IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 11, 2019, accepted December 25, 2019, date of publication January 6, 2020, date of current version January 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964294

A Novel Lane-Changing Decision Model for Autonomous Vehicles Based on Deep Autoencoder Network and XGBoost

XINPING GU, YUNPENG HAN, AND JUNFU YU
[1]Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Ministry of Education, Shandong University, Jinan 250061, China
[2]School of Mechanical Engineering, Shandong University, Jinan 250061, China

Corresponding author: Yunpeng Han (hanyp@sdu.edu.cn)

This work was supported by the Key Research and Development Projects of Shandong Province under Grant 2019GGX104100 and Grant 2016ZDJS02A04.

ABSTRACT Lane-changing (LC) is a critical task for autonomous driving, especially in complex dynamic environments. Numerous automatic LC algorithms have been proposed. This topic, however, has not been sufficiently addressed in existing on-road manoeuvre decision methods. Therefore, this paper presents a novel LC decision (LCD) model that gives autonomous vehicles the ability to make human-like decisions. This method combines a deep autoencoder (DAE) network with the XGBoost algorithm. First, a DAE is utilized to build a robust multivariate reconstruction model using time series data from multiple sensors; then, the reconstruction error of the DAE trained with normal data is analysed for LC identification (LCI) and training data extraction. Then, to address the multi-parametric and nonlinear problem of the autonomous LC decision-making process, an XGBoost algorithm with Bayesian parameter optimization is adopted. Meanwhile, to fully train our learning model with large-scale datasets, we proposed an online training strategy that updates the model parameters with data batches. The experimental results illustrate that the DAE-based LCI model is able to accurately identify the LC behaviour of vehicles. Furthermore, with the same input features, the proposed XGBoost-based LCD model achieves better performance than other popular approaches. Moreover, a simulation experiment is performed to verify the effectiveness of the decision model.

INDEX TERMS Autonomous vehicle, lane-changing identification, lane-changing decision-making, deep autoencoder network, XGBoost.

I. INTRODUCTION

Autonomous driving, which is an emerging and rapidly growing field, exhibits enormous potential to improve driving safety and transportation system efficiency and is the future direction for the development of vehicles [1]–[3]. The driving decision-making system is the key technology for ensuring the driving safety of autonomous vehicles (AVs), and the lane changing decision is a significant part of the research in this area [4]–[6]. Lane-changing (LC), which is a complex and potentially dangerous driving behaviour, significantly impacts the traffic capacity and road safety. According to research data from the US National Highway Traffic Safety Administration, traffic accidents caused by lane changing accounted for 27% of all traffic accidents [7]. Recently, human-like decision-making has become a popular topic in autonomous driving systems research. Although the fast development of depth sensors and machine learning methods have given a considerable boost to the self-driving research, making high-level LC decisions that conform to social norms is difficult, especially when vehicles are driving in complex dynamic environments [8], [9]. Therefore, this work mainly focuses on LC analyses.

An intelligent vehicle's monitoring and sensor system captures the parameters (e.g., location, velocity, acceleration, etc.) from its surrounding traffic participants and objects and performs remote or local monitoring. The system can change signals quickly and has numerous operating parameters. The motion characteristics of the vehicle are embedded in the motion variables that represent its operating state. Thus, the vehicle LC identification (LCI) and LC decision (LCD) can be studied by mining the information existing in the

The associate editor coordinating the review of this manuscript and approving it for publication was Ching-Ter Chang.

vehicle's historical motion data. Over the years, researchers have conducted extensive investigations to identify intelligent vehicle LC behaviour; for example, Moridpour *et al*. [10] proposed an LCD model and provided judgement criteria for the start and end points of an LC task. Wang *et al*. [11] studied the characteristics of various LC behaviours and designed a model to assess vehicle driving behaviours through lateral velocity. Additionally, Yang *et al*. [12] derived the mapping relationship between the vehicle driving state and the driving trajectory based on vehicle steering kinematics and then used a support vector machine (SVM) linear classifier to analyse the vehicle body transfer angle and maximum steering angle given by a moving direction vector model. Because deep neural networks (DNNs) have high learning abilities [13]–[16], Wang *et al*. [17] demonstrated that DNNs have the highest level of precision compared to other algorithms and are more suitable for detecting state changes through comparison using support vector regression (SVR) and shallow artificial neural networks (ANNs). Chen *et al*. [18] presented a procedure for selecting the key features and predicting the risks associated with a vehicle's LC behaviour, in which fault tree analysis, a k-means clustering algorithm and a random forest (RF) classifier were employed to model the LC process. Because of the correlations and coupling that occur among an AV's variables, using a deep learning method to mine the embedded information from the variables can significantly improve the LC behaviour identification performance.

The above studies mainly focus on detecting the LC behaviours of vehicles, but do not further assess the vehicle LC decision-making process. The earliest research study on the LC decision-making process, which mainly analysed the benefits, safety, and necessity of LC behaviours, was proposed by Gipps [19] in 1986. Based on Gipps' model, Yang and Koutsopoulos [20] considered the probability of lane changes and introduced random error. Hidas [21] classified LC behaviours into three categories, i.e., free, cooperative, and forced LC behaviours. Monteil *et al*. [22] introduced the state information of multiple preceding vehicles and developed an LCD framework that integrated a full velocity difference model (FVDM) with a general model (minimizing overall braking induced by lane changes, MOBIL). Moreover, Toledo *et al*. [23] used the acceptable gap model to study LC behaviour. Basically, most of the above models are based on the thought process of the driver. The weakness of these models is that it is difficult to accurately capture some potential decision-making modes and factors considered by the driver in the decision-making process. In recent years, with the development of artificial intelligence, many researchers have attempted to improve the accuracy of vehicle LCD models through machine learning and deep learning. Hou *et al*. [24] investigated two ensemble learning methods, i.e., RF and AdaBoost, to develop an LC assistance system. Qiu *et al*. [25] used a segmented discrete method to pre-process the vehicle trajectory measurement data and then established a Bayesian network (BN)-based LCD model. Most recently, Liu *et al*. [26] suggested that combining an SVM and a Bayesian optimization algorithm (BOA) could improve the performance of the decision-making system. Nie *et al*. [27] analysed the influence of a speed advantage and a space advantage on lane changes scenarios and designed a decision-making framework based on intention triggering and feasibility assessment. Xu *et al*. [28] proposed a fusion LCD model based on a gradient boosting decision tree (GBDT) and compared the effects of different features on the decision results. The XGBoost algorithm, which is based on the boosting tree, further improves the loss function, regularization and parallelization processes and exhibits excellent classification performance, which can effectively improve the precision of the LCD model [29]–[31].

At present, a number of LCI and LCD models have been proposed to improve the AVs' LC behaviour identification ability and LC decision-making ability, however, the existing approaches are very preliminary. Therefore, this study seeks to establish a novel LCD model for AVs by introducing deep learning and machine learning methods, thus improving the existing research and addressing the existing problems in LC behaviour identification and LC decision making for AVs. The main contributions of our work are summarized as follows:

1) We introduce the deep autoencoder (DAE) network to capture nonlinear correlations in the multivariate sensor data while providing a robust signal reconstruction. By analysing the trends in reconstruction errors and the situations in which a threshold is exceeded, the driving behaviours of vehicles can be identified. In addition, an adaptive threshold is adopted to improve the reliability of the recognition results and the robustness of the model. To the best of our knowledge, this is the first study to leverage DAE for LC behaviour identification applications in AVs.
2) To address the multi-parameter and nonlinear problems in the autonomous LC decision-making process, we propose a novel LCD model based on XGBoost and apply the Bayesian optimization algorithm to identify the optimal hyperparameters of XGBoost. A data extraction scheme for LCD based on the optimal time window is also proposed. The experimental results illustrate that a better human-like performance is achieved through our approach than through the previous approaches.
3) An online model rolling update strategy is proposed to guarantee the model prediction accuracy. The update strategy integrates the LCI and LCD modules. Specifically, the online model is obtained by updating the identified historical model with new samples collected by the DAE-based LCI algorithm, which enables the XGBoost-based LCD model to be fully trained with large-scale datasets and further improves the performance of the decision systems.

The remainder of this paper is organized as follows. In Section II, the LC behaviour identification and LC decision-making processes of AVs are analysed.

In Section III, a DAE-based LCI model and an XGBoost-based LCD model are established. In Section IV, the parameters of the DAE and XGBoost models are optimized, and the models are trained. In Section V, the simulation results and analyses are presented. In Section VI, this study is concluded.

II. VEHICLE LC ANALYSIS

A. VEHICLE LCI ANALYSIS

The LC process is divided into two phases based on the characteristics of free LC behaviours and the research objective. The first phase is the LC decision-making stage, which is the process that occurs between the driver's intention to change lanes and the implementation of an LC behaviour. The second phase is the LC execution stage, during which the vehicle starts to move towards the target lane continuously and horizontally until the continuous lateral movement of the vehicle ends.

The general purpose of the LC execution control algorithm is to divide the LC trajectory planning problem into the following two modules: longitudinal trajectory planning and lateral trajectory planning [32]. The control algorithm consists of the following five steps: 1) determine the acceptable inter-vehicle gap and time at which the manoeuvre should be performed; 2) determine the longitudinal safety channel corridor; 3) plan the longitudinal trajectory (e.g., x, v_x, a_x); 4) determine the lateral safety corridor; and 5) plan the lateral trajectory (e.g., y, v_y, a_y). A simplified schematic diagram of the intelligent vehicle LC control system is shown in Figure 1. The algorithm is based on the following assumptions:

- A1 The ego vehicle (E) is equipped with a sensor system, that can monitor the motion state and position (e.g., x, v_x, a_x, y, v_y, a_y) of E and the relative positions and motion states (e.g., x_{sj}, v_{sj}, y_{sj}, v_{sj}, road) of the surrounding traffic participants and objects;
- A2 E is equipped with a prediction system that estimates the trajectory of the surrounding traffic participants and objects over time;
- A3 E is equipped with a control system used to track the planned trajectory;
- A4 E is equipped with a decision-making system that provides the required manoeuvring commands.

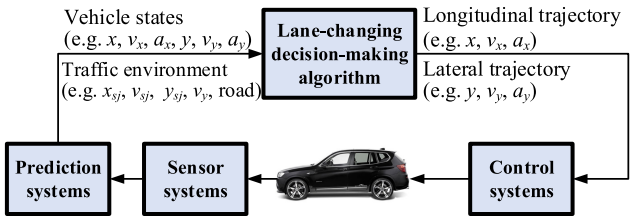

FIGURE 1. Schematic architecture of the LC manoeuvre system for intelligent vehicles.

Based on the LC direction, an LC can be considered one of two types, i.e., a "left-to-right LC" or a "right-to-left LC" [33]. Taking a right-to-left LC as an example, the LC process on a straight road is illustrated in Figure 2. Where E is the ego vehicle, P, R, TP, and TR are the preceding vehicle in the original lane, the rear vehicle in the original lane, the preceding vehicle in the target lane, and the rear vehicle in the target lane, respectively. Regarding the interaction between the car and the road, the distances (d_R and d_L) between the ego vehicle and the left and right lane lines will change according to certain rules during the LC process, and these changes directly reflect the changes in the driving behaviour of the vehicle.

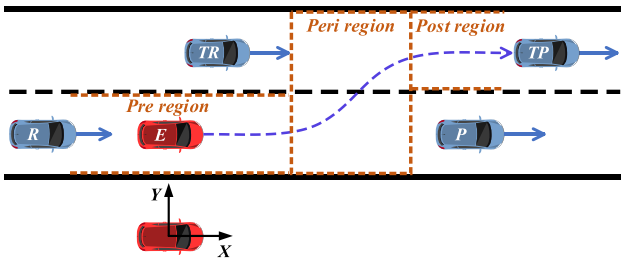

FIGURE 2. Schematic diagram of LC process phase division.

From the perspective of the road structure, most road sections have a certain degree of curvature. Taking a right-to-left LC on a right-turn road scenario as an example, the LC process is shown in Figure 3. Compared with straight road scenarios, curved road sections have road curvatures, so the curvature needs to be measured first. In this study, the following methods are used to measure road curvatures online during vehicle driving, as defined in (1).

$$C = \frac{\omega}{v} = \frac{1}{R} \tag{1}$$

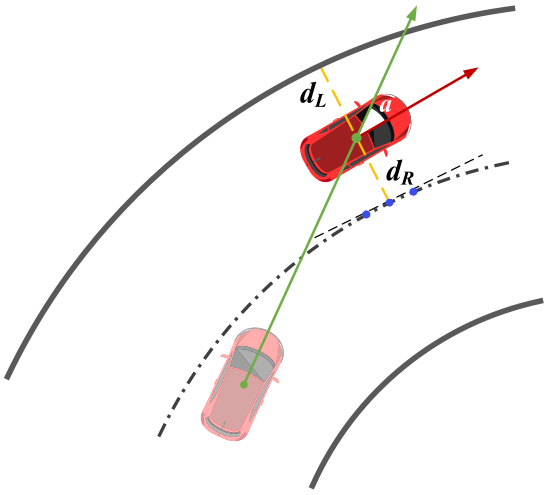

FIGURE 3. Schematic diagram of an LC in a curved road scenario.

where C is the curvature of the road, v is the velocity of the vehicle moving around the circumference, ω is the yaw angular velocity measured by a vehicle-mounted gyroscope, and R is the radius of the road curvature; $v = R\beta/t$, where t

is the time and β is the centre angle of the vehicle passing the arc length at time t.

The motion of the vehicles satisfies the kinematic and dynamic laws. Therefore, the historical motion state of the vehicle can be used to infer its possible future state [12], [18], [34]–[36]. Based on the above analysis of the road structure in straight and curved sections, the distances d_L and d_R between the ego vehicle and the left and right lanes, respectively, vehicle speed v_x, acceleration a_x, lateral velocity v_y, lateral acceleration a_y, yaw angular velocity ω, and yaw angle φ are utilized as the input features in the prediction model to identify the vehicle driving behaviours, determine the start and end time points of an LC and extract the decision-making data.

B. VEHICLE LCD ANALYSIS

The driver's decision-making behaviour in the stage in which the LC intention is generated is a comprehensive cognitive process that integrates the driver's environmental perceptions, analyses, judgements, and decision-making processes. The driver's decision-making behaviour is also a comprehensive reflection of the "driver-vehicle-road" interaction. In this work, the external factors affecting vehicle lane changes under expressway conditions are divided into the following categories: 1) physics-based features, 2) interaction-aware features, and 3) road structure-based features. The physics-based features mainly relate to the motion states (e.g., location, velocity, acceleration, etc.) of the surrounding vehicles. The interaction-aware features indicate how the vehicle's movements are affected by traffic interactions. To avoid colliding with the surrounding vehicles during an LC, the distances between the ego vehicle and the surrounding vehicles must be considered, and the trajectory of the surrounding vehicles must be predicted while the correlation between the vehicles is analysed. The road structure-based features mainly include the road topologies, road signs, and traffic rules.

The work presented in [37] defines the vehicle LC execution process as follows: the ego vehicle takes the adjacent gap in the target lane as the acceptable gap and enters the target gap through a lateral movement. After the characteristics and influencing factors of a vehicle's free LC behaviours on the expressway are analysed, ego vehicle E and the other three vehicles closely related to the LC behaviours, i.e., the current lane preceding vehicle P, target preceding vehicle TP, and target lane rear vehicle TR, are collectively regarded as a driving unit (the four vehicles), as shown in Figure 4.

As it is a microscopic traffic behaviour, vehicle LC is closely related to the surrounding traffic environment. Figure 4 shows that ego vehicle E's LC decision is affected by the surrounding vehicles (i.e., P, TP, and TR), but how these vehicles affect the ego vehicle in making the LCD requires further research. Gipps' LC model considers the following three main factors influencing the LC decision-making process: benefit factors, safety condition and necessity degree.

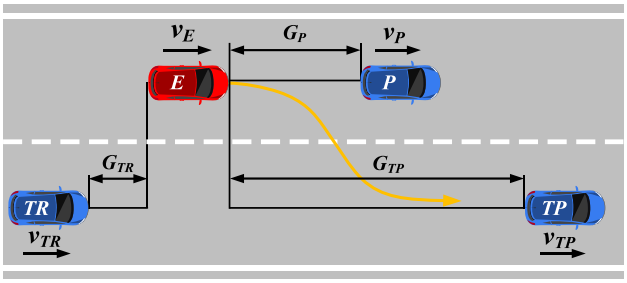

FIGURE 4. Schematic diagram of an LCD scenario. V_E, V_{TR}, V_{TP}, and V_P – the longitudinal velocity of E, TR, TP, and P, respectively; G_{TR}, G_{TP}, G_P – the longitudinal gap distance between E and TR, TP, P, respectively.

Therefore, free lane changes are analysed in this study from the aspects of the benefits, safety, and necessity.

1) LANE CHANGE BENEFIT

In a multi-lane environment, a driver's LCDs are to improve the driving speed or to obtain greater space ahead of the vehicle [38]. When the speed of an adjacent preceding vehicle in the current lane is low and the ego vehicle cannot reach the ideal speed for a long time, an LC motive occurs. Thus, the speed benefit can be expressed as follows:

$$v_{\text{benefit}} = \min(v_{\text{ideal}} - v_P, v_{TP} - v_P) \tag{2}$$

where v_{ideal} represents the optimal driving speed of the vehicle under this condition. The factors directly causing the driver to generate an LC motive mainly come from the driving state and the spatial relationship between the ego vehicle and the surrounding vehicles. When the distance between the ego vehicle E and the preceding vehicle P of the current lane is small, the ego vehicle aims to obtain more space by performing an LC. Thus, the distance benefit can be expressed as follows:

$$G_{\text{benefit}} = G_{TP} - G_P \tag{3}$$

Thus, the driving benefit model can be established as follows:

$$f_{\text{benefit}} = f(v_{\text{benefit}}, G_{\text{benefit}}) \tag{4}$$

2) SAFETY

To avoid a collision with the rear vehicle in the target lane during an LC, the safe distance between the ego vehicle and the rear vehicle in the target lane must be considered in judging the feasibility of an LC. This factor is essentially a component of the driver's spatial perception. Obviously, a larger gap and faster relative speed between E and TR lead to a safer LC process. In addition, lane changes require a minimum safety gap. Thus, the safety model can be established as follows:

$$f_{\text{safety}} = \begin{cases} -\infty, & TR < G_{TR\min} \\ f(G_{TR}, \Delta v_{TR}), & G_{TR} \geq G_{TR\min} \end{cases} \tag{5}$$

where $G_{TR\min}$ is the minimum safety gap between E and TR, which must satisfy $G_{TR\min} > 0$. Δv_{TR} is the relative speed of the ego vehicle E and the target lane rear vehicle TR, and $\Delta v_{TR} = v_E - v_{TR}$.

3) NECESSITY

The above analysis indicates that the decision variables of the internal operation's state transition are the driving factors of an LC, and the feasibility conditions are the constraints of an LC. However, if an LC is performed whenever the two conditions are satisfied, it might cause the AV to frequently change lanes. Therefore, a necessity model must be established. When the distance between E and P is quite large, the ego vehicle will follow the leading vehicle P in the adaptive cruise control (ACC) mode. The ideal following distance is determined by the speed and time headway. Thus, the necessity model can be established as follows:

$$f_{\text{necessity}} = f(G_P - v_E \cdot t_{THW}) \tag{6}$$

where t_{THW} is the time headway and $t_{THW} > 0$. However, LC decision-making is a multi-parameter and nonlinear problem for which a rule-based mathematical model is difficult to establish. Therefore, the LCD model of the vehicle should be expressed as follows:

$$f_{\text{decision}} = f(v_{income}, G_{TP} - G_P, G_{TR}, v_E - v_{TR}, G_P - v_E \cdot t_{THW}) \tag{7}$$

To deal with the abovementioned problems, we propose a novel LCD model for AVs based on a DAE and XGBoost. First, a DAE is utilized to monitor the vehicle's driving behaviours online and to collect new samples in real time. Then, a human-like LCD method based on an optimized XGBoost is proposed to address the multi-parameter and nonlinear problems in the AVs' LC decision-making process. In parallel, the new independent samples are applied to update the XGBoost-based LCD model. Thus, the model has an adaptive online update capability, which further enhances its robustness and generalization. The schematic of the proposed method in this paper is shown in Figure 5.

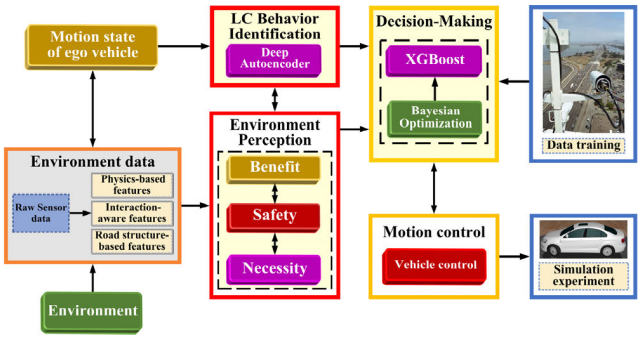

FIGURE 5. Schematic of the proposed method in the paper.

III. REFERENCE MODEL LEARNING

A. DAE ALGORITHM

At present, the emerging machine learning and deep learning methods provide us with new approaches to deal with the LCD problem. However, the analytical or data-driven research on LCD still has limitations, which are probably caused by the difficulties in collecting large-scale LCD data. Therefore, in this section, we present a deep learning-based LCI model to collect training data to enable the LCD model to be fully trained with a large-scale dataset.

The starting point of an LC executed by drivers is usually the starting point of a trajectory change. The most accurate method for determining the starting point is to calibrate the position at which the change occurs according to each trajectory. However, such data processing is cumbersome, and it is difficult to obtain a large amount of LC data for machine learning model training. Therefore, we propose a DAE-based LCI model to identify the LC behaviour of the vehicle and automatically extract the training data in real time. A DAE consists of several restricted Boltzmann machines (RBM) [39], [40]. An RBM is an undirected graph model consisting of a visible layer and a hidden layer. No internal connections exist between the units in the layer, and the layers of the units are externally connected. In this study, a standard RBM with a neuron value of 0 or 1 is used to construct the DAE.

The output of each lower-level RBM in the DAE is used as the input of the higher-level RBM to realize a layer-by-layer transmission of network learning and deep mining of the feature information [41]. Figure 6 shows that the deep autoencoder network includes an encoding process and a decoding process. The encoding process obtains the output value ($\boldsymbol{Y}$) or encoding result, which is abstract and has a characteristic representation ability; the decoding process is the inverse of the encoding process, and the data stream is transferred back to obtain the reconstructed value ($\hat{\boldsymbol{X}}$). The reconstructed value is similar to the vehicle motion state estimate, which has the same physical meaning as the input value.

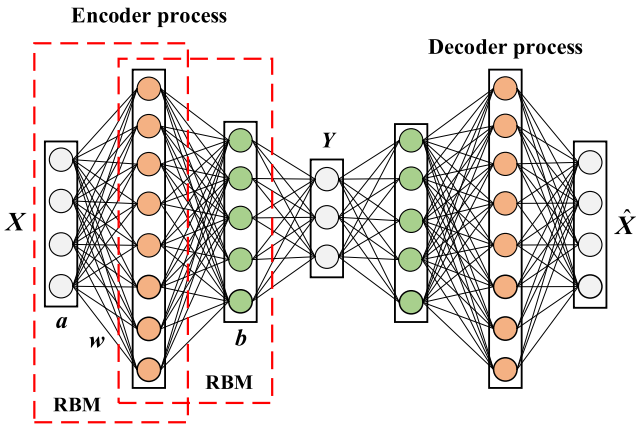

FIGURE 6. The layer-wise learning process of the DAE model.

The DAE model is trained by selecting the system data related to the long-term lane-keeping (LK) state of the vehicle. $\boldsymbol{X}_t$ is the motion state dataset associated with the vehicle at time t, which is generalized as follows:

$$\boldsymbol{X}_t = [\boldsymbol{x}_{t,1}, \boldsymbol{x}_{t,2}, \cdots, \boldsymbol{x}_{t,n}, \cdots, \boldsymbol{x}_{t,m}] \tag{8}$$

where $\boldsymbol{x}_{t,n}$ is the value of the n-th variable in the dataset of the state of the automobile at time t and m is the number of variables in the state dataset.

The reconstructed value ($\hat{\boldsymbol{X}}_t$) of the DAE network output is expressed as follows:

$$\hat{\boldsymbol{X}}_t = [\hat{\boldsymbol{x}}_{t,1}, \hat{\boldsymbol{x}}_{t,2}, \cdots, \hat{\boldsymbol{x}}_{t,n}, \cdots, \hat{\boldsymbol{x}}_{t,m}] \tag{9}$$

where $\hat{x}_{t,n}$ is the reconstructed value of the n-th variable at time t in the motion state dataset of the vehicle.

The reconstruction error (γ_{RE}) of the original input and the reconstructed value at time t are defined as the Euclidean norm of the difference between the respective normalized values, which is expressed as follows:

$$\gamma_{RE} = \left\| \boldsymbol{X}_t - \hat{\boldsymbol{X}}_t \right\| \tag{10}$$

In the normal LK state, a stable relationship exists between the various state variables of the vehicle. When the vehicle starts to change lanes, the correlation between variables is disrupted, and the reconstructed values deviate greatly. As the motion state changes further, an ascending or jittery reconstruction error occurs. Therefore, by analysing the changing trend of the reconstruction error, the changes in the driving behaviour of the vehicle can be detected, and the execution point of the LC can be identified. When the reconstruction error exceeds the alarm threshold, the vehicle begins to enter the LC state from the stable LK state, and this point is the boundary that divides the following two states: LK and LC.

B. XGBoost ALGORITHM

To enable AVs to change lanes in a human-like manner, we propose an LCD model based on XGBoost. XGBoost [42] is a supervised algorithm that consists of multiple base learners superimposed onto strong learners. The proposed LCD model trains the model on dataset $\boldsymbol{D} = \{(\boldsymbol{X}_i, \boldsymbol{y}_i)\}$ containing n samples and m lane change decision variables, where $\boldsymbol{X}_i \in \mathbf{R}^{\boldsymbol{m}}$, $\boldsymbol{y}_i \in \mathbf{R}$, and $\mathbf{R}^{\boldsymbol{m}}$ is an m-dimensional vector space. The vehicle lane change decision ($\hat{\boldsymbol{y}}_i$) is predicted by the trained model. $\boldsymbol{y}_i$ is the target value corresponding to $\boldsymbol{X}_i$, i.e., the decision category of vehicle LC; $\boldsymbol{X}_i$ is the decision data variable when $t = i$ in Equation (8), $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n]$.

The basic learner of XGBoost chooses a classification and regression tree (CART). A single CART is often excessively simple to make effective vehicle LCDs. K CART functions are added together to form an ensemble tree model for predicting the classification target value, and the formula is expressed as follows:

$$\hat{\boldsymbol{y}}_i = \sum_{k=1}^{K} f_k(\boldsymbol{X}_i) \quad f_k \in \Gamma \tag{11}$$

where $\boldsymbol{\Gamma} = \{f_k(\boldsymbol{X}) = \omega_{q(x)}\}$, $(\boldsymbol{\omega} \in \mathbf{R}^T, \boldsymbol{q}: \mathbf{R}^m \rightarrow \{1, 2, \ldots, T\})$ is a function space composed of a classification regression tree. $\boldsymbol{\omega}$ is the leaf weight. T is the number of leaf nodes on the tree. $\boldsymbol{q}$ is the structure of each tree, i.e., the mapping of the sample instances to the corresponding leaf node indexes. Each f_k corresponds to an independent tree structure ($\boldsymbol{q}_k$) and a leaf weight ($\boldsymbol{\omega}_k$).

To learn the set of functions in the model, the learning objective function of XGBoost is expressed as follows:

$$L = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{12}$$

where l is a differentiable convex function, which is used to measure the difference between the predicted classification value ($\hat{\boldsymbol{y}}_i$) and the real category ($\boldsymbol{y}_i$), which is called the loss function; the $\boldsymbol{\Omega}$ function represents the regular term used to punish the complexity of the model and prevent the model from overfitting. Equation (12) cannot be solved by traditional methods. Thus, the greedy method is used to learn f_k and minimize the objective function.

Let $\hat{\boldsymbol{y}}_i^{(t)}$ be the predicted value of the i-th instance at the t-th iteration. Then, we add a new function (f_t) in each iteration, i.e., we adopt the step-by-step forward additive model to minimize the objective function, as follows:

$$\boldsymbol{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_t^{(t-1)} + f_t(\boldsymbol{X}_i)) + \boldsymbol{\Omega}(f_t) \tag{13}$$

Taylor series expansion is performed on the loss function in Equation (13). Then, the constant term is removed and retained to the second-order term. Its expression is written as follows:

$$\tilde{\boldsymbol{L}}^{(t)} \approx \sum_{i=1}^{n} \left(g_i f_t(\boldsymbol{X}_i) + \frac{1}{2} h_i f_t^2(\boldsymbol{X}_i) \right) + \boldsymbol{\Omega}(f_t) \tag{14}$$

where $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ and $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$.

Let $\boldsymbol{I}_j = \{i | q(\boldsymbol{X}_I) = j\}$ be the index number of the instance corresponding to the j- th leaf node where the model complexity can be written as $\boldsymbol{\Omega}(f) = \gamma T + 1/2\lambda \sum_{j=1}^{T} \omega_j^2$. γ and λ are the normalization coefficients. Thus, the model objective function is rewritten as follows:

$$\tilde{\boldsymbol{L}}^{(t)} = \sum_{j=1}^{T} \left[\omega_j \sum_{i \in I_j} g_i + \frac{\omega_j^2}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) \right] + \gamma T \tag{15}$$

For a certain structure ($\boldsymbol{q}(\boldsymbol{X}_i)$), the optimization weight ($\boldsymbol{\omega}_i^*$) of leaf node j and the corresponding optimization objective function values are expressed as follows:

$$\boldsymbol{\omega}_j^* = \frac{-\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{16}$$

$$\tilde{\boldsymbol{L}}^{(t)}(\boldsymbol{q}) = -\frac{1}{2} \sum_{j=1}^{T} \left[\frac{\left(\sum_{i \in I_j} g_i \right)^2}{\left(\sum_{i \in I_j} h_i + \lambda \right)} \right] + \gamma T \tag{17}$$

Equation (17) can be utilized as a score function to measure the quality of the tree structure. Typically, all possible tree structures cannot be fully enumerated. Therefore, a greedy

algorithm that repeatedly adds branches from a leaf node is used. Assuming that $\boldsymbol{I}_L$ and $\boldsymbol{I}_R$ are examples of left and right sub-nodes, respectively, after splitting and $\boldsymbol{I} = \boldsymbol{I}_L \cup \boldsymbol{I}_R$, then the structural loss after splitting is expressed as Equation (18), which can be used to determine whether to split the nodes and the candidate points for splitting the nodes.

$$\boldsymbol{L}_s = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\left(\sum_{i\in I_L} h_i+\lambda\right)} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\left(\sum_{i\in I_R} h_i+\lambda\right)} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\left(\sum_{i\in I} h_i+\lambda\right)}\right] - \gamma \tag{18}$$

One of the key problems in tree learning is finding the optimal splitting condition, as shown in Equation (18). To this end, the split search algorithm enumerates all possible splits on all features. We call the split search algorithm the exact greedy algorithm, and it is computationally demanding to enumerate all the possible splits for continuous features. Thus, to perform this operation effectively, the algorithm must first rank all data according to each eigenvalue and access the data in the order of ranking to accumulate the gradient statistics of the structural scores shown in Equation (18). In many practical situations, input $\boldsymbol{X}_i$ is often sparse. Therefore, to ameliorate the robustness of the model, it is necessary to improve the algorithm, as shown in Algorithm 1.

Algorithm 1 Sparsity-Aware Split Finding [42]

Input: $\boldsymbol{I}$, instance set of current node
Input: $\boldsymbol{I}_\kappa = \{i \in I | x_{ik} \neq \text{missing}\}$
Input: d,feature dimension
Also applies to approximate setting, only collect statistics of non-missing entries into buckets
$gain \leftarrow 0$
$G \leftarrow \sum_{i\in I}, g_i, H \leftarrow \sum_{i\in I} h_i$
for $\kappa = 1$ **to** m **do**
  // enumerate missing value goto right
  $G_L \leftarrow 0, H_L \leftarrow 0$
  for *j in sorted*(I_κ, *ascent order by* x_{jk}) **do**
    $G_L \leftarrow G_L + g_j, H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L, H_R \leftarrow H - H_L$
    score $\leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
  // enumerate missing value goto left
  $G_R \leftarrow 0, H_R \leftarrow 0$
  for *j in sorted*($\boldsymbol{I}_\kappa$, *descent order by* x_{jk}) **do**
    $G_R \leftarrow G_R + g_j, H_R \leftarrow H_R + h_j$
    $G_L \leftarrow G - G_R, H_R \leftarrow H - H_R$
    score $\leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split and default directions with max gain

Sparseness has many possible causes (e.g., due to the complexity and uncertainty of the driving environment, some data may be lost during detection). Therefore, the algorithm must consider the sparse patterns in the data. Thus, we recommend adding a default orientation to each tree node. When a value is missing from a sparse matrix ($\boldsymbol{X}_i$), the sample is classified as the default direction. Each branch has two default direction choices, and the best default direction is learned from the data. The key improvement is to only visit the non-missing entries ($\boldsymbol{I}_{k1}$). The proposed algorithm regards the empty values as missing values and learns the best direction to deal with the missing values.

IV. VEHICLE LCI AND LCD ALGORITHM

A. ALGORITHM FRAMEWORK

The success of machine learning demonstrates that with larger-scale datasets, the network can learn a more complex non-linearity function to project the feature into a higher dimensional space and, thus, obtain a better performance [43]. Vehicles accumulate a large quantity of motion state data during driving, which is of great value to improve the model's performance and stability. Therefore, the vehicle can apply the newly accumulated data for online training, utilize system resources fully and avoid the state detection errors that are caused by the long-term application of the model after one round of training. To train our LCD model with large-scale datasets, we propose an online learning strategy that can update the model parameters by data batches, resulting in a new online LCD algorithm. First, the initial decision-making model is established based on an XGBoost algorithm and offline historical multivariate data. Meanwhile, the DAE-based LCI model can monitor the vehicle driving behaviour online and collect new samples in real time. Then, the online model is obtained by updating the identified historical model with new samples according to the approximate linear dependence (ALD) [44] condition. The proposed integrated decision framework includes the following three steps:

Step 1: Initialization of the LCI and LCD models. In the offline learning phase, an initial LCD model is generated offline based on the historical trajectory data. More specifically, the DAE-based LCI model is trained offline using the long-term LK driving data. The trained DAE model is applied to separate the LK and LC period data from the trajectory data to determine the demarcation points (i.e., the starting and ending points) of an LC. Then, the LC decision-making data are extracted at the optimal time window length ($\boldsymbol{w}$), and the XGBoost classification model is trained with multiple known LC label data.

Step 2: Real-time application of the LCI and LCD models. In this phase, the sensor system collects data in real time. For the new incoming testing data, we first apply the fixed time window with the optimal length used in the training phase to obtain the test vector as the input to the trained XGBoost algorithm. Then, the XGBoost-based LCD model is used to make safe and reasonable LCDs. In parallel, the newly

observed data are input into the trained DAE model to obtain their reconstruction error. The reconstruction errors are then compared with their corresponding control threshold given by the offline modelling stage to determine if LC behaviours have occurred. In addition, a DAE is also used to automatically extract training data during the operation of AVs and update the training datasets.

Step 3: Online updating of the XGBoost-based LCD model. To further enhance the performance of the LCD model, we designed an online training strategy to update the LCD model's parameters with data batches. The update strategy integrates the LCI module and the LCD module, which includes the following three sub-steps: 1) the LCI model monitors the driving behaviour of a vehicle online and collects new sample data in real time; 2) when a batch of new samples arrives, a comparison of the online monitoring results and the approximate linear dependence conditions is performed to determine whether to update the model; 3) when the new sample deviates considerably from the original data, the updated datasets are used to train the XGBoost model, and the historical model is updated. Then, the updated model replaces the previous model. Thus, we extend the XGBoost-based LCD model to an online update decision-making algorithm so that it is able to better address challenging decision tasks.

B. DATA DESCRIPTION

The data employed in this work were obtained by the next generation simulation (NGSIM) research programme initiated by the US Federal Highway Administration (FHWA) for studying microscopic traffic simulations [45]. The NGSIM data provide the motion information (e.g., speed, acceleration, position, etc.) of each vehicle on the surveillance road sections at a 0.1 s interval. It is an ideal dataset for parameter calibration and behaviour analysis of an LC model. Although the data were collected by the US Federal Highway Administration, the vehicle's LC characteristics are generic and are influenced very little by the populations in specific regions because the traffic parameter data were collected from typical environments. Therefore, this dataset has attracted significant attention from scholars and has also been used by researchers in many countries in research studies on LC theory. This study uses the US-101 and I-80 data in the NGSIM dataset as the original data when building the model. The two datasets contain six 15-min acquisition trajectory subsets (recorded as (I), (II),..., (VI)). Since the original data has a certain amount of error and noise [46], the Kalman filter [47] is first used to filter the original data.

C. EVALUATION CRITERION OF LCD

To quantitatively evaluate the rationality of the decision results made by the LCD model, based on the driver's LC decision mechanism, the expected velocity satisfaction, risk perception coefficient, and LC feasibility coefficient are proposed as the evaluation criteria. According to the conclusion of Section II, one of the triggers is the driver's desire for a speed advantage. When the average speed of the traffic in the current lane is lower than that in the target lane, drivers are likely to change lanes [48]. This study quantified a vehicle's expected velocity satisfaction by using the velocity ratio (RV) of the adjacent lanes, which is expressed as follows:

$$RV = \frac{V_{omean}}{V_{dmean}} \tag{19}$$

where V_{omean} represents the average speed of the traffic in the original lane and V_{dmean} represents the average speed of the traffic in the target lane.

The second trigger is the driver's expectation of the driving space advantage. When the preceding vehicle in the original lane decelerates, the distance between the ego vehicle and the preceding vehicle becomes shorter, and a demand for an LC is generated to avoid a collision or to obtain greater space ahead of the ego vehicle. According to the analysis of the driver's LC decision mechanism, we can know that the time headway (t_{THW}) and the inverse of time-to-collision (TTC_i) are highly correlated with the driver's final LC decision. Therefore, the risk perception (RP) is selected to quantify the degree of the risk perception and LC demand under current driving conditions. RP was first proposed by the Nissan Company, which is calculated by linearly combining the inverse of t_{THW} with TTC_i, as shown in Equation (20) [49], where the weighting coefficients a and b are taken as 1 and 5, respectively.

$$RP = \frac{a}{t_{THW}} + b \cdot TTC_i \tag{20}$$

where t_{THW} is the time headway, which is obtained by dividing the relative distance by the vehicle speed. TTC (i.e., time-to-collision) refers to the time that remains until a collision between two vehicles would have occurred, which is obtained by dividing the relative distance by the relative velocity. TTC_i is the inverse of the TTC.

Obviously, the LC motivation is a necessary but inadequate condition for making an LC decision. In terms of the safety factors, the safety of an LC is most closely related to the state of the rear vehicle in the target lane [50]. Therefore, the safety of an LC is quantified by the ratio (RS) of the actual distance and the safe distance between the ego vehicle and the rear vehicle of the target lane, which is expressed as follows:

$$RS = \frac{d}{d_{safeness}} \tag{21}$$

where d is the actual distance between the ego vehicle and the rear vehicle of the target lane and $d_{safeness}$ is the critical safety distance. In our study, the safety distance defined in reference [51] is used as the critical safety distance.

D. PARAMETER SETUP AND MODEL TRAINING

1) TRAINING OF THE DAE-BASED LCI MODEL

The training input of the DAE includes a number of variables that can represent the motion states of vehicles. Based on the analysis in Section II, we select the following eight variables as characterization parameters: the distances d_L and d_R between the ego vehicle and left and right lanes, respectively,

vehicle speed v_E, acceleration a_E, lateral velocity v_y, lateral acceleration a_y, yaw angular velocity ω, and yaw angle φ. The DAE network consists of seven hidden layers, and the number of nodes per layer in the encoding process are 1000, 500, 250, and 50. The collected raw data are filtered using the Kalman filter by considering the smoothness of the road and the various interferences in the external environment [52]. First, training is performed using the NGSIM dataset. After constructing the DAE for the self-driving vehicle, the DAE can be applied to identify the LC behaviour of the vehicle. The initial training and online identification process are presented in Figure 7.

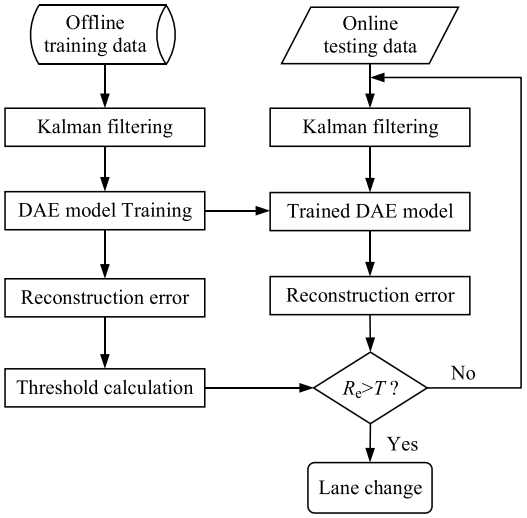

FIGURE 7. Flowchart of the proposed algorithm.

In this study, we first build the DAE model, which captures the nonlinear correlations among multiple sensor variables while providing a robust signal reconstruction. Normally, in the LK status, the reconstruction error (R_e) is very small and remains within a stable range. When an LC behaviour occurs, the reconstruction error jitters or increases dramatically. Therefore, by analysing the trend of the reconstruction error and its situation beyond the threshold, we can monitor the vehicle's motion state to identify an LC behaviour of an AV and to extract the LC samples. However, the vehicle driving process is dynamic and non-stationary, and the reconstruction error of the motion state data is also dynamic.

Thus, setting a fixed threshold may lead to misidentification. Therefore, the adaptive threshold of R_e is introduced as a judgment condition for identifying the LC behaviour.

The parameter confidence interval idea used in statistics is applied to the design of the adaptive threshold. The mean and variance of R_e are expressed as follows:

$$\mu(R_{ej}, t_k) = \frac{1}{n}\sum_{i=1}^{n} r_i(t_k)|_{R_{ej}} \tag{22}$$

$$\sigma^2(R_{ej}, t_k) = \frac{1}{n-1}\sum_{i=1}^{n}\left[r_i(t_k) - \mu(R_{ej}, t_k)\right]^2\Big|_{Rej} \tag{23}$$

where R_{ej} is the corresponding R_e at different times.

The confidence interval with a confidence degree of $(1-\alpha)$ mean can be expressed as follows:

$$P\{\bar{\mu} - z\alpha < \mu < \bar{\mu} + z\alpha\} = 1 - \alpha \tag{24}$$

where α is the confidence level and z is the coefficient associated with the confidence level. In practical applications, the confidence $(1-\alpha)$ is usually chosen as $95\% - 99\%$. In this work, the confidence is 96%, z is 2.06, and the threshold is obtained through Equation (24).

$$J_{\text{th}} = \mu(R_{ej}, t_k) \pm 2.06\sigma^2(R_{ej}, t_k) \tag{25}$$

Thus, the decision criteria for identifying the LC behaviour of AVs are obtained as follows:

$$\begin{cases} J \le J_{\text{th}}, & \text{lane-keeping} \\ J > J_{\text{th}}, & \text{lane-changing} \end{cases} \tag{26}$$

Figure 8(a) and (b) shows the principle of our proposed DAE-based LCI model. In Figure 8(a), the vehicle's reconstruction error R_e begins to fluctuate abnormally near time t_B, and the trend deviates from the original steady state and gradually approaches the upper control limit. After a period of time, R_e crosses the upper limit and remains above the threshold for a period of time, so the LC behaviour can be accurately detected when a fixed threshold is set. However, the LC occurs at time t_B when a fixed threshold (red line in Figure 8(b)) is set, misidentification occurs at time t_A, and the LC behaviour at time t_B cannot be detected. When the adaptive threshold (green line in Figure 8(b)) of R_e is adopted,

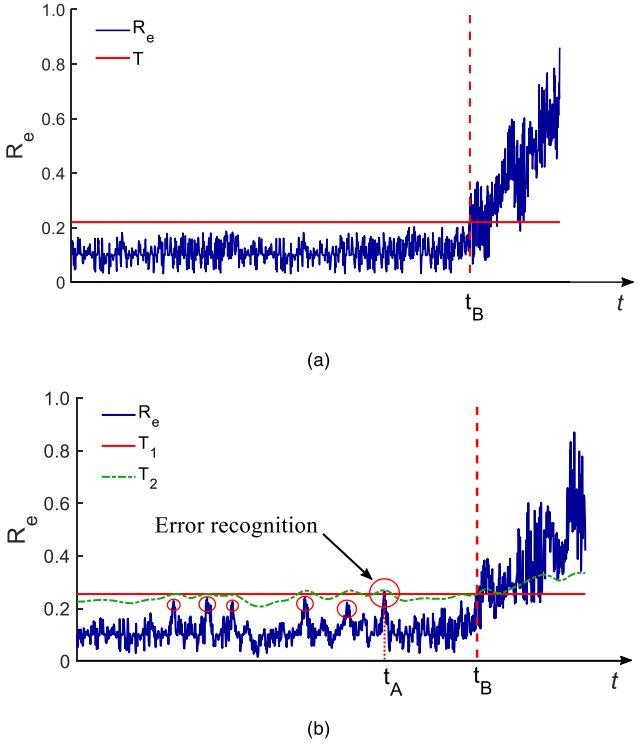

FIGURE 8. Principle of the DAE-based LCI model. (a) Principle of the fixed threshold; (b) principle of the adaptive threshold.

the LC behaviour at time t_B can be detected without misidentification, as shown in Figure 8(b). The results illustrate that the proposed deep learning-based LCI model combined with the adaptive threshold can significantly enhance the robustness of the model.

2) TRAINING OF THE XGBoost-BASED LCD MODEL

In real-world traffic, many factors affect the driver's LC behaviour. In addition to the microscopic running states of the subject and the adjacent vehicles, a driver with an LC motivation needs to immediately make the optimal LC decision according to the other related factors [53]. In this study, we selected 17 candidate features based on the experience [20], [29] of researchers in feature selection, as shown in Table 1. The vehicle motion parameters are directly transformed into vectors as the model input, which might result in data redundancy. Therefore, the dimensions of the training samples must be minimized to improve the training efficiency without losing important data. In this section, the statistical descriptions (e.g., correlation coefficient, feature importance score, etc.) are employed as evaluation indicators to select the key features from the candidate features.

TABLE 1. Candidate features of the XGBoost-based LCD model.

Features	Description
f_1	Speed of the ego vehicle E
f_2	Acceleration of the ego vehicle E
f_3	Distance between E and P
f_4	Speed difference between E and P
f_5	Acceleration difference between E and P
f_6	Collision time of E and P
f_7	Distance between E and R
f_8	Speed difference between E and R
f_9	Acceleration difference between E and R
f_{10}	Distance between E and TP
f_{11}	Speed difference between E and TP
f_{12}	Acceleration difference between E and TP
f_{13}	Collision time of E and TP
f_{14}	Distance between E and TP
f_{15}	Speed difference between E and TR
f_{16}	Acceleration difference between E and TR
f_{17}	Collision time of E and TR

To determine whether the information among the selected features is redundant, the Spearman correlation coefficient [54] is utilized to analyse the correlation between every two features. Spearman's correlation coefficient is a statistical measure of the strength of a monotonic relationship between paired data. The correlation matrix between the features is shown in Figure 9, where the values are correlation coefficients (c). The values of c range from -1 to 1. The closer c is to ± 1, the stronger the linear correlation between the two random variables fi and fj. If $|c| > 0.59$, it indicates that the two features have a strong correlation. When two variables have extremely high correlation magnitudes, indicating that they contain similar information, and the redundant variables need to be removed by correlation filtering. In our approach, if a pair of variables has high correlations ($|c| > 0.59$), one of the two will be removed to speed up the training process of the model.

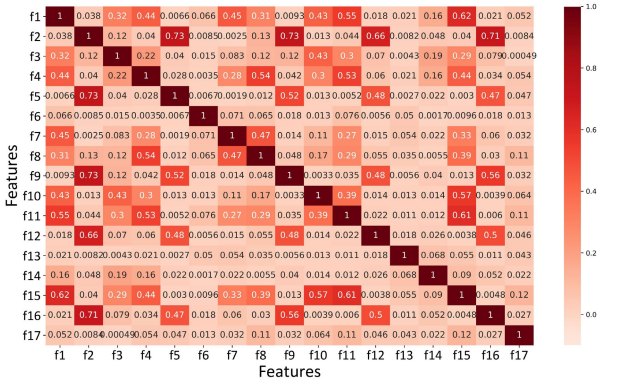

FIGURE 9. Correlation matrix between features.

First, the classification model considers all the candidate features to establish a classifier. During the training process, the LK and LC states of the driving behaviour are expressed as "1" and "2", respectively. Then, the Bayesian optimization algorithm [55] and k-fold cross-validation [56] with k equal to 5 are conducted to determine the optimal parameters of the XGBoost classifier. The main parameters of the model are as follows: the number of decision trees (DTs) is 200, the learning rate is 0.1, the maximum depth of the DT is 4, the weighted sum of the minimum leaf node samples is 0.5, the random and feature samplings are 0.8, and the penalty coefficients of L1 and L2 are 0.1 and 0.5, respectively.

The evaluation criteria of the model parameters are used with the logarithmic loss function [57] to evaluate the probability output of the classifier. The smaller the loss function is, the better the corresponding parameters are. The target parameters of XGBoost are multi-classified, and the logarithmic loss is as follows:

$$L_{\log}(\boldsymbol{Y}, \boldsymbol{P}(\boldsymbol{Y}|\boldsymbol{X})) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log p_{ij} \tag{27}$$

where N is the number of samples and M is the number of categories, which are the two states of the vehicle mentioned above. y_{ij} indicates the i-th sample, which is 1 when it belongs to category j; otherwise, it is 0. p_{ij} indicates the probability of the i-th sample being classified in category j.

The complexity of the XGBoost classifier increases after training. The 120th DT in the XGBoost model is taken as an example, as shown in Figure 10. In this DT, variables f_{14} and f_4 are split many times. The training model is based

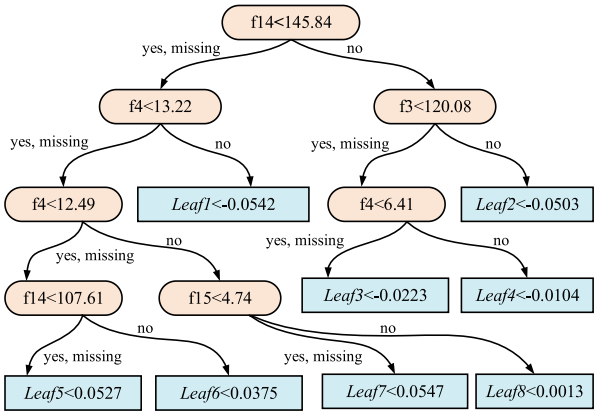

FIGURE 10. The 120th decision tree.

on the two lane-changing decisions mentioned above. The change in the velocity difference f_4 between the ego vehicle and the preceding vehicle of the current lane affects the changes in the other parameters. Therefore, f_{14}, f_4, and f_3 are split several times. The XGBoost model gives the relative importance scores of the input features, as shown in Figure 11.

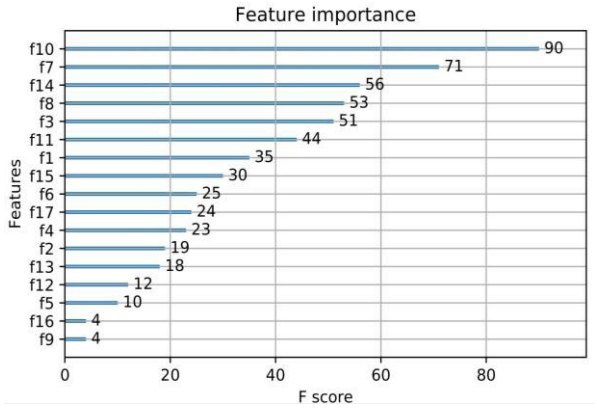

FIGURE 11. Feature importance scores.

The higher the importance score of a variable is, the larger its effect on the classification results. Therefore, f_{10} is indispensable and should be emphasized. By comparison, f_9 and f_{16} are not necessary features in the model. In addition to the correlation coefficient and importance score, the impact on model performance is also a vital factor in the feature selection. To further enhance the generalization ability of the decision model, a feature selection experiment was conducted. According to the relative importance score of the features in Figure 11, the features are added one by one from high to low and the corresponding prediction accuracy is calculated. The process of feature selection is shown in Figure 12.

As is observed, when the number of features reaches 11, the prediction performance tends to be stable, and the accuracy can reach 97%. Compared with the LCD model established

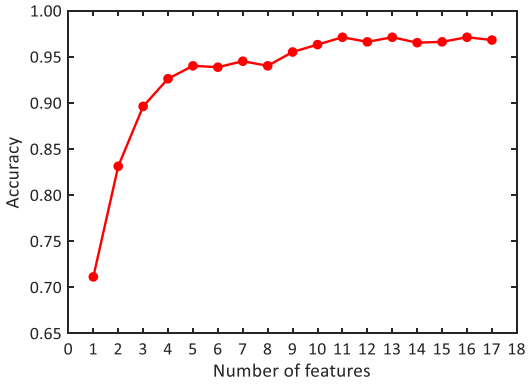

FIGURE 12. Feature selection process of the XGBoost-based LCD model.

by all 17 features, the prediction accuracy is slightly reduced, but the complexity of the model is significantly reduced after feature selection, and the generalization ability and efficiency of the model are also improved. Based on the above analysis, the key features are selected according to the feature correlation coefficients, feature importance scores and prediction performance to remove redundant variables. Finally, 11 features, i.e., f_1, f_3, f_4, f_6, f_7, f_8, f_{10}, f_{11}, f_{13}, f_{14}, and f_{17}, are selected as input variables for the XGBoost-based LCD model. Taking advantage of the powerful data feature extraction capabilities of the machine learning method, the information existing in decision data is fully mined. Then, a socially compliant decision, i.e., LK or LC, is made. The binary decision expression is written as follows:

$$Dec = \begin{cases} 1, & \text{lane-keeping} \\ 2, & \text{lane-changing} \end{cases} \tag{28}$$

V. MODEL EVALUATION

A. EVALUATION OF THE DAE-BASED LCI MODEL

For a randomly selected vehicle changing lanes in the DAE model test data, the identified driving behaviour is reflected in the trajectory diagram, as shown in Figure 13. To verify the performance of the DAE model, 506 groups of LC points identified by the DAE algorithm were compared with the manual extraction results, and the time error distribution is shown in Figure 14. Most of the data for the LC execution

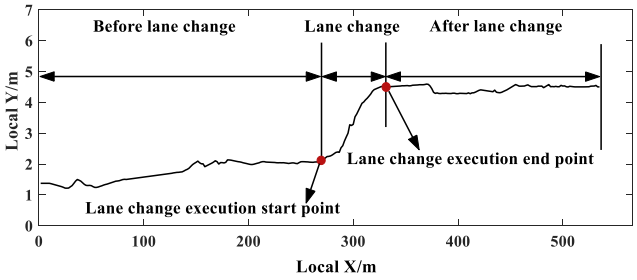

FIGURE 13. Trajectory of lane-changing vehicles.

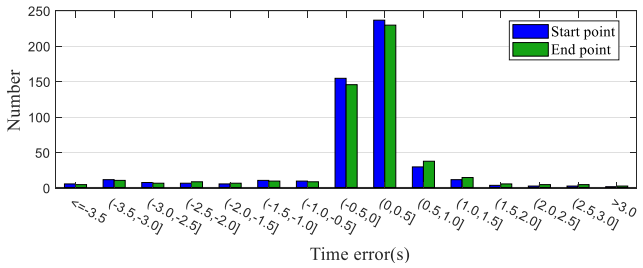

FIGURE 14. Time error distribution.

point extracted by the DAE model are concentrated near the ordinate origin. This result demonstrates that our proposed extraction approach is close to the manual pick method. Therefore, this method can be applied to identify the vehicle LC behaviour, determine the execution point of an LC, and extract training data for online updating of the XGBoost-based LCD model.

The LC decision data used in this study are pre-processed and filtered from the US-101 and I-80 segments of the NGSIM dataset, which included 540 LC data points and 845 counter-examples, representing a total of 1385 sample data points. Then, 80% of the cases are selected randomly from the sample data as training samples for the decision-making model, and the remaining 20% are used as testing samples.

In an actual driving environment, most of the vehicles remain in the original lane when there is no need to change lanes. Therefore, the number of LK vehicles in the dataset is more than that of LC vehicles, which is in accordance with the actual data distribution. There is no serious sample imbalance in the two parts of the data set, and the proportion of positive and negative samples is close to 1:2. In [58], researchers studied the impact of classification errors caused by sample imbalances of different proportions on multiple datasets. The results showed that the classification errors were very close when the positive and negative sample proportions were 1:1 and 1:5 in different data sets. It can be concluded that smaller sample imbalances do not lead to significant changes in the classification results. In addition, the study also compares the classification error when using the over-sampling and under-sampling methods to sample the data at different scales. The results show that the optimal classification effect can be obtained when the proportion of positive and negative samples is different from 1:1. Based on the above analysis, this study chooses to retain the distribution proportion of the original data, and not balance the samples.

B. EVALUATION OF THE XGBoost-BASED LCD MODEL

In this study, to improve the recognition accuracy and efficiency of the proposed LCD model, we use a fixed time window with an optimal length to segment the data. Because more traffic conflicts occur after the vehicle crosses the lane, identifying the LC state before the vehicle crosses the lane provides a warning to eliminate conflicts and prevent a dangerous LC. In addition, the intention recognition module also needs to identify the LC state of the vehicle as early as possible. If the distance from the ego vehicle to the lane is equal to 0, the vehicle has crossed the lane. Therefore, discerning the LC state after vehicles have crossed the lanes using the identification model is insignificant [59]. In summary, the time window should be less than the average time required for the vehicle to cross the lane.

XGBoost provides an intuitive and effective method to determine the complex nonlinear mapping relationship between the LC state of AVs and the motion parameters. The model can detect the LC status of AVs in a given time window based on the data characteristics. According to the above analysis of the LC decision-making mechanism, this study introduces the parameter selection method reported in reference [28] and divides the decision-making variables representing the vehicle motion state, road environment and states of the surrounding vehicles into the following three types: benefit factors (BF), safety condition (SC) and necessity degree (ND). To further explore the influence of different variables on the free LC decision, an experiment comparing different input combinations was conducted. The different input combinations are shown in Table 2. Finally, the prediction accuracy of the XGBoost model under different input combinations was evaluated to determine the optimal input combination.

TABLE 2. Combinations of input parameters.

Number	Combinations of input parameters
1	Benefit factors (BF)
2	Benefit factors (BF), Safety condition (SC)
3	Benefit factors (BF), Necessity degree (ND)
4	Safety condition (SC), Necessity degree (ND)
5	Benefit factors (BF), Safety condition (SC), Necessity degree (ND)

Based on the LC events in 1385 samples, the average time required for an LC is 7.42 s, and the average time required for lane crossing is 2.03 s. To improve the real-time performance and the accuracy of the decision-making model, the detection time should be as short as possible. Therefore, the upper limit of the time window is determined to be 2.0 s. In this work, we calculated the accuracy of the XGBoost models with different input parameters for time windows of 0.6, 0.8, 1.2, 1.4, 1.6, 1.8 and 2.0 s. The results are shown in Figure 15 and Table 3.

As seen from Table 3, when the combinations of the input parameters are (BF, SC, ND) and (BF, SC), the recognition model achieves a high recognition accuracy. The reason is that the relevant vehicles in the traffic flow form an interdependent whole, and their respective behaviours influence the decisions of the others. The decision module receives not only the state information of the ego vehicle but also the

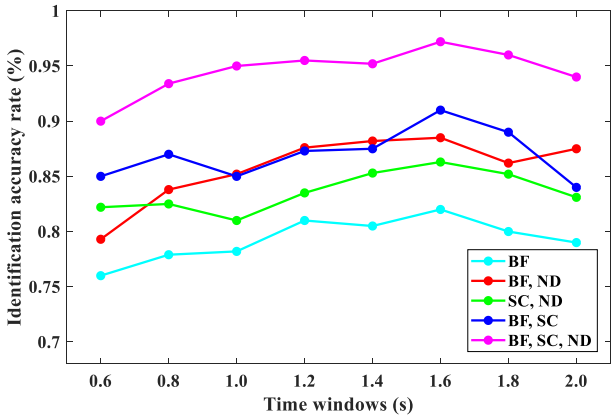

FIGURE 15. Accuracies of different input parameters and different time windows.

TABLE 3. Accuracies of different input parameters and different time windows.

Input parameter	0.6s	0.8s	1.0s	1.2s	1.4s	1.6s	1.8s	2.0s
BF	76%	78%	78%	81%	80%	82%	80%	79%
BF, SC	79%	84%	85%	88%	88%	89%	86%	88%
BF, ND	82%	83%	81%	84%	85%	86%	85%	83%
SC, ND	85%	87%	85%	87%	88%	91%	89%	84%
BF, SC, ND	90%	93%	95%	96%	95%	97%	96%	94%

state information of the surrounding vehicles, so the model can learn the correlation of multi-vehicle behaviour from the interactive information between vehicles and make a reasonable decision. Therefore, considering the effect of vehicle interactions on vehicle motions, the proposed model uses BF, SC and ND as the final combination of the input parameters.

Figure 15 demonstrates that the length of the time window significantly effects the classification performance. When the length of the time window is shorter than 1.6 s, the decision accuracy increases with an increase in the length of the time window. When the window length is longer than 1.6 s, the recognition accuracy gradually decreases. Although the decision accuracy is still very high for the time windows of 1.8 and 2.0 s, the application range of the proposed model becomes increasingly narrow as the time window increases. Therefore, synthetically considering the relationships among promptness, validity, and application range, we determine that 1.6 s is the optimum time window. As shown in Table 3, with 1.6 s as the time window and BF, SC, and ND as the input parameters, the recognition accuracy can reach 97%, which meets the accuracy requirements.

When XGBoost is trained with the sample data, the hidden state is calibrated for each moment. The starting point of an LC is defined as the time when the vehicle begins to approach the target lane continuously, and the end point is defined as the time when the vehicle first crosses the centre line of the target lane. The process from the starting point to the end point is defined as the LC state; the remaining time segments are marked as the LK state. The parameters of XGBoost are obtained after training. The performance of the algorithm is validated by the extracted LC data, and the results are shown in Table 4. The four performance evaluation metrics are as follows: correct recognition time, which is the number of times that the model decision results are consistent with the actual situation; error recognition time, which is the number of times an LC decision is made during the LK state; the recognition time interval, which is the time from the starting point identified by the LCD model to the actual starting point; and the recognition position interval, which is the distance between the starting point identified by the LCD model to the actual starting point. A positive value for the time interval indicates that the identified starting point occurs before the actual LC starting point, and a negative value indicates the reverse. A positive value for the position interval indicates that the vehicle has not crossed the actual LC starting point.

TABLE 4. Accuracy and recognition efficiency of different input parameters.

Input parameter	Correct recognition times	Error recognition times	Recognition time interval /s	Recognition position interval /m
BF	413	93	-1.56	-1.31
BF, SC	457	49	1.02	0.75
BF, SC, ND	492	14	0.42	0.32

Table 4 shows that the algorithm using only BF as the observation variable can recognize the intention to change lanes, but it produces multiple false identifications and has low recognition accuracy. The accuracy and recognition precision of introducing SC as the second observation variable have been improved to a certain extent; the introduction of ND as the third observed variable not only achieved higher accuracy but also significantly improved the recognition precision. When the algorithm without the ND input recognizes an LC intention, the average position interval is 0.75 m. When the input parameters are added with ND, the average position interval is reduced to 0.32 m, and the recognition accuracy is further improved, which is beneficial to the driving decision system for making accurate control decisions. In summary, the comparison between the human driver's LC starting point and the autonomous driving system's LC starting point can demonstrate that the LC decision-making process is similar to that of a human driver [60].

C. PERFORMANCE EVALUATION AND COMPARISON OF THE INTEGRATION MODEL

In this study, a deep learning-based LCI model is presented that can accurately identify the LC behaviour of AVs and address the difficulties of analysing large-scale collections of LCD data. Then, a novel data-driven LCD model is purposed to give AVs the ability to make human-like high-level decisions. It should be noted that both the

LCD model and the LCI model are simultaneously presented in a modelling framework. Thus, it is an integrated model. Obviously, the performance of the LCI model directly affects the quality of the LCD model. To analyse the comprehensive performance of the proposed integrated model, we applied the DAE-based LCI model to extract LC decision data and then tested the XGBoost-based LCD model. In our approach, the length of the time window w is an important parameter affecting the performance of the LCD model. Therefore, based on the above analysis, we trained and tested different decision algorithms under the following time windows: 0.6, 0.8, 1.2, 1.4, 1.6, 1.8 and 2.0 s. The results are shown in Figure 16.

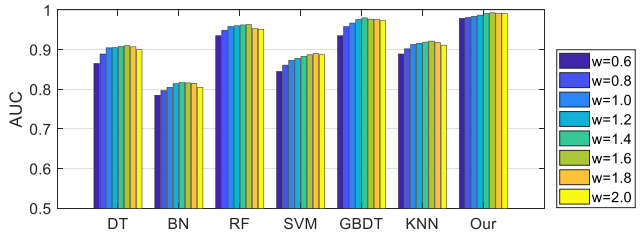

FIGURE 16. AUC at different time points.

As shown in Figure 16, for the different algorithms, the trend of the area under curve (AUC) value changing with the time window length is basically the same. The advantage of a larger window is that it produces a larger AUC in all datasets, which reduces the number of erroneous decisions, and the prediction is, therefore, more stable. When the length of the time window reaches 1.6 s, the AUC basically reaches its maximum value. With an increase in the length of the time window, the AUC begins to decline slowly. The experimental results are the basis of the time window selection in this study.

Finally, we take BF, SC and ND as the input parameters, and the length of time window is 1.6 s. To verify the effectiveness of our proposed XGBoost-based LCD approach, we compared it with other state-of-the-art approaches, which include a DT-based approach, a BN-based approach, a RF-based approach, an SVM-based approach, a GBDT-based approach, and a k-nearest neighbour (KNN)-based approach, on the six subsets {1,2,3,4,5,6} of the NGSIM dataset. The results are shown in Figure 17. Figure 17 demonstrates that when the time window length is 1.6 s, all decision algorithms have stable performances and can obtain large AUC values.

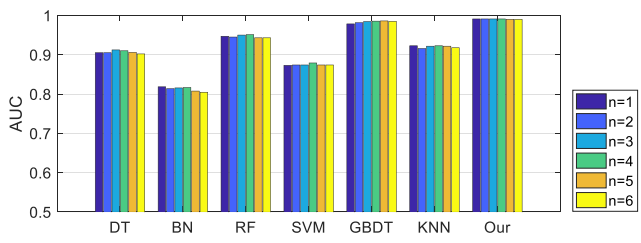

FIGURE 17. AUC of different data subsets.

In addition, it also illustrates that the DAE-based LCI model can accurately identify changes in vehicle driving behaviour and effectively address the problem associated with training data extraction with the XGBoost-based LCD model to realize online training and parameter updating of the model and further improve the performance of the decision system.

The receiver operating characteristic (ROC) curve for performance evaluation of the two datasets, the NGSIM US-101 and NGSIM I-80 datasets, using different methods are shown in Figure 18, where the numbers represent the AUC values for each method. Figure 18 shows that the performance of the RF and GBDT algorithms through ensembled training of the decision tree is much higher than that of the single decision tree represented by the DT and is obviously better than those of the SVM, KNN and other distance discriminant algorithms. With a boosting tree, the XGBoost algorithm has made further improvements in loss function, regularization and parallel processing and has better classification performance than the other algorithms. Specifically, the XGBoost model obtains a larger AUC value, which is close to 1. In other words, the performance of our XGBoost- based model is significantly better than that of other approaches. Among all approaches, BN has the worst performance due to its limited capabilities. In summary, this result further demonstrates the effectiveness and applicability of our approach for the LC decision-making of AVs.

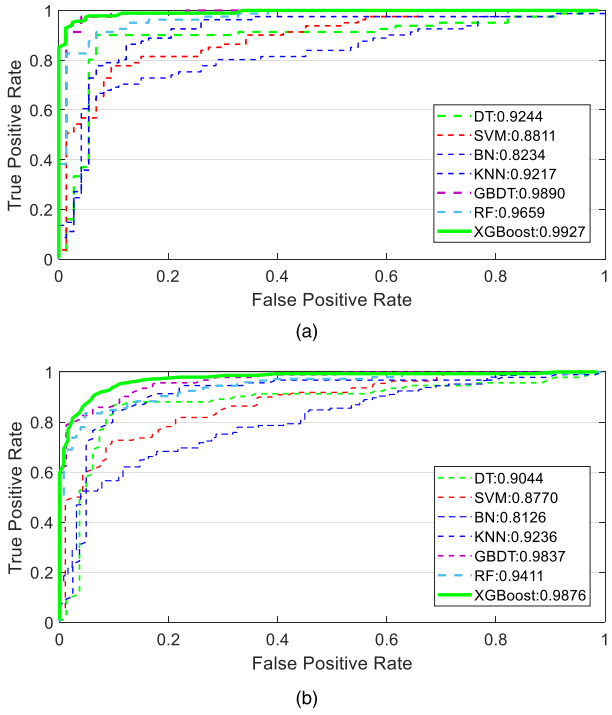

FIGURE 18. Performance comparison of the different approaches. ROC curve for each model with the NGSIM US-101 dataset (a) and the NGSIM I-80 dataset (b).

In this study, we apply the evaluation metrics of precision (PRE), recall (REC), F1 score (F1), and accuracy (ACC) to comprehensively measure the prediction (i.e., test)

performance of different models. The functions of the different evaluation metrics are presented as follows:

$$PRE = \frac{TP}{TP + FP} \tag{29}$$

$$REC = \frac{TP}{TP + FN} \tag{30}$$

$$F1 = \frac{2(PRE \cdot REC)}{PRE + REC} \tag{31}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{32}$$

where TP, FP, TN and FN refer to true positive, false positive, true negative and false negative, respectively. The average metrics of each model are presented in Table 5.

TABLE 5. Performance comparison of the different approaches.

Models	Test set				Training time (s)
	Precision	Recall	F1 score	Accuracy	
DT	91.60%	91.70%	91.56%	91.56%	0.61
SVM	82.50%	81.48%	81.99%	81.17%	2.52
BN	80.71%	79.77%	79.13%	79.22%	0.73
KNN	87.97%	87.50%	87.44%	87.50%	3.12
GBDT	95.66%	95.45%	95.46%	95.45%	2.52
RF	92.71%	92.21%	92.21%	92.21%	1.54
XGBoost	97.07%	97.16%	97.11%	97.12%	1.71

Table 5 illustrates that XGBoost has better classification accuracy than the other classification algorithms, with an average increase in accuracy by 9.27%. The training times of the DT, BN and RF algorithms were less than that of XGBoost. Because the DT and BN algorithms have low complexity, training their parameters takes a very little time, but both methods have unsatisfactory accuracy. The training speed of the RF algorithm is faster than that of XGBoost, but if we pursue reliable decision accuracy, the GBDT and RF algorithms are both DT-based algorithms, and their accuracy is slightly lower than that of XGBoost. Among all the approaches, the XGBoost retains the highest recall and F1 score (97.16% and 97.11%, respectively), which indicates that the proposed method achieves a more human-like performance than the previous methods.

D. SIMULATION EXPERIMENT OF THE XGBoost-BASED LCD MODEL

To further verify the effectiveness of the XGBoost-based LCD model in traffic scenarios, we performed simulation experiments with the model. We used Airsim as the simulation platform and used the TensorFlow framework to build the model based on the Python programming language. The simulation platform follows a modular design. The core components include an environment model, a vehicle model, a physical engine, a sensor model, a common API layer and a vehicle firmware interface layer. The environment system architecture for the XGBoost model test is shown in Figure 19. The test environment is a four-lane expressway with a lane width of 3.5 m. The test vehicle used in the experiment is a car. During the experiment, the test vehicle can obtain data such as the motion state and position of the vehicle in the current lane and the adjacent lane in real time through the Airsim platform, and the sectional data are shown in Table 6. The data includes the time, velocity and acceleration of the ego vehicle, the relative velocity between the ego vehicle and the preceding vehicle, the relative distance between the ego vehicle and the preceding vehicle, the distance between the ego vehicle and the left lane, and the distance between the ego vehicle the right lane.

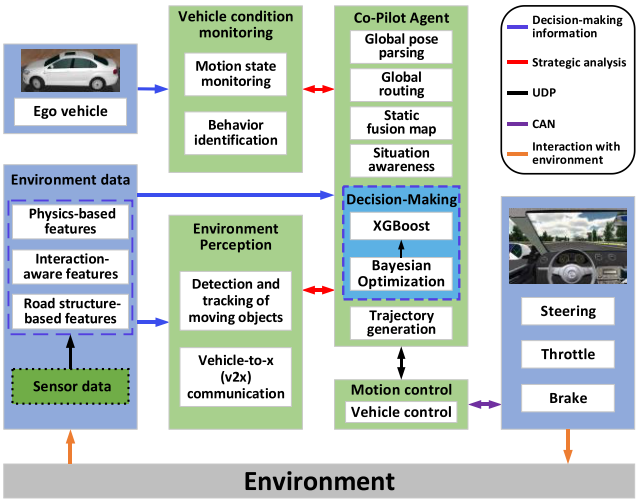

FIGURE 19. Architecture of the system that depicts the core components and their interactions.

During the experiment, the ego vehicle E travels on a straight road at a speed of 27.45 km/h, and the ideal speed is set to be $v_{\text{ideal}} = 80\ km/h$. The preceding vehicle P travels at 12.97 m ahead of E at a speed of 19.72 km/h. The experimental results are shown in Figures 20, 21 and 22. At the beginning, the speed of the ego vehicle was close to that of the preceding vehicle. At 27.1 s, the test vehicle began to accelerate, and the relative distance between the two vehicles continued to decrease. At 29.7 s, the test vehicle

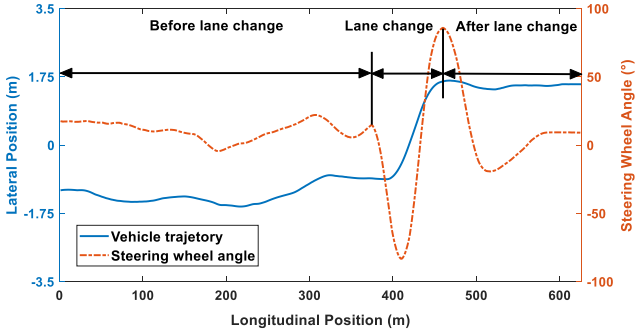

FIGURE 20. Steering wheel angle and vehicle trajectory.

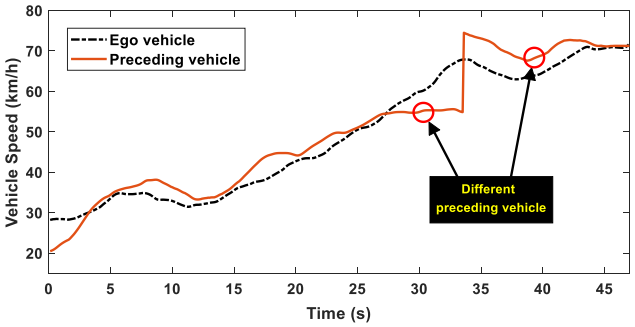

FIGURE 21. Vehicle speed changes during LC.

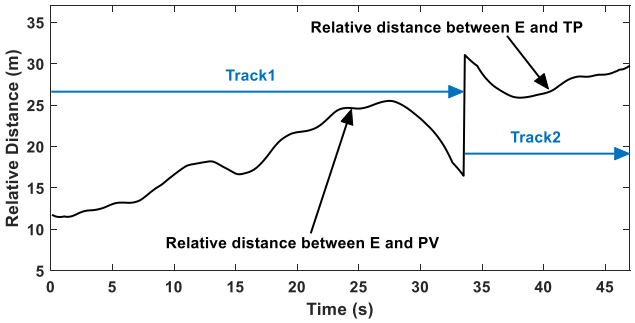

FIGURE 22. Relative distance.

speed increased to 58.07 km/h, the relative distance between the two vehicles decreased to 23.75 m, and the speed of the preceding vehicle was 54.71 km/h. At this time, the XGBoost-based LCD model made an LC decision. Then, the ego vehicle successfully changed lanes by adjusting the lateral speed (Figure 20), and the vehicle speed gradually increased to approach the predetermined ideal vehicle speed (Figure 21), reaching 72.54 km/h. Then, the leading vehicle became the preceding vehicle in the target lane *TP*, and the relative distance from the leading vehicle increased to 29.78 m, as shown in Figure 22. After the vehicle changed lanes, the speed of the test vehicle increased by 24.92%, and the relative distance from the preceding vehicle increased by 25.39%. The test vehicle LC decision-making process was consistent with the analysis of the LC decision mechanism described in Section II, which further verifies the correctness and validity of the proposed approach.

The LC behaviours of the 506 groups extracted manually are calculated and analysed. The results of analysing the real vehicle test data indicate that the quantitative indicators have different correlations with the LC decision. Table 7 shows the correlations between two spatial perception parameters and the decisions determined by point-biserial correlation analysis. As shown in Table 7, the correlation coefficient of *RP* is slightly higher than that of *RS,* which was determined by a quantitative method of assessing the spatial perception coefficient of the current lane. It is more appropriate to select *RP* as the risk perception parameter of the current lane. For the spatial perception of the target lane, the correlation coefficient of *RS* is significantly higher than that of *RP*, so *RS* is chosen as the parameter indicating the LC feasibility.

TABLE 6. Sectional data obtained by the system.

Time (s)	Distance to left lane (cm)	Distance to right lane (cm)	Speed (km/h)	Relative distance (m)	Relative velocity (m/s)	Acceleration (m/s)
0.1	-96	84	61.68	21.39	1.78	-1.02
0.2	-87	93	61.54	21.19	1.74	-0.02
0.3	-81	99	61.82	20.99	1.82	1.64
0.4	-76	104	62.61	20.78	2.04	2.88
0.5	-68	112	63.69	20.57	2.34	3.24
0.6	-63	117	64.70	20.34	2.62	2.61
0.7	-56	124	65.31	20.10	2.77	0.94
0.8	-48	132	65.35	19.84	2.77	-0.76
0.9	-45	135	65.07	19.57	2.68	-1.26
1.0	-41	139	64.84	19.30	2.61	-0.20
1.1	-39	141	65.04	19.00	2.65	1.37
1.2	-35	145	65.75	18.70	2.84	2.66
1.3	-31	149	66.93	18.39	3.16	3.41

TABLE 7. Comparison of the correlation between RP and RS on decision making.

Correlation coefficient	*RP*	*RS*
Current Lane Space Perception	0.540	-0.403
Target Lane Space Perception	0.129	0.497

The evaluation results of the test vehicle LC decision are shown in Table 8. It can be seen from Table 8 that the LC benefit evaluation metrics $RV = 0.688$, which shows that the average speed of the target lane is higher than the average speed of the current lane, and a larger speed advantage can be obtained after changing lanes. In addition, the value of the risk perception factor *RP* is larger, and the system perceives that the risk of LK in the current lane is higher, and the necessity for LC increases. The safety factor $RS = 2.738$. At this time, the actual distance between the ego vehicle and the rear vehicle of the target lane is much larger than the critical safety distance required for an LC, so it is safe to perform an LC. Therefore, the system decides to change lanes. From the analysis of the quantitative metrics of an LC decision, it can be seen that the process is in line with the driver's judgement of the potential benefits, safety and necessity of an LC in a real-world traffic situation. The test results show that the XGBoost-based LCD model is capable of changing lanes in a human-like manner.

TABLE 8. Evaluation results for the test vehicle lane change decision.

Evaluating indicator	*RV*	*RS*	*RP*
Lane change	0.688	2.738	0.571

VI. CONCLUSION

The LC decision-making system is significant for improving the safety and comfort of AVs. In this study, we used a DAE to identify the LC behaviour from actual on-road driving data and trained the XGBoost model by using the extracted data to establish a novel LCD model. First, a DAE-based method is proposed to enable efficient real-time LC behaviour identification and online model updating. The proposed approach builds the reference model offline using multivariate normal data and then identifies the driving behaviours online by comparing the monitoring indicators derived from the reconstruction errors. Meanwhile, an adaptive threshold is adopted to improve the robustness of the model. Then, the XGBoost algorithm and BOA are employed to establish a data-driven LCD method for AVs. By learning an initial knowledge base, the XGBoost-based LCD model can be applied in real time. In particular, our proposed LC framework integrates LCI and LCD modules, which enables the XGBoost-based LCD model to be fully trained with new samples collected by the DAE-based LCI algorithm and further improves the performance of the decision systems. Based on 1385 groups of LC events recorded during the naturalistic on-road experiment, the optimal time window length and input parameters of the LCD model are determined. Finally, both a benchmark dataset and a simulation experiment are used to evaluate our proposed approach. The experiments and simulations illustrate the outstanding performance and practicability of our approach.

At present, due to the complexity and uncertainty of real-world traffic, the proposed model is only applicable to the conventional LC decision-making process in straight lanes or curved lanes on expressways. In our future work, we will further extend the application scenarios of the model to make it suitable for more complex traffic environments. Additionally, we will also consider applying other pattern recognition methods to establish an LC model that can further improve the prediction accuracy and effectiveness.

REFERENCES

[1] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.

[2] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intell. Transport. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Jul. 2017.

[3] S. Shladover, "Cooperative (rather than autonomous) vehicle-highway automation systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 1, no. 1, pp. 10–19, Jul. 2009.

[4] C. Hatipoglu, U. Ozguner, and K. Redmill, "Automated lane change controller design," *IEEE Trans. Intell. Transport. Syst.*, vol. 4, no. 1, pp. 13–22, Mar. 2003.

[5] Y. Guo, Q. Sun, R. Fu, and C. Wang, "Improved car-following strategy based on merging behavior prediction of adjacent vehicle from naturalistic driving data," *IEEE Access*, vol. 7, pp. 44258–44268, 2019.

[6] *National Highway Traffic Safety Administration. (2015). Critical REASons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey (DOT HS 812 115)*, NHTSA's Nat. Center Statist. Anal., Washington, DC, USA. [Online]. Available: wwwnrd.nhtsa.dot.gov/pubs/812115.pdf

[7] X. Xiong, L. Chen, J. Liang, Y. Cai, H. Jiang, and Y. Chen, "A study on the driving behavior prediction of dangerous lane change," *Autom. Eng.*, vol. 39, no. 9, pp. 1040–1067, Sep. 2017.

[8] J. Li, B. Dai, X. Li, X. Xu, and D. Liu, "A dynamic Bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification," *Electronics*, vol. 8, no. 1, p. 40, Jan. 2019.

[9] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. A, Policy Pract.*, vol. 77, pp. 167–181, Jul. 2015.

[10] S. Moridpour, M. Sarvi, and G. Rose, "Modeling the lane-changing execution of multiclass vehicles under heavy traffic conditions," *Transp. Res. Rec.*, vol. 2161, no. 1, pp. 11–19, Jan. 2010.

[11] Q. Wang, Z. Li, and L. Li, "Investigation of discretionary lane-change characteristics using next-generation simulation data sets," *J. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 246–253, Jul. 2014.

[12] D. Yang, C. He, M. Li, and Q. He, "Vehicle steering and lane-change behavior Recognition based on a support vector machine," *J. Tsinghua Univ., Sci. Technol.*, vol. 55, no. 10, pp. 1093–1097, Sep. 2015.

[13] D. Yang, L. Zhu, F. Yang, and Y. Pu, "Modeling and analysis of lateral driver behavior in lane-changing execution," *Transp. Res. Rec.*, vol. 2490, no. 1, pp. 127–137, Jan. 2015.

[14] J. Zheng, K. Suzuki, and M. Fujita, "Predicting driver-lane-changing decisions using a neural network model," *Simul. Model. Pract. Theory*, vol. 42, pp. 73–83, Mar. 2014.

[15] J. Peng, Y. Guo, R. Fu, W. Yuan, and C. Wang, "Multi-parameter prediction of drivers' lane-changing behaviour with neural network model," *Appl. Ergonom.*, vol. 50, pp. 207–217, Sep. 2015.

[16] Y. Dou, F. Yan, and D. Feng, "Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2016, pp. 901–906.

[17] L. Wang, Z. Zhang, H. Long, J. Xu, and R. Liu, "Wind turbine gearbox failure identification with deep neural networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1360–1368, Jun. 2017.

[18] T. Chen, X. Shi, and Y. D. Wong, "Key feature selection and risk prediction for lane-changing behaviors based on vehicles-trajectory data," *Accident Anal. Prevention*, vol. 129, pp. 156–169, Aug. 2019.

[19] P. Gipps, "A model for the structure of lane-changing decisions," *Transp. Res. B, Methodol.*, vol. 20, no. 5, pp. 403–414, Oct. 1986.

[20] Q. Yang and H. N. Koutsopoulos, "A microscopic traffic simulator for evaluation of dynamic traffic management systems," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 3, pp. 113–129, Nov. 1996.

[21] P. Hidas, "Modelling vehicle interactions in microscopic simulation of merging and weaving," *Transp. Res. C, Emerg. Technol.*, vol. 13, no. 1, pp. 37–62, Feb. 2005.

[22] J. Monteil, R. Billot, N.-E. El Faouzi, A. Nantes, and J. Sau, "Microscopic cooperative traffic flow: Calibration and simulation based on a next generation simulation dataset," *IET Intell. Transp. Syst.*, vol. 8, no. 6, pp. 519–525, Sep. 2014.

[23] T. Toledo, H. N. Koutsopoulos, and M. Ben-Akiva, "Integrated driving behavior modeling," *Transp. Res. C Emerg. Technol.*, vol. 15, no. 2, pp. 96–112, Apr. 2007.

[24] Y. Hou, P. Edara, and C. Sun, "Situation assessment and decision making for lane change assistance using ensemble learning methods," *Expert Syst. Appl.*, vol. 42, no. 8, pp. 3875–3882, May 2015.

[25] X. Qiu, Y. Liu, L. Ma, and D. Yang, "A lane change model based on Bayesian networks," *Transp. Syst. Eng. Inform. Technol.*, vol. 15, no. 5, pp. 67–74, Oct. 2015.

[26] Y. Liu, X. Wang, L. Li, S. Cheng, and Z. Chen, "A novel lane change decision-making model of autonomous vehicle based on support vector machine," *IEEE Access*, vol. 7, pp. 26543–26550, 2019.

[27] J. Nie, J. Zhang, X. Wan, W. Ding, and B. Ran, "Modeling of decision-making behavior for discretionary lane-changing execution," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 707–712.

[28] B. Xu, X. Liu, Z. Wang, F. Liu, and J. Liang, "Fusion decision model for vehicle lane change with gradient boosting decision tree," *J. Zhejiang Univ., Engin. Sci.*, vol. 53, no. 6, pp. 1–11, Jun. 2019.

[29] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "A machine learning approach for personalized autonomous lane change initiation and control," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1590–1595.

[30] C. Vallon, Z. Ercan, A. Carvalho, and F. Borrelli, "Study on driving decision-making mechanism of autonomous vehicle based on an optimized support vector machine regression," *Appl. Sciences*, vol. 8, no. 1, p. 13, Dec. 2017.

[31] N. Motamedidehkordi, S. Amini, S. Hoffmann, F. Busch, and M. R. Fitriyanti, "Modeling tactical lane-change behavior for automated vehicles: A supervised machine learning approach," in *Proc. 5th IEEE Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, Jun. 2017, pp. 268–273.
[32] J. Nilsson, M. Brannstrom, E. Coelingh, and J. Fredriksson, "Lane change maneuvers for automated vehicles," *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 5, pp. 1087–1096, May 2017.
[33] H. Woo, Y. Ji, Y. Tamura, Y. Kuroda, T. Sugano, Y. Yamamoto, A. Yamashita, and H. Asama, "Lane-change detection based on individual driving style," *Adv. Robot.*, vol. 33, no. 20, pp. 1087–1098, Oct. 2019.
[34] T. Xu, C. Wen, L. Zhao, M. Liu, and X. Zhang, "The hybrid model for lane-changing detection at freeway off-ramps using naturalistic driving trajectories," *IEEE Access*, vol. 7, pp. 103716–103726, 2019.
[35] D. Bhatt and S. Gite, "Modeling techniques for driver's lane changing behaviour," in *Proc. 9th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2018.
[36] D. Yi, J. Su, C. Liu, M. Quddus, and W.-H. Chen, "A machine learning based personalized system for driving state recognition," *Transp. Res. C, Emerg. Technol.*, vol. 105, pp. 241–261, Aug. 2019.
[37] C. F. Choudhury, V. Ramanujam, and M. E. Ben-Akiva, "Modeling acceleration decisions for freeway merges," *Transp. Res. Rec.*, vol. 2124, no. 1, pp. 45–57, Jan. 2009.
[38] M. Park, K. Jang, J. Lee, and H. Yeo, "Logistic regression model for discretionary lane changing under congested traffic," *Transportmetrica A, Transp. Sci.*, vol. 11, no. 4, pp. 333–344, Apr. 2015.
[39] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 89–100, Feb. 2018.
[40] A. Fischer and C. Igel, "Training restricted Boltzmann machines: An introduction," *Pattern Recognit.*, vol. 47, no. 1, pp. 25–39, Jan. 2014.
[41] H. Zhao, H. Liu, H. Liu, and Y. Lin, "Condition monitoring and fault diagnosis of wind turbine generator based on stacked autoencoder network," *Autom. Electr. Power Syst.*, vol. 42, no. 11, pp. 102–108, Jan. 2018.
[42] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 85–794.
[43] Y. Xiang, G. Zhang, S. Gu, and J. Cai, "Online multi-layer dictionary pair learning for visual classification," *Expert Syst. Appl.*, vol. 105, pp. 174–182, Sep. 2018.
[44] L. Tan, P. Li, F. Tao, A. Miao, and M. Cao, "Cable joint fault detection for the ring main unit based on an adaptive TNPE algorithm," *WIREs Data Mining Knowl. Discovery*, vol. 10, no. 1, Jan. 2020, Art. no. e1336.
[45] FHWA U.S. Department of Transportation. (Jun. 2006). *NGSIM: Next Generation Simulation*. [Online]. Available: www.ngsim.fuwa.dot.gov
[46] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.
[47] C. Thiemann, M. Treiber, and A. Kesting, "Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data," *Transp. Res. Rec.*, vol. 2088, no. 1, pp. 90–101, Jan. 2008.
[48] C. Pek, P. Zahn, and M. Althoff, "Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1477–1483.
[49] T. Kondoh, T. Yamamura, S. Kitazaki, N. Kuge, and E. R. Boer, "Identification of visual cues and quantification of drivers' perception of proximity risk to the lead vehicle in car-following situations," *J. Mech. Syst. Transp. Logging*, vol. 1, no. 2, pp. 170–180, 2008.
[50] A. Shaout, D. Colella, and S. Awad, "Advanced driver assistance systems—Past, present and future," in *Proc. 7th Int. Comput. Eng. Conf. (ICENCO)*, Dec. 2011, pp. 72–82.
[51] R. Dang, J. Ding, B. Su, Q. Yao, Y. Tian, and K. Li, "A lane change warning system based on V2V communication," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1923–1928.
[52] F. Altche and A. De La Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 353–359.
[53] J.-H. Deng and H.-H. Feng, "A multilane cellular automaton multi-attribute lane-changing decision model," *Phys. A, Stat. Mech. Appl.*, vol. 529, Sep. 2019, Art. no. 121545.
[54] M. Chen, Q. Liu, S. Chen, Y. Liu, C.-H. Zhang, and R. Liu, "XGboost-based algorithm interpretation and application on post-fault transient stability status prediction of power system," *IEEE Access*, vol. 7, pp. 13149–13158, 2019.
[55] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.
[56] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Nov. 2012, pp. 2951–2959.
[57] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 1, pp. 2825–2830, Oct. 2011.
[58] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Comput. Intell.*, vol. 20, no. 1, pp. 18–36, Feb. 2004.
[59] C. Wang, Q. Sun, Z. Li, H. Zhang, and K. Ruan, "Cognitive competence improvement for autonomous vehicles: A lane change identification model for distant preceding vehicles," *IEEE Access*, vol. 7, pp. 83229–83242, 2019.
[60] X. Wang, J. Wu, Y. Gu, H. Sun, L. Xu, S. Kamijo, and N. Zheng, "Human-like maneuver decision using LSTM-CRF model for on-road self-driving," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 210–216.

XINPING GU received the B.S. degree in industrial engineering from Xi'an Polytechnic University, Xi'an, China, in 2016. He is currently pursuing the M.S. degree in industrial engineering with the School of Mechanical Engineering, Shandong University, Jinan, China.

His current research interests include intelligent vehicle behavior decision-making and path planning, and virtual reality technology.

YUNPENG HAN was born in Jinan, China, in 1962. He received the B.S. and M.S. degrees in mechanical manufacture and automation from the Shandong University of Technology, Jinan, in 1983 and 1991, respectively. He is currently a Professor and a Master's Supervisor with the School of Mechanical Engineering, Shandong University. He has leaded more than ten research projects, including the National Natural Science Foundation of China, the National High Technology Research and Development Program, and the Key Research and Development Programs of Shandong Province. His research interests include digital design, virtual reality, and 3D visualization. He is a Senior Member of the Chinese Mechanical Engineering Society and a member of the Shandong Safety Production Expert Group.

JUNFU YU received the B.S. degree in mechanical design manufacture and automation from Shandong University, Jinan, China, in 2018, where he is currently pursuing the M.S. degree in mechanical manufacture industrial engineering with the School of Mechanical Engineering.

His research interests include supply chain management and machine learning.