0.948
$\triangle x_{min}$	0.715	0.751	0.571	0.679
$\triangle \overline{x}$	0.255	0.446	0.278	0.326

Table 3 shows that velocity is the most significant parameter, even more significant than velocity difference, which suggests that in data-driven CF models, velocity is an indispensable variable for car-following modeling. The observation is consistent with the argument in a few studies [3,15] that incorporated optimal velocity function in optimal velocity (OV) model and asymmetric full velocity difference (AFVD) model. The distance gap is the least relevant, possibly because distance gap reflects the static information and affects driving behaviors only when it is combined with other kinetic variables such as the time gap. The parameter $\overline{t}$ represents the average time gap, which is computed as the division of the distance gap and the velocity. Time gaps were incorporated in our car-following model as they are significant parameters [2,29]. To our knowledge, this research may be the first one that incorporated the time gap parameter explicitly to model car-following behaviors.

4.3. Car-Following Model Based on LSTM

4.3.1. Long Short-Term Memory Neural Network in CF Models

LSTM is a type of recurrent neural network that specializes in handling time-sequence information. The special structure of LSTMs (a forget gate, a memory gate, and an output gate in each unit in the hidden layer) allow LSTMs to address the vanishing and exploding gradients problems in the training process. The way that LSTMs work is also in line with the decision-making process of humans: Keep updating memory by retaining, forgetting, and modifying some information [5,15] (see Figure 4).

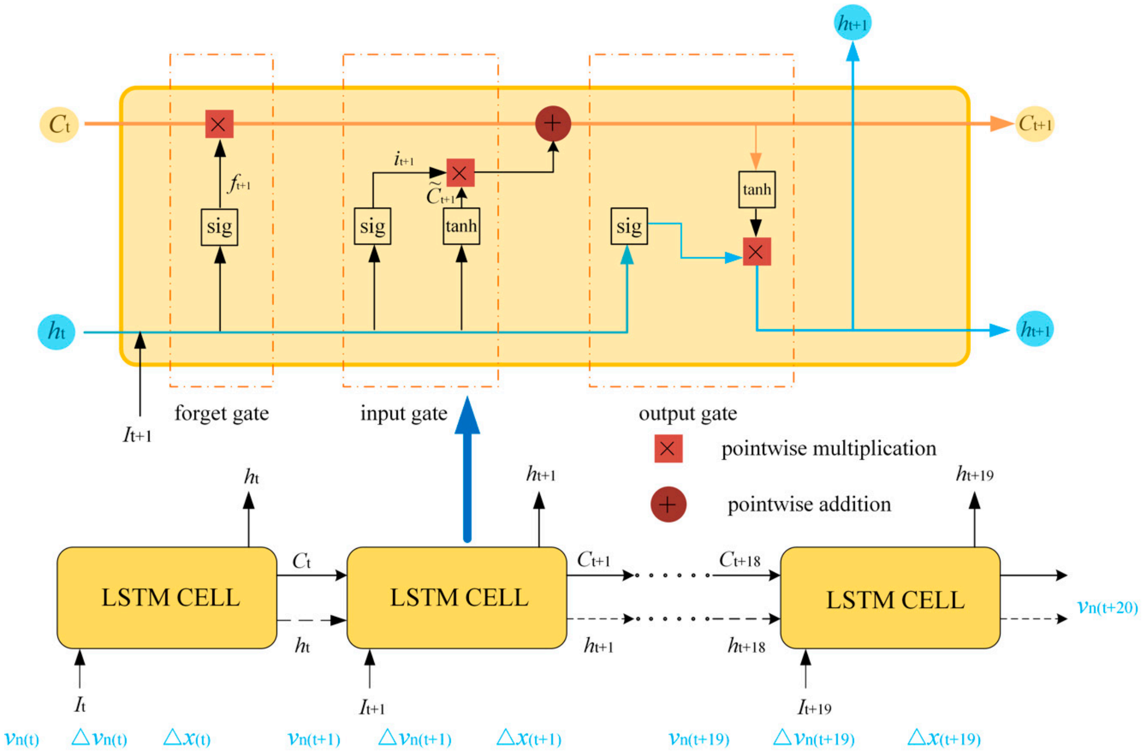


Figure 4. Schematic diagram of the long short-term memory (LSTM) cell and LSTM's mechanism.

Figure 4 illustrates the LSTM cell and LSTM's mechanism, where I_t denotes the input vectors containing the historical driving memory information. C_t and h_t denote the hidden state and output of the LSTM cell at time t. The *sig* and *tanh* in the figure indicate the standard sigmoid and hyperbolic tangent function transforming the input into ranges [0,1] and [−1,1], respectively. In the hidden layers, the information flow keeps being computed and updated, passing though the LSTM cells one by one until the final predictions are obtained.

A LSTM model was adopted to model the car-following behavior based on the historical driving memory: Velocity, relative velocity, and distance gap (or time gap) as input variables in time-series. ANNs have been widely used to model complicated relationships and many studies have focused on the prediction of acceleration. In this study, we are modeling the relationship between different short memory information and the response (see Equations (6) and (7) as examples). In two separate models, the velocity and distance gap in the time step directly following the sequence were chosen as output variables, respectively. In total, the dataset includes 506 trajectories (340,759 samples extracted) and each trajectory lasts for at least 60 s. 85% (430 trajectories) of the data were used for training the LSTM and the rest were used for validation. Dropout and batch normalization were applied to improve the model's performance by preventing overfitting [30]. As the time length of each trajectory differs, the batch size for training was not fixed. Both the LSTM models and SAE models were built using the Keras deep learning library with TensorFlow backend and implemented in Python 3.6.

$$v_i(t) = f(v_i(t-T:t-\Delta t), \Delta v_i(t-T:t-\Delta t), \Delta x_i(t-T:t-\Delta t)) \tag{6}$$

$$\Delta x_i(t) = f(v_i(t-T:t-\Delta t), \Delta v_i(t-T:t-\Delta t), t_i(t-T:t-\Delta t)) \tag{7}$$

where $f(\bullet)$ denotes the inference function that processes input variables through the LSTM architecture. $\triangle t$ denotes the minimum time step, which is 0.1 s in the trajectory data. T denotes the duration of one sequence, either 1.0 or 2.0 s.

4.3.2. Model Setups for CF Modeling

In order to analyze the impact of driving memory on CF behavior, we investigate several alternative modeling approaches. By comparing the corresponding performance of these models, we can infer how the driving memory affects CF behaviors at different time spans.

Figure 5 shows three approaches(patterns) we investigated in CF models. For pattern 1, the output variable is usually the velocity or acceleration of the subject vehicle at time t + T by supplying the relative velocity and the distance gap value at time t into mathematical equations such as the Gipps (outputs the velocity at time t + T) and GM model (outputs the acceleration at time t + T with the velocity at time t + T as inputs). The reaction time T of drivers should be well calibrated. When T is close to zero, the corresponding models are IDM, AFVD, OV models, etc. All these models do not take the subsequent times-series information (driving memory) and enough human factors into consideration. Under this prediction pattern, the parameters must be calibrated first with lower accuracy but better transferability. If combined with data-driven technology, the performance of models can also be satisfying [2,11,14]. As suggested in other research [31], we set the value of T as 1.0 s and implemented prediction pattern 1 based on a calibrated Gipps model. The velocity of the follower (n) at time t+1 denoted as $v_n(t+1)$ is given by the following equations when the reaction time T is 1.0 s.

$$v_n{}^{acc}(t+1) = v_n(t) + 2.5a_n(1-\frac{v_n(t)}{v_n d})\sqrt{0.025+\frac{v_n(t)}{v_n{}^d}} \tag{8}$$

$$v_n{}^{dec}(t+1) = -d_n + \sqrt{d_n{}^2 + d_n\left\{2[\Delta x_n(t) + l_{n-1} - S_{n-1}] - v_n(t) + \frac{v_{n-1}(t)^2}{d_{n-1}\prime}\right\}} \tag{9}$$

$$v_n(t+1) = \min\left\{v_n{}^{acc}(t+1), v_n{}^{dec}(t+1)\right\} \tag{10}$$

where a_n and $v_n{}^d$ are the follower's maximum acceleration and desired velocity. l_{n-1} is the length of the leading vehicle (n − 1). $v_{n-1}(t)$ is the velocity of the leading vehicle at time t. S_{n-1} is the effective length of the leading vehicle [32]. $d_{n-1}{}'$ and d_n are the follower's estimate of the leading vehicle's most severe braking capability and the follower's most severe braking capability.

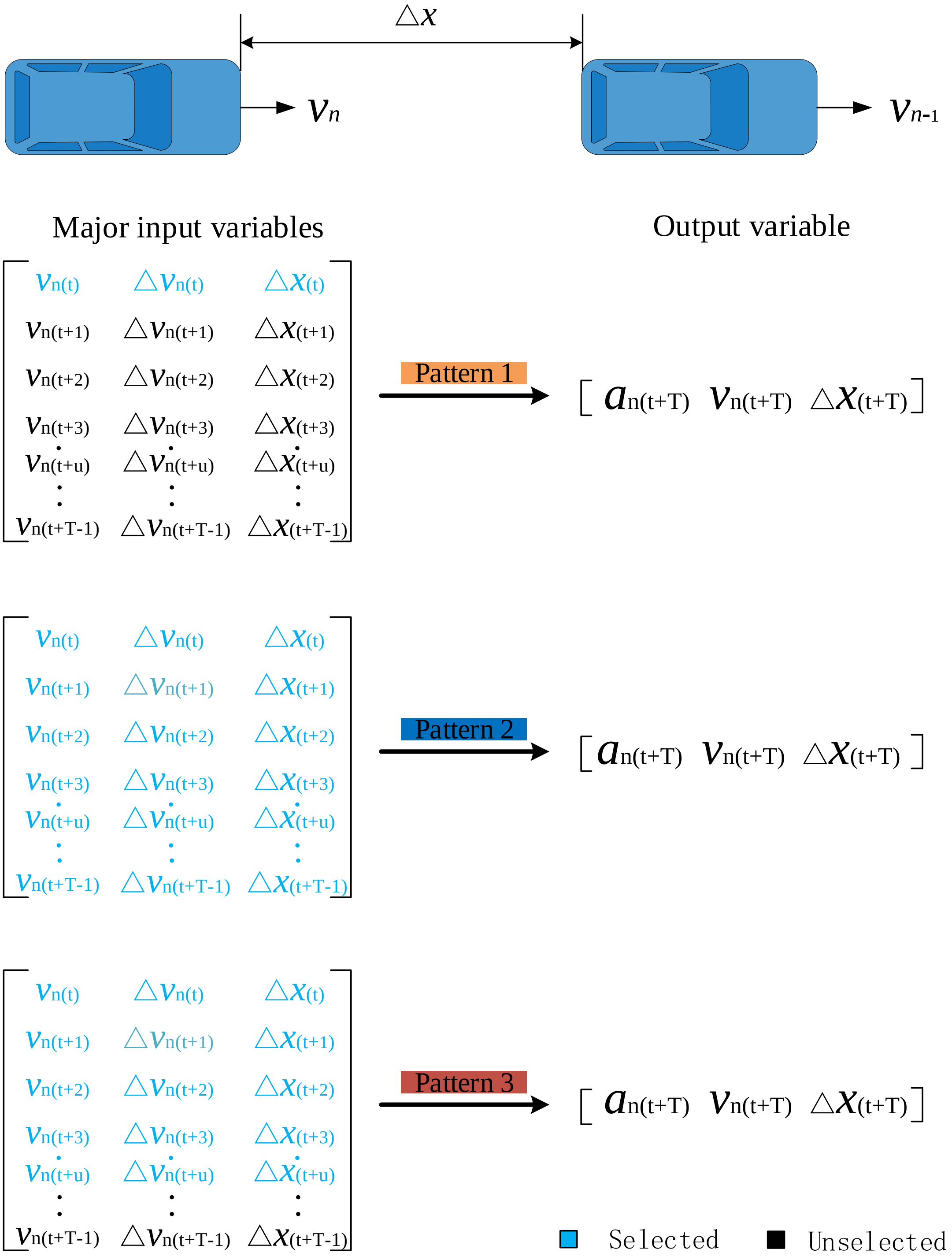


Figure 5. Three prediction patterns in car-following models. Generally, a model only predicts one of the three outcome variables.

For prediction pattern 2, its corresponding models are the main data-driven and deep learning models such as RNN, LSTM, and GRU [3,15,16]. Unlike the majority of existing CF models which take the instantaneous velocity, relative velocity, and distance headway as inputs, these advanced models use all information observed in the last few seconds and the output variables are not fixed. These models show very high accuracy in describing the car-following behaviors and only a few hyperparameters need to be calibrated, including the number of historical time steps *T*. Meanwhile, to obtain the optimal model massive high-quality data is needed and the training process is generally time-consuming and computationally expensive. The reaction time has not been considered explicitly unless the unit time interval is big enough (e.g., 1.0 s). We set the value of T as 10 and 20 (i.e., 1.0 and 2.0 s) and modeled this pattern using the LSTM model. Table 4 shows the parameters for LSTMs in pattern 2.

Table 4. Typical parameters for LSTMs in pattern 2.

Parameters	Value	Parameters	Value
Learning rate	0.002	Loss function	RMSE
Epoch	2	Optimizer	Adam
Number of LSTM cells	60	Batch size	≥587
Activation function	ReLU *	Time steps	20/10
Number of LSTM layers	4	Dropout rate	0.2

* ReLU denotes the Rectified Linear Unit function; RMSE means Root Mean Squared Error.

Prediction pattern 3 is a mix of patterns 1 and 2 to verify the short driving memory's impact on CF behaviors. Compared to pattern 1, it incorporated more historical memory information than the instantaneous case. Compared to pattern 2, pattern 3 removed part of historical memory information and incorporated reaction time explicitly, which is closer to the actual mechanism in driving behavior than patterns 1 and 2. By comparing the model performances, the importance and timeliness of historical memory information in different time spans can be investigated. We set pattern 3's sequence length u (see Figure 5) as 10 and 15 (i.e., 1.0 and 1.5 s), T = 20 (2.0 s) and implemented prediction pattern 3 based on the LSTM model.

To compare different variable sets and prediction patterns, mean squared error (MSE) and mean absolute percentage error (MAPE) were chosen as the performance indicators. These indicators are denoted as Equations (11) and (12), where y_i is the observed value of record i.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2 \tag{11}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{\mathrm{f}(x_i) - \mathrm{y_i}}{y_i}\right| \tag{12}$$

5. Results

The result of pattern 1 was given by the calibrated Gipps model whose output is velocity [30]. For calibration, we used Theil's U function as the objective function and a genetic algorithm to obtain the optimal parameters (see Equation (13)).

$$\mathrm{min}Fo = \frac{\sqrt{\sum_{i=1}^{m}(v_{s,i} - v_{a,i})^2}}{\sqrt{\sum_{i=1}^{m}{v_{s,i}}^2} + \sqrt{\sum_{i=1}^{m}{v_{a,i}}^2}} \tag{13}$$

where $v_{s,i}$ denotes the output velocity based on the calibrated Gipps model in the ith record; $v_{a,i}$ denotes the actual smoothed value of the corresponding velocity in the ith record. The calibration results showed that the follower's desired speed ${v_n}^d$ was 25.0 m/s and the follower's estimate of the leading vehicle's most severe braking capability d_{n-1}' was 1.0 m/s^2. The follower's most severe braking capability d_n was also 1.0 m/s^2, while the follower's maximum acceleration a_n was 2.4 m/s^2.

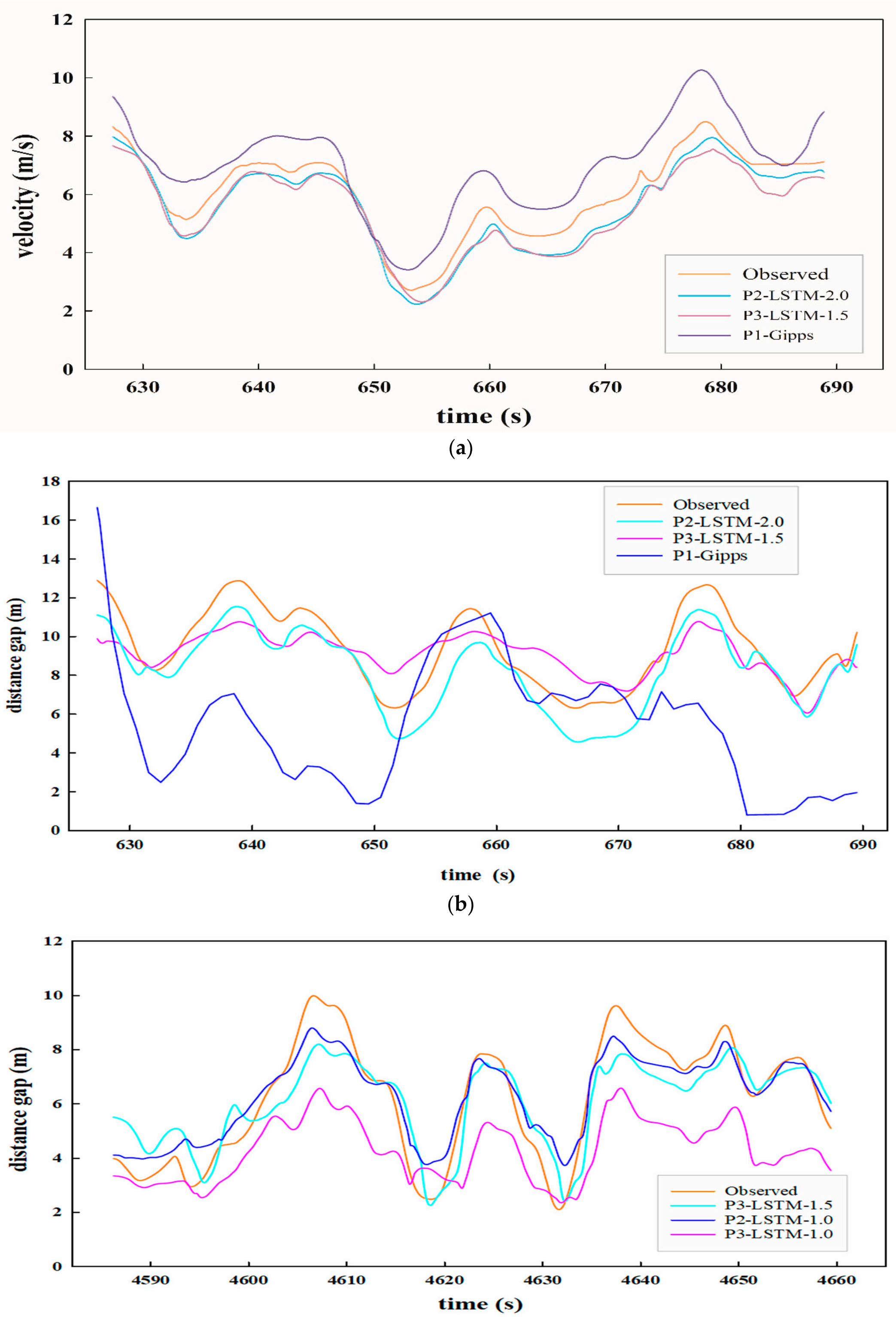
Table 5 compared the performance of the LSTM model with different configurations. The MSE and MAPE in Table 5 were computed using all validation trajectories. In Figure 6, two vehicles (ID 2072 and ID 241) in different traffic conditions and with different levels (regimes) of relative velocity and distance gap were selected for a detailed comparison. Based on the experiment results, Figure 6 and Table 5, we have the following observations.

1. The LSTM model can learn the driving memory information and describe car-following behaviors with high accuracy especially when using the prediction strategy of pattern 2, because of the LSTM's unique structure and the incorporation of important temporal information. Figure 6c shows that when v, $\triangle v$, t were used as inputs, the model did not perform as well in predicting the distance gap. The Gipps model showed acceptable performance in predicting the velocity gap but did not provide good indirect predictions of the distance gap, with estimates substantially lower than the observed value when v, $\triangle v$, $\triangle x$ were used as inputs (see Figure 6b).
2. The time gap parameter ranked highly in the autoencoder analysis, but did not lead to better results, because it is dependent on the velocity parameter. Taking both the time gap and velocity as inputs introduced redundant information and did not improve model performance. Therefore, the variable set of v, $\triangle v$ and $\triangle x$ is recommended instead.
3. In most cases, the model showed the best performance with 2.0 s as the time window in pattern 2. Only when predicting the distance gap, the model with a 1.0 s time window and with v, $\triangle v$, $\triangle x$ as inputs performed better. The optimal time window for predicting velocity and distance gaps needs to be further investigated. A larger time window may lead to better performance, but will also lead to increased computational requirements, increased training time, and (potentially) convergence difficulty during training.
4. For pattern 3, the reaction time and the historical driving memory were both considered explicitly, reflecting real driving behaviors. Some key temporal information has been removed, so the predictions are of slightly lower accuracy than using pattern 2 but are still acceptable (see Figure 6c, where v, $\triangle v$, t are used as inputs). The model showed increased prediction error with 1.0 s time window as a considerable amount of key information was lost. Therefore, incorporating reaction time and historical driving memory at the same time remains challenging for car-following modeling.
5. Results show that the prediction accuracy of the proposed LSTM model with patterns 2 and 3 was higher than that of existing models (i.e., the Gipps model).

Table 5. Performance comparison of models with different input variables and patterns for training the LSTM.

Pattern	Window Length (s) *	Input Variables	Output Variables			
			Velocity		Distance Gap	
			MSE	MAPE (%)	MSE	MAPE (%)
1	0.1	v $\triangle v$ $\triangle x$	2.494	24.396	–	–
2	2.0	v $\triangle v$ $\triangle x$	0.264	6.951	3.285	16.555
2	2.0	v $\triangle v$ t	0.589	9.689	4.855	17.192
2	1.0	v $\triangle v$ $\triangle x$	0.367	9.045	1.705	11.467
2	1.0	v $\triangle v$ t	1.227	16.474	2.795	14.044
3	1.0	v $\triangle v$ $\triangle x$	0.986	12.942	3.065	15.076
3	1.0	v $\triangle v$ t	2.353	22.163	12.543	31.612
3	1.5	v $\triangle v$ $\triangle x$	0.624	10.364	4.178	16.762
3	1.5	v $\triangle v$ t	2.148	18.337	4.954	20.771

* Window length refers to the number of historical time steps of input variables (1.0 s = 10 steps). The evaluation indictors MSE and MAPE represent the average error for all validation trajectories, with lower values representing better results.



(**a**)

(**b**)

(**c**)

Figure 6. Simulated and actual results of (**a**) velocity and (**b**) distance gap (Vehicle ID: 2072). Figure (**c**) shows the distance gap of Vehicle ID: 241. P1 means pattern 1.

In future research, the model can be assessed using more indicators than just accuracy. Such indicators may include training time, convergence difficulties, the generalization ability of the model, and three important asymmetric CF behavior characteristics identified from existing studies [15]: Hysteresis, discrete driving, and intensity difference.

6. Conclusions

This research developed a method incorporating driving memory to model car-following behaviors using long short-term memory neural networks. Driving memory is regarded as the information contributing to drivers' decision-making process. An autoencoder has been used to extract the main features underlying the time-series data in historical driving memory. Such main features include the velocity and time gaps. LSTM has been employed for car-following behavior modeling and to investigate the relationship between driving memory and car-following behavior. The results show that velocity, relative velocity, IPT, and time gap are the most relevant parameters, while the distance gap is insignificant. Furthermore, we compared the accuracy and robustness of three modeling approaches, including various driving memory information and span levels. The model with prediction pattern 2 and input variables of v, $\triangle v$ and $\triangle x$ outperformed the other patterns, showing the best accuracy and robustness in training and prediction results. Hence, this research bridges the gap between historical driving memory and car-following behavior.

Current methodologies for modelling car-following behaviour often employ relationships between speed, speed difference, and distance. Driving style has not been paid enough attention. By calibrating the LSTM model on sequences of a single driver (i.e., capturing the specific driving style), further increases in prediction accuracy could be obtained. This provides an opportunity to apply the LSTM method in future practical use. Such a model has the potential to provide personalized warnings of dangerous car-following distance over the next second. Future study will focus on investigating how the driving memory in different spans (timeliness) influences CF behaviors. A larger range of time windows should be tested to verify the model and the proposed idea. Based on our current research on LSTM models (pattern 3), the model presented can be incorporated into driving assistance systems to provide rear-end crash warnings in advance, potentially reducing road trauma.

Author Contributions: Conceptualization, P.F. and J.G.; Methodology, P.F.; Software, P.F. and J.G.; Validation, H.Z. and J.S.W.; Writing—original draft preparation, P.F.; Writing—review and editing, H.Z., J.S.W., and J.G.; Supervision, J.G.; Funding acquisition, J.G. and Y.W.

Funding: This research was funded by the National Key Research and Development Program of China (grant number 2018YFB1600500); the Opening Foundation of Key Laboratory of Road and Traffic Engineering of the Ministry of Education (2018). The authors would like to thank FHWA for providing the NGSIM data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. WHO. *Global Status Report on Road Safety 2018*; World Health Organization: Geneva, Switzerland, 2018.
2. Wang, J.; Zhang, L.; Lu, S.; Wang, Z. Developing a car-following model with consideration of driver's behavior based on an Adaptive Neuro-Fuzzy Inference System. *J. Intell. Fuzzy Syst.* **2015**, *30*, 461–466. [CrossRef]
3. Zhou, M.; Qu, X.; Li, X. A recurrent neural network based microscopic car following model to predict traffic oscillation. *Transp. Res. Part C Emerg. Technol.* **2017**, *84*, 245–264. [CrossRef]
4. Jiang, R.; Wu, Q.; Zhu, Z. Full velocity difference model for a car-following theory. *Phys. Rev. E* **2001**, *64*, 017101. [CrossRef] [PubMed]
5. Saifuzzaman, M.; Zheng, Z. Incorporating human-factors in car-following models: A review of recent developments and research needs. *Transp. Res. Part C Emerg. Technol.* **2014**, *48*, 379–403. [CrossRef]
6. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
7. Lee, G. A Generalization of Linear Car-Following Theory. *Oper. Res.* **1966**, *14*, 595–606. [CrossRef]
8. Durrani, U.; Lee, C.; Maoh, H. Calibrating the Wiedemann's vehicle-following model using mixed vehicle-pair interactions. *Transp. Res. Part C Emerg. Technol.* **2016**, *67*, 227–242. [CrossRef]
9. Gipps, P. A behavioural car-following model for computer simulation. *Transp. Res. Part B Methodol.* **1981**, *15*, 105–111. [CrossRef]
10. Newell, G. A simplified car-following theory: A lower order model. *Transp. Res. Part B Methodol.* **2002**, *36*, 195–205. [CrossRef]

11. Khodayari, A.; Ghaffari, A.; Kazemi, R.; Braunstingl, R. A Modified Car-Following Model Based on a Neural Network Model of the Human Driver Effects. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2012**, *42*, 1440–1449. [CrossRef]
12. Chong, L.; Abbas, M.M.; Medina, A. Simulation of Driver Behavior with Agent-Based Back-Propagation Neural Network. *Transp. Res. Rec. J. Transp. Res. Board* **2011**, *2249*, 44–51. [CrossRef]
13. Wei, D.; Liu, H. Analysis of asymmetric driving behavior using a self-learning approach. *Transp. Res. Part B Methodol.* **2013**, *47*, 1–14. [CrossRef]
14. Zhu, M.; Wang, X.; Wang, Y. Human-like autonomous car-following model with deep reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2018**, *97*, 348–368. [CrossRef]
15. Huang, X.; Sun, J.; Sun, J. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 346–362. [CrossRef]
16. Wang, X.; Jiang, R.; Li, L.; Lin, Y.-L.; Wang, F.-Y. Long memory is important: A test study on deep-learning based car-following model. *Phys. A Stat. Mech. Appl.* **2019**, *514*, 786–795. [CrossRef]
17. NGSIM: Next Generation Simulation. FHWA, U.S. Department of Transportation. Available online: www.ngsim.fhwa.dot.gov (accessed on 1 April 2019).
18. Thiemann, C.; Treiber, M.; Kesting, A. Estimating Acceleration and Lane-Changing Dynamics from Next Generation Simulation Trajectory Data. *Transp. Res. Rec. J. Transp. Res. Board* **2008**, *2088*, 90–101. [CrossRef]
19. Naseri, H.; Nahvi, A.; Karan, F.S.N. A new psychological methodology for modeling real-time car following maneuvers. *Travel Behav. Soc.* **2015**, *2*, 124–130. [CrossRef]
20. Van Winsum, W. The human element in car following models. *Transp. Res. Part F Traffic Psychol. Behav.* **1999**, *2*, 207–211. [CrossRef]
21. Zaky, A.B.; Gomaa, W. Car following regime taxonomy based on Markov switching. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 1329–1334.
22. Ma, X.; Andreasson, I. Behavior Measurement, Analysis, and Regime Classification in Car Following. *IEEE Trans. Intell. Transp. Syst.* **2007**, *8*, 144–156. [CrossRef]
23. Hosseini-Asl, E.; Zurada, J.M.; Nasraoui, O. Deep Learning of Part-based Representation of Data Using Sparse Autoencoders with Nonnegativity Constraints. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 2486–2498. [CrossRef]
24. Chen, J.; Wu, Z.; Zhang, J. Driver identification based on hidden feature extraction by using adaptive nonnegativity-constrained autoencoder. *Appl. Soft Comput.* **2019**, *74*, 1–9. [CrossRef]
25. Guo, J.; Liu, Y.; Zhang, L.; Wang, Y. Driving Behaviour Style Study with a Hybrid Deep Learning Framework Based on GPS Data. *Sustainability* **2018**, *10*, 2351. [CrossRef]
26. Liu, H.; Taniguchi, T.; Tanaka, Y.; Takenaka, K.; Bando, T. Visualization of Driving Behavior Based on Hidden Feature Extraction by Using Deep Learning. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2477–2489. [CrossRef]
27. Wiedemann, D. *Simulation des Straßenverkehrsflusses*; Schriftenreihe des Instituts für Verkehrswesen: Karlsruher, Germany, 1974.
28. Raju, N.; Kumar, P.; Arkatkar, S.; Joshi, G. Determining risk-based safety thresholds through naturalistic driving patterns using trajectory data on intercity expressways. *Saf. Sci.* **2019**, *119*, 117–125. [CrossRef]
29. Wang, F.; Dai, M.; Sun, L.; Jin, S. Mixed distribution model of vehicle headway based on multiclass car following. *J. Zhejiang Univ. Eng. Sci.* **2015**, *49*, 1288–1294.
30. Chollet, F. Keras Document. Available online: https://keras.io (accessed on 15 April 2019).
31. Zhang, X.; Bham, G.H. Estimation of driver reaction time from detailed vehicle trajectory data. In Proceedings of the 18th IASTED International Conference on Modelling and Simulation, Montreal, QC, Canada, 30 May–1 June 2007; pp. 574–579.
32. Vasconcelos, L.; Neto, L.; Santos, S.; Silva, A.B.; Seco, Á. Calibration of the Gipps Car-following Model Using Trajectory Data. *Transp. Res. Procedia* **2014**, *3*, 952–961. [CrossRef]