sensors

MDPI

Article

Time-Series-Based Personalized Lane-Changing Decision-Making Model

Ming Ye [1], Lei Pu [1,*], Pan Li [1], Xiangwei Lu [1] and Yonggang Liu [2]

[1] Key Laboratory of Advanced Manufacturing Technology for Automobile Parts, Ministry of Education, Chongqing University of Technology, Chongqing 400054, China
[2] State Key Laboratory of Mechanical Transmissions, College of Mechanical and Vehicle Engineering, Chonqing University, Chongqing 400044, China
* Correspondence: 20220621@stu.cqut.edu.cn

Abstract: In recent years, autonomous driving technology has been changing from "human adapting to vehicle" to "vehicle adapting to human". To improve the adaptability of autonomous driving systems to human drivers, a time-series-based personalized lane change decision (LCD) model is proposed. Firstly, according to the characteristics of the subject vehicle (SV) with respect to speed, acceleration and headway, an unsupervised clustering algorithm, namely, a Gaussian mixture model (GMM), is used to identify its three different driving styles. Secondly, considering the interaction between the SV and the surrounding vehicles, the lane change (LC) gain value is produced by developing a gain function to characterize their interaction. On the basis of the recognition of the driving style, this gain value and LC feature parameters are employed as model inputs to develop a personalized LCD model on the basis of a long short-term memory (LSTM) recurrent neural network model (RNN). The proposed method is tested using the US Open Driving Dataset NGSIM. The results show that the accuracy, F1 score, and macro-average area under the curve (macro-AUC) value of the proposed method for LC behavior prediction are 0.965, 0.951 and 0.983, respectively, and the performance is significantly better than that of other mainstream models. At the same time, the method is able to capture the LCD behavior of different human drivers, enabling personalized driving.

Keywords: autonomous vehicles; lane-change decision; driving style; LSTM; interaction

check for updates

Citation: Ye, M.; Pu, L.; Li, P.; Lu, X.; Liu, Y. Time-Series-Based Personalized Lane-Changing Decision-Making Model. *Sensors* **2022**, *22*, 6659. https://doi.org/10.3390/s22176659

Academic Editors: Peter Han Joo Chong and G. G. Md. Nawaz Ali

Received: 23 July 2022
Accepted: 1 September 2022
Published: 2 September 2022

1. Introduction

Autonomous driving is expected to help reduce traffic accidents, reduce the workload of drivers, and improve the quality of transportation, which has become a research hotspot in recent years [1]. However, the majority of people are skeptical regarding self-driving cars. A poll conducted in Germany found that roughly 25% of respondents were hesitant to utilize self-driving cars because their set driving patterns made them feel psychologically constrained and uneasy [2]. According to studies, in addition to legal and safety considerations, fulfilling user expectations, user acceptability, and accessibility to functional implementation are the fundamental requirements for the effective deployment of autonomous cars [3]. Therefore, enabling autonomous vehicles to have more accurate, comfortable, and personalized behavioral decisions is the key to improving autonomous driving technology.

The LCD problem is one of the more complicated and demanding challenges in autonomous driving technology, and has been intensively investigated by researchers at home and abroad [4]. Existing LCD methods can roughly be divided into two categories: rule-based methods and learning-based methods. Rule-based decision-making methods plan the behavior of autonomous vehicles, and establish a decision-making behavior library based on driving rules, knowledge, experience, and traffic rules [5]. The Gipps [6] model and the CORSIM [7] model are examples of traditional rule-based LCD models. Finite state machines have also frequently been employed in autonomous driving LCD systems, providing an example of rule-based decision-making techniques. Wang et al. [8] proposed a driving LCD model based on the finite state machine, which output the best driving behavior in the current scene by means of a benefit evaluation model. Qi et al. [9] designed a hierarchical finite state machine behavioral decision-making model. Rule-based decision-making methods can handle the general driving environment well, but they are designed in a fixed way, and their mode cannot be adjusted for drivers with different driving styles.

The monitors and sensors of autonomous cars are now able to record the motion state parameters of both the vehicle in front and other nearby vehicles thanks to advancements in autonomous driving decision-making technology. As a result, it is possible to study how vehicles make decisions by mining historical vehicle motion data [10], which is one of the key causes for the progressive rise in popularity of learning-based methodologies. Liu et al. [11] built an autonomous LCD model based on benefit, safety, and tolerance using a Bayesian parameter-optimized support vector machine method. Ma et al. [12] constructed a driving agent model based on a Bayesian network, which integrated vision and decision making. Tang et al. [13] proposed a lane change prediction method based on an adaptive fuzzy neural network. Xie et al. [14] utilized random forests to simulate LC maneuvers from the standpoint of traffic incidents. Díaz-Álvarez et al. [15] employed artificial neural networks to model the behavior of drivers performing LC tasks. Human driving behaviors are typically extremely nonlinear and complicated movements that are challenging to adequately simulate using traditional shallow machine learning models.

Deep neural networks (DNNs) are able to mimic discretionary lane changing (DLC) operations on roads more accurately, and may implicitly include memory effects in their structures [16]. DNNs have been shown to possess excellent application potential for behavioral decision making in complicated settings, such as urban roadways and crossroads [17]. On the basis of NGSIM data, Zhang et al. [18] established a vehicle following and LC simulation model using an LSTM model optimized by the Hybrid Retraining Constraint (HRC) training method. Zhang et al. [19] utilized XGBoost to construct an LC prediction model on the basis of the NGSIM dataset with selected high-dimensional driving feature data. Fang [20] used a deep belief network (DBN) to build an LCD model, and trained the network using NGSIM data, and the results showed that the network outperformed the BP neural network. Although these methods can achieve good LC predictions in driving scenarios, they have limitations with respect to modeling interactions with each other.

The analysis of the above decision-making approaches reveals that less study has been performed on personalized behavioral decision making and more emphasis has been placed on security and scene coverage in current LCD methods. At present, most of the personalized driving research focuses on the design of assisted driving systems and the identification of driving style. In fact, different drivers show different personal preferences in terms of risk perception, ride comfort and travel efficiency. Therefore, from a driving style perspective, enabling autonomous driving to capture human decision-making behavior is expected to provide drivers and occupants with personalized choices. In addition, the above methods only consider the feature quantities (e.g., speed, acceleration and distance) related to the motion state of the subject vehicle when modeling behavioral decisions. However, when an autonomous vehicle is operating in the driving environment, it forms a whole in which both it and the surrounding vehicles affect each other. Autonomous vehicles cannot make precise behavioral decisions by only considering the characteristics of the motion state of the subject vehicle.

To address these issues, on the basis of the LSTM recurrent neural network model, this paper establishes an LCD model that considers the driving style of the SV and includes interactions. Firstly, personalization factors (driving characteristics and driving style) are introduced into the algorithm model, enabling autonomous vehicles to capture individual characteristics and human decision making, with the aim of achieving personalized driving, while also improving the adaptability of automated driving systems to human drivers.

Then, the interaction between the autonomous vehicle and surrounding vehicles is modeled by constructing a gain function to improve the accuracy of the decision-making method. At the same time, the driving environment is extended to three lanes, with the autonomous vehicle being able to perform lane keeping and left–right LC, which is more general. The contributions of this study can be summarized as follows:

(1) A driving style recognition method for autonomous vehicles on highways is proposed, and characteristic variables such as speed, acceleration and headway are selected for quantitative analysis.

(2) The interaction between the main vehicle and the surrounding vehicles is modeled through the gain function, which enhances the understanding of the scenario by the autonomous vehicle.

(3) A personalized decision-making model with interaction is established. The behavioral feature data and LC gain values of different driving styles are used as the input of the LSTM model, and then the time series relationship of various states in the process of lane change is learned.

The rest of this paper is organized as follows: Section 2 briefly describes the overall framework and data processing modules of the paper; Section 3 describes the scheme for driving style identification; Section 4 constructs a personalized LCD model; Section 5 describes the model evaluation and analysis of the results; and Section 6 presents the conclusions.

2. The High-Level Framework of the Personalized LCD Model

This section introduces the high-level framework of an LSTM-based personalized LCD model and proposes a scheme for NGSIM data processing. The following content will be elaborated on the basis of this framework.

2.1. The High-Level Framework

The high-level architecture of the personalized LCD model proposed in this paper is shown in Figure 1, and includes four modules: data processing module, driving style recognition module, LC interaction module, and LCD module. Among them, the main function of the data processing module is to extract the state feature information between the SV and the surrounding vehicles. The information of the SV will be used to learn the driving style by means of the unsupervised clustering algorithm GMM. GMM is able to obtain the best cluster labels for each vehicle on the basis of handcrafted features. The LC interaction module is mainly used to model the interaction between the SV and the surrounding vehicles. Here, the gain function is used to describe its interaction, and the left LC gain value, lane keeping gain value, and right LC gain value of the SV are obtained. On the basis of driving style recognition, the LC feature parameters and LC income value are used as the input of the decision model, and then the LCD command is obtained.

2.2. NGSIM Data Processing

The NGSIM dataset provides actual driving trajectory data on US roadways. According to the study demands, the data of the I−80 and US−101 expressway portions were selected as the original data to develop the LC behavior sample library. The NGSIM information is gathered from various time periods for different places, and it is able to reflect congestion and moderate traffic conditions. A key concern with raw NGSIM data is that there are several substantial errors in the data, such as irregular and missing values for vehicle speed and acceleration. Significant outliers that are 20 times greater than the mean are termed outliers, and outliers are replaced by the mean. Then, the log−exponentially weighted moving average filtering technique (sEMA) introduced by Thiemann [21] is used to filter and preprocess the NGSIM data.

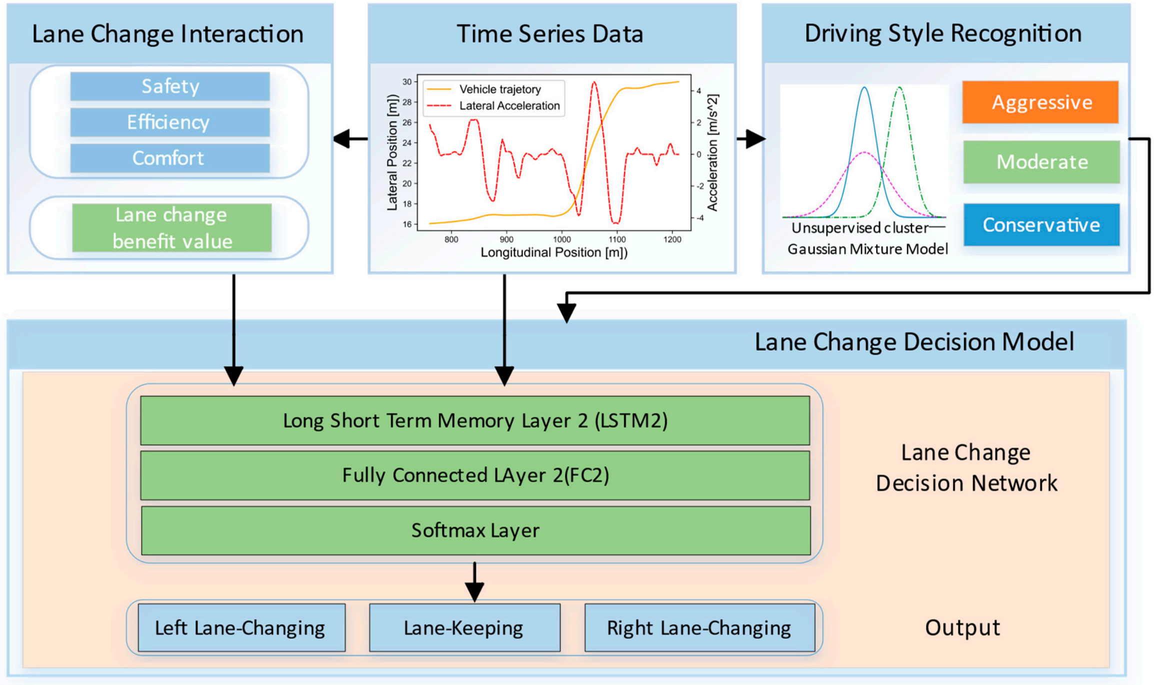


Figure 1. Illustration of the high−level framework for the personalized LCD model.

The NGSIM data provide the trajectory data of all cars on the road, including the speed, acceleration and location information of each vehicle every 0.1 s. It includes three vehicle types: motorbikes, cars and trucks. This research focuses solely on the trajectory data of cars. Here, we conduct the following data filtering:

- Eliminate the vehicle data without the preceding vehicle. Such data will lead to missing parameters for driving style clustering and LCD.
- Considering that there may be MLC vehicles in the auxiliary lanes (lanes 1 and 6), extract the relevant data of the middle lanes (lanes 2, 3, 4 and 5).
- Select the vehicle data of the 10 s complete LC trajectory, that is, 50 frames of data (5 s) before the lane crossing and 50 frames of data (5 s) after the crossing.
- Remove continuous lane change and non-adjacent LC behavior data.

After performing the data processing described above, 403 left LC samples, 121 right LC samples and 1400 lane keeping samples are extracted from the NGSIM dataset. Considering that there are fewer LC samples, this will affect the training accuracy of the decision model. Therefore, data augmentation is performed for the LC samples using a random sampling approach employing a random sampling rate between 0.8 and 0.9 per iteration. For each sample sequence, the sampling unit randomly creates a number within the range of the sampling rate and randomly extracts the data to construct a subsequence of the main sequence. Finally, 1208 left LC samples, 363 right LC samples and 1400 lane keeping samples are obtained, for a total of 2971 samples.

3. Driving Style Recognition Based on GMM

In this section, the Gaussian mixture model (GMM) is used to generate a unique driving style for each vehicle. The Gaussian mixture model is an extension of the Gaussian model, and it is also a linear combination of multiple Gaussian distribution functions. GMM is the fastest−learning probability model. Its principle is to construct the most suitable mixed multi-dimensional Gaussian distribution model by fitting the input dataset [22]. The Gaussian mixture model GMM can be described as follows:

$$p(x_i|\theta) = \sum_{n=1}^{k} \pi_k N(x_i|\mu_k, \Sigma_k) \tag{1}$$

Here, x_i denotes multidimensional feature data. θ is the parameter of the Gaussian mixture model, which can be expressed as $\theta = \{\pi_k, \mu_k, \Sigma_k\}$. π_k is the weight of each Gaussian distribution, meaning the probability of each cluster class being selected, and $\sum_{k=1}^{K} \pi_k = 1$, where μ and Σ are the mean and covariance parameters of the multivariate Gaussian function, and K is the number of models. $N(x_i|\mu_k, \Sigma_k)$ is the univariate Gaussian distribution function in this case, and its form is as follows, where $\Sigma_k = \sigma_k^2$, σ_k represents the standard deviation of the kth class:

$$N(x|\mu, \sum) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right] \tag{2}$$

When clustering datasets, the Gaussian mixture model for unknown parameters is not able to determine which potential components each data point originates from, so it is necessary to estimate the parameters of the Gaussian mixture model. The expectation maximization algorithm (EM) is a commonly used algorithm for GMM parameter estimation. EM is a maximum likelihood estimation algorithm that iteratively computes the maximum value of the cost function [23].

(1) Selection of the Number of Clusters: GMM−based clustering requires the number of clusters k to be pre−specified, and finding the optimal value of k is a challenge. To obtain the optimal number of clusters, this paper adopts the Akaike Information Criterion (*AIC*) and Bayesian Information Criterion (*BIC*) to evaluate the performance of GMM clustering, where k is the number of clusters, N is the number of data points, and L is the maximum likelihood of the objective function. The formulas for calculating *AIC* and *BIC* are as follows:

$$AIC = 2k - 2ln(L) \tag{3}$$

$$BIC = kln(N) - 2ln(L) \tag{4}$$

(2) Feature Selection: Different drivers have distinct driving styles when they drive. This paper classifies drivers by analyzing the identification parameters of each driver's driving characteristics. There is no uniform standard for the selection of driving style characterization parameters, and different scholars choose different indicators. The driving style index parameters selected in this paper are shown in Table 1, in order to fully characterize the driving style of the driver.

Table 1. Statistical description of the features used for driving style recognition.

Symbol	Features	Statistic Values		
ACC	Acceleration	Max Acceleration	Mean Acceleration	Acceleration STD
TH	Time Headway	Max Time Headway	Mean Time Headway	Time Headway STD
Jerk	Jerk	Max Jerk	Mean Jerk	Jerk STD
VX	Lateral speed	Max Lateral speed	Mean Lateral speed	Lateral speed STD
VY	Longitudinal speed	Mean Longitudinal speed	Longitudinal speed STD	
SH	Space Headway	Mean Space Headway	Space Headway STD	

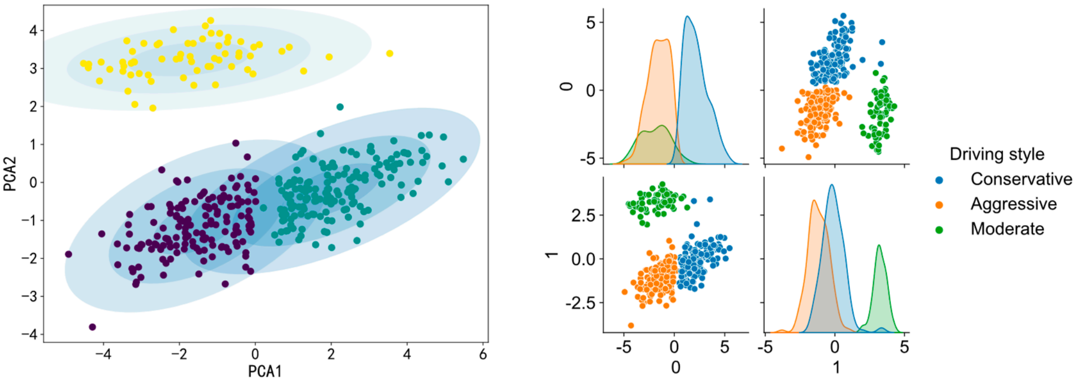
In this research, the influence of GMM on driving style recognition is investigated. The GMM−based driving style identification algorithm is able to determine the driving style of the SV on the basis of few driving behavior data. From Table 1, it can be seen that feature information such as speed, acceleration, lateral speed, and spatial headway time distance of the SV are extracted. At the same time, statistics such as standard deviation, mean and maximum values are introduced to strengthen the robustness of the GMM clustering method recognition.

Here, it is assumed that the driving style of each vehicle does not change at a particular time. The data for the statistical characteristics of the three SV driving styles are shown in Table 2. The driving styles can be defined as aggressive, moderate, and conservative on the basis of the characteristic quantities of lateral speed maximum, acceleration mean, and headway time distance. The definition of these three groups of names is only a reflection of the clustering of the SV driving styles and does not affect the further analysis of personalized LCD.

Table 2. Comparison of characteristic parameters of drivers with different driving styles.

Styles	VY Mean	VY STD	VX Max	VX mean	VX STD	ACC Max	ACC Mean	ACC STD
Conservative	9.391	3.305	1.260	0.595	0.406	3.122	2.670	0.519
Moderate	11.310	2.426	1.337	0.618	0.316	3.715	3.257	0.443
Aggressive	12.682	1.590	1.568	0.721	0.397	4.842	4.006	0.547
Styles	**TH Max**	**TH Mean**	**TH STD**	**Jerk Max**	**Jerk Mean**	**Jerk STD**	**SH Mean**	**SH STD**
Conservative	0.864	0.597	0.674	2.936	−0.056	1.025	34.240	4.764
Moderate	0.565	0.431	0.457	2.518	−0.073	1.106	28.973	3.787
Aggressive	0.540	0.276	0.982	2.637	−0.086	0.901	23.651	1.368

Using the built GMM clustering algorithm, 481 driving behaviors were classed as aggressive, 1286 were classed as moderate, and the remainder (1204) were classed as conservative. It can be seen from Table 2 that drivers with different driving styles exhibit significantly different driving behaviors. Aggressive drivers tend to drive at faster speeds, with shorter following distance and smaller headway. However, conservative drivers chose lower speeds, longer following distances and larger headway distances in order to drive carefully. Figure 2 presents a visualization of the three clusters of driving styles. As shown in the figure, the data for the three driving styles also exhibit varied significant distributions in the feature space.



(**a**) (**b**)

Figure 2. Cluster visualization for 3 driving styles. (**a**) Distribution of three driving styles with respect to two principal components. (**b**) Clustering results based on principal components.

4. Personalized Decision-Making Model Based on LSTM

This section presents the modeling of the LC interaction behavior of vehicles, the fundamental model of LSTM, and the personalized LCD model. Recent breakthroughs in deep learning have made it feasible to extract high-level characteristics from raw data [24]. In this paper, the capacity of LSTM networks to capture complex characteristics in LC scenes is exploited to model LCD behavior on the basis of temporal feature sequences.

4.1. Modeling of LC Interaction Behavior of Vehicles

4.1.1. Scenario Description of Vehicle LC Behavior

Vehicle LC behaviors are typical and crucial activities performed by drivers in response to their driving demands, driving environment conditions, and traffic flow. To describe the LC behavior in a complicated environment, the LC driving scenario represented in Figure 3, below, is adopted. The surrounding vehicles are dispersed according to their location relative to the SV. Special emphasis placed on the fact that the SV is fitted with the decision system and accompanying sensors described in this paper, which are capable of capturing information about the surrounding vehicles in real time.

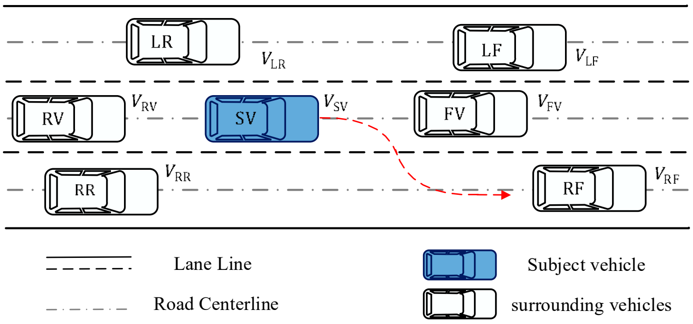


Figure 3. Illustration of lane change driving scenario.

Depending the features of LC behavior and the research aims, the LC process is frequently divided into two stages [16]. The first stage is the LCD stage, which is the process of driver intention creation and route selection. The second stage is the LC execution stage, which consists of the identification of collision−free pathways and the tracking of the produced paths. However, this research only addresses the first stage. The precise meanings of the parameters in Figure 3 are presented in Table 3.

Table 3. The meanings of parameters in the lane-changing scene.

SV	**The Subject Vehicle**
$LF/V/RF$	The vehicle in front of SV in left/middle/right lane
$LR/RV/RR$	The vehicle in front of SV in left/middle/right lane
$V_{LF}/V_{FV}/V_{RF}$	The speed of vehicle in front of SV in left/middle/right lane
$V_{LR}/V_{RV}/V_{RR}$	The speed of vehicle in behind of SV in left/middle/right lane

4.1.2. LC Interaction Behavior Modeling

On the basis of the interaction information, the surrounding vehicles in real traffic scenarios are regarded as an interdependent whole, and their maneuvering behaviors affect each other's decisions. To explain this interaction, the lane change benefit function described in [25] is borrowed, and the effects of SV travel efficiency, collision risk and driver comfort are considered. According to the literature [26], the behavioral decision making of a normal rational driver is a process of continuously pursuing the maximization of benefits.

The vehicle should be driven at as high a speed as possible under safe driving conditions, and the gain function of travel efficiency is set as shown in Equation (5), in which $V_{S,t}$ is the current speed of the SV, V_l is the current speed limit of the road section, where $V_l = 80$ Km/h is taken, and V_f is the current traffic speed of the road section.

$$U_{\text{efficiency}} = \left(V_{H,t} - V_f\right) / \left(V_l - V_f\right) \tag{5}$$

For the calculation of the collision risk benefit, it is necessary to first predict the trajectory of the SV and the surrounding vehicles during the future period T, and then judge the collision risk according to the trajectory.

- For the surrounding vehicles, use the constant turn rate and acceleration model (Constant Turn Rate and Acceleration, CTRA) to predict its trajectory during the future period T.
- For the SV, trajectory prediction needs to be performed according to its driving intention. If it is lane keeping, use the CTRA model to obtain its trajectory; if it is LC, use the quintic polynomial spline curve to fit its LC trajectory.

To improve the accuracy of the collision gain, here, the collision safety condition is defined. The collision benefit is characterized by the distance between vehicles, calculated on the basis of the collision safety condition. Assuming that the length and width of the SV and the surrounding vehicles V_q are L_0, D_0, L_q, and D_q, respectively, the coordinates and heading angles of the two vehicles at a certain time t are $(x_0^t, y_0^t, \varphi_0^t)$ and $\left(x_q^t, x_q^t, \varphi_q^t\right)$, and the difference between the heading angles of the two vehicles $\Delta\varphi = \varphi_0^t - \varphi_q^t$. Then, the conditions under which the SV and surrounding vehicles will not collide at any time t during the prediction period are:

$$\left|\left(x_0^t - x_q^t\right)cos\varphi_q^t + \left(y_0^t - y_q^t\right)sin\varphi_q^t\right| \geq \frac{\sqrt{L_0 + D_0}}{2}\sin(\alpha + |\Delta\varphi|) + \frac{L_q}{2} + \Delta s \tag{6}$$

Then, the gain function of the collision risk between the SV and the surrounding vehicle V_q is:

$$U_{SV,V_q} = \begin{cases} 0\,, & \text{match safety conditions} \\ -\frac{1}{d_{min}^{SV,V_q}}, & \text{dismatch safety conditions} \end{cases} \tag{7}$$

In the formula, d_{min}^{SV,V_q} is the minimum distance between the SV and the surrounding vehicles V_q(q$\epsilon\{1,2,3,\ldots,$m$\}$) during the prediction period T. The collision gain of the SV and all surrounding vehicles in the forecast period is as follows:

$$U_{\text{collision}} = \sum_{t\in T, t\neq 0} U_{SV,V_q} \tag{8}$$

During driving, drivers prefer to drive at a smooth speed; sharp acceleration, deceleration and frequent LC can affect their comfort. Here, the negative value of the sum of the squares of lateral acceleration a_x and longitudinal acceleration a_y integrated over the time period T is used as the comfort gain. The comfort gain can then be expressed as follows:

$$U_{comfort} = -\int_0^T \left(a_x^2 + a_y^2\right)dt \tag{9}$$

The travel efficiency gain $U_{efficiency}$, the collision gain $U_{\text{collision}}$, and the comfort gain $U_{comfort}$ of the SV can be calculated using Equations (5), (8) and (9), above; then, the total gain is

$$U_{total} = \omega_1 U_{efficiency} + \omega_2 U_{\text{collision}} + \omega_3 U_{comfort} \tag{10}$$

In the formula, ω_1, ω_2 and ω_3 are the weight coefficients of each gain. Using the NGSIM dataset, after normalizing the three returns, according to [26], the conjugate gradient method can be used to estimate the respective weight coefficients, as follows: ω_1 = 0.2834, ω_2 = 0.6426, ω_3 = 0.3182.

4.2. The Basic LSTM Model

The Long Short-Term Memory Neural Network (LSTM) is a recurrent neural network (RNN) structure [27]. LSTM overcomes the issue of gradient explosion and gradient disappearance present in RNN [28,29]. Unlike RNN, LSTM adds three additional 'gate' control structures, namely the forget gate, input gate and output gate. The storage unit is used to implement timing storage and timing prediction. The primary unit structure of LSTM is represented in Figure 4.

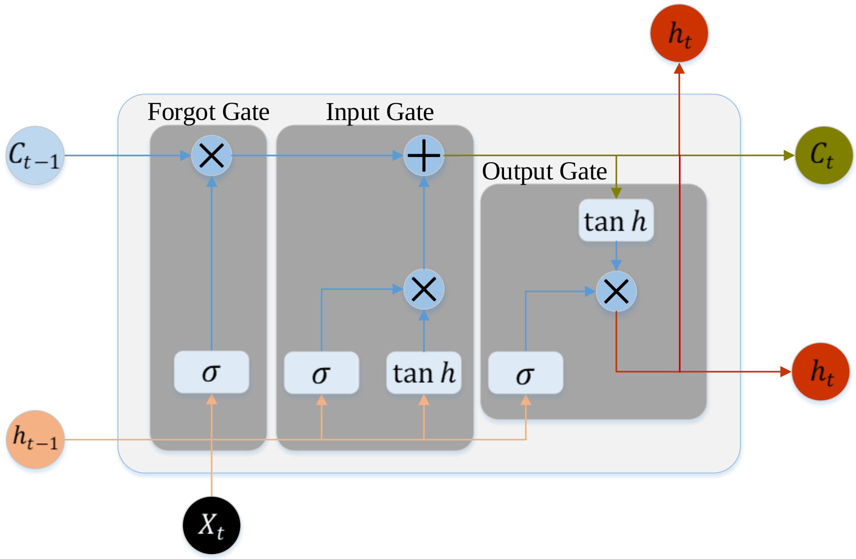


Figure 4. The LSTM cell structure.

In the following, the LSTM cell structure and its concepts are described in further detail. Among these, the sigmoid function and the tanh function are two significant activation functions in the LSTM network structure. Their expressions are as follows.

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{11}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{12}$$

(1) Forget gate: During the training period, the forget gate decides which information is to be abandoned. f_t stands for forget gate, which is defined as follows:

$$f_t = \sigma\left(W_f[h_{t-1}, X_t] + b_f\right) \tag{13}$$

where h_{t-1} is the output memory state of the LSTM in the previous cycle, and X_t represents the state of the input.

(2) Input gate: The input gate determines which fresh information will be input. Its expression is as follows:

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \tag{14}$$

$$\widetilde{C}_t = tanh(W_c[h_{t-1}, X_t] + b_c) \tag{15}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{16}$$

In the expression for the input gate, i_t represents the input gate, $\widetilde{C}_t$ is the estimate for the new cell, and C_t is the new long-term memory formed by this network.

(3) Output gate: The output gate regulates the output of the LSTM. Its expression is as follows:

$$O_t = \sigma(W_0[h_{t-1}, X_t] + b_0) \tag{17}$$

$$h_t = O_t * tanh(C_t) \tag{18}$$

where O_t represents the output of the network at this moment, and h_t is the output of the new hidden state information. $W_* = \left(W_f, W_i, W_c, W_0\right)$ represents the weight matrix of LSTM from unit to gate, and $b_* = \left(b_f, b_i, b_c, b_0\right)$ represents the bias vector of each gate.

4.3. Building the Personalized LCD Model

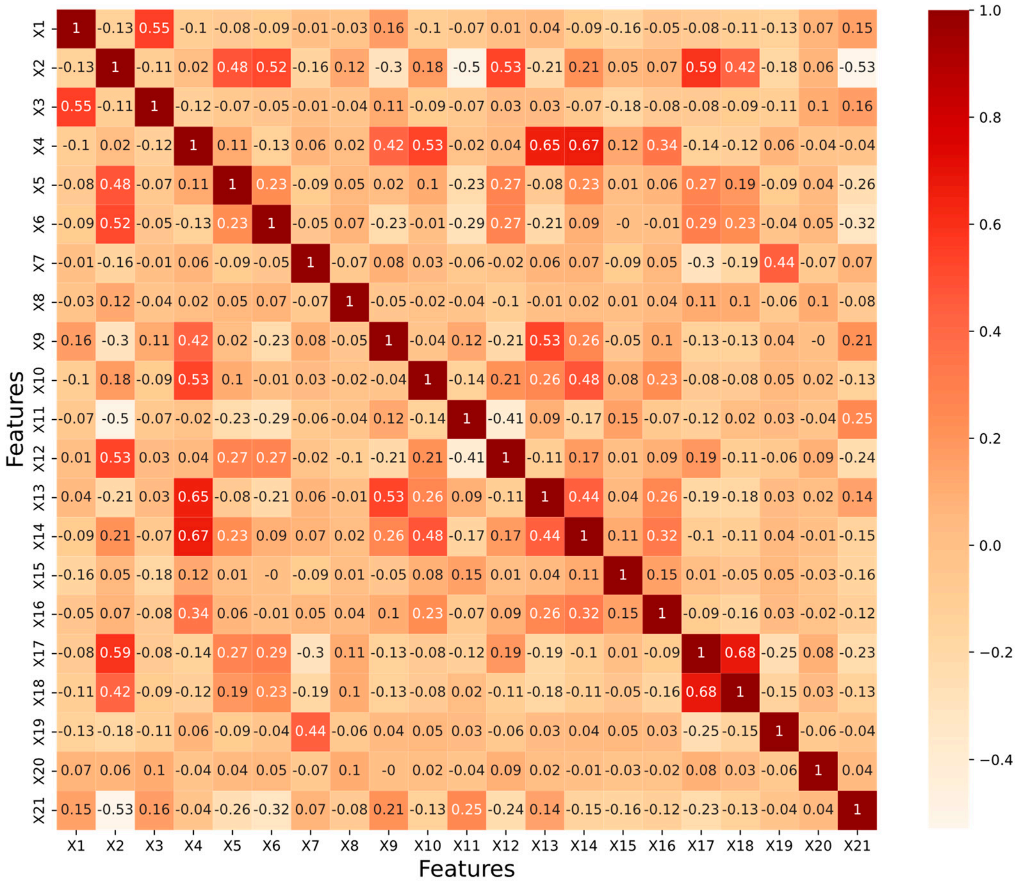
In this paper, LSTM is mainly used to process the LC feature vector and LC gain data of the SV's personalized and generate driving decision instructions. LSTM is able to obtain driving features at different timestamps, and then mine higher−level time series features to represent driving memory effects [30]. Due to the incorporation of historical driving information, the LSTM model is able to achieve improved predictions.

In actual traffic scenarios, the driver's LC conduct is impacted by several factors. In addition to the microscopic driving state of the driver themselves and the surrounding vehicles, drivers have to consider other factors in order to make the best LCD [31]. On the basis of the experience of previous researchers [32,33] regarding the selection of LCD feature parameters, 20 LCD feature parameters were selected and added to the LC gain values in this paper, as shown in Table 4. It is important to note, in particular, that the features used for the LCD are different from those used for the clustering of driving styles in Table 1. Since LC is a continuous process, its requirements need to be calculated at each time step. Driving style, on the other hand, is constant throughout a particular time period, and therefore does not need to be measured as regularly.

Table 4. Statistical description of characteristic parameters of driving decisions.

Number	Features	Description
X_1	V_{SV}	Speed of the SV
X_2	a_{SV}	Acceleration of the SV
X_3	ΔD_{SV_FV}	The distance between SV and FV
X_4	ΔV_{SV_FV}	The speed difference between SV and FV
X_5	Δa_{SV_FV}	The acceleration difference between SV and FV
X_6	ΔD_{SV_RV}	The distance between SV and RR
X_7	ΔV_{SV_RV}	The speed difference between SV and RR
X_8	Δa_{SV_RV}	The acceleration difference between SV and RR
X_9	ΔD_{SV_LF}	The distance between SV and LF
X_{10}	ΔV_{SV_LF}	The speed difference between SV and LF
X_{11}	Δa_{SV_LF}	The acceleration difference between SV and LF
X_{12}	ΔD_{SV_LR}	The distance between SV and LR
X_{13}	ΔV_{SV_LR}	The speed difference between SV and LR
X_{14}	Δa_{SV_LR}	The acceleration difference between SV and LR
X_{15}	ΔD_{SV_RF}	The distance between SV and RF
X_{16}	ΔV_{SV_RF}	The speed difference between SV and RF
X_{17}	Δa_{SV_RF}	The acceleration difference between SV and RF
X_{18}	ΔD_{SV_RR}	The distance between SV and RR
X_{19}	ΔV_{SV_RR}	The speed difference between SV and RR
X_{20}	Δa_{SV_RR}	The acceleration difference between SV and RR
X_{21}	U_{total}	The total lane change profit value

The feature parameters of the vehicle are directly transferred to vectors as model input, which may lead to data redundancy. To check whether there is redundancy of information between features, the correlation of feature parameters was analyzed using Spearman's coefficients [10]. The correlation matrix between the features is shown in Figure 5, where the values are the correlation coefficients m, and $m \in [-1, 1]$. The closer the value of m is to ±1, the higher the linear correlation between the random variables x_i and x_j. Usually, if $|m| > 0.5$, a significant correlation between the variables is implied. Similarly, the deeper the color in the correlation matrix, the higher the connection between the two traits, suggesting that they contain more related information. To filter out redundant variables, $m = 0.6$ was selected as the threshold value. If a pair of variables has a correlation coefficient $m \geq 0.6$, one of them will be deleted to speed up model training.



	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21
X1	1	-0.13	0.55	-0.1	-0.08	-0.09	-0.01	-0.03	0.16	-0.1	-0.07	0.01	0.04	-0.09	-0.16	-0.05	-0.08	-0.11	-0.13	0.07	0.15
X2	-0.13	1	-0.11	0.02	0.48	0.52	-0.16	0.12	-0.3	0.18	-0.5	0.53	-0.21	0.21	0.05	0.07	0.59	0.42	-0.18	0.06	-0.53
X3	0.55	-0.11	1	-0.12	-0.07	-0.05	-0.01	-0.04	0.11	-0.09	-0.07	0.03	0.03	-0.07	-0.18	-0.08	-0.08	-0.09	-0.11	0.1	0.16
X4	-0.1	0.02	-0.12	1	0.11	-0.13	0.06	0.02	0.42	0.53	-0.02	0.04	0.65	0.67	0.12	0.34	-0.14	-0.12	0.06	-0.04	-0.04
X5	-0.08	0.48	-0.07	0.11	1	0.23	-0.09	0.05	0.02	0.1	-0.23	0.27	-0.08	0.23	0.01	0.06	0.27	0.19	-0.09	0.04	-0.26
X6	-0.09	0.52	-0.05	-0.13	0.23	1	-0.05	0.07	-0.23	-0.01	-0.29	0.27	-0.21	0.09	-0	-0.01	0.29	0.23	-0.04	0.05	-0.32
X7	-0.01	-0.16	-0.01	0.06	-0.09	-0.05	1	-0.07	0.08	0.03	-0.06	-0.02	0.06	0.07	-0.09	0.05	-0.3	-0.19	0.44	-0.07	0.07
X8	-0.03	0.12	-0.04	0.02	0.05	0.07	-0.07	1	-0.05	-0.02	-0.04	-0.1	-0.01	0.02	0.01	0.04	0.11	0.1	-0.06	0.1	-0.08
X9	0.16	-0.3	0.11	0.42	0.02	-0.23	0.08	-0.05	1	-0.04	0.12	-0.21	0.53	0.26	-0.05	0.1	-0.13	-0.13	0.04	-0	0.21
X10	-0.1	0.18	-0.09	0.53	0.1	-0.01	0.03	-0.02	-0.04	1	-0.14	0.21	0.26	0.48	0.08	0.23	-0.08	-0.08	0.05	0.02	-0.13
X11	-0.07	-0.5	-0.07	-0.02	-0.23	-0.29	-0.06	-0.04	0.12	-0.14	1	-0.41	0.09	-0.17	0.15	-0.07	-0.12	0.02	0.03	-0.04	0.25
X12	0.01	0.53	0.03	0.04	0.27	0.27	-0.02	-0.1	-0.21	0.21	-0.41	1	-0.11	0.17	0.01	0.09	0.19	-0.11	-0.06	0.09	-0.24
X13	0.04	-0.21	0.03	0.65	-0.08	-0.21	0.06	-0.01	0.53	0.26	0.09	-0.11	1	0.44	0.04	0.26	-0.19	-0.18	0.03	0.02	0.14
X14	-0.09	0.21	-0.07	0.67	0.23	0.09	0.07	0.02	0.26	0.48	-0.17	0.17	0.44	1	0.11	0.32	-0.1	-0.11	0.04	-0.01	-0.15
X15	-0.16	0.05	-0.18	0.12	0.01	-0	-0.09	0.01	-0.05	0.08	0.15	0.01	0.04	0.11	1	0.15	0.01	-0.05	0.05	-0.03	-0.16
X16	-0.05	0.07	-0.08	0.34	0.06	-0.01	0.05	0.04	0.1	0.23	-0.07	0.09	0.26	0.32	0.15	1	-0.09	-0.16	0.03	-0.02	-0.12
X17	-0.08	0.59	-0.08	-0.14	0.27	0.29	-0.3	0.11	-0.13	-0.08	-0.12	0.19	-0.19	-0.1	0.01	-0.09	1	0.68	-0.25	0.08	-0.23
X18	-0.11	0.42	-0.09	-0.12	0.19	0.23	-0.19	0.1	-0.13	-0.08	0.02	-0.11	-0.18	-0.11	-0.05	-0.16	0.68	1	-0.15	0.03	-0.13
X19	-0.13	-0.18	-0.11	0.06	-0.09	-0.04	0.44	-0.06	0.04	0.05	0.03	-0.06	0.03	0.04	0.05	0.03	-0.25	-0.15	1	-0.06	-0.04
X20	0.07	0.06	0.1	-0.04	0.04	0.05	-0.07	0.1	-0	0.02	-0.04	0.09	0.02	-0.01	-0.03	-0.02	0.08	0.03	-0.06	1	0.04
X21	0.15	-0.53	0.16	-0.04	-0.26	-0.32	0.07	-0.08	0.21	-0.13	0.25	-0.24	0.14	-0.15	-0.16	-0.12	-0.23	-0.13	-0.04	0.04	1

Figure 5. Correlation matrix among features.

To achieve a personalized LCD, in this paper, a decision model is developed, as shown in Figure 6. The structure of the model contains five layers, where the input layer is used to receive information on the state of the SV and the surrounding vehicles. The data accepted by the input layer are time series variables that characterize the lane changing behavior of the SV, the length of which is denoted by t_p, and the data for each time step are as follows:

$$X = \left[X_1^t, X_2^t, X_3^t, \ldots, U_{LLC}^t, U_{LK}^t, U_{RLC}^t\right] \tag{19}$$

where $\left[X_1^t, X_2^t, X_3^t, \ldots\ X_{20}^t\right]$ is the decision feature vector extracted from the SV trajectory with known driving style, and U_{LLC}^t, U_{LK}^t and U_{RLC}^t are the gain values of the SV performing left LC, lane keeping and right LC at time t, respectively.

The second layer of the model's structure is composed of a two−layer LSTM neural network. This LSTM network layer captures the key factors influencing vehicle behavior decisions from the high−dimensional time series data and then classifies the input data appropriately. The third layer is a fully connected layer, which is used to transform the dimensions of the transmitted data. The fourth layer is the Softmax layer, which is processed by the Softmax classifier to obtain the LC behavior probability matrix of the SV. This probability matrix is as follows:

$$P = [P_1, P_2, P_3] \tag{20}$$

where P_1, P_2 and P_3 are the probabilities of left LC, lane keeping and right LC for the SV, respectively. The fifth layer is the output layer of the model, which obtains the lane change decision results at the prediction time t_p. According to [34], the LC time in a highway environment is generally between 3.5 s and 6 s, and a complete LC process can be performed in an average of 5 s. However, this paper only considers the LC decision, so $t_p = 3$ s is chosen.

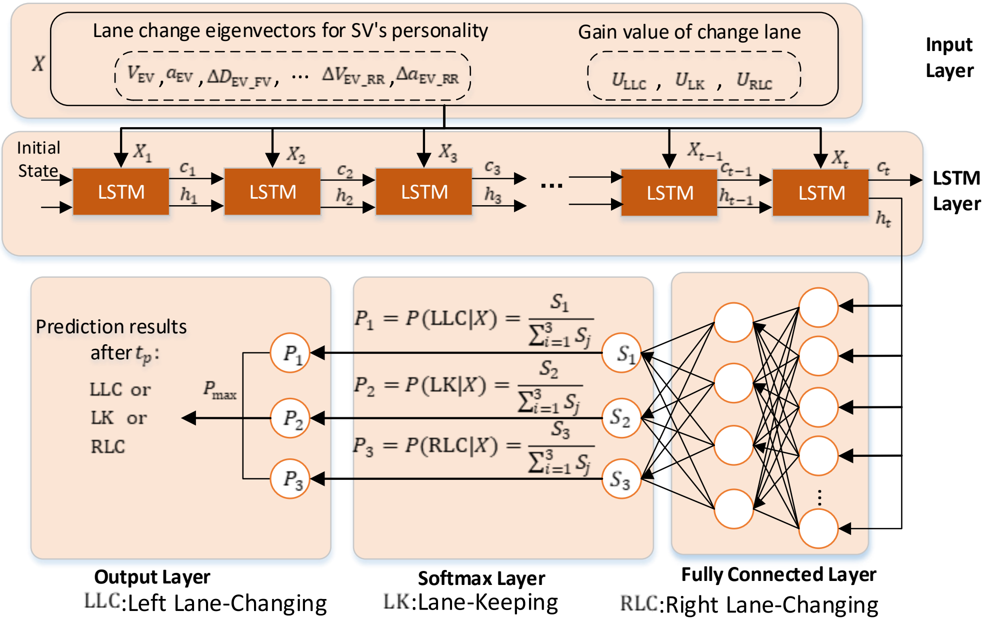

Lane change eigenvectors for SV's personality
X
$V_{EV}, a_{EV}, \Delta D_{EV\_FV}, \cdots \Delta V_{EV\_RR}, \Delta a_{EV\_RR}$
Gain value of change lane
U_{LLC}, U_{LK}, U_{RLC}
Input Layer
Initial State
X_1 X_2 X_3 X_{t-1} X_t
c_1 h_1 c_2 h_2 c_3 h_3 c_{t-1} h_{t-1} c_t h_t
LSTM
LSTM Layer
Prediction results after t_p:
LLC or
LK or
RLC
P_{max}
P_1 P_2 P_3
$P_1 = P(LLC|X) = \frac{S_1}{\sum_{i=1}^{3} S_j}$
$P_2 = P(LK|X) = \frac{S_2}{\sum_{i=1}^{3} S_j}$
$P_3 = P(RLC|X) = \frac{S_3}{\sum_{i=1}^{3} S_j}$
S_1 S_2 S_3
Output Layer
Softmax Layer
Fully Connected Layer
LLC:Left Lane-Changing
LK:Lane-Keeping
RLC:Right Lane-Changing

Figure 6. Structure of the lane change decision model.

The aim of this LCD model is to enable the SV to generate safe, stable, and personalized LC decisions on the basis of its own driving state and the location information of surrounding vehicles. Firstly, the LC feature parameters of the SV are extracted on the basis of GMM clustering. Then, the personalized LC feature parameters and LC gain values are utilized as feature inputs to train the model and generate the resulting LCD.

5. Model Evaluation and Analysis

In this section, the LCD model is evaluated using multiple evaluation metrics, and a comparison with other models is performed. Furthermore, LCD models considering various driving styles are evaluated. The training tasks for the dataset pre−processing and decision models in this paper were performed by MATLAB 2020a and pytorch-based jupyter on a laptop in the Windows 10 environment.

5.1. Data Preparation

In this paper, data were extracted from the NGSIM dataset to train the decision model. On the basis of the recognition of SV driving style, further LC decision parameters were extracted. Sequential data with a predicted time length of 3 s between the SV and surrounding vehicles were extracted using the sliding window method. At the same time, the personalized LCD samples were labeled with data. The left LC, lane keeping, and right LC were represented using 0, 1 and 2, respectively. Then, 80% of the cases were randomly selected from the total sample data as training samples for the decision model, while the remaining 20% were used as test samples.

5.2. Model Parameter Settings

In this paper, the LSTM network is used to learn the interaction characteristics of the vehicle, and the behavioral decision of the SV is output through Softmax. The parameter settings for the LSTM network model are shown in Table 5. To achieve better model training accuracy, we set the batch size to 512, the number of nodes in the hidden layer to 256, and the number of iterations for model training to 300. The cross−entropy function is used to train the LSTM network model. Importantly, a learning rate that is too high will cause the loss function to ignore the minimum loss, while a learning rate that is too small will make training exceedingly sluggish. On the basis of previous research and experience [35], the optimal learning rate was finally determined to be 0.0124 by plotting the learning rate against training loss.

Table 5. The settings of the LSTM network model.

Item	Description	Value
Batch size	Batches per training	512
Hidden size	Number of hidden neural units of LSTM	256
Epoch	The number of iterations for LSTM model training	300
Output size	LSTM model output size	3
Learning rate	Learning rate	0.0124
Loss function	Instruct LSTM to update weight parameters	CrossEntropy

5.3. Results and Analysis

5.3.1. Evaluation Indicators of the Model

The model is trained using the training set, and the model performance is evaluated using the test set. Performance metric testing involves the following.

(1) Accuracy rate A_{cc}: Indicates the proportion of the number of correct predictions in the model test instance to the total number of test instances, and its calculation formula is as follows:

$$A_{cc} = \frac{T_p + T_N}{T_p + T_N + F_P + F_N} \tag{21}$$

where T_p is the number of samples where the true label is positive and the prediction is also positive; T_N is the number of samples where the true label is negative and the prediction is also negative; F_P is the number of samples where the true label is negative and the prediction is positive; and F_N is the number of samples where the true label is positive and the prediction is negative.

(2) F_1 score: Refers to the harmonic mean of the precision rate P and the recall rate R, and its calculation formula is as follows:

$$P = \frac{T_P}{T_p + F_p} \tag{22}$$

$$R = \frac{T_P}{T_p + F_N} \tag{23}$$

$$F_1 = \frac{2 * P * R}{P + R} \tag{24}$$

(3) Test loss T_{loss}: The cross−entropy loss of the decision model on the test set; its expression is as follows:

$$T_{loss} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{n=1}^{N} P_{iN}\log(q_{iN}) \tag{25}$$

where m is the number of samples tested by the model, n is the LC decision intention, P_{iN} is the true label of the ith sample in the test set belonging to the Nth decision intention, and q_{iN} is the probability of the model predicting that the *i*th sample belongs to the Nth decision intention.

In the NGSIM dataset, the number of cars executing lane-maintaining actions is substantially larger than the number of cars performing LC. From the previous LC data extraction, it can be recognized that the number of right LC is smaller than the number of left LC. When the multi−class data are uneven, the accuracy evaluation method is flawed. Therefore, this study presents assessment measures such as accuracy rate, recall rate, and F1 score. Meanwhile, the Macro-average receiver operating characteristic curve (ROC curve) and the area under the curve (Macro-AUC) are also utilized to assess the performance of the LCD model.

5.3.2. Decision Model Evaluation and Comparison

To verify the effectiveness of the LCD method proposed in this paper, the relevant performance metrics of several models are compared and analyzed using the same dataset. Here, the LCD model LSTM_I proposed in this paper is compared with five other algorithmic models. These are: the LSTM model, the SVM model [11], the Logistic Regression (LR) model [36], the KNN model [37] and the XGB model [10]. Of these, the LSTM model is a single LCD model that does not take into consideration the introduction of interaction feature parameters. In addition, the other models (SVM, LR, KNN and XGB) are applied using Python and the "scikit-learn" package. It should be noted that this part of the dataset does not specify the driving style of each vehicle.

The optimization of model parameters is an issue that needs to be considered here. Common model parameter optimization methods include genetic algorithm, random search, and Bayesian optimization. Compared with other optimization algorithms, the Bayesian approach exhibits superior performance in fewer iterations [38]. Therefore, the Bayesian method is selected here to optimize the model. A selection of critical parameters and variables following the optimization of each model is shown in Table 6.

Table 6. Key parameters following the optimization of each model.

Model	Optimal Parameter
SVM	C = 7.42; gamma = 0.001; kernel: 'rbf'
LR	Penalty: 'l2'; tol = 1e − 4; max_iter = 640; solver: 'sag'; multi_class = 'multinomial'
XGB	learning_rate = 0.0124; max_depth = 6; min_child_weight = 0.1; gamma = 0.36
KNN	n_neighbors = 5; leaf_size = 30, weights: 'uniform'; algorithm = 'auto'

In this paper, the performance of the six decision models is tested using data under different time windows. To improve the real-time performance and accuracy of the decision models, the upper limit of the time length was chosen to be 3 s. The accuracy of the different LCD models was calculated for time windows of 0.5, 1.0, 1.5, 2.0, 2.5, and 3 s, respectively, as shown in Figure 7. As can be seen from Figure 7, the overall accuracy of the model increases as the dataset increases. The accuracy of all models stabilized when the data time length reached 2.5 s. Furthermore, the accuracy of our models already exceeded 95% at a time length of 2.0 s for the test data. Compared with the LSTM model without the introduction of interactions, our model demonstrated an improvement in overall accuracy by 4.26%. Compared with SVM, LR, KNN and XGB, the overall accuracy was improved by 8.89%, 11.04%, 14.65% and 4.96%, respectively. This shows that the introduction of the interaction module did improve the accuracy of the autonomous driving LCD.

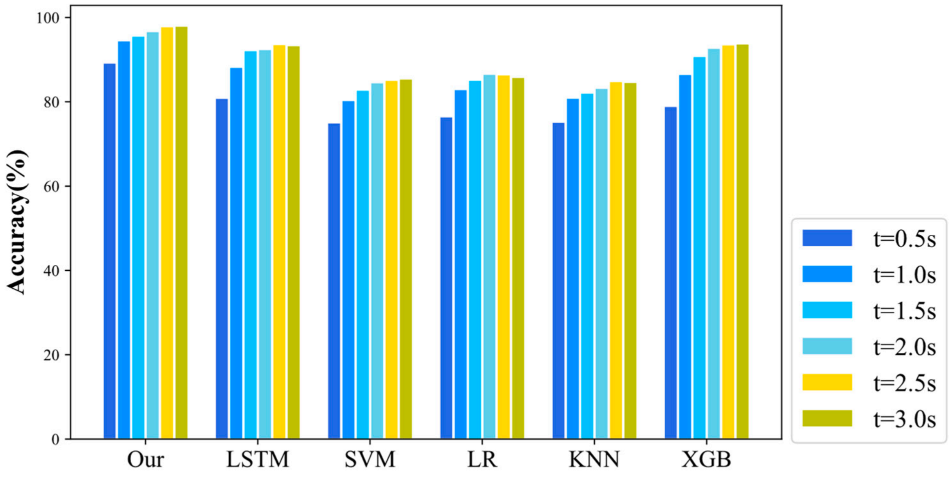


Figure 7. Accuracy at different time points.

The LCD model is established 3 s before LC. The six LC models are trained and tested on the basis of datasets I−80 and US−101, and the results are shown in Table 7. As can be seen from Table 7, the LSTM_I model has combined accuracy, recall, and F1 score values of 0.965, 0.951 and 0.956, respectively, demonstrating better classification accuracy than the other models. This indicates that the model is better able to mine the potential features in the LC process and generate more accurate decision results. Additionally, the test loss values of different decision models are presented in Table 7. From Table 7, it can be seen that our model also shows better performance in terms of test loss values. This further demonstrates that the decision model is capable of making more accurate predictions.

Table 7. Performance comparison of the different approaches.

Model	**Precision (%)**			**Recall (%)**			**F1-score (%)**			T_{loss}
	LCL	**LK**	**LCR**	**LCL**	**LK**	**LCR**	**LCL**	**LK**	**LCR**	
LSTM_I	96.54	97.33	95.46	98.24	96.57	90.37	97.28	97.10	92.84	0.127
LSTM	94.42	92.26	91.51	88.04	94.01	89.70	93.67	93.19	92.60	0.210
SVM	89.03	86.49	88.43	84.18	90.07	80.59	87.08	88.44	83.23	0.267
LR	86.21	90.54	80.60	84.36	83.28	83.16	84.50	85.39	81.62	0.368
XGB	93.86	91.47	90.33	95.06	92.39	81.64	94.16	91.22	85.35	0.156
KNN	76.63	81.68	84.20	80.17	82.54	83.66	78.84	81.40	83.59	0.423

Figure 8 shows the macroscopic receiver operating characteristic curves of the six models. Among them, the KNN algorithm model has the worst performance, with a Macro−AUC value of 0.815. The Macro−AUC value of the SVM model is slightly higher than that of the logistic regression model, with Macro−AUC values of 0.894 and 0.865, respectively. In addition, the LSTM model algorithm exhibits better performance than the XGB model algorithm, with Macro−AUC values of 0.952 and 0.937, respectively. Meanwhile, the LCD model LSTM_I proposed in this paper shows the best performance, with a Macro−AUC value of 0.983.

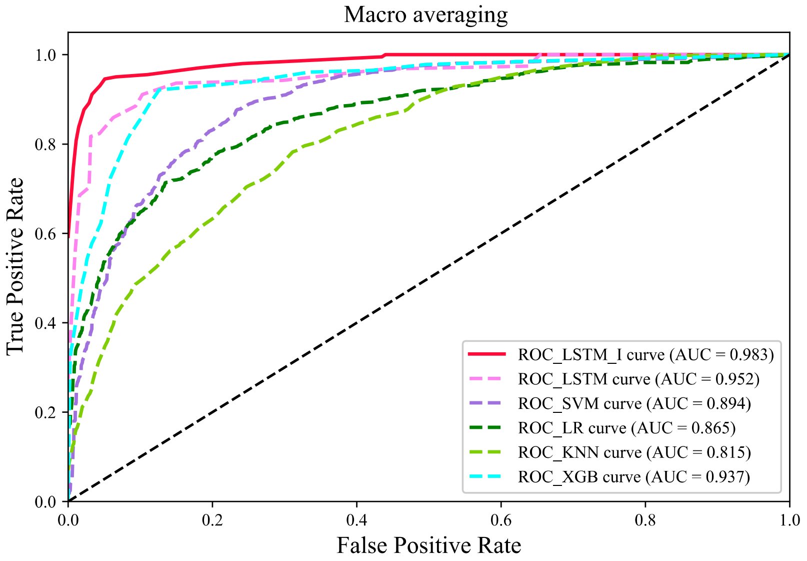


Figure 8. The Macro-average ROC curve of each model.

In general, the LCD model proposed in this paper shows better performance than the other decision models. The reason for this is that, during the process of traffic flow, the SV and the surrounding vehicles form an interacting whole. The behavioral decisions of the SV are not only influenced by its own state, but also by the driving environment and the behavior of the surrounding vehicles. In addition, the LCD model proposed in this paper is able to learn the correlation between vehicles on the basis of the interaction and mine the key features affecting LC. Under the same driving environment, this decision model is able to make more reasonable behavioral decisions on the basis of the surrounding vehicle information and its own state.

5.3.3. Model Evaluation Considering Driving Style

In this section, the LSTM_I model is further evaluated using the various driving style data obtained in Section 3. The classification results of the test data for different driving styles are shown in Table 8. From Table 8, it can be seen that the decision model has high accuracy in predicting non-LC behavior, while it has the lowest accuracy in predicting right LC behavior. Meanwhile, among the different driving styles, moderate has the highest overall accuracy, followed by conservative and aggressive, from high to low, respectively. This is because there is an imbalance in the data of the three driving styles extracted in this paper, and more feature data could improve the accuracy of the model to a certain extent. In addition, the prediction accuracy when taking different driving styles into consideration was overall higher than that for samples without driving style classification.

Table 8. Prediction accuracy of LCD model considering driving style.

Decision Results	**Driving Style**		
	Left Lane Changing	**Lane Keeping**	**Right Lane Changing**
Conservative	96.95%	98.87%	94.57%
Moderate	97.94%	99.79%	96.59%
Aggressive	94.85%	97.68%	92.29%
Mean	96.58%	98.78%	94.48%
Non-Classified	94.18%	98.14%	92.10%

Table 9 shows the prediction accuracy of the six models for predictions based on data related to driving behavioral characteristics for different driving styles. The data in the table are the average of the prediction accuracy of each model for behaviors such as LCL, LK and RCL. From Table 9, it can be seen that the overall accuracy of the LSTM_I model is better than that of the other five models, both for classified and non-classified samples. Among them, the average accuracy of the LSTM_I model prediction, performed on the basis of three driving style feature data points, is 97.28%, while the accuracies of the other five models are 92.98%, 86.29%, 84.32%, 91.69% and 81.35%, respectively. In addition, the same conclusion as that presented in Table 8 can also be obtained: for the six LCD models, the prediction accuracy following classification is higher than that for the unclassified samples.

Table 9. The prediction accuracy of 6 models considering the driving style of the SV.

Driving styles	**LSTM_I**	**LSTM**	**SVM**	**LR**	**XGB**	**KNN**
Conservative	96.80%	91.76%	84.61%	85.65%	90.58%	81.94%
Moderate	98.46%	95.31%	92.62%	90.03%	95.31%	86.43%
Aggressive	96.57%	91.89%	81.64%	77.27%	89.18%	74.67%
Mean	97.28%	92.98%	86.29%	84.32%	91.69%	81.35%
Non-Classified	96.51%	91.59%	83.96%	82.81%	89.89%	77.2%

Figure 9 shows several examples of lane change predictions for vehicles with different driving styles. Here, we use the LSTM_I model to predict the driving behavior of real vehicles found in the NGSIM data, and we color-code the actual trajectories of the predicted vehicles according to the model's predictions. In this case, the data segments that correctly predict the LK and LC behavior are represented in purple and green, respectively. Data segments that are not involved in the prediction are marked in black, while data that predict LK and LC incorrectly are shown in red. As can be seen from Figure 9, the model has an overall prediction accuracy of over 94% for vehicles with different driving styles. In contrast to Table 8, the prediction accuracy is higher for aggressive driver behavior. In addition, the model has a prediction time range of 12 ms to 18 ms, which is sufficient for real-world situations. Overall, the decision model proposed in this paper is able to produce good lane change predictions for drivers with different driving styles.

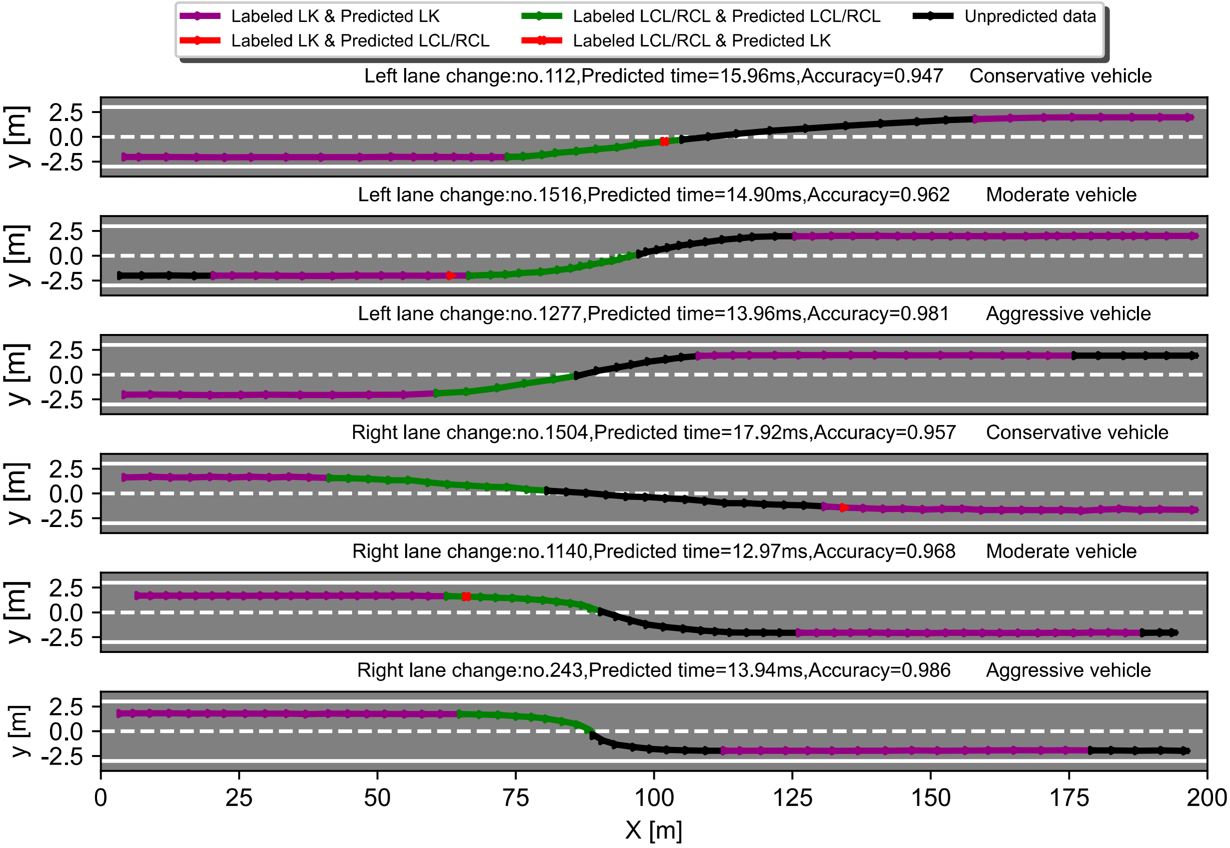


Figure 9. Examples of lane change prediction for vehicles with different driving styles. Purple and green represent correctly predicted data segments for LC and LK, respectively, black represents unpredicted data, and red represents incorrectly predicted data.

6. Conclusions

This research proposes an LCD model that considers SV driving styles as well as interactions. Feature variables such as speed, acceleration and headway time distance are selected, and the style type is identified for each vehicle sample on the basis of the unsupervised clustering algorithm GMM. The interaction between the SV and the surrounding vehicles is described by constructing a gain function, which takes into account the safety, driving efficiency, and comfort of the SV. The LC feature variables with different driving styles and the LC gain values are used as model inputs to construct an LSTM−based personalized LCD model.

To verify the effectiveness of the model, real vehicle trajectory data, MGSIM, were used to evaluate the model. The model was also compared with other models, and the results showed that the model outperformed the other models in terms of accuracy, F1 score, and macro−AUC value. This indicates that the model is able to make more accurate behavioral decisions on the basis of the state of its own vehicle and information regarding the surrounding vehicles. Behavioral decisions were also evaluated for different driving styles. The test results showed that the personalized LCD framework is able to make sound decisions on the basis of different driving styles, and it can provide personalized options for different drivers.

Future work will focus on the improvement of the proposed model algorithms and the application of real−time on−board hardware systems for algorithm validation. Additionally, other traffic participants (e.g., pedestrians, non-motorized vehicles) will be taken into consideration in the driving environment in order to further improve the decision−making capability of autonomous vehicles.

Author Contributions: Conceptualization and methodology, M.Y. and L.P.; Formal analysis and writing—manuscript preparation, L.P.; Dataset collection and analysis, P.L. and X.L. Supervision and writing—review and editing, M.Y. and Y.L. All authors discussed and commented on the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Chongqing Municipal Technology Innovation and Application Development Project (Grant No. cstc2021jscx-gksbX0043).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no competing interest.

References

1. Badue, C.; Guidolini, R.; Carneiro, R.V.; Azevedo, P.; Cardoso, V.B.; Forechi, A.; Jesus, L.; Berriel, R.; Paixao, T.M.; Mutz, F. Self-driving cars: A survey. *Expert Syst. Appl.* **2021**, *165*, 113816. [CrossRef]
2. Ossig, J.; Cramer, S.; Eckl, A.; Bengler, K. Tactical decisions for lane changes or lane following: Assessment of automated driving styles under real-world conditions. *IEEE Trans. Intell. Veh.* 2022; *in press*. [CrossRef]
3. Tang, X.; Yang, K.; Wang, H.; Wu, J.; Qin, Y.; Yu, W.; Cao, D. Prediction-uncertainty-aware decision-making for autonomous vehicles. *IEEE Trans. Intell. Veh.* 2022; *in press*. [CrossRef]
4. Li, X.; Sun, J.-Q. Studies of vehicle lane-changing dynamics and its effect on traffic efficiency, safety and environmental impact. *Phys. A Stat. Mech. Its Appl.* **2017**, *467*, 41–58. [CrossRef]
5. Jin, C.-J.; Knoop, V.L.; Li, D.; Meng, L.-Y.; Wang, H. Discretionary lane-changing behavior: Empirical validation for one realistic rule-based model. *Transp. A* **2019**, *15*, 244–262. [CrossRef]
6. Gipps, P.G. A model for the structure of lane-changing decisions. *Transp. Res. Part B Methodol.* **1986**, *20*, 403–414. [CrossRef]
7. Bloomberg, L.; Dale, J. Comparison of vissim and corsim traffic simulation models on a congested network. *Transport. Res. Rec.* **2000**, *1727*, 52–60. [CrossRef]
8. Wang, P.; Gao, S.; Li, L.; Cheng, S.; Zhao, H. Research on driving behavior decision making system of autonomous driving vehicle based on benefit evaluation model. *Arch. Transp.* **2020**, *53*, 21–36. [CrossRef]
9. Wang, X.; Qi, X.; Wang, P.; Yang, J. Decision making framework for autonomous vehicles driving behavior in complex scenarios via hierarchical state machine. *Auton. Intell. Syst.* **2021**, *1*, 1–12. [CrossRef]
10. Gu, X.; Han, Y.; Yu, J. A novel lane-changing decision model for autonomous vehicles based on deep autoencoder network and xgboost. *IEEE Access* **2020**, *8*, 9846–9863. [CrossRef]
11. Liu, Y.; Wang, X.; Li, L.; Cheng, S.; Chen, Z. A novel lane change decision-making model of autonomous vehicle based on support vector machine. *IEEE Access* **2019**, *7*, 26543–26550. [CrossRef]
12. Ma, J.; Xie, H.; Song, K.; Liu, H. A bayesian driver agent model for autonomous vehicles system based on knowledge-aware and real-time data. *Sensors* **2021**, *21*, 331. [CrossRef]
13. Tang, J.; Liu, F.; Zhang, W.; Ke, R.; Zou, Y. Lane-changes prediction based on adaptive fuzzy neural network. *Expert Syst. Appl.* **2018**, *91*, 452–463. [CrossRef]
14. Xie, J.; Zhu, M. Maneuver-based driving behavior classification based on random forest. *IEEE Sens. Lett.* **2019**, *3*, 1–4. [CrossRef]
15. Díaz-Álvarez, A.; Clavijo, M.; Jiménez, F.; Talavera, E.; Serradilla, F. Modelling the human lane-change execution behaviour through multilayer perceptrons and convolutional neural networks. *Transport. Res. F Traf.* **2018**, *56*, 134–148. [CrossRef]
16. Liu, X.; Liang, J.; Xu, B. A deep learning method for lane changing situation assessment and decision making. *IEEE Access* **2019**, *7*, 133749–133759. [CrossRef]
17. Shengbo, L.; Yang, G.; Lian, H.; Hongbo, G.; Jingliang, D.; Shuang, L.; Yu, W.; Bo, C.; Keqiang, L.; Wei, R.; et al. The key technology of deep neural network and its application in the field of autonomous driving. *China J. Automot. Saf. Energy* **2019**, *10*, 119–145.
18. Zhang, X.; Sun, J.; Qi, X.; Sun, J. Simultaneous modeling of car-following and lane-changing behaviors using deep learning. *Transp. Res. Part C Emerg. Technol.* **2019**, *104*, 287–304. [CrossRef]
19. Zhang, Y.; Shi, X.; Zhang, S.; Abraham, A. A xgboost-based lane change prediction on time series data using feature engineering for autopilot vehicles. *IEEE Trans. Intell. Transp. Syst.* 2022; *in press*. [CrossRef]
20. Zhezhe, F. Modeling and analysis of lane changing behavior based on deep learning. Master's Thesis, Beijing Jiaotong University, Beijing, China, 2018.
21. Montanino, M.; Punzo, V. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transport. Res. B Meth.* **2015**, *80*, 82–106. [CrossRef]

22. McDowell, I.C.; Manandhar, D.; Vockley, C.M.; Schmid, A.K.; Reddy, T.E.; Engelhardt, B.E. Clustering gene expression time series data using an infinite gaussian process mixture model. *PLoS Comput. Biol.* **2018**, *14*, e1005896. [CrossRef]
23. An, P.; Wang, Z.; Zhang, C. Ensemble unsupervised autoencoders and gaussian mixture model for cyberattack detection. *Inform. Process. Manag.* **2022**, *59*, 102844. [CrossRef]
24. Gao, J.; Yi, J.; Zhu, H.; Murphey, Y.L. A personalized lane-changing model for advanced driver assistance system based on deep learning and spatial-temporal modeling. *SAE Int. J. Transp. Saf.* **2019**, *7*, 163–174. [CrossRef]
25. Xiaolin, S.; Yanbing, Z.; Haotian, C.; Mingjun, L.; Binlin, Y. A lane-changing intention recognition method based on long short-term memory network. *China J. Highw. Transp.* **2021**, *34*, 236–245.
26. Ali, Y.; Zheng, Z.; Haque, M.M.; Yildirimoglu, M.; Washington, S. Understanding the discretionary lane-changing behaviour in the connected environment. *Accid. Anal. Prev.* **2020**, *137*, 105463. [CrossRef] [PubMed]
27. Zhao, J.; Gao, Y.; Bai, Z.; Wang, H.; Lu, S. Traffic speed prediction under non-recurrent congestion: Based on lstm method and beidou navigation satellite system data. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 70–81. [CrossRef]
28. Mozaffari, S.; Al-Jarrah, O.Y.; Dianati, M.; Jennings, P.; Mouzakitis, A. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE trans. Intell. Transp. Syst.* **2020**, *23*, 33–47. [CrossRef]
29. Staudemeyer, R.C.; Morris, E.R. Understanding lstm—A tutorial into long short-term memory recurrent neural networks. *arXiv* **2019**, arXiv:1909.09586.
30. Xie, D.-F.; Fang, Z.-Z.; Jia, B.; He, Z. A data-driven lane-changing model based on deep learning. *Transp. Res. Part C Emerg. Technol.* **2019**, *106*, 41–60. [CrossRef]
31. Deng, J.-H.; Feng, H.-H. A multilane cellular automaton multi-attribute lane-changing decision model. *Physica. A* **2019**, *529*, 121545. [CrossRef]
32. Ma, Y.; Zhang, Z.; Ihler, A. Multi-lane short-term traffic forecasting with convolutional lstm network. *IEEE Access* **2020**, *8*, 34629–34643. [CrossRef]
33. Guo, Y.; Zhang, H.; Wang, C.; Sun, Q.; Li, W.; Applications, I. Driver lane change intention recognition in the connected environment. *Physica. A* **2021**, *575*, 126057. [CrossRef]
34. Yang, D.; Zheng, S.; Wen, C.; Jin, P.J.; Ran, B. A dynamic lane-changing trajectory planning model for automated vehicles. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 228–247. [CrossRef]
35. Tang, L.; Wang, H.; Zhang, W.; Mei, Z.; Li, L. Driver lane change intention recognition of intelligent vehicle based on long short-term memory network. *IEEE Access* **2020**, *8*, 136898–136905. [CrossRef]
36. Ghasemzadeh, A.; Ahmed, M.M. Utilizing naturalistic driving data for in-depth analysis of driver lane-keeping behavior in rain: Non-parametric mars and parametric logistic regression modeling approaches. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 379–392. [CrossRef]
37. Yang, S.; Wang, W.; Lu, C.; Gong, J.; Xi, J. A time-efficient approach for decision-making style recognition in lane-changing behavior. *IEEE Trans. Hum. Mach. Syst.* **2019**, *49*, 579–588. [CrossRef]
38. Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A.G.; Bakshy, E. Botorch: A framework for efficient monte-carlo bayesian optimization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21524–21538.