International Journal of Intelligent Transportation Systems Research (2023) 21:86–98
https://doi.org/10.1007/s13177-022-00339-9

Data-driven Modeling of Car-Following Behavior on Freeways Considering Spatio-Time Effects: A Comparison of Different Neural Network Structures

Masahiro Kinoshita[1] · Yasuhiro Shiomi[2]

Received: 1 June 2022 / Revised: 13 September 2022 / Accepted: 10 December 2022 / Published online: 4 March 2023

Abstract

Car-following behavior models based on conventional mathematical models cannot adequately reproduce traffic phenomena, such as traffic breakdown, capacity drop, and oscillations, and require parameter setting. Therefore, this study aims to construct a highly accurate car-following behavior model using deep learning. We evaluated the influence of variables in the dataset using a random forest. Furthermore, we constructed models to predict the acceleration in one second using the deep learning methods, deep neural network, long short-term memory, one-dimensional convolution neural network (1DCNN), and 2DCNN models. The models were evaluated using root mean square error, MAE, yyplot, and loss plot. The results showed that spatiotemporally structuring the data increased the accuracy of the predictions.

Keywords Car-following · Deep learning · Data-driven · Vehicle trajectory data · Freeways

1 Introduction

Traffic congestion on freeways remains one of the social problems to be solved. Although the amount of traffic congestion on freeways temporally reduced due to the COVID-19 pandemic, it has mostly returned to the level of before the pandemic in many countries, including Japan. The speed reduction caused by traffic congestion results in huge economic losses and environmental burdens, making a carbon–neutral society difficult to achieve. Therefore, it is essential to implement effective traffic management measures to alleviate traffic congestion based on existing infrastructure. Additionally, a microscopic traffic simulation is a useful tool for evaluating the effectiveness and impacts of such measures in advance, especially in the era of connected and autonomous vehicles.

Generally, in microscopic traffic simulations, vehicle movement is described by combining multiple mathematical models according to many IF–THEN rules [1, 2]. Therefore, there is a nonlinear and discontinuous relationship between parameter values and vehicle behavior, and even a slight change in parameters or the order of vehicles with different parameters causes unrealistic behavior, such as rear-end collisions [3, 4]. It is still a challenging issue to establish a logical method for calibrating parameters in traffic simulation [5–8].

However, it is possible to collect large amounts of detailed data on vehicle trajectories along freeways, and deep learning methods have been recently established to learn complex vehicular behaviors. These methods enable data-driven vehicle behavior modeling, which does not require any assumptions. For example, Zhou et al. [9] showed that a car-following behavior model using a recurrent neural network (RNN) trained on the next generation simulation (NGSIM) I80 datasets achieved higher reproducibility than the intelligent driver model (IDM) [10]. Wang et al. [11] developed a deep learning-based car-following model using NGSIM and vehicle trajectory data generated by an IDM calibrated based on the NGSIM data. They verified that the proposed model could reproduce traffic congestion phenomena, including hysteresis loops. Additionally, they claimed the importance

✉ Masahiro Kinoshita
ce0003sp@ed.ritsumei.ac.jp

Yasuhiro Shiomi
shiomi@fc.ritsumei.ac.jp

[1] Graduate School of Science and Engineering, Ritsumeikan University, 1-1-1, Nojihigashi, Kusatsu City, Shiga 525-8577, Japan

[2] Department of Science and Engineering, Ritsumeikan University, 1-1-1, Nojihigashi, Kusatsu City, Shiga 525-8577, Japan

of considering the time-series factors using long short-term memory (LSTM). Fan et al. [12] investigated the parameters necessary for improving the accuracy of RNN.

The above studies used the NGSIM dataset. However, this dataset has the following issues on the coverage of the dataset, i.e., this data did not contain the free-flow state and shockwave generation. Thus, it is not unveiled that the data-driven approach can represent the vehicle maneuvers in such traffic conditions. Additionally, although the above studies emphasized the importance of capturing time-series effects, they ignored the spatial effect on vehicle maneuvers. Traffic congestion is a spatiotemporal phenomenon; thus, the spatial factors, such as road geometric features and the traffic state of vehicles ahead, should be captured in the model, as previous studies suggested [13–15].

In this study, we develop a car-following model using deep learning-based on open data sources of vehicle trajectories on a freeway, Zen traffic data (ZTD) [16], provided by Hanshin Expressway Co. Ltd. in Japan. This dataset provides 5 h of vehicle trajectory data over a 2 km section and includes various traffic conditions, such as free-flow conditions and the generation and propagation of shockwaves. Using this data, we develop a deep learning-based model that explicitly considers the gradient and curvature as road structure characteristics, states of multiple vehicles in front and behind to reflect the spatial characteristics of the traffic conditions, and the time-series effect. First, we evaluate the importance of the input variables using a random forest (RF) and select them for further modeling. Then, we employed four types of neural network (NN) models, e.g., fully connected deep neural network (DNN), LSTM, one-dimensional convolutional neural network (1DCNN), and two-dimensional CNN (2DCNN), to train the car-following behavior. In previous studies, many models have been developed by considering time series data, but in this study, each model was selected to verify how structurization of the data set affect the prediction accuracy. Finally, their prediction accuracy is compared.

The remainder of the paper is organized as follows. Section 6 provides a modeling framework for car-following behavior and outlines of RF and NN models employed in this paper. Section 3 describes the ZTD trajectory dataset used in this paper and its preprocessing procedures. In Section 4, we describe the results of variable selection based on RF. Section 5 presents the comparisons of the prediction accuracy given by the four NN models and the discussion. Finally, Section 6 presents the conclusion and future studies.

2 Model overview

The following section provides a concept of car-following modeling using a data-driven approach. It also briefly introduces each machine learning and NN algorithm used in this study.

2.1 Overview of Data-Driven Car-Following Model

Car-following models describe traffic dynamics from the perspective of individual driver-vehicle units [17]. This model mimics the accel and brake pedal operation of human drivers that perceive the surrounding environment and react to the vehicle ahead and behind. The model outputs the longitudinal acceleration of the target vehicle and requires the variables about stimuli and environment a driver may perceive and react to as input variables. Thus, the model can be formulated by Eq. (1).

$$a_i(t + \Delta t) = f\left(\boldsymbol{s}_i(x, t)|\boldsymbol{e}_i(x, t)\right), \tag{1}$$

where $a_i(t)$ is the acceleration of vehicle i at time t; Δt is the reaction delay; $\boldsymbol{s}_i(x, t)$ and $\boldsymbol{e}_i(x, t)$ are vectors of variables representing stimuli that vehicle i perceived and the surrounding environment at the time and location (x, t), respectively; $f(\bullet)$ is an arbitrary function that transforms the stimuli to an acceleration after Δt.

Generally, relative speeds and space gaps with vehicles ahead are considered as variables of stimuli $\boldsymbol{s}$. $\boldsymbol{e}$ includes the geometric features, such as a curvature radius, gradient, and current traffic states. For example, the effect of changes in gradient on acceleration is accounted in the model formulation by Bernat et al. [18], and the free-flow and interact terms are considered in IDM+[19]. These factors are represented as a vector $\boldsymbol{e}_i$.

In the conventional mathematical model of car-following behavior, an arbitrary function $f(\bullet)$ is explicitly defined given a priori principles and assumptions on human driving behavior. However, there is no guarantee that the principles and assumptions are always consistent, instead they may depend on drivers, vehicle characteristics, and traffic state. Meanwhile, in this study, we use a data-driven approach to model the behavior by applying NNs, which enable assumption-free modeling.

2.2 Outline of RF and NN Algorithms

In this study, we use RF to select the important input variables and criteria for the car-following model. We referred Degenhardt et al. [20] and the importance values estimated by random forests has been used as a method of variable selection. Then, we employ DNN, LSTM, 1DCNN, and 2DCNN to predict acceleration after a 1-s reaction delay and compare the prediction accuracies. In existing studies, the reaction delay is often assumed to be 1 s or longer. Therefore, in this study, we also assumed 1 s. DNN deals with input variables independently, i.e., the order and structure of input variables are not considered in the model. Meanwhile, STM and 1DCNN can capture the time-series effect, and 2DCNN can capture the time-series and spatial effects by appropriately formatting the input variables. A brief explanation of each model is described as follows.

2.2.1 Random Forest

RF is a machine learning algorithm that combines multiple decision trees to make predictions. Decision trees are methods for creating conditional branches in order from the top; however, it is prone to overlearning. RF can create models with high generalization performance by averaging the results of multiple decision trees; thereby, minimizing overlearning. RFs are also characterized by the ability to obtain variable importance. Importance is a relative measure of the magnitude of the prediction error when randomly replacing a numerical value for a selected variable from all variables; the importance of all variables added together equals 1. In other words, the higher the value of importance, the more important the variable is for the forecast.

2.2.2 DNN

DNN is a kind of NN with four or more layers. It is one of the deep learning methods that enable complex processing. In the DNN model, neurons in each layer are fully coupled, and if appropriate weights can be derived, the method can provide highly accurate predictions. For the details of DNN, please refer to the literature [21].

2.2.3 LSTM

LSTM is a type of RNN that can learn long-term dependencies. RNNs can only handle relatively short datasets, whereas LSTM models can be trained on datasets with longer time-series information. For the details of LSTM, please refer to the literature [22].

2.2.4 CNNs

CNN is a type of deep learning method that adds convolution to NNs. Convolution is the process of extracting local features and generating a "feature map" that summarizes the data features using numerical data on a small lattice called a kernel. There are two types of CNN models: the 1DCNN model, which can numerically predict time-series data, and the 2DCNN model, which is successful in the field of image recognition. The 2DCNN model can structure input variables in two dimensions in time and space; therefore, it is suitable for representing traffic phenomena. In this study, we use the 1DCNN and 2DCNN models for prediction. For the details of CNN, please refer to the literature [23].

3 Data and Dataset Construction

3.1 ZTD

ZTD [16], provided by Hanshin Expressway, is used for modeling car-following behaviors. ZTD utilizes image sensing to create data on the driving behavior of all vehicles driving in the target section in 0.1-s increments as vehicle trajectory data. ZTD records the position and speed of each vehicle.

Figure 1 shows a schematic road map of the vehicle trajectory data used in this study. The target section includes the Tsukamoto on-ramp on the Hanshin Expressway Route 11 Ikeda Line (Osaka direction). This section is an important route that connects Osaka Airport, Meishin Expressway, Chugoku Expressway, and other national trunk roads with Osaka City. In this section, traffic congestions recurrently occur due to the complex road environment with shaped curves, sags, and a merging section with the Tsukamoto on-ramp.

Our proposed car-following behavior model predicts the acceleration based on the conditions of vehicles in front and behind the target vehicle and road structure data. To eliminate the influence of lane changes and merging vehicles on acceleration, the target section is limited to the overtaking lane where a lane change is prohibited. The starting and ending points of the no lane change zone are 4.2 and 3.5 kp, respectively. In the overtaking lane, congestion mainly occurs near the Tsukamoto on-ramp at 3.8 kp. In this study,

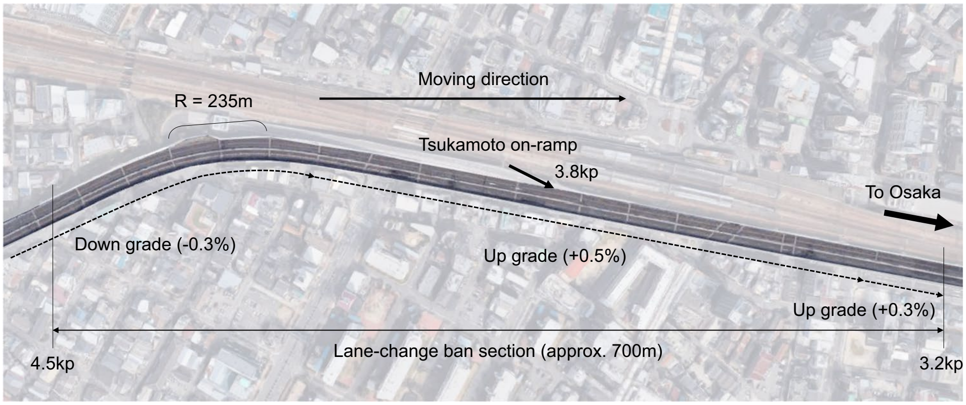


Fig. 1 Target section

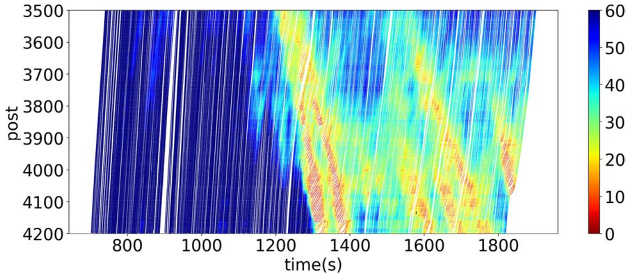


Fig. 2 Vehicle trajectory diagram

we used approximately 20 min of data, as shown in Fig. 2. This figure shows the free-flow state and the generation and propagation of the shockwave.

3.2 Input Data

In this study, data every 0.1-s is used to learn detailed vehicle motion and predict the acceleration of the target vehicle one second later. The target vehicles are divided into small and large vehicles, and models are built for each.

For each vehicle for which acceleration was to be predicted after 1 s, the datasets for each model were obtained by adding the driving history data for ten vehicles in the front and three vehicles in the rear and road structure data, as shown in Table 1. Only the DNN models were given the time-series information of “speed change, acceleration change, and distance traveled from 0.5 to 3 s ago (in 0.5-s increments). The road structure data was the average longitudinal slope of the road 50 m in front and behind the target vehicle and the curvature 50 m ahead. The number of time steps considered in the LSTM, 1DCNN, and 2DCNN models was set to 30 steps, corresponding to 3 s.

To create the dataset, we first organized the data as described above and then removed outliers and unnecessary data. Next, we applied normalization according to Eq. (2) to eliminate differences in the magnitude of each variable value and level the data.

$$x_i^{'} = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}(i = 1 \cdots n) \tag{2}$$

where x_i represents the i^{th} value of the original input data **x**, and $x_i^{'}$ represents the value after normalization. This gives all data a maximum value of 1 and a minimum value of 0. Then, all data were randomly split 8:2, with the former used for model training and the latter for validation.

In this study, we used DNN, LSTM, 1DCNN, and 2DCNN models, but the shape of the data required for training differs depending on the model. Therefore, it is necessary to create an array structure that is suitable for each model. Figure 3 illustrates each array structure. The variable y_t in the figure is the variable to be predicted, which represents the acceleration of the target vehicle at time t in 1-s. Additionally, $x^k_{i,t-j}$ represents the kth variable at j time steps ($0 \le j \le n$) ahead ($-m_f \le j < 0$) or behind ($0 < j \le m_b$) of the vehicle from time t in terms of the target vehicle. The vector notation for this variable is $\mathbf{x}_{i,t-j}$, and it is expressed in Eq. (3). where N is the total number of input variables.

$$\mathbf{x}_{i,t-j} = \{x^1_{i,t-j}, x^2_{i,t-j}, \dots, x^N_{i,t-j}\} \tag{3}$$

The DNN model uses a 1D array of all input variables as input variables. For the LSTM and 1DCNN models, we created a 2D array with the time-series on the vertical axis and the values of each explanatory variable on the horizontal axis to allow for time-series variation. The 2DCNN model takes as input a 3D array of data for each variable, organized in a 2D array in time and space, to explicitly take into account the spatiotemporal dynamics of traffic conditions.

4 Variable Selection By Random Forest

4.1 Model Settings

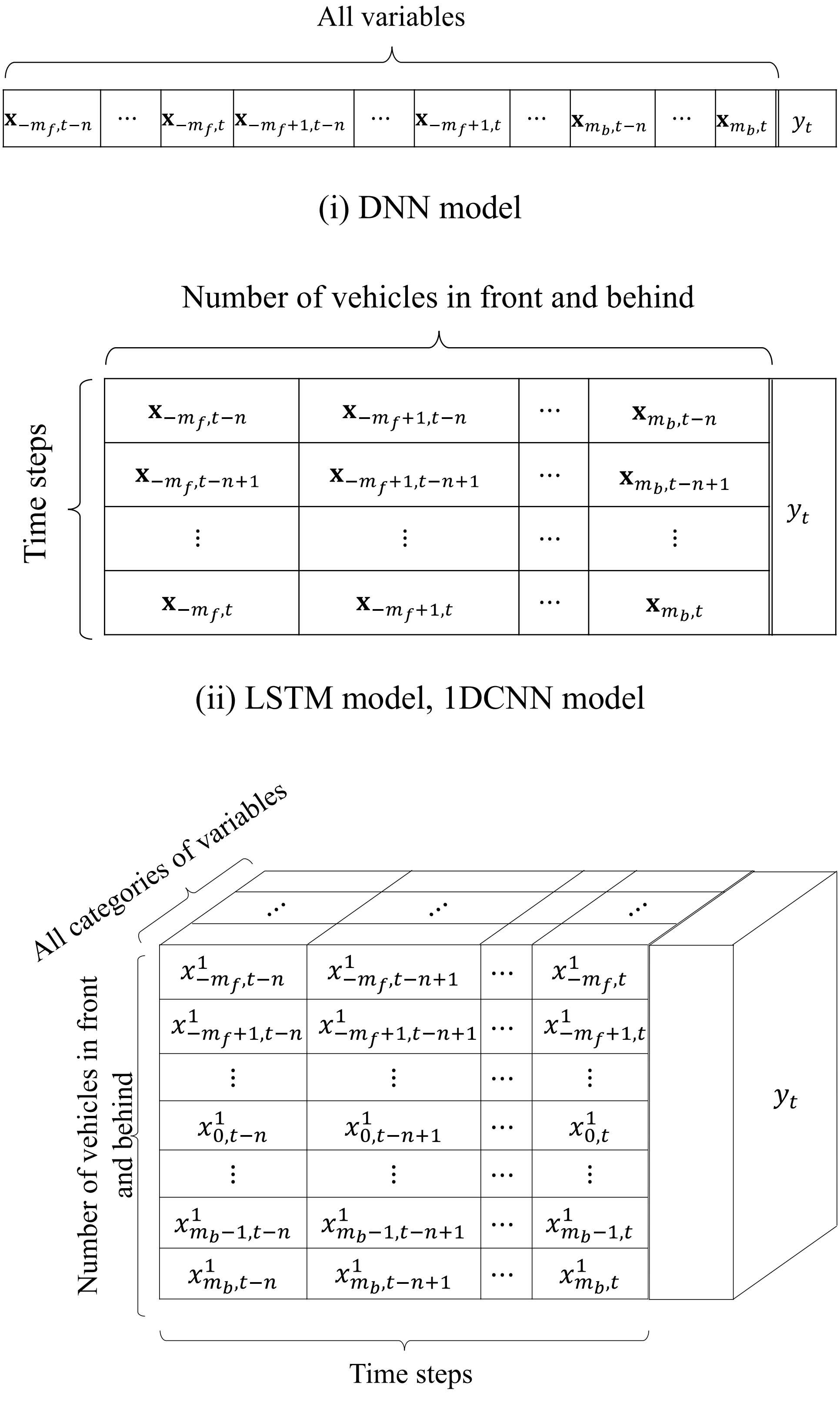
This study evaluates the degree of influence of a variable on a dataset using the importance of RF, a type of machine learning. The purpose of the influence evaluation is to determine how a variable affects the predicted acceleration at 1-s after. We used a dataset of small vehicles to evaluate the influence of the variables. The numbers of training data, validation data, and variables are 238,634, 59,659, and 296, respectively.

The accuracy of RF can be improved by specifying hyperparameters during training. In this study, we set the “number of decision trees” and “depth of decision trees” as

Table 1 Layout of the dataset

	Data content
Driving history data	Current velocity
	Distance and relative speed to vehicle in front
	Speed change, acceleration change, and distance traveled from 0.5 to 3 s ago (in 0.5-s increments)
Road structure data	Curvature at 50 m
	Average of longitudinal gradient of 50 m before and after

Fig. 3 Structures of input data



(i) DNN model

(ii) LSTM model, 1DCNN model

(iii) 2DCNNmodel

hyperparameters that can be set after trial and error. Increasing the "number of decision trees" improves the prediction accuracy, but accuracy reaches a certain limit value. "Depth of decision trees" allows one to specify the maximum depth of each decision tree. If it is too large, over-fitting will occur on the training data, decreasing the prediction accuracy for unknown data. We set the "number of decision trees" and "depth of decision trees" to 200 and 50, respectively.

4.2 Results

Figure 4 shows the scatter plots between the prediction results of accelerations using RF and the overserved acceleration (hereafter referred to as "yyplot"). Although the prediction accuracy is not good in the large deceleration and acceleration phases, the model prediction plots are mostly on the diagonal line. R^2 and the inclination of the yyplot are 0.739 and 0.629, respectively, which is high enough for analyzing the variable selection.

Table 2 presents the top 20 and bottom 20 importance variables calculated. As presented in the table, relative speed and distance used in conventional following models were selected as the important variables for prediction. In addition to driving historical data of the target vehicle, time-series data from multiple seconds before for the first to third vehicles in front of the target vehicle was shown to be important. For the road structure data, the average of the longitudinal gradient of 50 m before and after the road was also shown to have a significant impact on the prediction. However, variables related to the fifth and subsequent vehicles in front were less important. This result shows that variables related to the fifth and subsequent vehicles in front are not important for prediction. Therefore, we will compare the prediction accuracy using a dataset with only five vehicles in front of the vehicle for prediction using deep learning.

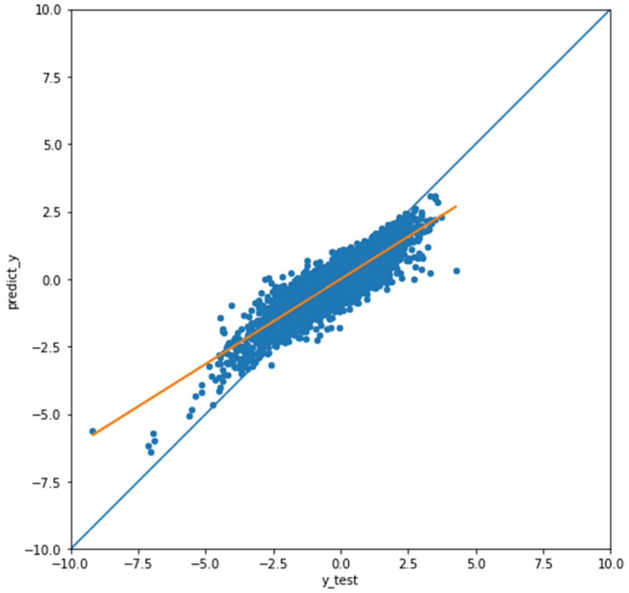


Fig. 4 Prediction results of RF

5 Model Setup and Forecast Results

5.1 Model Setup

In this study, we used Python3, which allows for easy construction of a learning environment, and implemented the model using Keras. We used Adam as the optimization method. The training process of a model is visualized by Loss Plot to check for over-fitting. Loss Plot shows the sequence of loss values of each iteration and can determine that no over-fitting has occurred if the loss of training and validation data converges. RMSE, MAE, yyplot, and the inclination of yyplot were used to evaluate the prediction accuracy. The closer RMSE and MAE are to 0, the more accurate the predictions are. yyplot is a scatterplot of the observed and predicted values, and the closer the slope is to 1, the more accurate the prediction is in all regions. The structure and hyperparameters of each model were determined after trial and error. The learning rate is a parameter that determines how much to update the weights in the optimization. Batch size is the number of training data used in one training run. Epochs are the number of iterations over which the training data are trained. The models were constructed for small and large vehicles.

5.2 Results

5.2.1 DNN Model

Figure 5 shows the structure of the DNN model constructed in this study. The model setup and each evaluation index are presented in Table 3, and the loss plot and yyplot are shown in Fig. 6. The loss plots show that the loss converged for the training and validation data, confirming that no over-fitting occurred. For small vehicles, the inclination of yyplot is far from 1.0, and some plots are far off the diagonal, which raises questions about the accuracy of the prediction. However, for large vehicles, the inclination of yyplot is close to 1.0. Although there are some plots that are far from the diagonal, the trend of acceleration and deceleration can be grasped.

5.2.2 LSTM Model

Figure 7 shows the structure of the LSTM model constructed in this study. The model setup and each evaluation index are presented in Table 4, and the loss plot and yyplot are shown in Fig. 8. The loss plots show that the loss converged for the training and validation data, confirming that no over-fitting

Table 2 Importance of variables to predict acceleration after 1 s

Rank	Variable	Importance index	Rank	Variable	Importance index
1	Relative velocity of the target vehicle at present	0.2866	277	Distance traveled from 2.5 s before of the 5th vehicle in front	0.0014
2	Average of longitudinal gradient of 50 m before and after at present	0.0505	278	Distance traveled from 1.5 s before of the 6th vehicle in front	0.0014
3	Distance of the target vehicle at present	0.0163	279	Distance traveled from 2.5 s before of the 8th vehicle in front	0.0014
4	Speed change from 3 s before of the 2nd vehicle in front	0.0126	280	Distance traveled from 0.5 s before of the 6th vehicle in front	0.0013
5	Speed change from 1 s before of the target vehicle	0.0114	281	Distance traveled from 1.5 s before of the 10th vehicle in front	0.0011
6	Speed change from 2 s before of the 1st vehicle in front	0.0088	282	Distance traveled from 0.5 s before of the 5th vehicle in front	0.0011
7	Speed change from 3 s before of the 3rd vehicle in front	0.0088	283	Distance traveled from 2.5 s before of the 6th vehicle in front	0.0011
8	Speed change from 3 s before of the target vehicle	0.0081	284	Distance traveled from 1 s before of the 7th vehicle in front	0.0011
9	Acceleration change from 1 s before of the 2nd vehicle in front	0.0068	285	Distance traveled from 1 s before of the 6th vehicle in front	0.0011
10	Speed change from 2.5 s before of the 2nd vehicle in front	0.0066	286	Distance traveled from 2.5 s before of the 7th vehicle in front	0.0011
11	Speed change from 3 s before of the 1st vehicle in front	0.0064	287	Distance traveled from 0.5 s before of the 7th vehicle in front	0.0011
12	Distance traveled from 1 s before of the target vehicle	0.0063	288	Distance traveled from 2.5 s before of the 4th vehicle in front	0.0011
13	Speed of the target vehicle at present	0.0058	289	Distance traveled from 2.5 s before of the 5th vehicle in front	0.0011
14	Distance of the 2nd vehicle in front at present	0.0056	290	Distance traveled from 2 s before of the 9th vehicle in front	0.0011
15	Distance traveled from 0.5 s before of the target vehicle	0.0055	291	Distance traveled from 2 s before of the 10th vehicle in front	0.0011
16	Acceleration change from 1.5 s before of the 2nd vehicle in front	0.0051	292	Distance traveled from 2 s before of the 6th vehicle in front	0.0010
17	Speed change from 1.5 s before of the target vehicle	0.0050	293	Distance traveled from 2 s before of the 7th vehicle in front	0.0010
18	Speed change from 1.5 s before of the 1st vehicle in front	0.0049	294	Distance traveled from 1.5 s before of the 7th vehicle in front	0.0010
19	Relative velocity of the 2nd vehicle in front at present	0.0048	295	Distance traveled from 2.5 s before of the 10th vehicle in front	0.0010
20	Speed change from 2.5 s before of the target vehicle	0.0046	296	Distance traveled from 2 s before of the 4th vehicle in front	0.0010

occurred. As shown in the yyplot, the plots are concentrated on the diagonal for the small and large vehicles. Additionally, there are no plots that are far off the prediction in the rapid deceleration and rapid acceleration regions, indicating that the prediction is very accurate.

5.2.3 1DCNN Model

Figure 9 shows the structure of the 1DCNN model constructed in this study. The model setup and each evaluation index are presented in Table 5, and the loss plot and yyplot are shown in Fig. 10. The loss plots show that the loss converged for the training and validation data, confirming that no over-fitting occurred. The slope of yyplot is larger for the large vehicle than that of the small vehicle, resulting in higher prediction accuracy. Additionally, compared with the yyplot of the LSTM model, the small and large vehicles have plots that miss predictions mainly in the region of rapid deceleration.

5.2.4 2DCNN Model

Figure 11 shows the structure of the 2DCNN model constructed in this study. The model setup and each evaluation

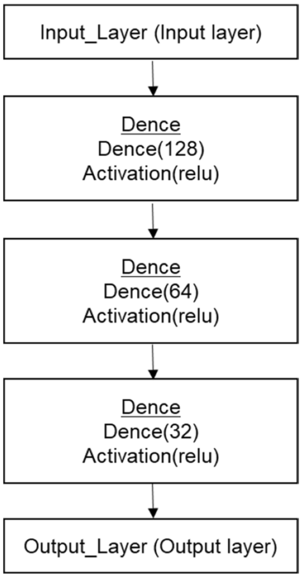


Fig. 5 DNN model composition

Table 3 DNN model result

	Small car	Large car
Number of training data	244,684	15,135
Number of test data	61,171	3,784
Learning rate	0.0005	0.0001
Batch size	128	128
epoch	500	1000
RMSE	0.0304	0.0432
MAE	0.0232	0.0324
Slope of yyplot	0.6526	0.8298

index are presented in Table 6, and the loss plot and yyplot are shown in Fig. 12. The loss plots show that the loss converged for the training and validation data, confirming that no over-fitting occurred. The inclination of the yyplot approaches 1.0 for the small and large vehicles, and the plots are concentrated on the diagonal. The 2DCNN model can predict with relatively high accuracy by learning from structured data with time-series information and spatial information of the vehicles in front and behind.

5.3 Discussions

Figures 13 and 14 compare the RMSE, MAE, and inclination of the yyplot for each NN method.

The DNN model showed the worst prediction accuracy compared to the other three models for the small and large vehicles. This is because unlike the LSTM, 1DCNN, and 2DCNN models, the time-series effect is not explicitly captured in DNN, although the dataset used in the DNN model was trained with historical vehicle information. This clearly shows the importance of structuring the time-series factors.

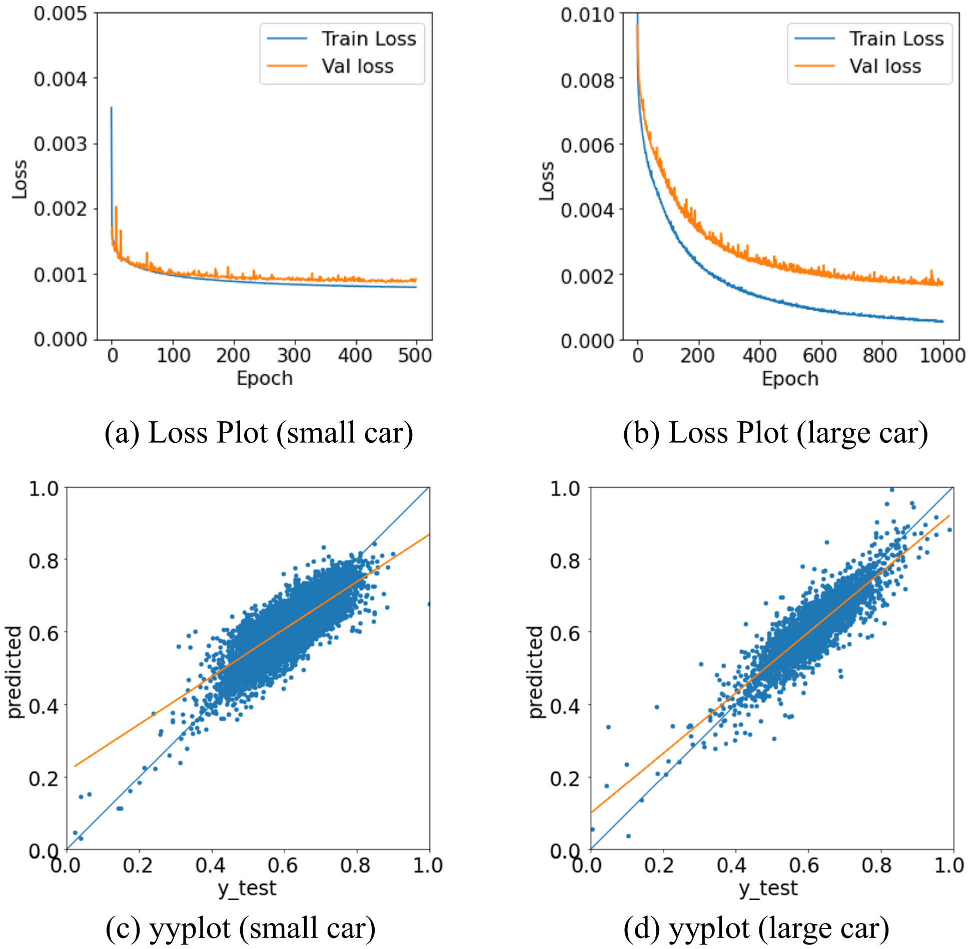
It was also shown that the prediction of heavy-duty vehicles was highly accurate even though the number of data was smaller than that of small vehicles. This may be because large vehicles are driven by a higher percentage of skilled drivers and have less variability in driving behavior. Among LSTM and CNNs, the LSTM model had the smallest error for the small and large vehicles, and the inclination of the yyplot was closest to 1. Additionally, the LSTM model's yyplot plots were more densely distributed on the diagonal than those of other methods. There were no significant outliers in the LSTM model, even in the rapid deceleration and acceleration regions. LSTM can capture the long-term memory, where the characteristics of current traffic states are represented, which may work well to predict the driving behaviors. The 2DCNN model gives the competitive prediction accuracy with the LSTM and higher accuracy than 1DCNN, even though it does not consider the long-term memory. This might be because the 2DCNN model is arranged to explicitly capture the spatial effect of traffic state and time-series effect. In this study, the LSTM model achieved the highest accuracy in predicting car-following behavior; however, further refinement of LSTM to explicitly capture the spatial effects would be expected to yield higher accuracy. LSTM can consider time-series data, but it cannot handle structured spatial data, so we believe that combining LSTM and CNN can construct a model that also considers spatial effects.

6 Conclusion

In this study, we developed a car-following behavior model using vehicle trajectory data on expressways from the data-driven approach. We applied a series of NN algorithms to predict the vehicle acceleration after 1 s. The main findings of this study are summarized as follows.

1. Capturing the time-series and spatial effects is essential in improving the prediction accuracy of acceleration behavior; however, the information of 5th and more ahead vehicles maneuver is not necessary as input variables.
2. Spatiotemporal data structuring and long-term memory are effective in improving prediction accuracy.
3. The highest prediction accuracy was obtained for the LSTM model, confirming the high-prediction accuracy in the low- and high-acceleration bands.

Fig. 6 Loss Plot and yyplot of DNN model



(a) Loss Plot (small car)

(b) Loss Plot (large car)

(c) yyplot (small car)

(d) yyplot (large car)

Fig. 7 LSTM model composition

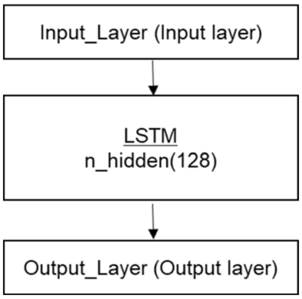


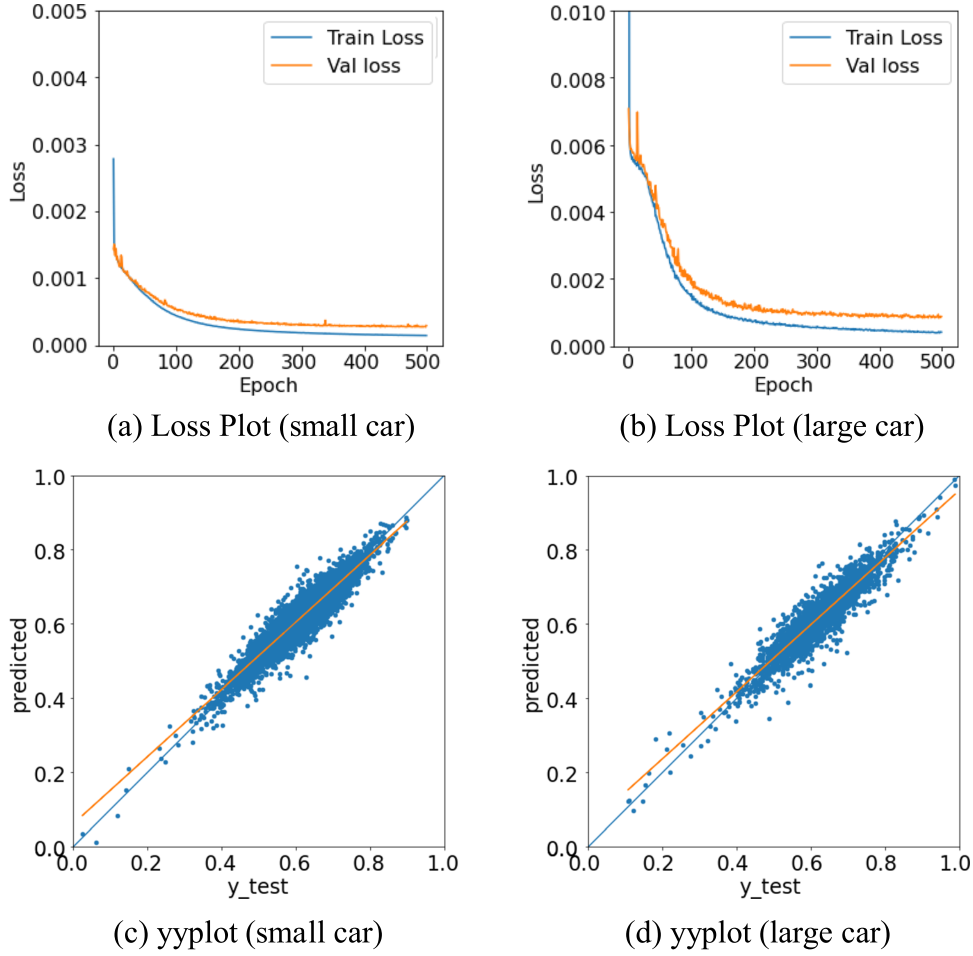
Table 4 LSTM model result

	Small car	Large car
Number of training data	225,883	14,021
Number of test data	56,471	3506
Time steps	30	30
Learning rate	0.001	0.0005
Batch size	128	128
epoch	500	500
RMSE	0.0173	0.0303
MAE	0.0130	0.0223
Slope of yyplot	0.9059	0.9038

4. The prediction accuracy of large vehicles is much higher than that of small vehicles, suggesting that the variability and heterogeneity in driving behavior significantly impact the accuracy of the models, which should be captured in the models.

These results show that a car-following behavior model based on deep learning using vehicle trajectory data can predict acceleration with high accuracy. It means that a data-driven approach for modeling car-following behavior can advance traffic simulation.

Fig. 8 Loss plot and yyplot of LSTM model



(a) Loss Plot (small car)

(b) Loss Plot (large car)

(c) yyplot (small car)

(d) yyplot (large car)

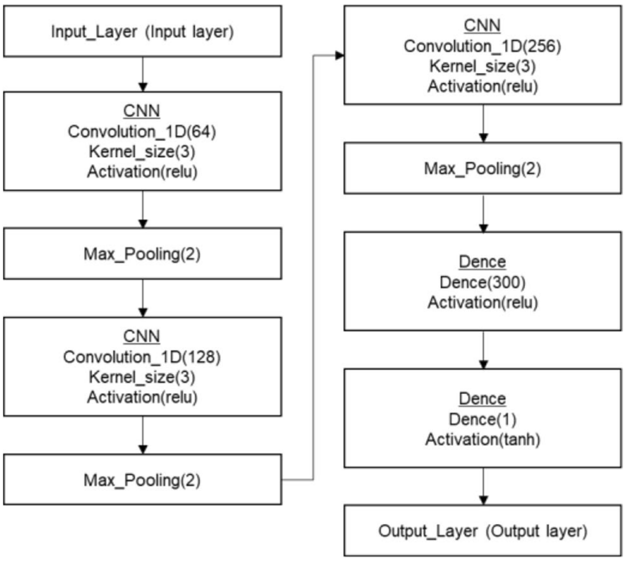


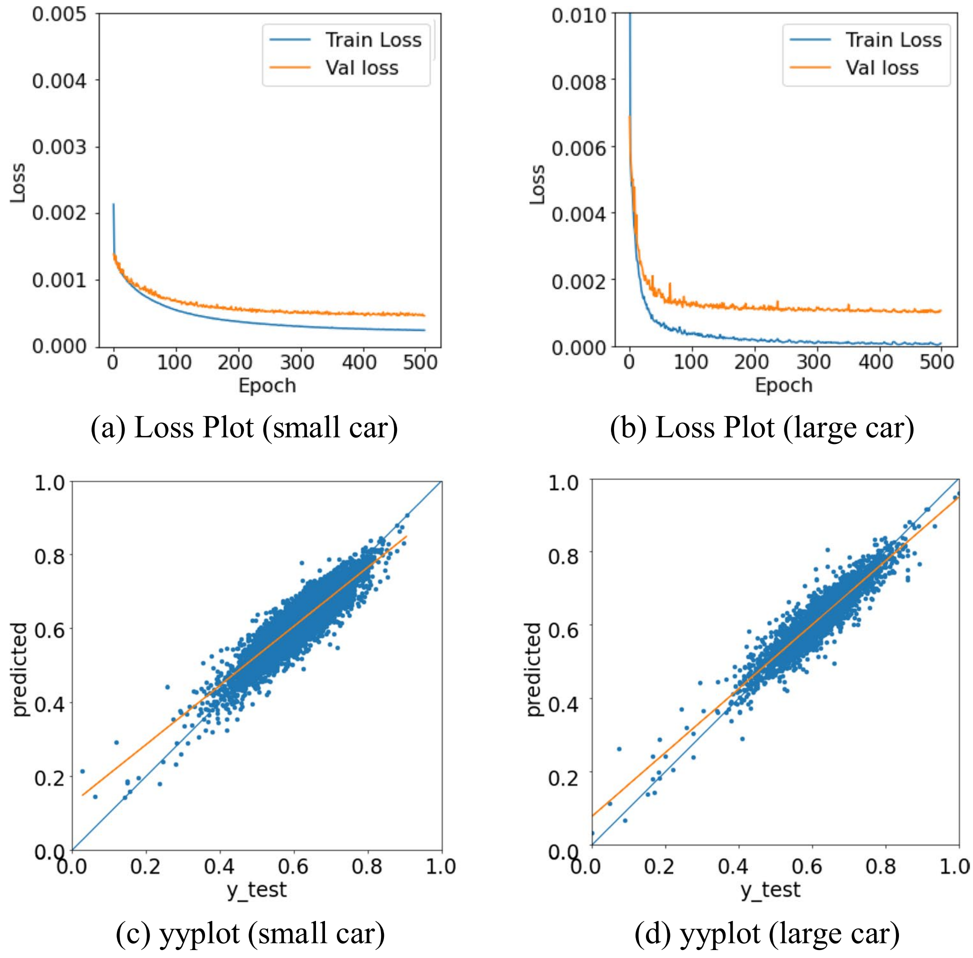
Fig. 9 1DCNN model composition

Table 5 1DCNN model result

	Small car	Large car
Number of training data	225,883	14,021
Number of test data	56,471	3,506
Time steps	30	30
Learning rate	0.001	0.0005
Batch size	128	128
Epoch	500	500
RMSE	0.0214	0.0313
MAE	0.0163	0.0230
Slope of yyplot	0.7990	0.8708

However, since the data used in this study was limited to the overtaking lane in a lane change prohibited zone, it is necessary to verify the prediction accuracy in other sections and at other times of the day. It is also necessary to analyze the cases where the LSTM model misses forecasts under what traffic conditions.

Fig. 10 Loss Plot and yyplot of 1DCNN model



(a) Loss Plot (small car)

(b) Loss Plot (large car)

(c) yyplot (small car)

(d) yyplot (large car)

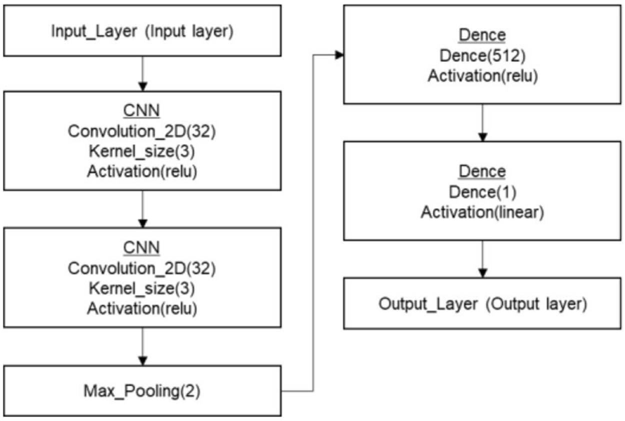
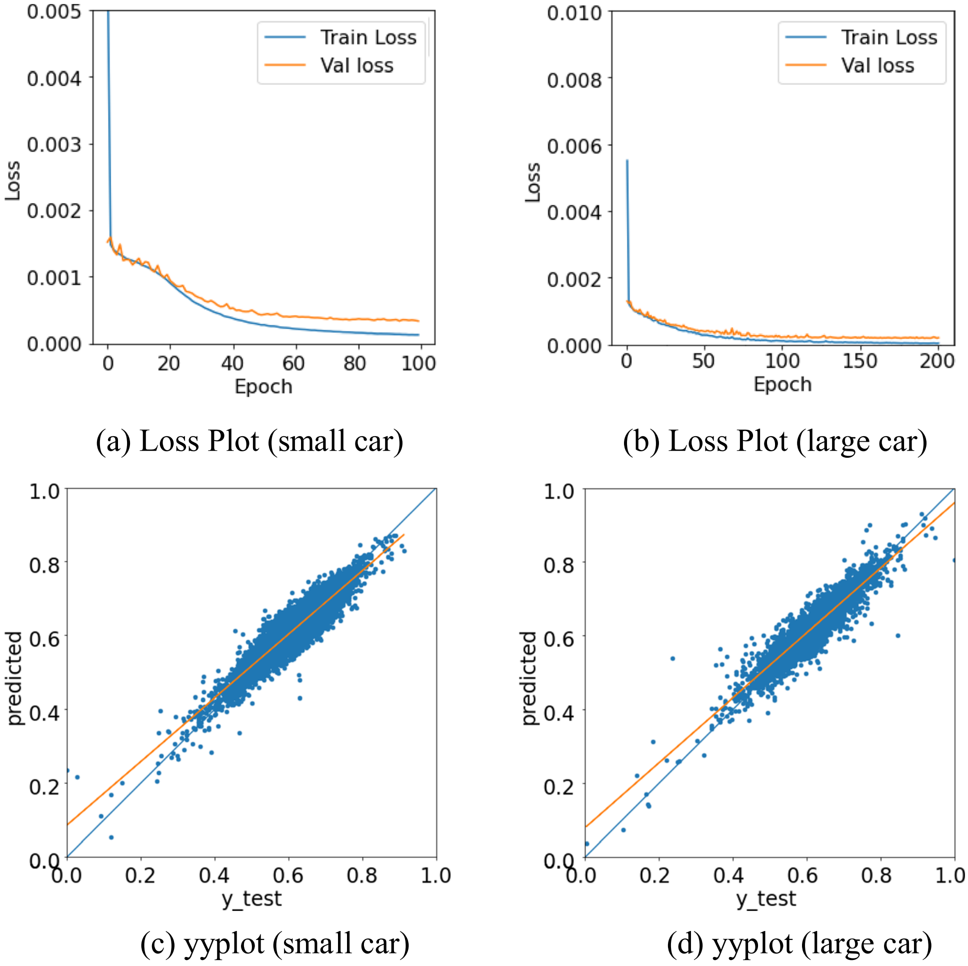


Fig. 11 2DCNN model composition

Table 6 2DCNN model result

	Small car	Large car
Number of training data	227,140	14,021
Number of test data	56,786	3,506
Time steps	30	30
Learning rate	0.001	0.0005
Batch size	256	128
epoch	100	200
RMSE	0.0183	0.0329
MAE	0.0139	0.0244
Slope of yyplot	0.8618	0.8804



(a) Loss Plot (small car)

(b) Loss Plot (large car)

(c) yyplot (small car)

(d) yyplot (large car)

Fig. 12 Loss Plot and yyplot of 2DCNN model

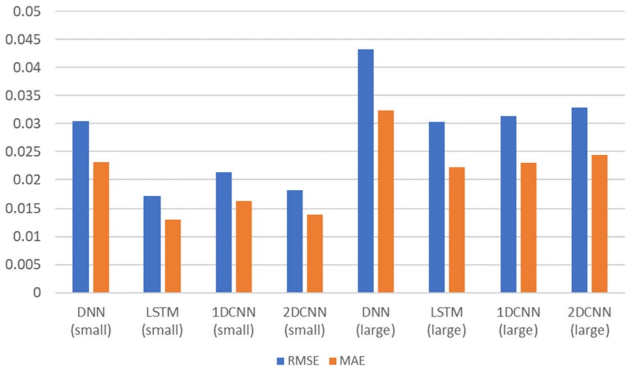


Fig. 13 RMSE, MAE of each NN method

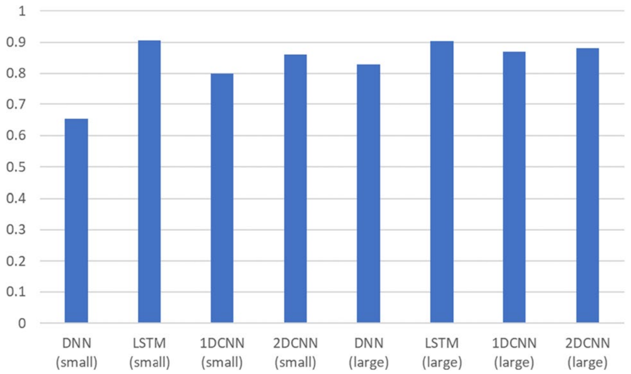


Fig. 14 Inclination of the yyplot for each NN method

In future studies, we will improve the car-following behavior model by increasing the training data and lengthening the time-series information to be considered.

Acknowledgements This research was supported by JSPS Grant-in-Aid for Scientific Research 19H02268. The vehicle trajectory data used in this study was provided by Hanshin Expressway Co. The authors wish to thank Hanshin Expressway Co.

Funding JSPS Grant-in-Aid for Scientific Research 19H02268.

Data Availability The data that support the findings of this study are available in Zen Traffic Data at https://zen-traffic-data.net, reference. These data are available for research and development organizations to contribute to the development of basic research, technology and services that will make road traffic more safe, secure and comfortable for the next generation.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. "Aimsun next users manual: Modeling vehicle movement." https://docs.aimsun.com/next/22.0.1/UsersManual/MicrosimulationModellingVehicleMovement.html#car_following_model (1st June, 2022)
2. Wiedemann, R.: "Simulation des Strassenverkehrsflusses." Schriftenreihe des Instituts fur Verkehrswesen der Universitat Karlsruhe, Vol. 8, Karlsruhe, Germany (1974)
3. Xing, J., Koshi, M.: A study on the Bottleneck phenomena and car-following behaviour on sags of motorways. Doboku Gakkai Ronbunshu. **506**(IV-26), 45–55 (1995)
4. Oguchi, T.: Analysis of bottleneck phenomena at basic freeway segments - car-following model and future exploration. Doboku Gakkai Ronbunshu. **660**(IV-49), 39–51 (2000)
5. Aghabayk Eagely, S. K., Sarvi, M., Young, W., Kautzsch, L.: A novel methodology for evolutionary calibration of vissim by multi-threading. In B. O'Keeffe (ed.) Australasian Transport Research Forum 2013 Proceedings, pp. 1–15. Australian National Audit Office (2013)
6. Rrecaj, A.A., Bombol, K.M.: Calibration and validation of the VISSIM parameters-state of the art. TEM J. **4**(3), 255–269 (2015)
7. Durrani, U., Lee, C., and Maoh, H.: Calibrating the Wiedemann's vehicle-following model using mixed vehicle-pair interactions. Transp. Res. C Emerg. Technol. **67**, 227–242 (2016)
8. Nakai, M., Nakazawa, K., and Shiomi, Y.: Representation of sag bottleneck phenomena with a commercial microscopic traffic simulator. Transp. Res. Procedia. **34**, 99–106 (2018)
9. Zhou, M., Qu, X., Li, X.: A recurrent neural network based microscopic car following model to predict traffic oscillation. Transp. Res. C Emerg. Technol. **84**, 245–264 (2017)
10. Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top. **62**(2 Pt A), 1805–1824 (2000)
11. Wang, X., Jiang, R., Li, L., Lin, Y., Zheng, X., Wang, F.-Y.: Capturing Car-Following Behaviors by Deep Learning. IEEE Trans. Intell. Transp. Syst. **19**(3), 910–920 (2018)
12. Fan, P., Guo, J., Zhao, H., Wijnands, J.S., Wang, Y.: Car-following modeling incorporating driving memory based on autoencoder and long short-term memory neural networks. Sustainability **11**(23), 1–15 (2019)
13. Koshi, M., Kuwahara, M., and Akahane, H.: Capacity of Sags and Tunnel on Japanese Motorways. ITE J. (Institute of Transportation Engineers). **62**(5), 17–22 (1992). https://doi.org/10.1109/ITSC.2010.5625133
14. Ossen, S., Hoogendoorn, S.P.: Driver heterogeneity in car following and its impact on modeling traffic dynamics. Transp. Res. Rec. J. Transp. Res. Board **1**, 95–103 (1999)
15. Sun, D.H., Liao, X.Y., Peng, G.H.: Effect of looking backward on traffic flow in an extended multiple car-following model. Physica A **390**(4), 631–635 (2011)
16. Hanshin Expressway Co: Zen Traffic Data, https://zen-traffic-data.net (1st June, 2022)
17. Treiber, M., Kesting, A.: Traffic flow dynamics. Springer, Germany, 157 (2012)
18. Goñi Ros, B., Knoop, V.L., Shiomi, Y., Takahashi, T., van Arem, B., Hoogendoorn, S.P.: Modeling traffic at sags. Int. J. Intell. Transp. Syst. Res. **14**(1), 64–74 (2016)
19. Schakel, W. J., van Arem, B., and Netten, B. D.: Effects of adaptive cruise control and cooperative adaptive cruise control on traffic flow. Proc. 13th International IEEE Annual Conference on Intelligent Transportation Systems, pp. 759–764 (2010). https://doi.org/10.1109/ITSC.2010.5625133
20. Degenhardt, F., Seifert, S., Szymczak, S.: Evaluation of variable selection methods for random forests and omics data sets. Brief. Bioinform. **20**(2), 492–503 (2019). https://doi.org/10.1093/bib/bbx124
21. Philipp Christian Petersen: Lecture notes for Neural network theory lecture, (2022).http://pc-petersen.eu/Neural_Network_Theory.pdf. Accessed 1 Mar 2023
22. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Comput. **9**(8), 1735–1780 (1997)
23. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Masahiro Kinoshita is a Master's course student at the Graduate School of Science and Engineering, Ritsumeikan University. He received his Bachelor of Engineering degree from Ritsumeikan University in 2022.

Yasuhiro Shiomi is a Professor at Ritsumeikan University. He received his Doctor of Engineering degree from Kyoto University in 2008. His research interests include traffic flow analysis and freeway operation and management.