Transportation Research Part C 106 (2019) 41–60

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

# A data-driven lane-changing model based on deep learning☆

Dong-Fan Xie[a,b], Zhe-Zhe Fang[a], Bin Jia[a], Zhengbing He[c,*]

[a] Institute of System Science, Beijing Jiaotong University, China
[b] Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Beijing Jiaotong University, China
[c] Beijing Key Laboratory of Traffic Engineering, College of Metropolitan Transportation, Beijing University of Technology, China

ARTICLE INFO

*Keywords:*
Traffic flow
Driving behavior
Vehicle trajectory
Deep belief network
Long short-term memory

ABSTRACT

Lane-changing (LC), which is one of the basic driving behavior, largely impacts on traffic efficiency and safety. Modeling an LC process is challenging due to the complexity and uncertainty of driving behavior. To address this issue, this paper proposes a data-driven LC model based on deep learning models. Deep belief network (DBN) and long short-term memory (LSTM) neural network are employed to model the LC process that is composed of LC decisions (LCD) and LC implementation (LCI). The empirical LC data provided by Next Generation Simulation project (NGSIM) is utilized to train and test the proposed DBN-based LCD model and LSTM-based LCI model. The results indicate that the proposed data-driven model is able to accurately predict the LC process of a vehicle. The sensitivity analysis shows that the most important factor associated with LCD is the relative position of the preceding vehicle in the target lane. This may be the first work that comprehensively models LC using deep learning approaches.

## 1. Introduction

Lane-changing (LC) is one of the most basic driving behavior. It takes place when a vehicle moves from one lane to another lane, with the purpose of improving driving conditions or leaving a road. Both longitudinal and lateral movements are involved in an LC process, making LC closely relate to traffic safety (Van Winsum et al., 1999; Mattes, 2003; Pande and Abdel-Aty, 2006; Zheng et al., 2010) and significantly impact on traffic flow dynamics (Kerner and Rehborn, 1996; Mauch and Cassidy, 2002; Cassidy and Rudjanakanoknad, 2005; Laval and Daganzo, 2006; Ahn and Cassidy, 2007; Zheng, 2014; Yang et al., 2019). Therefore, LC modeling is significant for both transportation research and practice.

Generally speaking, an LC process can be divided into the following two stages (Kesting et al., 2007). The first stage is regarded as LC decisions (LCD), in which a driver is mentally motivated to change lanes based on the surrounding traffic; The second stage, which is called as LC implementation (LCI), is the physical process that a vehicle moves from the current lane to the target lane.

As one of the most important branches of transportation research, a number of LC models have been proposed over the past decades. Basically, those models can be divided into two categories, i.e., analytical models and data-driven models. The most prevailing analytical LCD models include rule-based models (Gipps, 1986; Hidas, 2002, 2005; Laval and Leclercq, 2008; Kesting et al., 2007; Choudhury and Ben-Akiva, 2013) and discrete choice-based models (Ahmed et al., 1996; Toledo et al., 2003; Sun and Elefteriadou, 2011, 2012, 2014; Knoop and Buisson, 2015; Lint et al., 2016; Keyvan-Ekbatani et al., 2016b; Knoop et al., 2018). The data-driven LCD models are proposed based on Neural Networks (NN), Support Vector Machine (SVM), and Bayesian Filtering (BF)

☆ This article belongs to the Virtual Special Issue on Traffic flow modeling.

* Corresponding author.
*E-mail addresses:* dfxie@bjtu.edu.cn (D.-F. Xie), he.zb@hotmail.com (Z. He).

https://doi.org/10.1016/j.trc.2019.07.002
Received 30 October 2018; Received in revised form 11 June 2019; Accepted 10 July 2019
Available online 17 July 2019

(Hunt and Lyons, 1994; Kumar et al., 2013; Dou et al., 2016; Li et al., 2016b,a; Gao et al., 2018). The analytical or data-driven studies regarding LCI are relatively limited, probably due to the difficulties of large-scale collections of LCI data (Enke, 1979; Nelson, 1989; Chovan et al., 1994; Shamir, 2004; Yao et al., 2013; Ding et al., 2013; Zhou et al., 2017). More details regarding the LCD and LCI models will be reviewed in Section 2. One may also refer to literature review articles such as Moridpour et al. (2010), Rahman et al. (2013), and Zheng (2014).

Although a number of LCD and LCI models have been proposed, there are still many open questions for understanding LC behavior. The problems that are closely related to this study are summarized as follows.

- The prediction accuracy of the analytical LC models can only reach 70–80% (Olsen, 2013), and there are large discrepancies between the principles modeled and observations (Knoop et al., 2012). Obviously, more accurate models are expected.
- Only a little literature that carefully investigates LCI is found. For the analytical models, it is difficult to accurately consider the diversity and uncertainty of the LC process (Enke, 1979; Chovan et al., 1994; Shamir, 2004; Zhou et al., 2017). For the data-driven models, only the impact factors at a specific time is considered so far (Ding et al., 2013), while it is known that the historical data that is immediately before driver's instantaneous motion also plays an important role (Wang et al., 2019).
- Most of the analytical and data-driven models separately consider LCD or LCI, which cannot reproduce a complete LC process that is composed of LCD and LCI and its impact on traffic flow.

Recently emerging deep learning approaches provide us with a new opportunity to understand various traffic-related phenomena. For example, deep belief network (DBN) was reported to reduce traffic flow prediction errors by more than 50% compared with autoregressive integrated moving average model (ARIMA) (Arief et al., 2016). Long short-term memory (LSTM) NN, which is a typical recurrent NN, has been extensively applied in traffic flow modeling (Huang et al., 2018) and real-time prediction (Zhao et al., 2017; Jia et al., 2017); Convolutional Neural Networks (CNN) achieved great success in traffic sign recognition (Zhang et al., 2017), traffic speed prediction (Ma et al., 2017), and transportation demand prediction (Lin et al., 2018). A large number of studies have sufficiently demonstrated the outperformance of deep learning approaches. The success of deep learning is also one of the motivations of the study, i.e., whether could deep learning improve LC modeling?

To address the abovementioned problems, this paper proposes an integrated LC model[1] for the whole LC process of a vehicle. The contributions of the model are as follows.

- Advanced deep learning approaches, i.e., DBN and LSTM NN, are employed to depict LCD and LCI, respectively, making the proposed model more accurate in describing and predicting the LC process. In particular, the LSTM-based LCI model takes into account a series of historical trajectory data rather than the instantaneous data at a specific time. This may be the first work that comprehensively models the LC process using deep learning approaches, based on our literature review presented in Section 2 and other review articles (Moridpour et al., 2010; Rahman et al., 2013; Zheng, 2014).
- Both the LCD model and the LCI model are proposed in a modeling framework, meaning that these two models can be applied in one simulation environment. Therefore, it is an integrated LC model.
- The proposed model is able to excavate the underlying features from high-dimensional data, making it possible to analyze and unveil the impacts of various impact factors on LC. The proposed model shows the most important factor regarding LCD is the relative position of the preceding vehicle in the target lane. Moreover, compared with speed, drivers pay more attention on the relative positions of the surrounding vehicles in making LCD.

In summary, this paper proposes a deep learning-based LC model that can accurately capture LC behavior for the first time. The proposed LCD and LCI models can be easily trained using real-world vehicle trajectory data and then be applied in microscopic traffic flow simulations. The proposed model also sheds light on the underlying mechanism of LC behavior, which is important basis of LC modeling.

The rest of the paper is organized as follows. Section 2 introduces the existing studies regarding the LCD models, the LCI models, and the data-driven LC models. Section 3 proposes the data-driven LC model including the DBN-based LCD model and the LSTM-based LCI model. Section 4 carefully evaluates the proposed models by using empirical LC data. Finally, conclusions are made in Section 5.

## 2. Literature review

### 2.1. LCD models

The rule-based and discrete choice-based models may be the most prevalent analytical LCD models. Both of them can be easily embedded into microscopic traffic flow models for multi-lane traffic simulations and thus the LC impact on traffic flow can be reflected.

[1] LC is usually categorized into discretionary LC (to obtain a better driving condition) and mandatory LC (to move to the desired destination) (Rahman et al., 2013; Zheng, 2014). Only the discretionary LC is considered in this study.

#### 2.1.1. Rule-based models

In general, the rule-based LCD models are proposed from the perspective of lane selections and gap acceptances (Choudhury and Ben-Akiva, 2013). Gipps (1986) developed one of the most classic LCD models with a two-level LC framework. The model involves a series of LC rules. However, the rules are deterministic, and some essential factors are ignored, such as the LC trade-offs, driver heterogeneity, and drivers' time-dependent behavior.

Following Gipps' seminal model, many efforts have been made to consider more traffic situations. For example, Hidas (2002, 2005) proposed a framework by classifying the LCD into three cases, namely, free, cooperative and forced LC cases. The models assume that the followers on the target lane would accept a certain maximum speed decrease, while ignore the other factors related to the followers. Kesting et al. (2007) developed an LCD model, named as Minimizing Overall Braking Induced by Lane change (MOBIL), by integrating LC benefits and risks into an acceleration function based on car-following models. Due to the acceleration-based rules in MOBIL, it can be easily integrated into a car-following framework. However, the MOBIL was not calibrated or tested in real traffic conditions. Laval and Leclercq (2008) recently incorporated lane-specific macroscopic quantities into microscopic LC modeling to consider various traffic situations.

#### 2.1.2. Discrete choice-based models

Different from the rule-based LCD models, the discrete choice-based models are proposed based on LC probability and utility theory. Ahmed et al. (1996) may be one of the earliest works proposing the discrete choice-based LCD model. The study takes into account driver heterogeneity by defining a utility function for making LCD. Similar models are proposed by taking into account various factors of the subject vehicle and its surrounding vehicles. For instance, Toledo et al. (2003) proposed an integrated utility function for both mandatory and discretionary LCD. The result indicates the necessity of considering the trade-offs between mandatory and discretionary LCD.

Apparently, driver characteristics (such as aggressiveness, gender, and alertness) have significant impact on the utility function of LCD models. Unfortunately, many of them are neglected in most of the LCD models. To address this problem, Sun and Elefteriadou (2011, 2012) carried out a series of studies with consideration of driver types and various impact factors for each driver type in making LCD. They extracted the most important factors to define the utility function for making LCD. Furthermore, Sun and Elefteriadou (2014) applied a similar approach to develop the LCD model for the vehicles on urban arterial streets. With the consideration of driver characteristics, it was reported the models can better reproduce LC in real traffic. Nevertheless, it may be difficult to incorporate the model into traffic simulations due to the fact that many scenarios with distinct impact factors are involved.

In addition, Keyvan-Ekbatani et al. (2016b) summarized the four distinct LC strategies revealed in real traffic, and investigated the impact of these strategies on traffic dynamics based on a microscopic simulation tool called MOTUS. Lint et al. (2016) developed an advanced open-source simulation framework called OpenTraficSim, by using which the LC behavior can be modeled and validated. Although various probabilistic LC models are proposed, calibration and validation are not commonly conducted for those analytical models. In addition, it is not clear what calibration and validation entail for probabilistic LC models. To address this problem, Knoop and Buisson (2015) illustrated a calibration and validation process for a probabilistic discretionary LC model, indicating that it is the best to use physically interpretable measures during validation.

Although a number of LCD models have been developed to capture LCD behavior, few studies are dedicated to investigate LCD behavior except Keyvan-Ekbatani et al. (2016a) and Knoop et al. (2018) that carried out an online survey and analyzed the driving strategies during LC in detail.

### 2.2. LCI models

Different from the extensively studied LCD models, only a few models are proposed to capture LC dynamics. Early studies attempt to describe LC dynamics by using mathematical functions. For example, Enke (1979), Nelson (1989), Chovan et al. (1994), and Shamir (2004) described LC trajectories using quantic polynomials, sinusoidal and trapezoidal functions, etc. In general, these models can provide smoothing LC trajectories and well fit real traffic data. Thus, they can be applied in trajectory planning for autonomous vehicles. However, most of them do not have clear physical meanings and thus cannot be used to interpret drivers' LC behavior. Furthermore, they are developed based on simulation data with no consideration of the heterogeneity of LC processes.

Most recently, Zhou et al. (2017) carried out a field experiment with 11 taxis in Beijing, China. Reference angle data is extracted from the videos recorded by the cameras installed in the taxis, and an LC trajectory model is proposed only based on the reference angle. Nevertheless, the data used in the study is only from those 11 taxis, making the result not very solid. In addition, this modeling approach may not be easily incorporated into the existing traffic flow models.

### 2.3. Data-driven LC models

In contrast to the analytical LC models, the data-driven LC models are proposed based on data training and machine learning. The emphasis is to extract LC dynamics from massive data instead of describing the nature of things.

Among the data-driven LCD studies, NN is the most popular tool so far. For example, Hunt and Lyons (1994) employed NN to model LCD on dual carriageways. In the model, LCD are predicted based on surrounding dynamic traffic. Both simulated and real traffic data are used to demonstrate the NN-based model. The amount of empirical traffic data is very limited due to the difficulty of high-fidelity data collection in the 1990s. The results show that the model based on simulated data performs better than that based on real data, indicating the necessity of sufficient data for NN-based models.

To further improve the prediction accuracy, several studies combined NN with other machine learning approaches. For example, Dou et al. (2016) proposed a mandatory LC prediction model by combining NN with SVM at highway lane drops. The prediction accuracy reaches 94% for non-merging behavior and 78% for merging behavior, through inputting speed difference, vehicle gap, and the positions. However, the employed NN structure is not clearly discussed. Li et al. (2016b) employed BF and NN to predict the LCD. The parameters involved are associated with lane line sensors, steering wheel angle sensors, and in-vehicle CAN bus acquisition characterization. The accuracy reaches 91.38% based on real data.

In addition to NN-based LCD models, Kumar et al. (2013) proposed an LC behavior prediction model based on SVM and BF. To recognize driver's LC intention, Li et al. (2016a) developed an LCD model by combining the hidden Markov model with BF. The inputs of the model are steering angles, lateral acceleration and yaw rates, and the output is behavior classification. The results show that the proposed model can achieve an average prediction accuracy of 91.9%.

More recently, rapid development of deep learning provides us a new opportunity to make a step forward in this direction of data-driven modeling. Gao et al. (2018) proposed a group-wise convolutional NN framework to predict LC before the event actually occurs. In particular, the inputs of this model are three types of physiological signals (i.e., electrocardiogram, galvanic skin response, and respiration rate) from the driver. The deep learning-based LC studies are few, probably due to the fact that the deep learning is a relatively new concept.

Compared with the data-driven LCD models, only a few data-driven studies are found to investigate an LCI process. For example, to evaluate the risk of making LC, Yao et al. (2013) developed an LC trajectory prediction model by integrating the $k$ nearest real LC instances. Considering the fact that some LDI models can only predict the future positions of an LC vehicle, Ding et al. (2013) developed a two-layer Tansig and Linear Backpropagation (BP) NN model to predict LC trajectories in a real-time manner. Although those models can well predict the trajectories of LC vehicles, all of them need to know LCD first. It turns out that those models cannot be used if LCD are not made or predicted in advance.

In summary, although many data-driven LC models have been proposed, the following gaps are found.

- Most of the existing data-driven LC models are still based on traditional machine learning methods such as NN, SVM, and BF. Few LC models are proposed based on deep learning approaches, although deep learning approaches are attracting unprecedented attentions in various fields.
- Most of the existing models are proposed to describe LCD making. Although the LCD and LCI are closely related parts in an LC process, an integrated LCD and LCI model is still lacking.
- Compared with the analytical models, the traditional machine learning methods have improved the LCD prediction accuracy (80–90%). The prediction could be further improved by taking advantage of recently prevailing and powerful deep learning approaches.
- Most of the existing data-driven models do not explicitly analyze the factors that impacting on LCD and LCI, making latent LC mechanism unclear.

## 3. Data-driven LC model

### 3.1. LC stages

An LC process usually lasts for several seconds as will be shown in Section 4.1 and also reported in Toledo and Zohar (2007) and Yang et al. (2015). Basically, it begins with the generation of LC motivations and completes when the vehicle is laterally stable in the target lane. The process composed of LCD and LCI stages can be modeled as follows.

**Stage 1: LCD**. This stage begins with the generation of LC motivations and ends with a decision made. In the decision-making process, an LC intention is first generated. Then, LC benefits and safety conditions are evaluated by considering the surrounding traffic conditions. Finally, a decision, i.e., lane-keeping (LK) or LC, is made. The binary decision is written as follows.

$$Dec = \begin{cases} 1, & \text{lane-keeping} \\ 2, & \text{lane-changing} \end{cases} \tag{1}$$

**Stage 2: LCI**. The LCI is comprised of two sub-processes in practice. First, the LC driver steers his/her vehicle to move longitudinally and laterally towards the target lane until the whole body of the vehicle is within the target lane. Then, the driver rotates the steering wheel to complete the lateral shift and then the vehicle moves directly along the target lane. Accordingly, the trajectory of an LC vehicle during the LC process can be formulated as follows.

$$\begin{cases} x(t+\Delta t) = x(t) + \bar{v}_x(t)\Delta t \\ y(t+\Delta t) = y(t) + \bar{v}_y(t)\Delta t \end{cases} \tag{2}$$

where $\Delta t$ is the length of each time step; $x(t)$ and $y(t)$ are the longitudinal position and lateral position of the vehicle at time $t$, respectively; $\bar{v}_x(t)$ and $\bar{v}_y(t)$ are the longitudinal and lateral average speed of the vehicle during the time step $\Delta t$ just after time $t$, respectively.

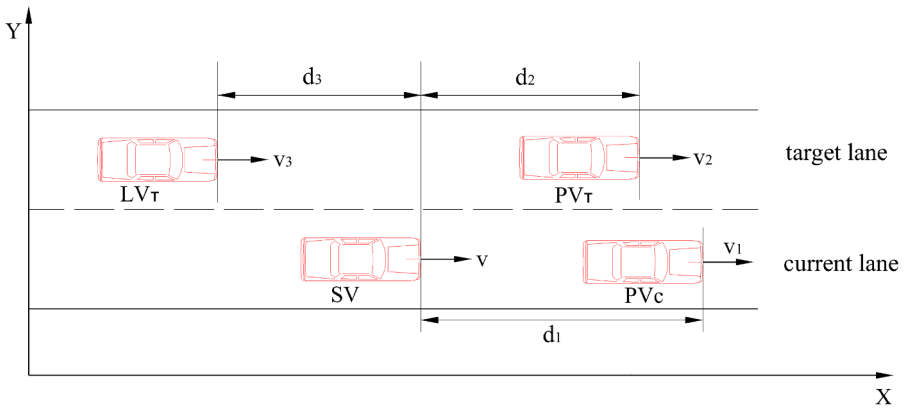


**Fig. 1.** A schematic diagram of an LCD scenario.

### 3.2. Notations

The notations used in this study are listed as follows (also see Fig. 1).

- *SV*, the subject vehicle of changing lanes;
- $PV_C$, the preceding vehicle on the current lane;
- $PV_T$, the preceding vehicle on the target lane;
- $LV_T$, the lag vehicle on the target lane.
- $d_1$, $d_2$, $d_3$, the space headway between *SV* and $PV_C$, $PV_T$, $LV_T$, respectively;
- $v$, $v_1$, $v_2$, $v_3$, the velocity of vehicles *SV*, $PV_C$, $PV_T$, and $LV_T$.

The velocity differences are calculated as follows.

$$\begin{cases} \Delta v_1 = v_1 - v \\ \Delta v_2 = v_2 - v \\ \Delta v_3 = v_3 - v \end{cases} \tag{3}$$

### 3.3. Modeling framework

Fig. 2 presents the framework of the proposed LC model, which contains three parts: data preparation, LCD model, and LCI model. In detail, normalized trajectory data is first prepared for training and testing of the data-driven LC model; Then, the DBN-based LCD model is employed to predict driver's LCD; If the driver decides to change lanes, the proposed LSTM-based LCI model is applied to generate the LC trajectory; Otherwise, a car-following model is used to depict the moving dynamics of the LK vehicle (note that a car-following model is not discussed here, since it is out of the scope of this study).

It is necessary to clarify that the proposed model can be easily implemented in a multi-lane microscopic simulation platform. In detail, we introduce a state label for each vehicle, namely, 'LC' or 'LK'. Before updating vehicle positions, we first update the state labels of all vehicles as follows.

- Label update. For each vehicle with the state label of 'LK', the proposed LCD model is applied to determine LC or LK. If LC is obtained, we set the state label as 'LC'; Otherwise, keep it as 'LK'. For each vehicle with the state label of 'LC', we first determine whether the LC process is completed. If yes, change the state label to be 'LK'; Otherwise, keep it as 'LC'.
- Position update. The proposed LCI model is applied to update the position of each vehicle with the label of 'LC'; A car-following model is applied to update the position of each vehicle with the label of 'LK'.

### 3.4. DBN-based LCD model

DBN is a generative graphical model developed by Hinton et al. (2006). It is regarded as a composition of Restricted Boltzmann Machines (RBMs) (Hinton, 2009). A RBM is a generative energy-based model with a visible input layer, a hidden layer, and connections between but not within layers. The hidden layer of each sub-network serves as the visible layer for the next. Therefore, DBN can be trained layer-by-layer, which significantly promotes the training process. In addition, DBN have simple network structure which can easily be designed in applications. In particular, the existing studies have been demonstrated that DBN can well predict stochastic events such as traffic accidents (Arief et al., 2016; Zhang et al., 2018). Therefore, this paper employs DBN to model LCD.

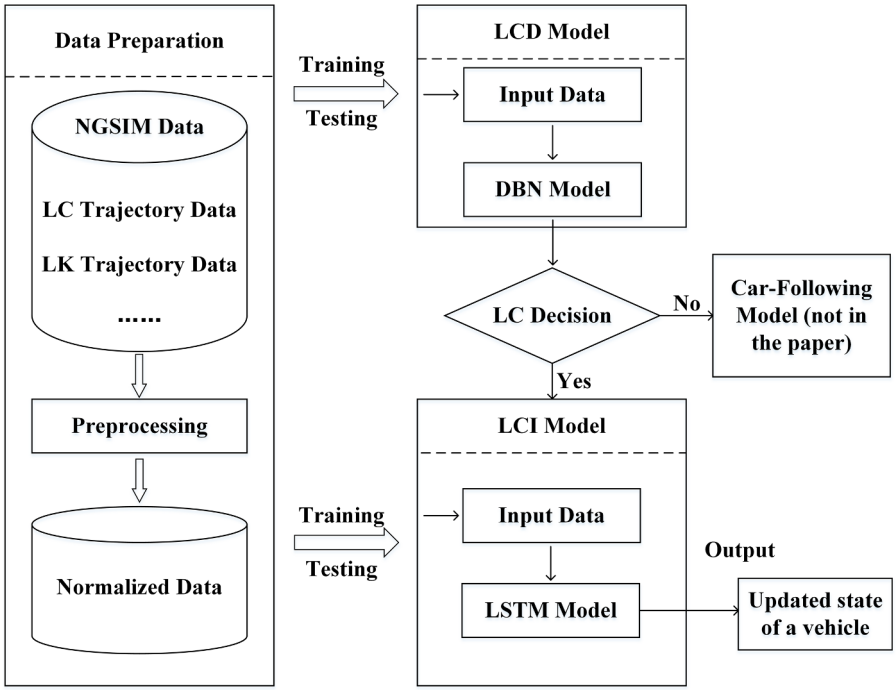


**Fig. 2.** Modeling framework based on deep learning.

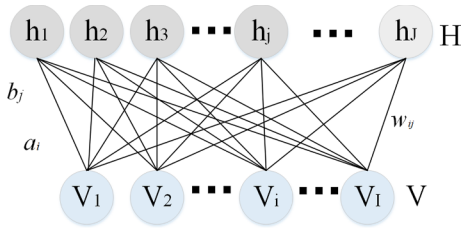


**Fig. 3.** Structure of the restricted Boltzmann machine.

Fig. 3 presents a schematic diagram of a RBM in which there are $I$ neurons in visible layer $V$ and $J$ neurons in hidden layer $H$. The parameter vector can thus be obtained as follows.

$$\theta = \{\omega_{ij}, a_i, b_j\}, \quad i = 1, 2..., I, \quad j = 1, 2, ..., J \tag{4}$$

where $\omega_{ij}$ is the weights between visible unit $i$ and hidden unit $j$; $a_i$ is the bias threshold of visible unit $i$; $b_j$ is the bias threshold of hidden unit $j$. During the training process, the parameters $\theta$ are modified based on the sample data. Finally, the well trained parameters $\theta$ can make the probability distribution that is represented by the corresponding RBM consistent with the training sample data as far as possible.

It is known that the DBN training is unsupervised learning, and thus an NN structure based on the BP algorithm is added to the last layer for supervised learning. The DBN structure in the LCD model is proposed as shown in Fig. 4, and the DBN-based LCD model is written as follows.

$$Dec = f_{LCD}(\mathbf{Z}_{LCD}|\theta) \tag{5}$$

where $Dec$, which is defined in Eq. (1), is the output variable of the DBN; $\mathbf{Z}_{LCD}$ is the input vector of the DBN, which is

$$\mathbf{Z}_{LCD} = \{z_1, z_2, ..., z_n\} \tag{6}$$

where $n$ in the number of input variables.

It should be noted that the hidden layer of the last RBM and the output layer of DBN constitute a BP NN, which can be trained by the BP algorithm. The last output of RBM is the input of BP NN, and then the BP NN is trained with supervision to achieve the purpose of sample classification. In particular, the BP NN errors can also be transmitted to the previous RBM, so that the whole DBN can achieve the optimized performance.

The following widely-used Sigmoid function is employed to be the transformation function

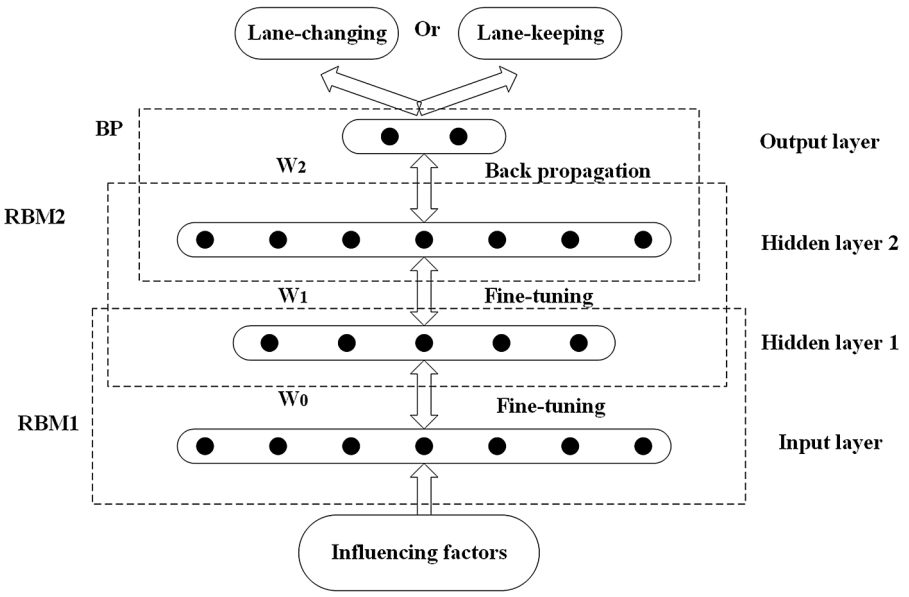


**Fig. 4.** Structure of the DBN in the LCD model.

$$g(z) = \frac{1}{1 + e^{-z}}. \tag{7}$$

Taking advantage of deep learning models, we input all variables regarding the traffic conditions on both the current and target lanes. The output of the LCD model is the binary decision that changing lane or not, as presented in Eq. (1). Table 1 presents the inputs and outputs in details. In particular, the input variables can be written as follows.

$$\mathbf{Z}_{LCD} = \{v(t_{LC} - t),\ \Delta v_1(t_{LC} - t),\ \ \Delta v_2(t_{LC} - t),\ \ \Delta v_3(t_{LC} - t),\ d_1(t_{LC} - t),\ \ d_2(t_{LC} - t),\ \ d_3(t_{LC} - t)\} \tag{8}$$

where $(t_{LC} - t)$ indicates the selected variables that are $t$ seconds before the LC time $t_{LC}$.

The widely used Mean Squared Error (MSE) is taken as the performance index in traing and testing the proposed DBN-based LCD model. It reads

$$MSE = \frac{\sum_{k=1}^{M} (Dec_k - Dec_k')^2}{M} \tag{9}$$

where $M$ is the total number of trajectories that are used in the training and testing; $Dec_t$ and $Dec_t'$ are the ground-truth and predicted binary decision values at time step $k$, respectively. The goal of training the DBN-based model is to obtain the optimal $\theta$ such that the MSE can be minimized. It reads

$$\min_{\theta \in \Omega} = MSE_{\theta} \tag{10}$$

where $\Omega$ is the solution space of $\theta$.

Furthermore, to intuitively unveil the performance of the proposed LCD model, the following prediction accuracy rate $R_A$ is proposed in the training and testing of the model.

**Table 1**
Inputs and outputs of the DBN-based LCD model.

| | Variables | Variable descriptions |
|---|---|---|
| Inputs | $v(t_{LC} - t)$ | Velocity of $SV$ |
| | $\Delta v_1(t_{LC} - t)$ | Velocity differences between $SV$ and $PV_c$ |
| | $\Delta v_2(t_{LC} - t)$ | Velocity differences between $SV$ and $PV_T$ |
| | $\Delta v_3(t_{LC} - t)$ | Velocity differences between $SV$ and $LV_T$ |
| | $d_1(t_{LC} - t)$ | Space headway between $SV$ and $PV_c$ |
| | $d_2(t_{LC} - t)$ | Space headway between $SV$ and $PV_T$ |
| | $d_3(t_{LC} - t)$ | Space headway between $SV$ and $LV_T$ |
| Outputs | $Dec$ | Binary decision: changing lane or not (Eq. (1)) |

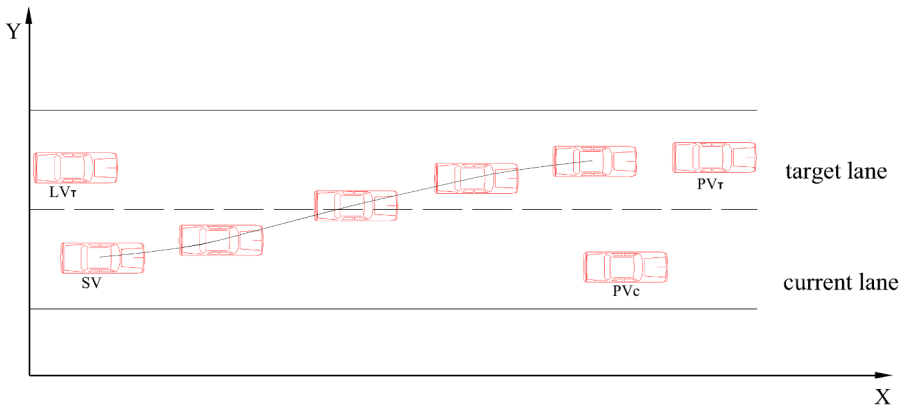


**Fig. 5.** A schematic diagram of an LCI scenario.

$$R_A = \frac{N_{pred}}{N_{sample}} 100\% \tag{11}$$

where $N_{pred}$ is the number of predictions that are consistent with the actual value of the test samples, and $N_{sample}$ is the total number of the test samples.

Regarding the DBN structure, the following two aspects are considered. One is the number of the hidden layers contained in the DBN; The other is the number of neurons in each hidden layer. Generally speaking, it is difficult to find the best values for the parameters in a data-driven model. Therefore, an iterative method is applied to determine the parameters in the study; see Appendix A.

### 3.5. LSTM-based LCI model

A vehicle moves on a two-dimensional plane during an LC process, rather than a one-dimension car-following motion. It is closely related to the status of the subject vehicle and its surrounding vehicles (Fig. 5).

Historical movements that are seconds before current time should be taken into account, since LC is a continuous driving process. Therefore, we employ LSTM to model the two-dimensional LC trajectory by taking its advantage in considering historical data.

Fig. 6 presents the structure of the LSTM-based LCI model, which is composed of three layers, namely, an input layer, an output layer, and a hidden layer. The input layer is designed to receive traffic state information of the subject vehicle and its surrounding vehicles, and the output layer exports the predicted trajectory of the subject vehicle. In particular, the neurons of the hidden layer connect each other to form a directed cycle, creating an internal state of the network.

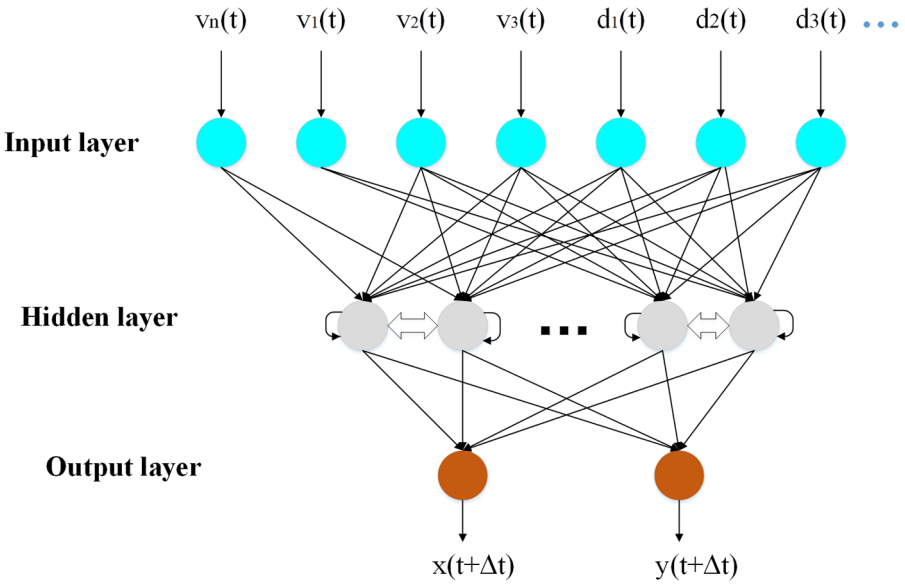


**Fig. 6.** Structure of the LSTM for the LCI model.

**Table 2**
Inputs and outputs of the LSTM-based LCI model.

| | Variables | Variable descriptions |
|---|---|---|
| Inputs | $v(t)$ | Velocity of $SV$ |
| | $v_1(t)$ | Velocity of $PV_c$ |
| | $v_2(t)$ | Velocity of $PV_T$ |
| | $v_3(t)$ | Velocity of $LV_T$ |
| | $d_1(t)$ | Space headway between $SV$ and $PV_c$ |
| | $d_2(t)$ | Space headway between $SV$ and $PV_T$ |
| | $d_3(t)$ | Space headway between $SV$ and $LV_T$ |
| | $x(t)$ | Longitudinal position of the subject vehicle |
| | $y(t)$ | Lateral position of the subject vehicle |
| | $X$ | Historical longitudinal positions |
| | $Y$ | Historical longitudinal positions |
| Outputs | $x(t+\Delta t)$ | Longitudinal position of the subject vehicle at next time step |
| | $x(t+\Delta t)$ | Lateral position of the subject vehicle at next time step |

The proposed LSTM-based LCI model is written as follows.

$$\{x(t+\Delta t), y(t+\Delta t)\} = f_{LDI}(\mathbf{Z}_{LCI}|\gamma) \tag{12}$$

where $\gamma$ is the parameter vector of the LSTM; $\mathbf{Z}_{LCI}$ is the vector of the decision variables that is generalized as follows.

$$\mathbf{Z}_{LCI} = \{z_1, z_2, ..., z_m\} \tag{13}$$

where $m$ is the number of the decision variables.

Table 2 lists the inputs and outputs of the LSTM-based LCI model (Eq. (12)). In detail, the inputs are written as follows.

$$\mathbf{Z}_{LCI} = \{v(t),\ v_1(t),\ v_2(t),\ v_3(t), d_1(t),\ d_2(t),\ d_3(t), x(t),\ y(t),\ \mathbf{X},\ \mathbf{Y}\} \tag{14}$$

where the positions (including both the longitudinal and lateral positions) of $K$ preceding time steps are employed to be the model inputs. They are written as follows.

$$\begin{aligned}\mathbf{X} &= \{x(t-\Delta t),\ x(t-2\Delta t), ...,\ x(t-K\Delta t)\}\\ \mathbf{Y} &= \{y(t-\Delta t),\ y(t-2\Delta t), ...,\ y(t-K\Delta t)\}\end{aligned} \tag{15}$$

The output variables of the LCI model are the longitudinal and lateral positions of the subject vehicle at next time step $t+\Delta t$, i.e., $\{x(t+\Delta t), y(t+\Delta t)\}$.

Likewise, the MSE is taken as the performance index of the model training and testing. It reads

$$MSE = \frac{\sum_{k=1}^{T}[(x_k - x_k')^2 + (y_k - y_k')^2]}{T} \tag{16}$$

where $T$ is the time length of discretizing the LC process; $x_k$ and $x_k'$ are the ground-truth and simulated longitudinal positions at time step $k$, respectively; $y_k$ and $y_k'$ are the ground-truth and simulated lateral positions at time step $k$, respectively.

The structure of the LSTM is determined by using the same method as the DBN presented in Appendix A; see Appendix B for the details.

## 4. Model evaluation

### 4.1. Data description

The data employed in this study is provided by the Next Generation Simulation program (NGSIM, 2006). It is a high-fidelity trajectory dataset widely used in various traffic flow studies over the past ten years (Toledo and Zohar, 2007; Thiemann et al., 2008; Choudhury and Ben-Akiva, 2013; Zheng et al., 2013a,b; He et al., 2015; Wang et al., 2018; He et al., 2019a,b). As Thiemann et al. (2008) stated, the NGSIM trajectory data provides a good basis for LC investigation. Many LCD models has been calibrated and validated by using the NGSIM trajectory data (Kesting et al., 2007; Laval and Leclercq, 2008; Yang et al., 2015; Dou et al., 2016), and the prediction accuracy could reach 91.38% (Li et al., 2016b).

The NGSIM data includes the trajectory data of all vehicles on the surveillance road sections, including speed, acceleration, and position of each vehicle at a 0.1-s interval. It contains three types of vehicles, i.e., motorcycles, cars, and trucks. Only cars (96.5% and 94.9% contained in US-101 and I-80 datasets, respectively) are taken into account in this study.

Initially, the trajectory data of 1078 LC vehicles and 1120 LK vehicles are obtained from the NGSIM data. We further filter the data as follows.

**Table 3**
Statistical results of LC trajectories contained in the NGSIM data.

| Duration (s) | Total number of vehicles | Proportion (%) |
|---|---|---|
| ⩽2 | 7 | 1.32 |
| 2–4 | 29 | 5.47 |
| 4–6 | 118 | 22.26 |
| 6–8 | 237 | 44.72 |
| 8–10 | 109 | 20.57 |
| 10–12 | 23 | 4.34 |
| >12 | 7 | 1.32 |
| Total | 530 | 100 |

- The vehicles on rightmost Lanes 5 and 6 are discarded to remove the impact of ramps.
- The vehicles entering or leaving from ramps are discarded, since we only focus on discretionary LC.
- The vehicles that make LC more than one time or cross more than one lane are discarded.

As a consequence, 530 vehicles with complete LC processes are used here. It is sufficient to calibrate and validate an LC model.

To ensure that the selected LC and LK trajectories are in similar traffic conditions, we extract the trajectories of the LK vehicles that appear in the spatiotemporal neighbor of the LC vehicles.

In this paper, LC duration is defined as the time required for continuous lateral moving during the LC process. Table 3 summarizes statistical results of the LC duration of those selected vehicles. It can be seen that the average duration of the LC processes is 6.97 s. The LC duration for all vehicles ranges from 1.5 s to 13.9 s, and nearly 95% are less than 10 s. Therefore, this study employs the trajectory data with the LC duration less than 10 s to exclude various noises and irregular behavior. Denote by $t_{LC}$ the lane switching time, i.e., the moment before which the vehicle is in the current lane and after which it is in the target lane. Then, the selected trajectory data consists of two time intervals, that is, 5 s before $t_{LC}$ and 5 s after $t_{LC}$.

Statistical results of the LC-related variables are listed in Table 4 to further reveal the characteristics of an LC process based on the NGSIM data. It can be seen from Table 4 that most of these variables have relatively large standard deviations, indicating the diversity and complexity of LC behavior.

It is worth noting that the lateral position data provided by NGSIM may be not very accurate. However, as we will show, the proposed model could still satisfactorily capture LC dynamics, implying that the model will perform better if data with higher lateral accuracy could be provided.

### 4.2. Evaluation of DBN-based LCD model

An appropriate algorithm to train NN should be applied to update the weight coefficients, so that the objective function could gradually converge to a global minimum. To the end, both the proposed LCD and LCI models are trained in the framework of TensorFlow in this study.

In addition, the following cross-validation method is applied to the proposed models to avoid over-training: If both the validation and training errors steadily decrease, the model is still under-training; if the validation error increases while the training error steadily decreases, the model is over-training.

#### 4.2.1. Data preparation

As mentioned above, total 2198 LC decision-making samples are extracted, including 1078 LC samples and 1120 LK samples. 80% of the samples are used for training, which is 1758 samples including 862 LC samples and 896 LK samples. The remaining 20% samples are used for testing, which are 440 samples consisting of 216 LC samples and 224 LK samples.

The pre-processed data including all the input and output variables is first normalized to avoid the differences in magnitude. The

**Table 4**
Statistical results of the LC-related variables applied for validations.

| Variavle | Mean value | Standard deviation |
|---|---|---|
| $v_n$ (m/s) | 11.48 | 3.56 |
| $v_1$ (m/s) | 11.30 | 3.43 |
| $v_2$ (m/s) | 12.62 | 3.28 |
| $v_3$ (m/s) | 11.36 | 3.44 |
| $\Delta v_1$ (m/s) | −0.18 | 1.45 |
| $\Delta v_2$ (m/s) | 1.14 | 1.83 |
| $\Delta v_3$ (m/s) | −0.12 | 2.27 |
| $d_1$ (m) | 21.60 | 10.35 |
| $d_2$ (m) | 18.39 | 15.64 |
| $d_3$ (m) | 22.65 | 13.39 |

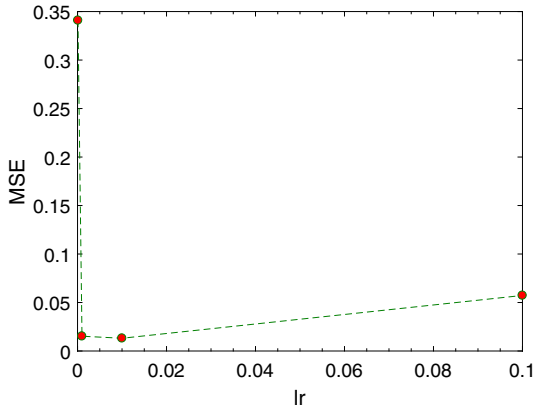


**Fig. 7.** Variation of the MSE against various $l_r$ values.

min-max normalization approach is applied, which is written as follows.

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{17}$$

where $x^*$ is the normalized value; $x_{max}$ and $x_{min}$ are the maximum and minimum values in the dataset, respectively. As a consequence, the normalized data ranges from 0 to 1, and it is ready for the following training and testing.

#### 4.2.2. Training of DBN-based LCD model

The training of the proposed DBN-based LCD model in this paper is divided into two stages:

- Stage 1: Unsupervised training of each RBM in DBN. It is to maximize the retention of feature information when input samples are mapped to different feature spaces.
- Stage 2: BP algorithm is used to train the network composed of the last RBM hiding layer and the DBN output layer.

According to the results in Appendix A, a four-layer DBN is constructed, which is named as '7-5-7-2' DBN. The first layer is the input layer comprised of 7 neurons that correspond to the 7 input variables in Eq. (6). The second and third layers are the hidden layers, which contain 5 and 7 neurons, respectively. The last layer is the output layer with 2 neurons, representing the binary decision of LK or LC.

A learning rate (denoted by $l_r$) is an important parameter that needs to be determined for the training. A large learning rate may induce a large reconstruction error, resulting in the unstable variation of the weighted parameters. However, an excessively small learning rate may greatly decrease the training speed. Since the optimal learning rate cannot be theoretically derived, an appropriate $l_r$ value is selected based on simulation tests. Fig. 7 shows the variation of the MSE against various $l_r$ values (including 0.0001, 0.001, 0.01 and 0.1). It can be found that the minimum MSE is obtained when $l_r = 0.01$. Therefore, $l_r$ is set to be 0.01 for the proposed DBN-based LCD model.

Furthermore, the convergence process of training the DBN-based LCD model is simulated and presented in Fig. 8. At the beginning, the performance index, i.e., the MSE, decreases quickly with the increase of the number of iterations; The MSE changes very

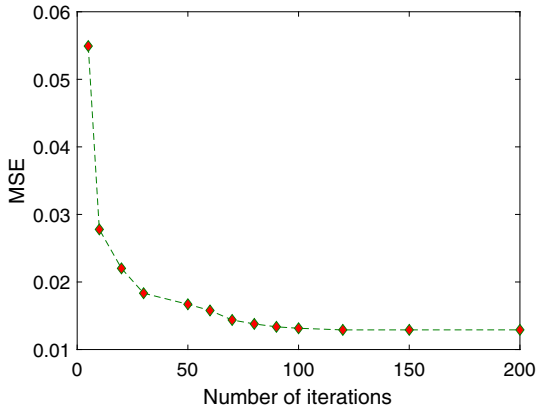


**Fig. 8.** The convergence process of the training of the DBN-based LCD model.

**Table 5**
Training results of the DBN-based lane-changing model.

| | MSE | Accuracy |
|---|---|---|
| Training 1 | 0.0132 | 99.32% |
| Training 2 | 0.0148 | 97.05% |
| Training 3 | 0.0158 | 96.59% |
| Training 4 | 0.0135 | 98.64% |
| Training 5 | 0.0160 | 96.82% |

slowly until a critical value (it is 100, approximately) is reached. Therefore, the training of the DBN-based LCD model lasts for 100 iterations in this study, which is also fit the criterion of cross-validation. In this study, a personal computer (PC) with CPU of 2.2 GHz is used to train and test the DBN-based LCD model, and the computational time is 23 s for obtaining the best performing model. Table 5 shows the MSE and the accuracy rate of the training results, from which it can be concluded that the training results are satisfactory.

#### 4.2.3. Testing of DBN-based LCD model

Table 6 presents the prediction results of the proposed DBN-based model, which correspond to the five trainings in Table 5. It can be seen that the accuracy rate is as high as 99.32%. Furthermore, we compare them with the BP-based LCD model (Appendix C) and the Logit-based LCD model (Appendix D), which are classic data-driven and analytical LCD models, respectively. As shown in Table 6, the accuracy of the BP-based and Logit-based LCD models are just 76.82% and 61.30%, respectively, indicating the remarkable advantages of the proposed DBN-based LCD model.

Fig. 9 compares the LCD prediction of all 440 testing samples by using the DBN-based and BP-based LCD models. It is obvious that the BP-based LCD model has 102 error predictions, while the DBN-based LCD model only has 2 error predictions. For most of the error predictions, the BP-based LCD model outputs LK decisions for the actual LC vehicles, indicating that the BP NN cannot well capture the motivation of LC. Therefore, it can be concluded that the DBN-based LCD model is able to well extract LC-related information from data and has better prediction capability.

Taking Test 1 as an example, Table 7 presents a comparison between the predicted and ground-truth values for the 440 test samples. It can be seen that the predicted values of 437 samples are consistent with the ground-truth values, turning out the accuracy rate of 99.32%. Of the 224 samples with the ground-truth value of 1 (i.e., LK), the prediction values of 222 samples are 1, turning out the accuracy rate of 99.11%; Of 216 samples with the ground-truth value of 2 (i.e., LC), the prediction values of 215 samples are 2, turning out the accuracy rate of 99.54%.

#### 4.2.4. Sensitivity analysis of decision variables

Sensitivity analysis is carried out to validate the rationality of the decision variable selection in proposing the DBN-based LCD model. The testing is presented as follows. One of the 7 decision variables (i.e., the input variables) is alternately removed, resulting in 7 new DBNs with the NN structure of '6-5-7-2'. The 7 DBNs with 6 input variables are trained and tested based on the same data samples. Table 8 presents the MSE of the 6-input DBN-based LCD models. It can be seen that the MSE increases greatly, compared with the MSE resulted from the original 7-input model (which is 0.01315 as shown in Table 5). In particular, the MSE increases more than 4 times when $d_2$ is removed.

It is known that LCD mainly depend on the conditions of the surrounding vehicles. However, it is still unclear which one of the surrounding vehicles has the most essential impact on LCD. This is an important question but attracts little attention previously. At present, the proposed DBN-based LCD model can answer the question. Fig. 10 presents the incremental MSE when one variable of the surrounding vehicle is removed. It can be seen that $PV_T$ (i.e., the preceding vehicle on the target lane) is the most important vehicle for LCD, while $PV_C$ (i.e., the preceding vehicle on the current lane) is the least important vehicle. In addition, for the same vehicle, removing the input variables regarding space headway induces more incremental MSE than those regarding speed, indicating that drivers pay more attention on space headway than speed when making LCD.

**Table 6**
Prediction results and model comparison.

| Model | MSE | Accuracy |
|---|---|---|
| DBN (Test 1) | 0.0132 | 99.32% [†] |
| DBN (Test 2) | 0.0168 | 96.04% |
| DBN (Test 3) | 0.0162 | 96.14% |
| DBN (Test 4) | 0.0169 | 95.16% |
| DBN (Test 5) | 0.0143 | 97.73% |
| BP | 0.3102 | 76.82% |
| Logit | 0.39 | 61.30% |

[†] We have carefully checked the results, and we are sure that the prediction accuracy is equal to the training accuracy by chance.

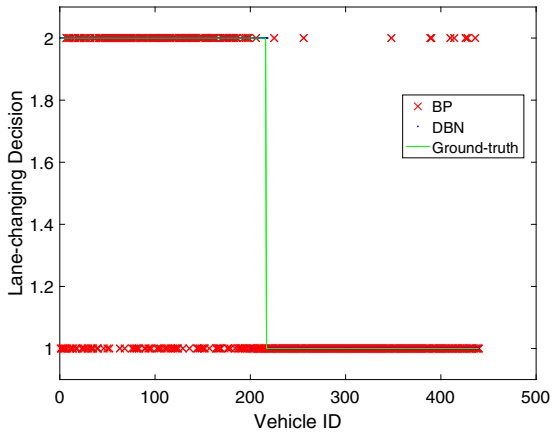


**Fig. 9.** LC prediction using the DBN-based and BP-based LCD models (Note that 1 represents LK and 2 represents LC).

**Table 7**
Prediction accuracy of the DBN-based LCD model (Test 1).

| Ground-truth value | Predicted value ($Dec = 1$) | Predicted value ($Dec = 2$) | Total |
|---|---|---|---|
| lane-keeping ($Dec = 1$) | 222 (99.11%) | 2 (0.89%) | 224 |
| lane-changing ($Dec = 2$) | 215 (99.54%) | 1 (0.46%) | 216 |
| Total | 437 (99.32%) | 3 (0.68%) | 440 |

**Table 8**
MSE of the DBN-based LCD model when one of the decision variables is removed.

| Removed variable | MSE | Removed variable | MSE |
|---|---|---|---|
| v | 0.0230 | $d_1$ | 0.0447 |
| $\Delta v_1$ | 0.0249 | $d_2$ | 0.0574 |
| $\Delta v_2$ | 0.0312 | $d_3$ | 0.0387 |
| $\Delta v_3$ | 0.0308 | | |

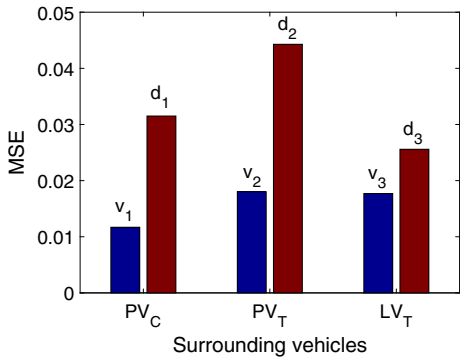


**Fig. 10.** MSE of the DBN-based LCD model when one of the variables of the surrounding vehicle is removed.

In summary, the most important factor regarding LCD is the relative position of the preceding vehicle in the target lane. Compared with speed, drivers pay more attention on the relative positions of the surrounding vehicles. The data mining results based on the proposed model are consistent with our real-life driving experience.

### *4.3. Evaluation of LSTM-based LCI model*

#### *4.3.1. Data preparation*

As introduced in Section 4.1, 530 LC samples with a whole LC process are extracted from the NGSIM trajectory datasets. 90% of the samples (i.e., 477 samples) are used for training. The remaining 10% samples (i.e., 53 samples) are used for testing. Each of the

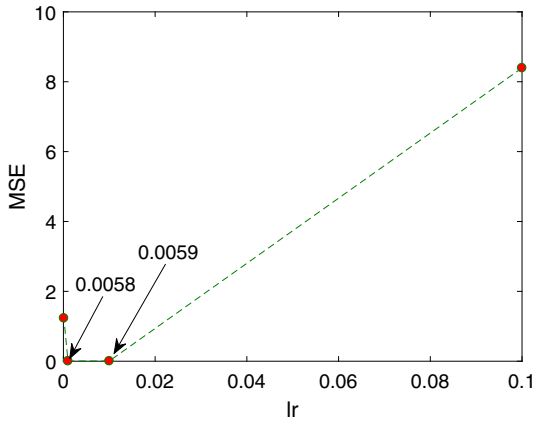


**Fig. 11.** Variation of the MSE against various learning rates.

sample contains 10-s LC trajectory data with 0.1-s time intervals, and thus each sample is composed of 100 data records. Accordingly, 47700 data records are obtained for training, while 5300 records for testing.

Likewise, the pre-processed data is normalized before the training and testing. Here, the following Z-score approach is applied.

$$x^* = \frac{x - \mu}{\sigma} \tag{18}$$

where $x^*$ is the normalized value; $\mu$ and $\sigma$ are the mean and standard deviation of the sample data, respectively.

*4.3.2. Training of LSTM-based LCI model*

Based on the method in Appendix C, the network structure of the LSTM-based LCI model is '(7 + $K$)-7-3-2'. More specifically, the input layer has 7 + $K$ neurons corresponding to the number of the input variables given in Eq. (14); the two hidden layers have 7 and 3 neurons, respectively; and the output layer has 2 neurons corresponding to the longitudinal and lateral positions of the subject vehicle.

To train the LSTM-based LCI model, the following parameters have to be determined first.

- Learning Rate. The LSTM NN is trained with the learning rates of 0.0001, 0.001, 0.01 and 0.1, respectively. The resulting MSEs are presented in Fig. 11. As a consequence, the learning rate $l_r$ is set to be 0.001, since it results in the minimum MSE.
- Time Step. The LSTM NN searches from the samples within the last time step $t_s$ and considers its impacts on current samples. The time step is thus regarded as an impact factor of the current training. Since the time step plays an important role in training the model, it is necessary to choose a proper value for it. Fig. 12 shows the variation of the MSE when training with different time steps. Obviously, the minimum MSE is obtained at the time step of 40. Therefore, the time step is set to be 40 in this study.
- Batch size. The batch size represents the size of samples within an training iteration. Generally speaking, the larger the batch size, the better the training results. Nevertheless, it requires higher computational ability if increasing the size of the samples. In the following training process, the batch size is set to be 40 here.

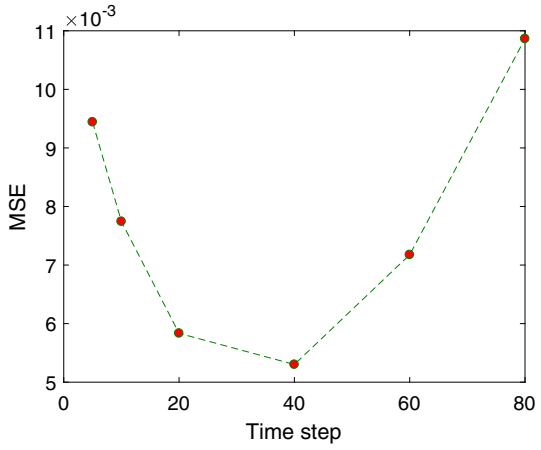


**Fig. 12.** Variation of the MSE against the time step of the LSTM-based LCI model.

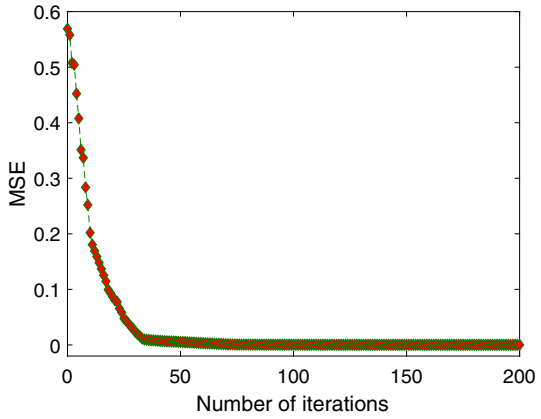


**Fig. 13.** The convergence process of training the LSTM-based LCI model.

In addition, the convergence process of training the LSTM-based LCI model is presented in Fig. 13. It can be found that the MSE decreases quickly at the beginning, and then gradually converges to a very small MSE after 80 iterations. Therefore, the number of iterations should be larger than 80. We set the number of iterations here to be 100. Likewise, the PC with CPU of 2.2 GHz is applied to train and test the LSTM-based LCI model, and the computational time is 519 s for obtaining the best model.

#### *4.3.3. Testing of LSTM-based LCI model*

By using the LSTM-based LCI model, numerical tests are carried out to predict the trajectories of the 53 testing vehicles. The MSE is used as the performance index during the whole LC process. For any testing vehicle, the MSE of the predicted trajectory varies near the value of 0.0002; see Table 9. The average MSE of all tested vehicles is about 0.00015. Such small MSEs indicate that the proposed LCI model can well capture the whole LC process.

Fig. 14 presents the ground-truth and predicted LC trajectories of 6 vehicles. Among the vehicles, vehicle 26 has the smallest MSE (Fig. 14(a)), while vehicle 11 has the largest MSE (Fig. 14(b)). The other 4 vehicles are randomly selected. Intuitively, the LC trajectories predicted by using the proposed LSTM-based LCI model almost coincide with the ground-truth trajectories.

To further evaluate the performance of the proposed LSTM-based LCI model, the LC process of each vehicle is divided into the following three LC stages.

- Stage 1: Pre-LC stage. It is a stage of preparing LC before the continuous lateral moving.
- Stage 2: LC stage. It is a continuous lateral moving process.
- Stage 3: Adjusting stage. During the stage LC drivers adjust the direction and speed of their vehicles after the continuous lateral moving.

Fig. 15 presents the MSEs at the three stages for the 6 vehicles shown in Fig. 14. It can be found from Fig. 15 that the MSEs at Stage 2 are very small for all vehicles, indicating that the proposed LSTM-based LCI model can well predict the LC trajectory at the LC stage. However, the prediction errors are relatively higher and not stable for Stages 1 and 3. In particular, Stage 1 is the intermediate process between LCD and LCI. During this process, the complexity of impact factors (i.e., surrounding traffic states) may lead to large uncertainty in driving behavior. The heterogeneity of driving behavior additionally increases the uncertainty. As a consequence, the MSE of Stage 1 is relatively fluctuating in Fig. 15. It implies that there are more random impact factors during pre-LC and adjusting stages, and thus it is more difficult to make predictions. Generally speaking, the MSEs at all stages are smaller than 0.002 for all vehicles, indicating that the proposed model can provide satisfactory predictions for all the three stages.

**Table 9**
MSE of predicting the LC trajectories of 20 randomly-selected vehicles.

| Vehicle ID | MSE | Vehicle ID | MSE |
|---|---|---|---|
| 1 | 0.00016 | 21 | 0.00011 |
| 3 | 0.00013 | 23 | 0.00020 |
| 5 | 0.00009 | 25 | 0.00020 |
| 7 | 0.00027 | 27 | 0.00014 |
| 9 | 0.00006 | 29 | 0.00010 |
| 11 | 0.00027 | 31 | 0.00011 |
| 13 | 0.00013 | 33 | 0.00011 |
| 15 | 0.00014 | 35 | 0.00015 |
| 17 | 0.00023 | 37 | 0.00021 |
| 19 | 0.00010 | 39 | 0.00007 |

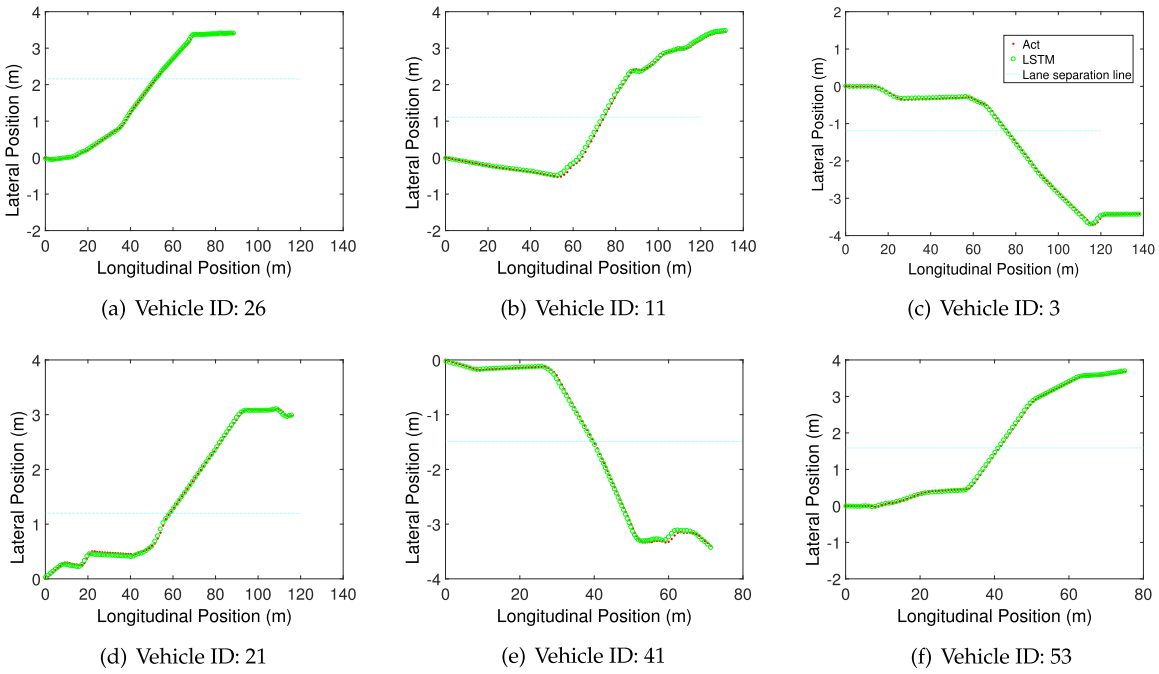


**Fig. 14.** Comparisons of ground-truth and predicted trajectories of vehicles. Point (0, 0) denotes the end of LC processes, all of which are from US-101 dataset (8:20 A.M.–8:35 A.M.). The abscissa denotes the longitudinal distance from the point (0, 0), and the longitudinal coordinate is the lateral distance from the point (0, 0). The directions of all vehicles are from right to left.

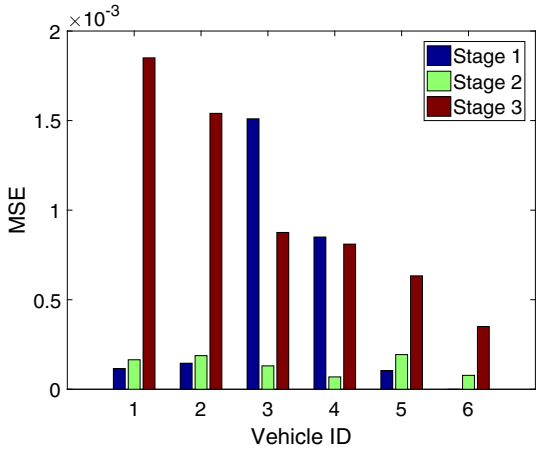


**Fig. 15.** MSE of the predicted LC trajectories in all stages. Note that there is no Stage 1 for vehicle 53 due to the lack of data.

At last, we compare the proposed LSTM-based LCI model with the LCI models based on the classic BP NN (E). Table 10, which presents the MSE resulted from both the two models, indicates that the LSTM-based LCI model results in smaller MSE compared to the BP-based LCI model. Fig. 16 illustrates the ground-truth and predicted trajectories. It is obvious that the predicted trajectory resulted from the LSTM-based LCI model overlays much better with the ground-truth trajectory than that of the BP-based model. The results well demonstrate that the LSTM-based LCI model outperforms the traditional BP-based model.

**Table 10**
MSE of the LSTM-based and BP-based LCI models.

| Model | MSE |
|---|---|
| BP | 0.01137 |
| LSTM | 0.00014 |

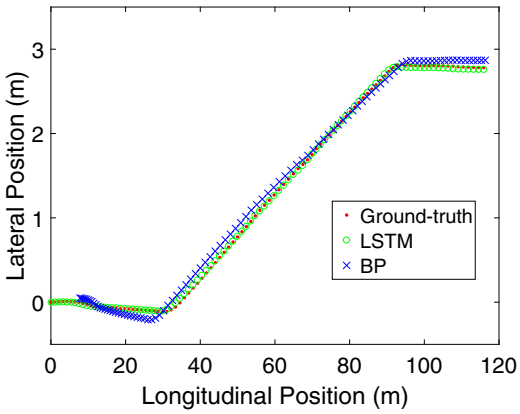


**Fig. 16.** LC trajectory comparison between the LSTM-based and BP-based LCI models.

## 5. Conclusions

LC that is composed of LCD and LCI is a basic driving behavior that largely impacts on traffic efficiency and safety. LC is complicated since it is influenced by various factors, such as driver's random behavior, the position and speed of the subject vehicle, and the status of the surrounding vehicles in the current and target lanes. Therefore, it may be difficult to well capture the detailed LC dynamics by using mathematical models and traditional machine learning methods such as NN and SVM.

Taking advantages of the prevailing deep learning approaches, this paper proposes a data-driven discretionary LC model, which is composed of the DBN-based LCD model and the LSTM-based LCI model. The DBN-based LCD model takes into account various factors that impact on driver's LCD. The LSTM-based LCI model particularly incorporates the historical data immediately before LCI. Model tests based on empirical vehicle trajectory data demonstrate that the proposed deep learning-based LC model is able to accurately predict the LC process of a vehicle, and it outperforms the traditional machine learning method.

The proposed deep learning-based LC model is able to excavate the underlying features from high-dimensional data. More specifically, we find, by taking advantage of the proposed model, that the most important factor associated with LCD is the relative position of the preceding vehicle in the target lane. Moreover, drivers in making LCD pay more attention on the relative positions of the surrounding vehicles rather than the speed.

Since LC usually plays significant impact on traffic flow dynamics and safety, future works will be carried out to develop data-driven multi-lane traffic flow models and to analyze the impact of various LC behavior on traffic flow.

## Acknowledgements

This work was partially supported by the National Key R&D Program of China (Grant Nos. 2018YFB1600900), and National Natural Science Foundation of China (Grant Nos. 71621001, 71671014, 71871010, 71771012, 71631007).

## Appendix A. DBN construction

The following algorithm is applied to determine the DBN structure.

**Step 1:** Set the number of the hidden layers to be $h = 1$ for the DBN. Prepare data samples for training.
**Step 2:** The neuron number of the hidden layer $h$ is set to be $n_h = 1, 2, ..., n_L$, iteratively, where $n_L$ is a large enough number.
  **Step 2.1** For each $n_h$, the DBN is trained with the same data until the training error converges to a stable value.
  **Step 2.2** Find the minimum training error $e_h$ and the corresponding neuron number $n_{opt}(h)$, which is selected as the neuron number of the corresponding hidden layer for the proposed DBN-based LCD model.
**Step 3:** Let $h = h + 1$, and repeat Step 2.
**Step 4:** If $e_{h+1} > e_h$, terminate the algorithm; Otherwise, turn to Step 3.

The MSE is taken as the training error $e_h$ and the proposed algorithm is applied to design the DBN structure. Fig. A.1 shows the variation of the training MSE with the neuron number in hidden layers 1, 2, and 3, respectively. Obviously, the minimum MSE can be obtained with 2 hidden layers. Corresponding to hidden layers 1 and 2, the MSE reaches the minimum value with 5 and 7 neurons, respectively. Therefore, the optimized DBN structure is '7-5-7-2' by combining the input and output layers.

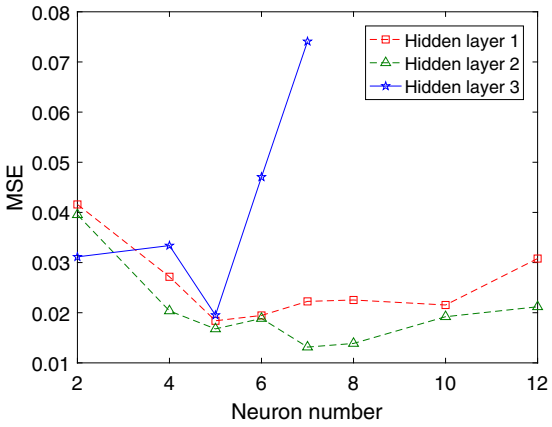


**Fig. A.1.** Variation of the MSE with the different neuron numbers of hidden layers.

## Appendix B. LSTM construction

The algorithm in A is utilized to design the LSTM structure. The training MSE is calculated for various neuron numbers of different hidden layers, as shown in Fig. B.1. Since a 3 hidden layer LSTM network cannot further decrease the training MSE, the number of hidden layers is set to be 2. Accordingly, the neuron numbers are 3 and 7, corresponding to the minimum MSE as shown in Fig. B.1. In short, the optimized structure of the LSTM network is '(7 + $K$)-7-3-2'.

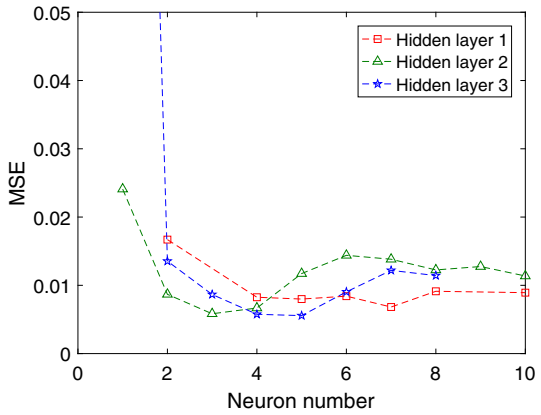


**Fig. B.1.** Variation of the MSE with the different neuron numbers of the hidden layers.

## Appendix C. BP-based LCD model

The inputs and outputs of the BP-based LCD model are the same as the DBN-based LCD model, as shown in Table 1. The following Tan-Sigmoid function is employed to be the transfer function at the input layer.

$$g(z) = \frac{2}{1 + e^{-2z}} - 1. \tag{C.1}$$

The Sigmoid function in Eq. (7) is employed as the transfer function of the output layer. Levenberg-Marquadt BP algorithms are applied for training. Likewise, the approach of constructing the DBN is applied to design the hidden layers of the BP NN, and a 7-3-2 network is obtained.

## Appendix D. Logit-based LCD model

The probability function of the LCD is formulated as follows.

$$P(LC) = \frac{e^{V}(LC)}{1 + e^{V(LC)}} \tag{D.1}$$

where $V(LC)$ is the LC utility that is written as follows.

**Table D.11**
Regression results of the Logit-based LCD model.

| Independent variable | Regression coefficient | Independent variable | Regression coefficient |
|---|---|---|---|
| $v_n$ | −0.076 | $d_1$ | −0.025 |
| $v_1$ | −0.161 | $d_2$ | 0.020 |
| $v_2$ | 0.129 | $d_3$ | −0.012 |
| $v_3$ | 0.063 | | |

$$V(LC) = \beta_0 + \beta^T X \tag{D.2}$$

where $X$ is the vector of independent variables; $\beta_0$ is the constant; and $\beta^T$ is the corresponding coefficients of X.

Generally speaking, the independent variables are related to the traffic states on current and target lanes. Table D.11 shows the selected independent variables and the regression results.

## Appendix E. BP-based LCI model

The input and output variables listed in Table 2 are used. Accordingly, the input and output neurons are 9 and 2, respectively. Tan-Sigmoid function in Eq. (C.1) is employed as the transfer function of the input layer. The linear Purelin function is used as the output layer, which is written as follows.

$$g(z) = z. \tag{E.1}$$

Levenberg-Marquadt BP algorithms are applied for training. Similar approach of constructing the LSTM network is applied to design the hidden layers of BP NN, and finally a 9-3-2 network is constructed.

## References


Ahmed, K.I., Ben-Akiva, M.E., Kousopoulos, H.N., Mishalani, R.G., 1996. Models of freeway lane changing and gap acceptance behaviour. In: The 13th International Symposium on the Theory of Traffic Flow and Transportation, pp. 501–515.

Ahn, S., Cassidy, M., 2007. Freeway traffic oscillations and vehicle lane-change maneuvers. In: 17th International Symposium of Transportation and Traffic Theory, pp. 1–23.

Arief, K., Ridha, S., Fakhreddine, K., 2016. Improving trafic ow prediction with weather information in connected cars: a deep learning approach. In: TVT, pp. 9508–9517.

Cassidy, M.J., Rudjanakanoknad, J., 2005. Increasing the capacity of an isolated merge by metering its on-ramp. Transp. Res. Part B: Methodol. 39, 896–913.

Choudhury, C., Ben-Akiva, M.T., 2013. Modelling driving decisions: a latent plan approach 9, 546–566.

Chovan, J.D., Tijerina, L., Alexander, G., Hendricks, D.L., 1994. Examination of Lane Change Crashes and Potential IVHS Countermeasures.

Ding, C., Wang, W., Wang, X., Baumann, M., 2013. A neural network model for driver's lane-changing trajectory prediction in urban traffic flow. Math. Problems Eng. 2013, 1–8.

Dou, Y., Yan, F., Feng, D., 2016. Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, pp. 901–906.

Enke, K., 1979. Possibilities for improving safety within the driver vehicle environment loop. In: 7th Int. Technical Conference On Experimental Safety Vehicle.

Gao, J., Murphey, Y.L., Zhu, H., 2018. Multivariate time series prediction of lane changing behavior using deep neural network. Appl. Intell. 48, 3523–3537.

Gipps, P.G., 1986. A model for the structure of lane-changing decisions 20B, 403–414.

He, Z., Lv, Y., Lu, L., Guan, W., 2019a. Constructing spatiotemporal speed contour diagrams: using rectangular or non-rectangular parallelogram cells? Transportmetrica B 7, 44–60.

He, Z., Zhang, W., Jia, N., 2019b. Estimating carbon dioxide emissions of freeway traffic: a spatiotemporal cell-based model. IEEE Trans. Intell. Transp. Syst.

He, Z., Zheng, L., Guan, W., 2015. A simple nonparametric car-following model driven by field data. Transp. Res. Part B: Methodol. 80, 185–201.

Hidas, P., 2002. Modelling lane changing and merging in microscopic traffic simulation. Transp. Res. Part C: Emerg. Technol. 10, 351–371.

Hidas, P., 2005. Modelling vehicle interactions in microscopic simulation of merging and weaving. Transp. Res. Part C: Emerg. Technol. 13, 37–62.

Hinton, G.E., 2009. Deep belief networks. Scholarpedia 4, 5947.

Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18, 1527–1554.

Huang, X., Sun, J., Sun, J., 2018. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. Transp. Res. Part C: Emerg. Technol. 95, 346–362.

Hunt, J.G., Lyons, G.D., 1994. Modelling dual carriageway lane changing using neural networks. Transp. Res. Part C: Emerg. Technol. 2, 231–245.

Jia, Y., Wu, J., Ben-akiva, M., Seshadri, R., Du, Y., 2017. Rainfall-integrated traffic speed prediction using deep learning method.

Kerner, B.S., Rehborn, H., 1996. Experimental features and characteristics of traffic jams. Phys. Rev. E Stat. Phys. Plasmas Fluids Related Interdiscipl. Top. 53, R1297–R1300.

Kesting, A., Treiber, M., Helbing, D., 2007. General lane-changing model MOBIL for car-following models. Transp. Res. Rec. J. Transp. Res. Board 1999, 86–94.

Keyvan-Ekbatani, M., Knoop, V.L., Daamen, W., 2016a. Categorization of the lane change decision process on freeways. Transp. Res. Part C: Emerg. Technol. 69, 515–526.

Keyvan-Ekbatani, M., Knoop, V.L., Grébert, V., Daamen, W., 2016b. Lane change strategies on freeways: a microscopic simulation study. Traffic and granular. Flow 15, 395–402.

Knoop, V.L., Buisson, C., 2015. Calibration and validation of probabilistic discretionary lane-change models. IEEE Trans. Intell. Transp. Syst. 16, 834–843.

Knoop, V.L., Hoogendoorn, S.P., Shiomi, Y., Buisson, C., 2012. Quantifying the number of lane changes in traffic Empirical analysis. Transp. Res. Board.

Knoop, V.L., Keyvan-Ekbatani, M., Baat, M.D., Taale, H., Hoogendoorn, S.P., 2018. Lane change behavior on freeways: an online survey using video clips. J. Adv. Transp. 2018, 1–11.

Kumar, P., Perrollaz, M., Lefèvre, S., Laugier, C., 2013. Learning-based approach for online lane change intention prediction. In: IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia.

Laval, J., Leclercq, L., 2008. Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model. Transp. Res. Part B: Methodol. 42, 511–522.

Laval, J.A., Daganzo, C.F., 2006. Lane-changing in traffic stream. Transp. Res. Part B: Methodol. 40, 251–264.
Li, K., Wang, X., Xu, Y., Wang, J., 2016a. Lane changing intention recognition based on speech recognition models. Transp. Res. Part C 69, 497–514.
Li, L., Zhang, M., Liu, R., 2016b. The application of Bayesian filter and neural networks in lane changing prediction. In: In: Kim, Y.H. (Ed.), Proceedings of the 5th International Conference on Civil Engineering and Transportation 2015, vol. 30. pp. 2004–2007.
Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: a graph convolutional neural network approach. Transp. Res. Part C: Emerg. Technol. 97, 258–276.
Lint, H.V., Schakel, W., Tamminga, G., Knoppers, P., Verbraeck, A., 2016. Getting the human factor into traffic flow models: new open-source design to simulate next generation of traffic operations. Transp. Res. Rec. J. Transp. Res. Board 2561, 25–33.
Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y., 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors (Switzerland) 17.
Mattes, S., 2003. The lane change task as a tool for driver distraction evaluation. DaimlerChrysler AG Research & Technology.
Mauch, M., Cassidy, M.J., 2002. Freeway traffic oscillations: observations and predictions - eScholarship. In: Taylor, M.A.P. (Ed.), Proceedings of the15th International Symposium on Transportation and Traffic Theory. Pergamon-Elsevier, Oxford, UK.
Moridpour, S., Sarvi, M., Rose, G., 2010. Lane changing models: a critical review. Transp. Lett. Int. J. Transp. Res. 2, 157–173.
Nelson, W.L., 1989. Continuous-curvature path for autonomous vehicles. In: IEEE International Conference on Robotics and Automation, 1989. Proceedings, vol. 3. pp. 1260–1264.
NGSIM, 2006. The Next Generation Simulation Program. http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.
Olsen, E.C.B., 2013. Modeling Slow Lead Vehicle Lane Changing (Ph.D. thesis). Virginia Polytechnic Institute and State University.
Pande, A., Abdel-Aty, M., 2006. Assessment of freeway traffic parameters leading to lane-change related collisions. Accident Anal. Prevent. 38, 936–948.
Rahman, M., Chowdhury, M., Xie, Y., He, Y., 2013. Review of microscopic lane-changing models and future research opportunities. IEEE Trans. Intell. Transp. Syst. 14, 1942–1956.
Shamir, T., 2004. How should an autonomous vehicle overtake a slower moving vehicle: design and analysis of an optimal trajectory. IEEE Trans. Autom. Control 49, 607–610.
Sun, D., Elefteriadou, L., 2011. Lane-changing behavior on urban streets: a focus group-based study. Appl. Ergon. 42, 682–691.
Sun, D., Elefteriadou, L., 2012. Lane-changing behavior on urban streets: an in-vehicle field experiment-based study. Comput.-Aided Civ. Infrastruct. Eng. 27, 525–542.
Sun, D., Elefteriadou, L., 2014. A driver behavior-based lane-changing model for urban arterial streets. Transp. Sci. 48, 184–205.
Thiemann, C., Treiber, M., Kesting, A., 2008. Estimating acceleration and lane-changing dynamics based on NGSIM trajectory data. Transp. Res. Rec. 90–101.
Toledo, T., Koutsopoulos, H., Ben-Akiva, M., 2003. Modeling integrated lane-changing behavior. Transp. Res. Rec. J. Transp. Res. Board 1857, 30–38.
Toledo, T., Zohar, D., 2007. Modeling duration of lane changes. Transp. Res. Rec. J. Transp. Res. Board 1999, 71–78.
Van Winsum, W., De Waard, D., Brookhuis, K.A., 1999. Lane change manoeuvres and safety margins. Transp. Res. Part F Traffic Psychol. Behav. 2, 139–149.
Wang, X., Jiang, R., Li, L., Lin, Y., Zheng, X., Wang, F.Y., 2018. Capturing car-following behaviors by deep learning. IEEE Trans. Intell. Transp. Syst. 19, 910–920.
Wang, Y., Zhang, D., Liu, Y., Dai, B., Lee, L.H., 2019. Enhancing transportation systems via deep learning: a survey. Transp. Res. Part C: Emerg. Technol. 99, 144–163.
Yang, D., Zhu, L., Yang, F., Pu, Y., 2015. Modeling and analysis of lateral driver behavior in lane-changing execution. Transp. Res. Rec. J. Transp. Res. Board 2490, 127–137.
Yang, M., Wang, X., Quddus, M., 2019. Examining lane change gap acceptance, duration and impact using naturalistic driving data. Transp. Res. Part C: Emerg. Technol. 104, 317–331.
Yao, W., Zhao, H., Bonnifait, P., Zha, H., 2013. Lane change trajectory prediction by using recorded human driving data. IEEE Intelligent Vehicles Symposium, Proceedings 430–436.
Zhang, J., Huang, Q., Wu, H., Liu, Y., 2017. A shallow network with combined pooling for fast traffic sign recognition. Information (Switzerland) 8.
Zhang, Z., He, Q., Gao, J., Ni, M., 2018. A deep learning approach for detecting traffic accidents from social media data. Transp. Res. Part C: Emerg. Technol. 86, 580–596.
Zhao, Z., Chen, W., Wu, X., Chen, P.C.Y., Liu, J., 2017. LSTM network: a deep learning approach for short-term traffic forecast. IET Intel. Transp. Syst. 11, 68–75.
Zheng, J., Suzuki, K., Fujita, M., 2013a. Car-following behavior with instantaneous driver-vehicle reaction delay: a neural-network-based methodology. Transp. Res. Part C: Emerg. Technol. 36, 339–351.
Zheng, Z., 2014. Recent developments and research needs in modeling lane changing. Transp. Res. Part B: Methodol. 60, 16–32.
Zheng, Z., Ahn, S., Chen, D., Laval, J., 2013b. The effects of lane-changing on the immediate follower: anticipation, relaxation, and change in driver characteristics. Transp. Res. Part C 26, 367–379.
Zheng, Z., Ahn, S., Monsere, C.M., 2010. Impact of traffic oscillations on freeway crash occurrences. Accident Anal. Prevent. 42, 626–636.
Zhou, B., Wang, Y., Yu, G., Wu, X., 2017. A lane-change trajectory model from drivers' vision view. Transp. Res. Part C: Emerg. Technol. 85, 609–627.