Cite this article:
Sun, H., Cheng, Q., Wang, P., Huang, Y., & Liu, Z. (2024). Lane change decision prediction: An efficient BO-XGB modeling approach with interpretable Shapley additive explanation analysis. Accepted by *Transportmetrica A*. https://doi.org/10.1080/23249935.2024.2372020

Lane Change Decision Prediction: An Efficient BO-XGB Modeling Approach with SHAP Analysis

*Haobo Sun [a], Qixiu Cheng [b], Pu Wang [c], Yongqi Huang [d], Zhiyuan Liu [a, *]*

[a] Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, School of Transportation, Southeast University, Nanjing 210096, China.

[b] University of Bristol Business School, University of Bristol, Bristol BS8 1PY, UK.

[c] School of Traffic and Transportation Engineering, Central South University, Changsha 410000, China.

[d] Key Laboratory of Concrete and Prestressed Concrete Structures of Ministry of Education, Southeast University, Nanjing 211189, China.

[*] Corresponding author.

E-mail: h_b_sun@163.com (H. Sun); qixiu.kevin.cheng@gmail.com (Q. Cheng); wangpu@csu.edu.cn (P. Wang); yancy7@yeah.net (Y. Huang); zhiyuanl@seu.edu.cn (Z. Liu).

Abstract

The lane-change decision (LCD) is a critical aspect of driving behaviour. This study proposes an LCD model based on a Bayesian optimization (BO) framework and extreme gradient boosting (XGBoost) to predict whether a vehicle should change lanes. First, an LCD point extraction method is proposed to refine the exact LCD points with a highD dataset to increase model learning accuracy. Subsequently, an efficient XGBoost with BO (BO-XGB) was used to learn the LCD principles. The prediction accuracy on the highD dataset was 99.14% with a computation time of 66.837s. The accuracy on the CQSkyEyeX dataset was 99.45%. Model explanation using the shapley additive explanation (SHAP) method was developed to analyse the mechanism of the BO-XGB's LCD prediction results, including global and sample explanations. The former indicates the particular contribution of each feature to the model prediction throughout the entire dataset. The latter denotes each feature's contribution to a single sample.

Keywords: Lane change decision-making; XGBoost; Bayesian optimization; Model explanation; Integrated learning; SHAP.

1 Introduction

The explosive growth in the number of private vehicles has caused considerable problems for road traffic managers, including congestion and accidents. Approximately 539,000 traffic accidents occur annually in the United States due to risky lane changes (Liu, 2019). Modeling lane-change behavior is a critical aspect of microscopic traffic flow theory (Li et al., 2020; Wang et al., 2024, 2023). Lane-change behaviors are more complex and subjective than car-following behaviors (Gao et al., 2022; Jin et al., 2019; Zhu et al., 2022), resulting in a greater likelihood of accidents (Ji and Levinson, 2020; Zheng et al., 2010). Statistically, drivers driving at 90 km/h on freeways make approximately 50 lane changes per 100 km, especially under free-flow conditions (Liu, 2019). U.S. National Highway Traffic Safety Administration statistics reveal that lane changes cause 27% of traffic accidents (Zhao et al., 2021). Highly accurate lane-change decision (LCD) prediction helps improve proactive driving safety and protection (Schomakers et al., 2023; Tang et al., 2019). Consequently, it is vital to establish an advanced LCD prediction model that predicts when vehicles should change lanes (Moridpour et al., 2010).

However, most current studies on lane-change models use traditional rule-based models or artificial intelligence algorithms to portray vehicle lane-change trajectories (Sun and Elefteriadou, 2014). Although the LCD principle in traditional models is well-defined; their application scenarios are too specific (Zheng, 2014). In addition, the factors considered in rule-based models are limited, which significantly reduces the applicability of these lane change models. Instead, models based on artificial intelligence have a more comprehensive range of applications, and more factors can be incorporated into them. However, the principles and interpretation of these models remain unclear. Therefore, it is necessary to formulate a mechanically well-defined LCD model that can be applied to many different scenarios by combining the mechanisms of traditional models with artificial intelligence methods, such as the supervised-learning used in this study.

Most supervised-learning methods aim to derive a model that performs satisfactorily in varying conditions (Huang et al., 2024; Tang et al., 2024, 2023). However, this is often unsatisfactory because the interpretability of most machine learning methods is disappointing, and sometimes (Sun and Huang, 2024), we can only generate weakly supervised models with different preferences. However, integrated learning combines these weakly supervised models to derive a more comprehensive supervised model (Galar et al., 2012). Although a weak classifier could provide an inaccurate classification prediction, other weak classifiers in integrated learning can quickly remedy the error, allowing the inaccurate classification prediction to be adjusted in the final prediction result (He et al., 2023; Sagi and Rokach, 2018). Consequently, it has excellent strength and potential for use in automated decision-making systems (Hou et al., 2015).

Building an accurate data-driven LCD model with integrated learning requires excellent model construction and optimization as well as precise inputs and outputs (Tajdari et al., 2019). Therefore, it is crucial to extract the exact point at which the driver intends to change lanes (i.e., lane change intention point). It is widely recognized that the vehicle environment at that point is the precise input and output required for an LCD model (Hornberger et al., 2018). Lateral vehicle positions and lateral velocities are typically used to extract lane-change intention points from the trajectory dataset (Li et al., 2016; Shangguan et al., 2022; Xing et al., 2020). However, this ignores the incidental lateral deflections of vehicles owing to drivers' heterogeneous habits rather than lane-change intentions. In addition, high-speed vehicles may suddenly appear from the back of their target lane, forcing some lane-change vehicles to cease their lane-change behavior temporarily. Studies have not considered the effect of these temporary avoidance behaviors on identifying lane-change intention points (Shangguan et al., 2022). Many existing lane-changing studies only consider the lane-change intention points at which lane-changing succeeds but ignore many lane-change intention points at which lane-changing fails due to avoidance behavior (Ali et al., 2022). However, the lane-change intention point of a failed lane change is also the moment when the driver makes a lane-change decision in a real lane-change scenario (Ali et al., 2020).

This study applies extreme gradient boosting (XGBoost), one of the most accurate integrated learning models for autonomous decision-making, with Bayesian Optimization (BO) to construct an innovative LCD model (BO-XGB). Moreover, the lane-change vehicle velocity, relative distance, and relative velocity to its surrounding vehicles were extracted from the highD dataset and input into XGBoost (Chen and Guestrin, 2016). After tuning the hyperparameters of XGBoost using BO (Shahriari et al., 2016; Snoek et al., 2012), the accuracy reached 99.14%. To improve the interpretability of the BO-XGB model, a game theory-based model explanation method, i.e., the shapley additive explanation analysis (SHAP) (Shapley, 2016), is used to explain its prediction results and model principles. To prove that the model is widely applied, we further tested it on the open-source dataset CQSkyEyeX in China, which resulted in an accuracy of 99.45%.

In addition, this study addresses three deficiencies frequently observed in many existing LCD models.

(1) An innovative and accurate LCD model, BO-XGB, was developed using the XGBoost model optimized using BO in multilane scenarios. Existing studies (Zhang et al., 2022) have used similar principles but focused on feature combinations rather than accurate decision-making and in-depth model explanations (Ali et al., 2022). They lack precise LCD point extraction (i.e., ignore lane-change intention points of failed lane changes), hyperparameter optimization, and complete model explanation methods (i.e., some studies focus on the global model explanation and ignore the sample explanation).

(2) A new approach for extracting exact LCD points considering the trajectory of the subject vehicle and its surrounding vehicles was designed for high-resolution vehicle trajectory data.

(3) A framework is proposed from a game theory perspective to explain the quantitative LCD prediction results of the machine learning model in a single sample and the entire dataset to improve model interpretability, which can aid researchers in analyzing the causes of abnormal samples.

The rest of this paper is structured as follows: The literature review related to this study is presented in Section 2, followed by the research problem statement in Section 3. The data processing, LCD point extraction method, and principles of XGBoost, BO, and SHAP are explained in detail in Section 4. Section 5 introduces the data used in this study, the performance of BO-XGB in LCD prediction along with a comparison to other methods. SHAP is used to conduct model explanation experiments. The results, conclusions, and future research directions are discussed in Section 6.

2 Literature Review

Most traditional LCD models are rule-based, and their targets include improving safety (Li et al., 2020) and providing the benefit of lane changes (Ben-Akiva et al., 2012). Gipps (1986) first proposed a fundamental LCD model that focused on improving road safety. In Gipps' model, deceleration is regarded as an indicator of lane-change feasibility. However, it ignores the stochastic variations between the behavioral characteristics of different drivers, which are essential for studying LCDs (Zheng et al., 2010). To address this issue, Yang and Koutsopoulos (1996) proposed a microscopic traffic simulator model by adding the lane change probability based on Gipps' model. The corridor simulation model (Halati et al., 1997), developed by the Federal Highway Administration, combines the freeway simulation model and the network simulation model, which consist of motivation, benefit, and urgency. Based on previous studies, this model categorizes lane-change behaviors into mandatory lane change, discretionary lane change, and random lane change. Based on these studies, Toledo et al. (2003, 2007) developed a practical framework that integrated mandatory and discretionary lane changes. These models focus on improving lane-change safety.

Some researchers have modeled LCD behavior to enhance lane-change benefits (Ben-Akiva et al., 2012; Hidas, 2005; Kesting et al., 2007; Monteil et al., 2014). A theory of LCD based on acceleration control was proposed to derive lane-change rules for various types of vehicles (Kesting et al., 2007), referred to as the minimizing overall braking induced by lane changes (MOBIL) model. This model incorporates game theory in mathematics to assess the synergy and conflict between the subject vehicle and its surrounding vehicles. It then determines whether to switch lanes by calculating the variation in the system benefits of some involved vehicles when changing lanes. Ben-Akiva et al. (2012) proposed a benefit-based LCD model by artificially defining a benefit function that calculates the benefit and then selects the behavior with the highest benefit. Based on Kesting's research, Monteil et al. (2014) proposed an LCD framework incorporating the full velocity difference model and MOBIL model. Furthermore, some studies (Hidas, 2005; Monteil et al., 2014) used an innovative approach to model lane-change behaviors by integrating several factors (e.g., velocity and acceptable clearance) and developing a southern integrated transportation system model.

Although these traditional models' accuracy and explanatory power are guaranteed (Cheng et al., 2024b, 2024a; Mo et al., 2024, 2023), their performance in different scenarios is unsatisfactory because of rigid modeling scenarios and assumptions (Tang et al., 2018; Zhang et al., 1998). In addition, the factors (e.g., safe clearance and acceptable acceleration) considered in these traditional models are somewhat limited (Rahman et al., 2013; Tang et al., 2018), making it challenging to incorporate potential elements of drivers in decision-making and expand the model's applicability (Zheng, 2014).

Most cutting-edge studies have combined lane-change models with artificial intelligence algorithms. Among them, three main types are classified according to their methodology (Das et al., 2020; Guo et al., 2021; Nie et al., 2016; Wang et al., 2021; Xie et al., 2019; Y. Zhang et al., 2023): traditional machine learning models, deep learning models, and reinforcement learning models. Nevertheless, their research excluded multilane scenarios and was limited to two-lane settings. Liu et al. (2019) established a support vector machine model using a BO. This model solves the problem of the multiparameter and nonlinear characteristics of the autonomous LCD process. However, this study is still based on a two-lane scenario. By integrating long short-term memory and Bayesian methods, an LCD model that can adapt to different driving environments was developed (Wang et al., 2021). Fei et al. (2020) combined a generative adversarial network and imitation learning to derive a lane-change trajectory model with 80% accuracy. One popular paradigm is the lateral decision-making schemes with the deep Q-network (DQN) or its variants (Li et al., 2022). To improve their superior generalization and robustness, the latest research uses reinforcement learning to study the robust decision-making problem of self-driving cars (He et al., 2023). Some lane change decision-making studies based on vehicle trajectories use time-series data-driven deep learning methods, such as Long Short-Term Memory (Ashfaq et al., 2023; Gao et al., 2020; Li et al., 2022; Zhang et al., 2024a) and Transformer (Gao et al., 2023; Zhang et al., 2024b). They can make suggestions and alerts to drivers well in advance as the time window passes. However, this is where the problem lies. Giving advance advice on lane change decisions can provide drivers with more reaction time. However, it ignores the influence of surrounding vehicles on lane change decision-making (Ashfaq et al., 2023; Guo et al., 2022; Zhang et al., 2024a, 2024b), especially the sudden entry of the rear vehicle in the target lane (Gao et al., 2020). Due to this situation, the lane change decision problem needs to be considered as a classification problem. In the classification problem of lane-changing decisions, the integrated model may outperform the methods of advanced models such as Transformer (Zhang et al., 2024b, 2024a). Moreover, all such methods suffer from weak explanatory modeling principles, as seen in Table 1. Many studies have integrated deep and reinforcement learning models to obtain more accurate decision-making results (Li et al., 2022; Peng et al., 2022). However, they ignore the LCD process (which acknowledges when and where the driver can change lanes) in favor of focusing on the trajectory learning of lane-change behaviors, that is, the process of changing lanes (such as the acceleration and steering angle). This study fills this gap and uses the BO-XGB model to predict LCD precisely with the global and sample explanations provided by SHAP.

It is evident that most studies on machine learning and lane-change models have been conducted on learning lane-change trajectories, and fewer studies have been conducted on LCDs. Although traditional LCD models are concise and easy to understand, some limitations exist, such as numerous assumptions and application scenarios that are too specific. Furthermore, most LCD models based on artificial intelligence (AI) algorithms focus on two-lane scenarios (Li and Sun, 2017), ignoring the interpretability of the model and the influence of vehicles in the lane adjacent to the lane-change vehicle (Fei et al., 2020; Gu et al., 2020; S. Li et al., 2022; Peng et al., 2022). Moreover, most existing studies for lane change decision-making take the point where the vehicle presses over the lane line as the lane change decision point (Ali et al., 2023), which is indeed imprecise. This is one of the problems we are trying to solve.

3 Problem Statement

This study focuses on the LCD problem encountered during a drive on an expressway. According to previous studies (Gipps, 1986; Hidas, 2005; Peng et al., 2022), most LCD problems can be normalized to a function mapping problem, as follows:

$$f:\mathbf{X}\rightarrow\mathbf{Y}\,, \tag{1}$$

where $\mathbf{X}$ is the input dataset that includes factors affecting the LCD, $\mathbf{Y}$ is the output dataset, and y is an element of $\mathbf{Y}$. $y=1$ implies that the vehicle does not change lanes, whereas $y=2$ and $y=0$ denote that the vehicle changes lanes to the left and right, respectively. Additionally, the vehicles surrounding a lane-change vehicle are defined as shown in Fig. 1. This can be easily reduced to a two-lane scenario with only a left or right lane adjacent to the subject lane.

Table1. Comparison table of the latest typical studies on lane change decisions

Research	LCD point extraction method	Parameter optimization methods	Vehicle environment	Model explanation
Ashfaq et al. (2023)	✕	✕	✕	✕
Gao et al. (2022)	✕	✕	incomplete	✕
Wang et al. (2022)	✕	✕	✓	sensitivity analysis, lack of sample explanation
Xue et al. (2022)	✕	single parameter	✓	✕
Li et al. (2022)	✕	✕	✓	✕
Zhang et al. (2022)	✕	✕	✕	✕
Gao et al. (2020)	✕	✕	incomplete	✕
Li et al. (2022)	✕	✕	✓	✕
Atagoziev et al. (2023)	✕	✕	✕	✕
Zhang et al. (2024a)	✕	Adam+grid search	✕	✕
Zhang et al. (2024b)	✕	✓	✕	✕
Guo et al. (2022)	✓	✓	✕	✕
Gao et al. (2023)	✕	✓	✓	✕
Our model	✓	Bayesian optimization	✓	SHAP

Note: ✕ indicates a lack of relevant content, ✓ indicates there is no mention of this aspect in the research.

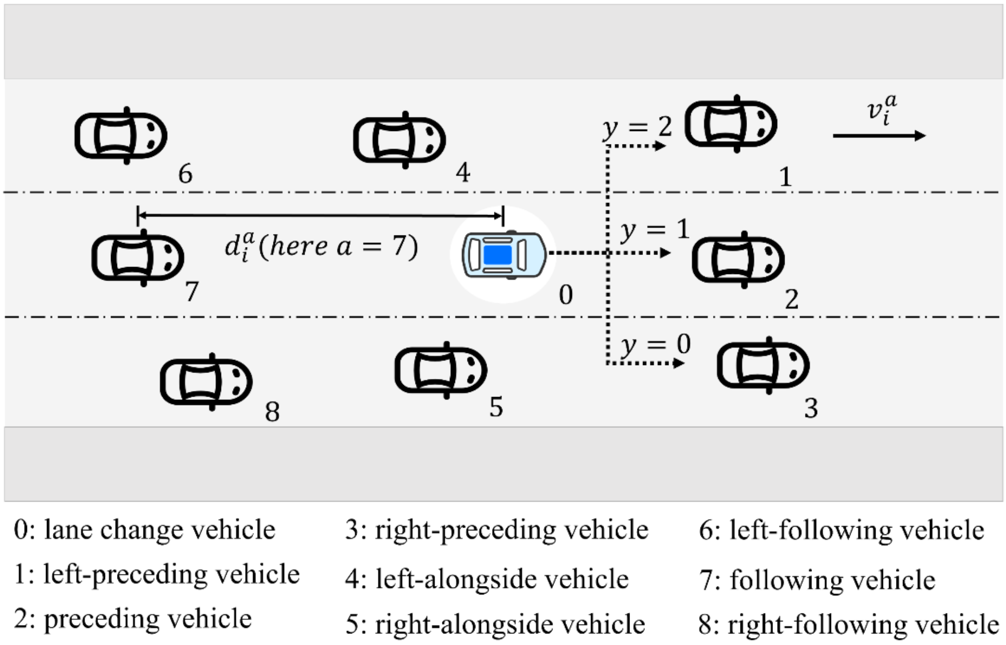

6
4
y = 2
1
v_i^a
d_i^a(here a = 7)
7
0
y = 1
2
y = 0
8
5
3
0: lane change vehicle
1: left-preceding vehicle
2: preceding vehicle
3: right-preceding vehicle
4: left-alongside vehicle
5: right-alongside vehicle
6: left-following vehicle
7: following vehicle
8: right-following vehicle

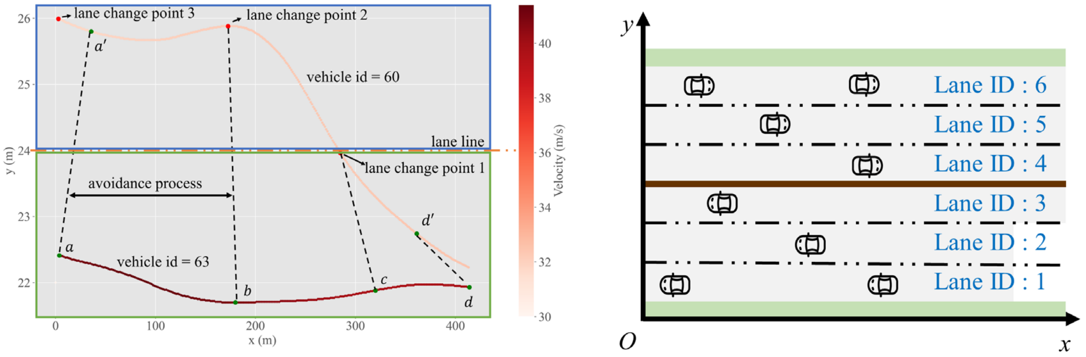
Fig. 1. Illustration of lane-change factors

The input factor $\mathbf{X}$ influencing the LCD of vehicle i from Section 2 can be summarized into the following four aspects: the velocity v_i^0 of the lane change vehicle i. The safety space (expressed by the vector of relative distances $\mathbf{D}_i = \{d_i^a \mid a = 1, 2, \cdots, 8\}$ between vehicle i and its surrounding vehicles). The velocity gains of lane change behavior (expressed by the vector of relative velocities $\mathbf{U}_i = \{\mu_i^a \mid a = 1, 2, \cdots, 8\}$ between vehicle i and its surrounding vehicles), and the limit of safety acceleration. We can eliminate the limit of safety acceleration from the input component because it is constrained by velocity and distance derived from the kinematic equations (Whelan and Hodgson, 1978). Consequently, the input to the LCD of vehicle i can be defined as follows:

$$\mathbf{X}_i = \left[v_i^0, \mathbf{D}_i, \mathbf{U}_i \right], \tag{2}$$

where v_i^0 denotes the velocity of vehicle i.

It is essential to extract exact LCD points (called lane-change intention points) where drivers intend to change lanes from the vehicle's input environment information (Venthuruthiyil and Chunchu, 2022; Wang et al., 2022). However, extracting the exact LCD points due to vehicles' avoidance behaviors is difficult, as shown in Fig. 2(a). It shows the paths traversed by vehicle 60, which switches lanes, and vehicle 63, which accelerates forward from behind vehicle 60. The positions of these two vehicles in the same frame are shown in the figure as points paired with dashed lines. Vehicle 60 initially decided to change lanes before the arrival of vehicle 63. However, it is forced to avoid and give up lane space because high-speed vehicle 63 arrives rapidly from behind in its target lane. Consequently, there is a section of y-axis displacement in the direction opposite to the lane-change process. A lane change is only possible when the driver finds sufficient space. Therefore, for vehicle 60, the exact LCD point was the initial point (i.e., lane-change point 3 in Fig. 2(a)) of the first section in its lane-change trajectory.



a b

Fig. 2. Schematic of the exact LCD point extraction: (a) A real case in the highD dataset (trajectory points connected by dashed lines represent the trajectory points in the same frame), and (b) An illustration of the coordinate system construction in this study.

Therefore, the primary challenges of this research are learning complex function mapping relationships and extracting precise LCD points from a large volume of trajectory data. This study proposes a novel LCD point extraction method for the highD dataset. The extracted data were integrated into the XGBoost model (Chen and Guestrin, 2016) with BO for training and optimization to obtain a data-driven LCD model (BO-XGB model). Furthermore, we explain the broad concepts of the BO-XGB model and the contribution of each feature to the single-sample prediction result for LCDs using the SHAP method. Table 2 summarizes the primary notation used in this paper.

Table 2. Notations

Notations	Description
$\mathbf{X}$	Set of input variables
$\mathbf{Y}$	Set of output variables
$\mathbf{Q}$	Raw trajectory dataset

$\mathbf{D}_i$	Vector of relative distances between lane change vehicle *i* and its surrounding vehicles
$\mathbf{U}_i$	Vector of relative velocities between lane change vehicle *i* and its surrounding vehicles
$\mathbf{V}_x$	Velocity vector of the lane change vehicle along the x-axis
$\mathbf{V}_y$	Velocity vector of the lane change vehicle along the y-axis
$\mathbf{T}$	Frame order vector of lane change vehicle trajectory data
$\mathbf{l}$	Vector of lane *id*s
$\mathbf{L}$	Position vector of the lane change vehicle on the x-axis
$\mathbf{q}_{i,t}^{env}$	Vector of vehicle environment of vehicle i at frame t
$\mathbf{V}_{i,t}$	Surrounding vehicles' velocities related to vehicle i along the x-axis at frame t
$\mathbf{P}_{i,t}$	Surrounding vehicles' x-axis coordinates related to vehicle i along the x-axis at frame t
b_0, b_1	Frame guides for recording the point frame
l_t	Lane *id* of lane change vehicle at frame t
i	Vehicle id
t'	Frame of exact LCD points
v_x^t	Velocity of the vehicle along the y-axis at frame t
v_y^t	Velocity of the vehicle along the y-axis at frame t
a	Index for vehicles around a lane change vehicle, as shown in Fig. 1, where $a = 1 \sim 8$ indicate respectively the left-preceding vehicle, preceding vehicle, right-preceding vehicle, left-alongside vehicle, right-alongside vehicle, left-following vehicle, following vehicle, and right-following vehicle
$VID_{i,t}^a$	*id* of the surrounding vehicle a for vehicle i at frame t
d_i^a	Relative distance between vehicle i and the surrounding vehicle a
μ_i^a	Relative velocity between vehicle i and the surrounding vehicle a
p_i^a	x-axis coordinate of the vehicle a around vehicle i

v_i^a	x-axis velocity of the vehicle a around vehicle i
p_i^0	x-axis coordinate of vehicle i in the image coordinate system
v_i^0	Velocity of vehicle i along x-axis

4 Methodology

This section describes the three subsections that establish the BO-XGB model for LCDs. The detailed framework of the methodology is shown in Fig. 3. In Section 4.1, an LCD point extraction method is designed by considering the changes in the vehicle velocity and trajectory line segments of each lateral offset. The detailed algorithm is described in Appendix A. Then, in Appendix B, the *id*s of vehicles surrounding the lane-changing vehicles are linked to the lane-changing vehicle to address the problem of insufficient information related to the environment of the raw vehicle trajectory data. Based on the first two parts, the driving environment of the vehicle is quantified and input into the model. Section 4.2 introduces the XGBoost model, the basis for learning the mapping relationship between input and output. Section 4.3 illustrates the BO method used to tune the hyperparameters in XGBoost and derive a convergent and well-behaved model for LCD. Section 4.4 describes the SHAP method used in this study to explain the model (primarily the equations used for calculating the SHAP values).

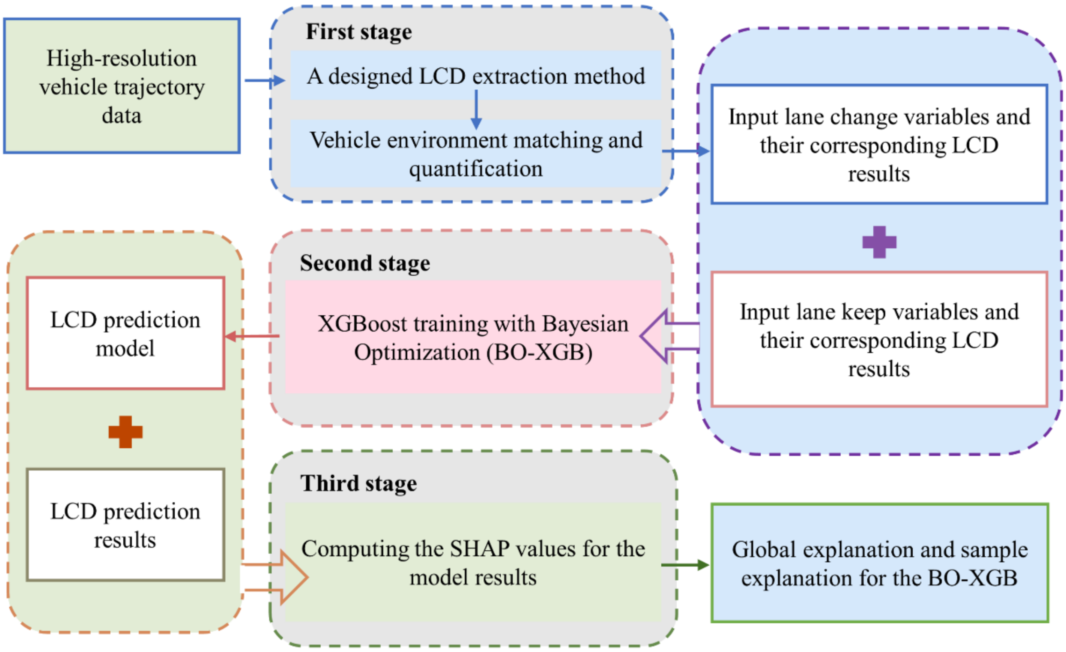

High-resolution vehicle trajectory data
First stage
A designed LCD extraction method
Vehicle environment matching and quantification
Input lane change variables and their corresponding LCD results
Input lane keep variables and their corresponding LCD results
Second stage
XGBoost training with Bayesian Optimization (BO-XGB)
LCD prediction model
LCD prediction results
Third stage
Computing the SHAP values for the model results
Global explanation and sample explanation for the BO-XGB

Fig. 3. Framework of the proposed method.

4.1 Lane change decision point extraction method

The major goal of this subsection is to extract the exact frame wherein drivers intend to change lanes during the lane-change process after identifying lane-changing vehicles from a large volume of raw vehicle trajectory data $\mathbf{Q}$. The closer the LCD model is to the frame wherein the driver intends to change lanes, the more relevant the decision information (Chauhan et al., 2022; Zhang et al., 2022). The point in this frame is considered the exact LCD point in this research where drivers originally intended to change lanes.

As mentioned in Section 3, considering the avoidance behaviors of certain lane-change vehicles, the LCD point extraction method can be divided into three steps. The "lane change point 1" in Fig. 2(a) is the first step to extract an inexact point based on the shift of the lane *id*. This is considered a rough LCD point because the point calculated by this method is the point at which the lane-change action is almost complete rather than the initial point at which the driver intends to change lanes. In the second step, we search forward along the track, from the "lane change point 1" to the point where the vehicle lateral velocity changes from a negative (positive) value to a positive (negative) value for the first time, which is the "lane change point 2" in Fig. 2(a). The third step is a forward search based on the second step to judge whether there is a section of significant and continuous trajectory in the same direction as the lane change between "lane change point 2" and the track's starting point. The exact LCD point is "lane change point 2" if there is no such trajectory. If not, the exact LCD point is the starting point of such a trajectory ("lane change point 3" in Fig. 2(a)). The proposed LCD point extraction algorithm is illustrated in Appendix A.

We can then obtain the output of this model using Equation (3) and Fig. 2(b).

$$y = \begin{cases} 0\text{，} \text{ if } v_x^{t'+1} \cdot v_y^{t'+1} < 0 \text{ means the vehicle turns right} \\ 1\text{，} \text{ if the vehicle does not change the lane} \\ 2\text{，} \text{ if } v_x^{t'+1} \cdot v_y^{t'+1} > 0 \text{ means the vehicle turns left} \end{cases}, \tag{3}$$

where t' denotes the exact frame of the LCD point, and $v_x^{t'+1}$ and $v_y^{t'+1}$ indicate respectively the velocity of the vehicle along the x-axis and y-axis at frame $t'+1$.

Subsequent data processing, such as data matching and quantification of the vehicle-driving environment, is presented in Appendix B.

4.2 XGBoost model

This study used XGBoost, one of the most frequently used integrated learning algorithms (Cai et al., 2022; Mohammadi et al., 2019; Shi et al., 2019; Wang et al., 2022), to learn the relationship between the input information X and output decision Y. This section introduces XGBoost so that readers can understand the proposed LCD model. XGBoost efficiently implements gradient boosting decision trees, which are structured similarly to trees with root nodes, internal nodes, and leaves. Decision tree algorithms (Parsa et al., 2020) typically use simple rules that start at the root node, branch out, pass through the internal nodes, and finally reach the leaf nodes. In contrast, gradient-boosting decision trees are integrated learning techniques that utilize a series of decision trees. In this method, each decision tree learns from the previous decision tree and influences the next tree to improve the model and build a robust learning framework. Next, the principle of XGBoost is introduced; interested readers can refer to Chen and Guestrin (2016) for further details.

Given a dataset with a sample size of n, there are the independent variables x_i ($i \in [1, n]$) with m environmental features and $x_i \in \mathbf{X}^{n \times m}$. For each LCD environment variable x_i, there is a true LCD variable y_i. Its tree models predict the value of $\overline{y_i}$ through x_i, which can be formulated as

$$\overline{y_i^K} = \sum_{k=1}^{K} f_k\left(x_i\right) = \overline{y_i^{K-1}} + f_K(x_i), f_k \in F, \tag{4}$$

where f_k represents an independent tree structure, k is the tree index, K indicates the total number of trees used for addition, F denotes the space of the trees, and $\overline{y_i^K}$ indicates the prediction result after K iterations. This additive procedure is called "boosting." Each unit of the model is described as a tree. The objective was to obtain the learning parameters by minimizing the loss function, as shown in Equation (5).

$$\{\varphi\}_1^K = \arg\min_{\varphi} \sum_{i=1}^{n} l\left(y_i, \sum_{k=1}^{K} \left[f_k(x_i) + \Omega(f_k) \right] \right), \tag{5}$$

where φ depicts the learning parameters of tree k, $\sum_{i=1}^{n} l\left(y_i, \sum_{k=1}^{K} \left[f_k(x_i) + \Omega(f_k) \right] \right)$ indicates the loss function of the model wherein split cross-entropy is used as the loss value to quantify the error and the penalty term $\sum_{k=1}^{K}\Omega(f_k)$ for model complexity to reduce the risk of overfitting.

The model complexity of a single-base tree is defined by Equation (6).

$$\Omega(f_k) = \gamma R + \frac{1}{2}\lambda\sum_{j=1}^{R}\omega_j^2 , \tag{6}$$

where R represents the number of leaves, ω_j is the score of the leaf j; γ and λ are model parameters, and the optimal solution ω_j^* is derived by solving Equations (4)–(6): The corresponding solution values are as follows:

$$\omega_j^* = -\frac{\sum_{i\in\mathbf{I}_j}\partial_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right)}{\sum_{i\in\mathbf{I}_j}\partial^2_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right) + \lambda} , \tag{7}$$

$$\xi^{(K)} = -\frac{1}{2}\sum_{j=1}^{R}\frac{\left(\sum_{i\in\mathbf{I}_j}\partial_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right)\right)^2}{\sum_{i\in\mathbf{I}_j}\partial^2_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right) + \lambda} + \gamma R , \tag{8}$$

where $\mathbf{I}_j$ represents the sample dataset of the leaf j. It is difficult to compute this optimal solution ω_j^* for all possible trees in practical situations; therefore, Equation (9) is frequently used (Parsa et al., 2020).

$$\xi_{prac} = \frac{1}{2}\left[\begin{array}{c}\dfrac{(\sum_{i\in\mathbf{X}_L}\partial_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right))^2}{\sum_{i\in\mathbf{X}_L}\partial^2_{\overline{y}_i^{(r-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right) + \lambda} + \dfrac{(\sum_{i\in\mathbf{X}_R}\partial_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right))^2}{\sum_{i\in\mathbf{X}_R}\partial^2_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right) + \lambda} \\ -\dfrac{(\sum_{i\in\mathbf{X}}\partial_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right))^2}{\sum_{i\in\mathbf{X}}\partial^2_{\overline{y}_i^{(K-1)}} l\left(y_i, \overline{y}_i^{(K-1)}\right) + \lambda}\end{array}\right] - \gamma , \tag{9}$$

where $\mathbf{X} = \mathbf{X}_L \bigcup \mathbf{X}_R$, $\mathbf{X}_L$, and $\mathbf{X}_R$ indicate the sample sets of the left and right nodes after splitting, respectively. One advantage of XGBoost over the other algorithms is that it is not affected by multicollinearity (Liu et al., 2019). Thus, although the two variables yielded the same results in the system, they should be retained. This is necessary for the vehicle lane-change decision because there may be a linear relationship between the factors influencing the decision (e.g., there may be a positive relationship between relative distance and relative velocity).

Based on the concept of integrated learning, XGBoost synthesizes the decision classification results of a series of classification and regression trees (CARTs) with deficiencies in learning ability. After all iterations, these classification results are input into the model with specific weights, *i.e.,* the base learner iterates until the training is complete or the model residuals are below a predefined threshold. The specific process of model construction is illustrated in Fig. 4.

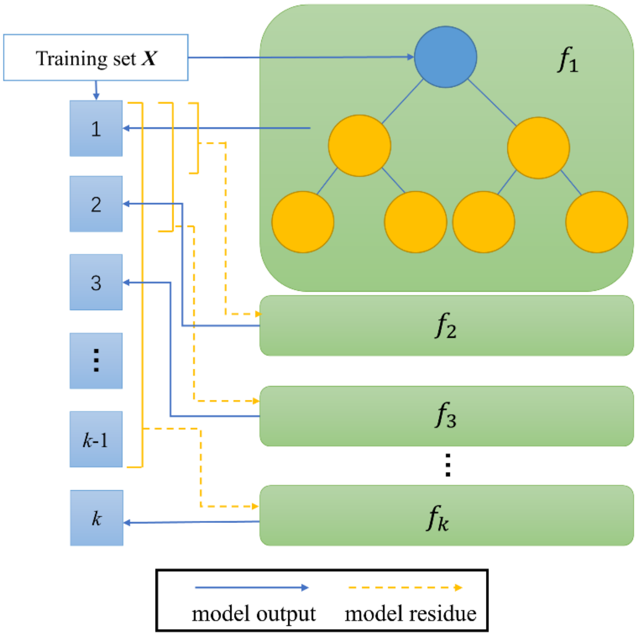


Fig. 4. XGBoost model construction diagram.

Finally, according to Equation (2), the data after processing with the fields listed in Table 9 were selected as the model input features. The LCD results extracted by the method described in Section 4.1 are chosen as the output. The input and output can then be input into the XGBoost model for training purposes.

Table 3. Description of XGBoost hyperparameters

Hyperparameter	Description
n_estimators (K)	Number of base learners (early stop iteration)
learning rate (ε)	Learning rate (eta)
max_depth	Maximum tree depth
min_child_weight	Sum of the minimum leaf sample weights
gamma (γ)	Node split gain threshold
alpha	L1 regularization parameter
lambda (λ)	L2 regularization parameter
subsample	Sampling ratio

colsample	Column sampling ratio
seed	Number of random seeds

XGBoost has several hyperparameters that can be tuned and optimized (as summarized in Table 3), such as the learning rate, maximum depth, and minimum child weight. The hyperparameters of trees can control model complexity and prevent model overfitting efficiently, for example K , max_depth, and min_child_weight. The hyperparameter K represents the number of base learners in XGBoost, *i.e.*, the number of trees built. The larger the value, the better the learning ability of the model. However, the model is more likely to be overfitted. The learning rate controlled the iteration rate. The larger the value, the faster the iteration speed. However, there is a risk that it will not converge to the true optimum and will become overfitted. An appropriate subsample enables the model to focus more on critical samples. The node split gain threshold (γ), L1 regularization parameter (alpha), and L2 regularization parameter (λ) were used to prevent overfitting.

4.3 Bayesian optimization

However, the objective function for the hyperparameter tuning of XGBoost is unknown. BO (Jones, 2001) is acknowledged as one of the most famous extensible applications of the Bayesian network because of the following listed advantages (Fakhrmoosavi et al., 2022; Liu et al., 2019; Liu et al., 2022; Wu et al., 2020; Yin et al., 2022): (1) BO can find a better combination of hyperparameters in a small number of steps, and (2) BO is a gradient-free global optimization method (therefore, it is extremely suitable for problems where the gradient is inaccessible). Prior knowledge was used to approximate the posterior distribution of the unknown objective function, and the hyperparameter combination was then selected for subsequent sampling according to the distribution. Thus, in this study, BO was used to optimize the hyperparameters in XGBoost to derive an accurate and well-fitted model, resulting in the BO-XGB model.

The Bayesian optimization method consists of two principal components: the prior and acquisition functions (Frazier, 2018). This method uses the Gaussian process (Seeger, 2004) as the prior distribution function. The acquisition function combines the estimated value and error to determine the point in the definition domain that maximizes its function value as the next evaluation point. The acquisition function (Yin et al., 2022) actively selects the most “promising” sampling points for evaluation based on the

Gaussian process and efficiently uses complete historical information to improve search efficiency.

4.4 SHAP

A fundamental challenge in the application of AI in transportation is the interpretability of machine learning models, which can improve trust in the model's predictions (Ayoub et al., 2021a, 2021b). In this study, some limitations exist even though the constructed BO-XGB model can predict LCD accurately. For example, two major difficulties are the manner in wherein drivers typically make LCDs based on their surrounding environment and the method to be used to quantify the impact of each factor in a specific sample. Furthermore, analyzing the causes of the abnormal decision results of the model is a key issue. To address these issues, SHAP was introduced to explain the model. SHAP explains model decisions using the Shapley value in game theory (Shapley, 2016), which is the most popular method for improving the interpretability of machine-learning models and has been applied to transportation problems (Ali et al., 2022; Ayoub et al., 2022; Guimaraes et al., 2022; Kong et al., 2022; Oseni et al., 2022; Song et al., 2022; Yan et al., 2022; Zhou et al., 2022). SHAP values indicate the contribution of each input feature to the model decision process (Equation (10)). For XGBoost, the total number of feature splits, representing its contribution to the final prediction, quantifies the importance of each input feature.

$$\phi_j(val) = \sum_{S \subseteq \{1,\cdots,M\} \backslash \{j\}} \frac{|S|!(M-|S|-1)!}{M!} \left(val\left(S \cup \{j\}\right) - val\left(S\right) \right), \tag{10}$$

$$F_j(val) = \frac{1}{n} \sum_{i=1}^{n} \left| \phi_j^{(i)}(val) \right|, \tag{11}$$

$$f(x_i) = E(f(x)) + \sum_{j=1}^{M} \phi_j^{(i)}(val), \tag{12}$$

where $\phi_j(val)$ represents the Shapley value of feature j of a sample under the condition that the value function is val; M is the number of the input features; S is a subset of the features used in this model; $|S|$ denotes the number of elements in subset S; $val(S)$ is the contribution of combination S in predicting lane change decision in this study; $F_j(val)$ means the global Shapley value of feature j in the sample set; n represents the number of the sample set, and i denotes the sample index; $f(x_i)$

represents the model's output as the input is x_i; and $E(f(x))$ is the mean prediction value (base value) in the whole dataset.

To interpret the prediction of the BO-XGB model, SHAP was utilized in two ways: *i.e.,* a global explanation (Equation (11)) and a sample explanation (Equation (10)). In the global explanation, the importance ranking of the input factors and interaction effects between the input factors on the output variable can be obtained (the interaction SHAP value can be calculated using Equation (13)). Furthermore, the sample explanation clarifies the individual predictions by showing the contribution of each factor to the output in a specific sample.

$$\begin{gathered}\phi_{i,j}(val)=\sum\nolimits_{S\subseteq\{i,j\}}\frac{|S|!(M-|S|-2)!}{M!}\times \\ \left[val\left(S\cup\{i,j\}\right)-val\left(S\cup\{i\}\right)-val\left(S\cup\{j\}\right)+val(S)\right]\end{gathered}, \tag{13}$$

In particular, for complex models with many variables, SHAP is more suitable than other popular methods such as sensitivity analysis for analyzing the influence of input variables on output variables. SHAP can provide a comprehensive analysis and explanation of the contribution of all input features, from both a macroscopic perspective (using the entire dataset) and a microscopic perspective (analyzing different features in a single sample). Inspired by this research (Simon Zhou et al., 2022), we also want to use SHAP to improve the interpretability of the lane-change model from both macroscopic and microscopic perspectives.

5 Case Study

5.1 Data description

The dataset used in this study is derived from the highD trajectory dataset. According to the Institute of Automotive Engineering at RWTH Aachen University, Germany, the extraction method for this dataset (Fig. 5) is an excellent technique (Krajewski et al., 2018) for measuring vehicle data from an aerial perspective. The dataset has 11.5 h of recording acquired across 45,000 Km and 110,000 vehicle track data points from six different sections of German freeways 4, 41, and 61. Using sophisticated computer vision algorithms, researchers can localize a dataset with a typical error of less than 10 cm. In this study, we first classified vehicles in the highD dataset into three classes (lane-keeping, left lane change, and right lane change) using the exact LCD extraction method. For the lane-keeping vehicle, the driving environment in the first frame was considered a group of lane-keeping data. As for the left (right) lane change vehicle, the exact LCD point is selected as a group of the left (right) LCD data (*e.g.*, “lane change point 3” of vehicle 60 in Fig. 2(a), is a group of LCD data). There were 97,184 groups of lane-keeping data, 7,213 groups of right lane-change data, and 6,119 groups of left lane-change data in the highD dataset, resulting in imbalanced sampling. To address this issue of imbalanced classification affecting the fitting of the BO-XGB model, a Synthetic Minority Oversampling Technique (SMOTE) is applied to resample the lane-change data (Chen et al., 2019). Following this resampling process, there were 97,184 groups of lane-keeping data, left-lane-change data, and right-lane-change data for processing and model learning.

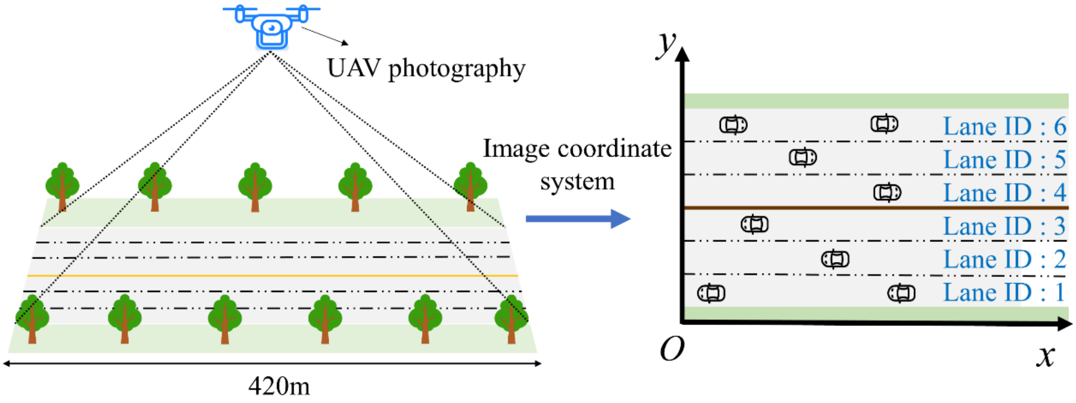


Fig. 5. Illustration of an example image coordinate system.

5.2 Lane change decision point extraction

Using the method described in Section 4.1, we extracted the LCD points for the highD trajectory dataset and obtained the green points shown in Fig. 6, which are the results of one of the highD dataset files named 01_tracks. The example figure demonstrates the extraction of a two-lane scenario; our data extraction and model training processes include both two- and three-lane scenarios. In Fig. 6, the red points obtained using the lane *id* shift methods are the vehicle trajectory intersection points and lane lines in the image coordinate system. These red points are mostly the points in the intermediate process of lane-change after the driver decides to change lanes and are not the initial points where the driver decides to change lanes. The green dots are the exact LCD points extracted by our proposed method, considering vehicle avoidance behavior and lateral velocity. Owing to data recording limitations, many vehicles tended to change lanes at the start of their trajectory, as recorded by the UAV, resulting in numerous green data points being clustered at the beginning of the recording period.

The difference in the extracted decision points between the two methods is shown in Fig. 6. The average time between the LCD points extracted in some previous study (i.e., points at which the subject vehicle presses over the lane line, represented by red dots) (Ali et al., 2023) and the LCD points extracted in this study (represented by green dots) in Fig. 6 is 8.9s.

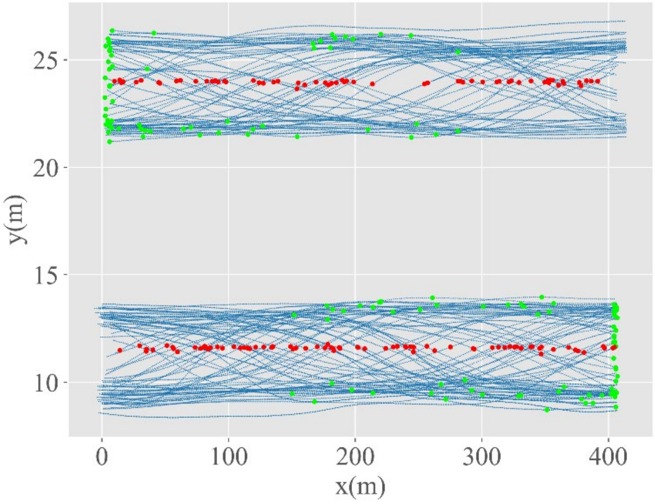


Fig. 6. LCD point extraction results in one of the highD dataset files named 01_tracks (red dots represent the points obtained by the lane id shift method, and green dots denote the points extracted by our proposed method).

5.3 Model optimization and performance

The typical process of model construction using highD datasets is as follows: Based on the proposed LCD point extraction method, 97,184 groups of lane-keeping data, 97,184 groups of right-lane-change data, and 97,184 groups of left-lane-change data were selected. 75% of these data were randomly input as training data for the XGBoost model, and the remaining 25% were used as the test dataset. The hyperparameters of the XGBoost-based LCD model were auto-tuned using BO.

The detailed optimization process is shown in Figs. 7–9. The LCD prediction model was trained and optimized based on the exact extracted LCD points. The modeling of LCD prediction is improved in two ways: model training with more insightful data and hyperparameters auto-tuned by BO.

Herein, the objective function was to minimize the opposite of the classification accuracy of the XGBoost model using specific hyperparameters via 5-fold stratified cross-validation. One hundred iterations were selected. The optimization process for each step is shown in Fig. 7, and it can be observed that the eventual accuracy can reach

99.14%. The specific points at which the BO evaluated the objective are shown in Fig. 8. The histograms display the sample distribution of each hyperparameter. The two-dimensional scatter plots demonstrate the order of the evaluated points. The figure illustrates the development of the search process. The order in which the points were evaluated was encoded by the color of each point. Darker colors (purple) correspond to sampling points for the earlier search process, and lighter colors (yellow) correspond to sampling points for the later search process. Red pentagons indicate the positions of the optimal values determined by the optimization process. The partial dependency figure (Fig. 9) shows an input variable value (model hyperparameter) mapping to the objective function after averaging all other variables. Eventually, several critical hyperparameters that affect the XGBoost model were obtained, including K , ε , max_depth, γ , subsamples, and colsample_bytree. The optimal parameter settings obtained using this method are listed in Table 4.

Table 4. Optimized XGBoost hyperparameters with BO

Hyperparameter	Values	Hyperparameter	Values
n_iterations (K)	165	alpha	0.3096
learning rate (ε)	0.0918	lambda (λ)	0.2465
max_depth	27	subsample	0.4633
min_child_weight	5	colsample	0.2203
gamma (γ)	0.6105	seed	666

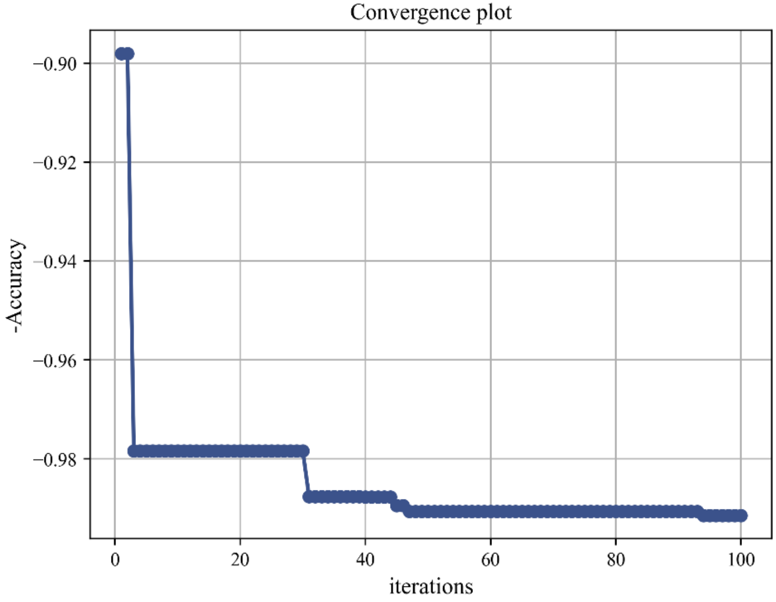


Fig. 7. Convergence plot of BO process for 100 iterations.

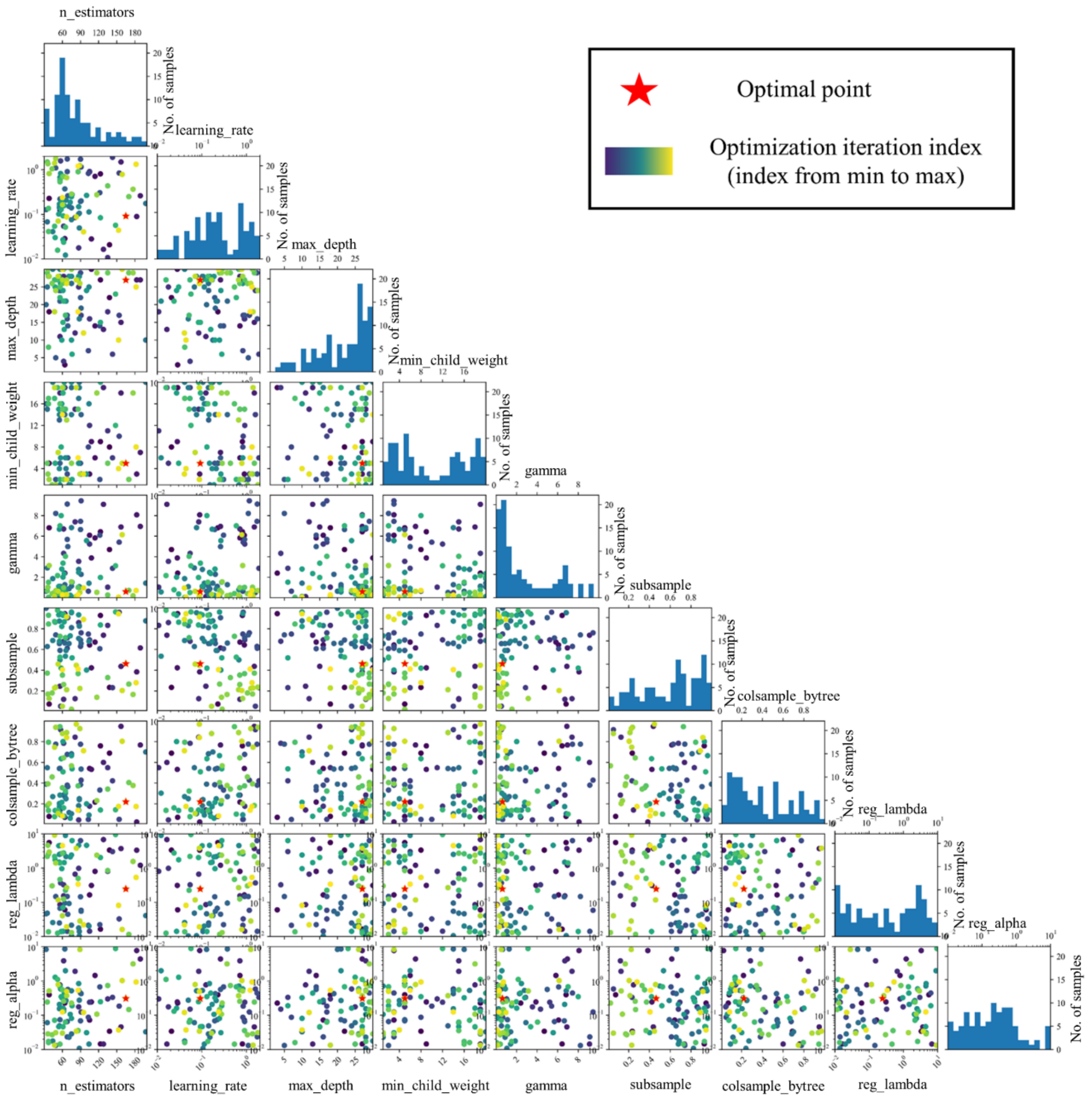


Fig. 8. Sampling point order diagram for Bayesian optimization process.

The performance of the model can be observed in Figs. 10 and 11, and Tables 5 and 6. Fig. 10 shows that the model with BO eventually fits the decision-making behavior satisfactorily and does not overfit. In addition, Figs. 7 and 10 indicate that for LCD in the test set with the hyperparameters optimized by the BO method, a prediction accuracy of 99.14% can be achieved. The detailed results of the BO-XGB model for LCD on the highD dataset are shown in Fig. 11. In the right lane-change, keep straight, and left lane-change test data of the highD dataset, 24,039, 23,741, and 24,482 groups of vehicles made correct decisions with accuracies of 99.98%, 97.45%, and 100.00%, respectively. In conclusion, after training and hyperparameter tuning, this model exhibited excellent performance, with an accuracy of 99.14%.

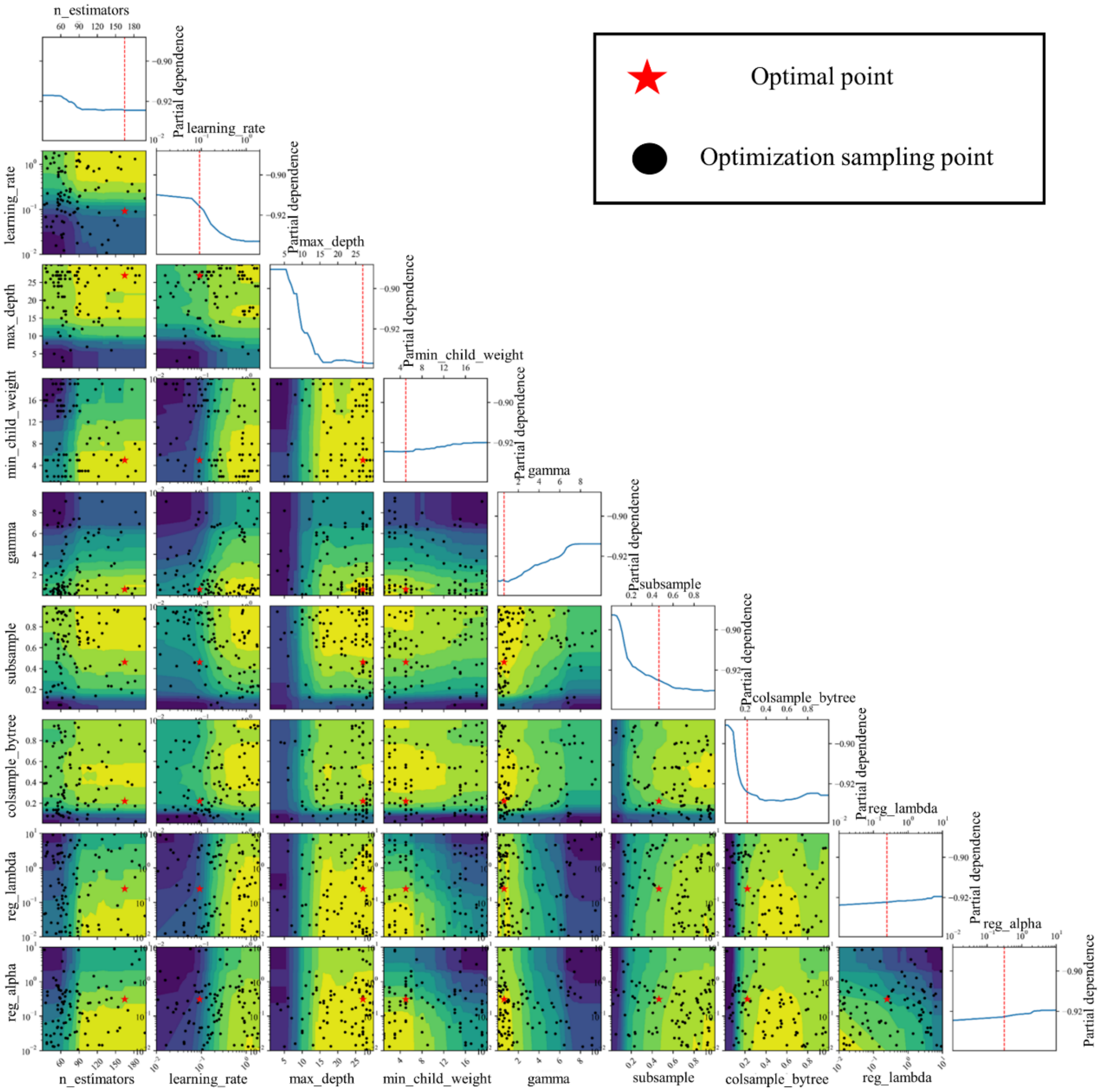


Fig. 9. Hyperparameters auto-tuning and corresponding performance.

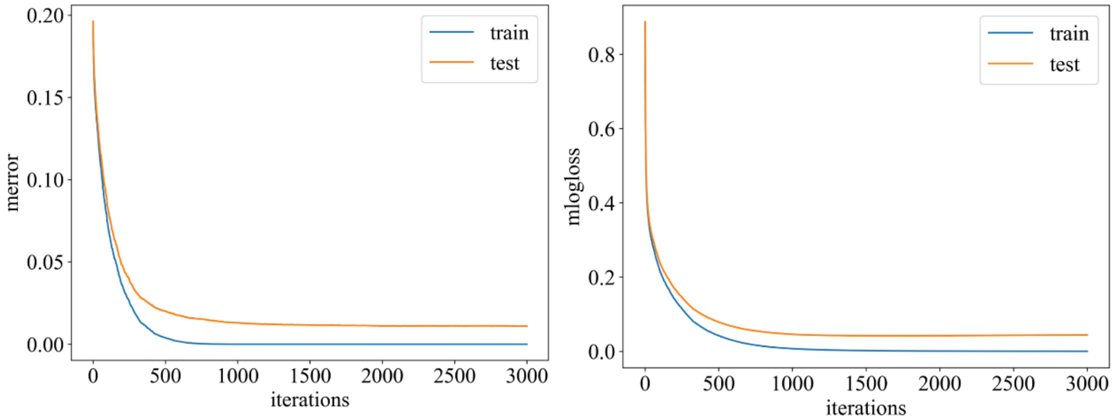


Fig. 10. Learning curve of the BO-XGB model based on the classification error rate and log loss (merror denotes the error rate of this multiclassification problem and mlogloss represents the multiclass log loss in model metrics).

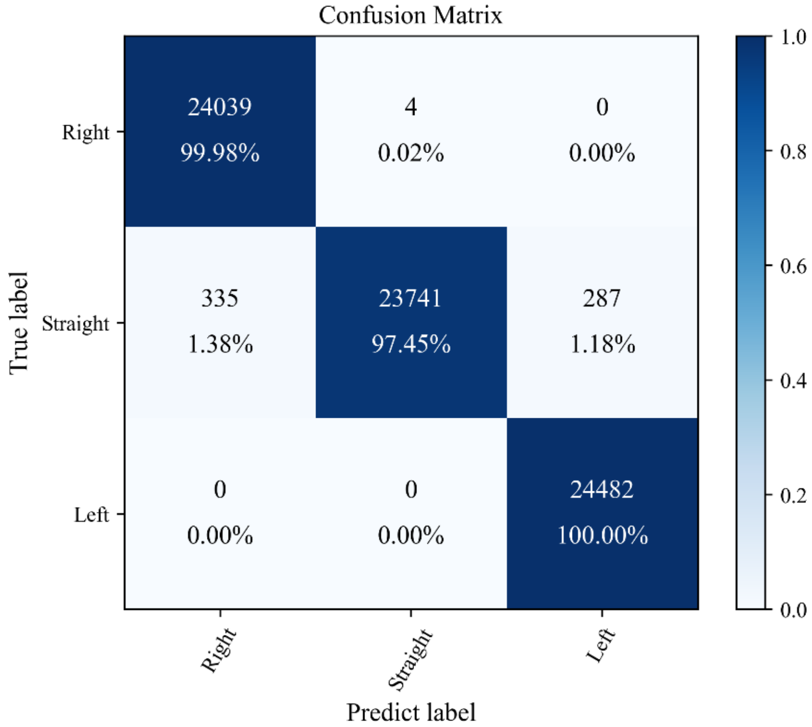


Fig. 11. Lane change decision results on the highD dataset.

Table 5. Evaluation metrics of the proposed method on the highD dataset

Evaluation metrics	Precision	Recall	F1 score	Accuracy
Right lane change	0.9863	0.9998	0.9930	0.9914
Keep straight	0.9998	0.9745	0.9869	
Left lane change	0.9884	1.0000	0.9942	

Table 6. Performance of different methods on the highD dataset

Algorithms	Left lane change		Right lane change		Lane keeping		Accuracy	Computation time (s)
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC		
SVM	0.432	0.573	0.588	0.706	0.746	0.780	0.591	87.357
Random forest	0.600	0.755	0.805	0.811	0.892	0.772	0.756	79.825
AdaBoost	0.588	0.688	0.724	0.779	0.802	0.856	0.711	65.923
Gradient boost	0.941	0.950	0.931	0.950	0.932	0.955	0.935	417.312
XGBoost	1.000	0.997	0.999	0.996	0.975	0.987	0.991	66.837

Notes: Area under curve (AUC) is defined as the area enclosed by the coordinate axis under the receiver operating characteristic curve.

The performance of various machine learning techniques for LCD was investigated using the same training and test data. Accuracy was chosen as the evaluation metric. Table 6 lists the results of the test sets for SVM, random forest (RF), adaptive boosting (AdaBoost), gradient boosting tree, and XGBoost with BO. Evidently, the SVM model performs much weaker than tree-based models, and XGBoost achieves the best performance, with an overall accuracy of 99.14%. Gradient boosting was the second-best method, whereas RF and AdaBoost had comparable decision abilities.

The computation times for all models with the same dataset and parameter optimization methods are listed in Table 6. Evidently, AdaBoost was the fastest on the LCD (65.923 s computation time), but its overall performance was not the best considering its accuracy. However, the gradient boost requires the longest time to train a model and only achieves the second-highest accuracy. In contrast, XGBoost, owing to its optimization design, only requires 66.837 s to find the model with the highest LCD decision accuracy. Owing to its powerful prediction ability with an accuracy of 99.14%, a small additional computation time is acceptable.

5.4 Model explanation of XGBoost-based LCD model based on SHAP

5.4.1 Global explanation

SHAP was used to explain the BO-XGB model for LCD. To acknowledge the importance of these factors in the LCD model, we examined the SHAP summary plot shown in Fig. 12(a). The SHAP value in Fig. 12(a) denotes the factors contributing to LCD prediction. The variables are ranked according to their global feature importance. The classes “right,” “left,” and “straight” indicate that the vehicle changes lanes to the right, left, or stays straight ahead, respectively. The first and third most important features of the BO-XGB model, “precedingXVelocity” and “precedingD” respectively, are shown in this figure as the relative velocity and distance between the subject vehicle and its preceding vehicle. The velocity and distance between the subject vehicle and the preceding vehicle on its target lane, represented by “rightPrecedingD” and “rightPrecedingV” or “leftPrecedingV” and “leftPrecedingD” (depending on whether its target lane is right or left), are crucial in the LCD model.

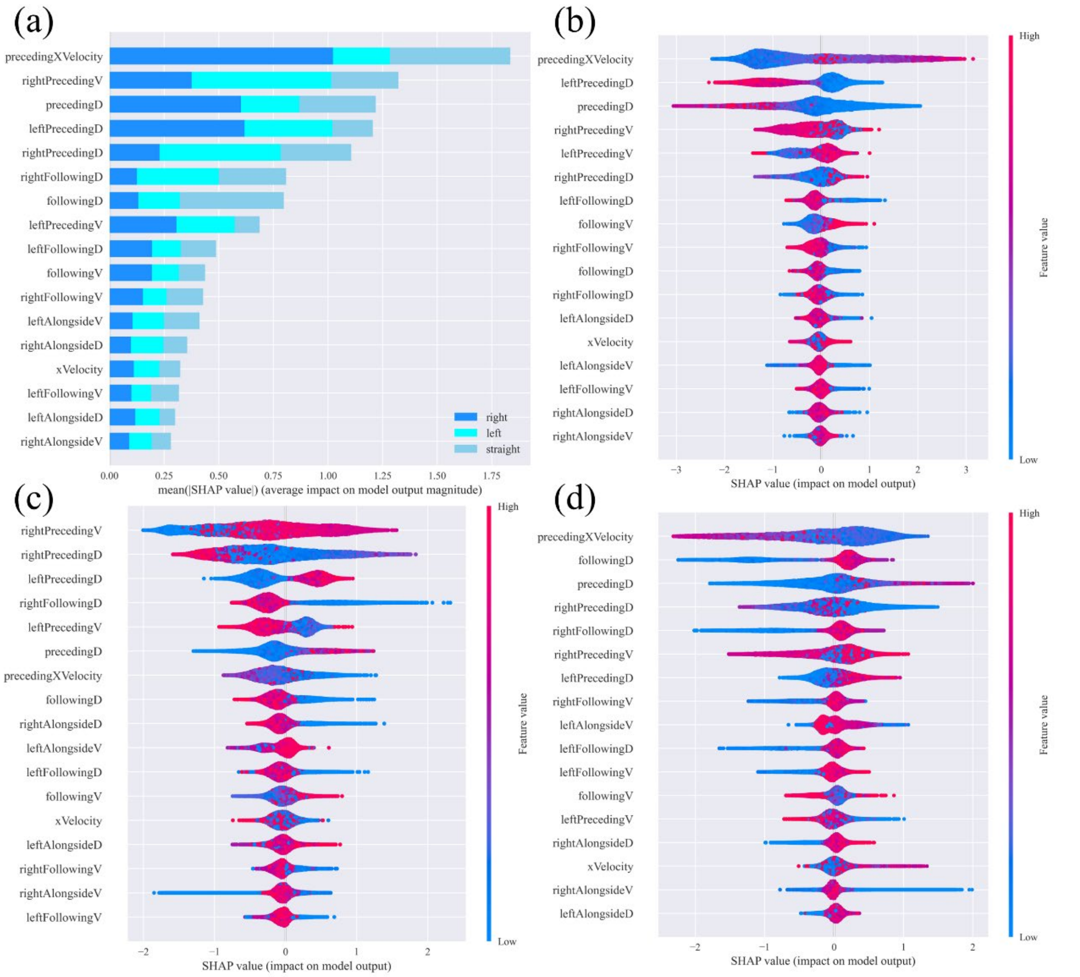
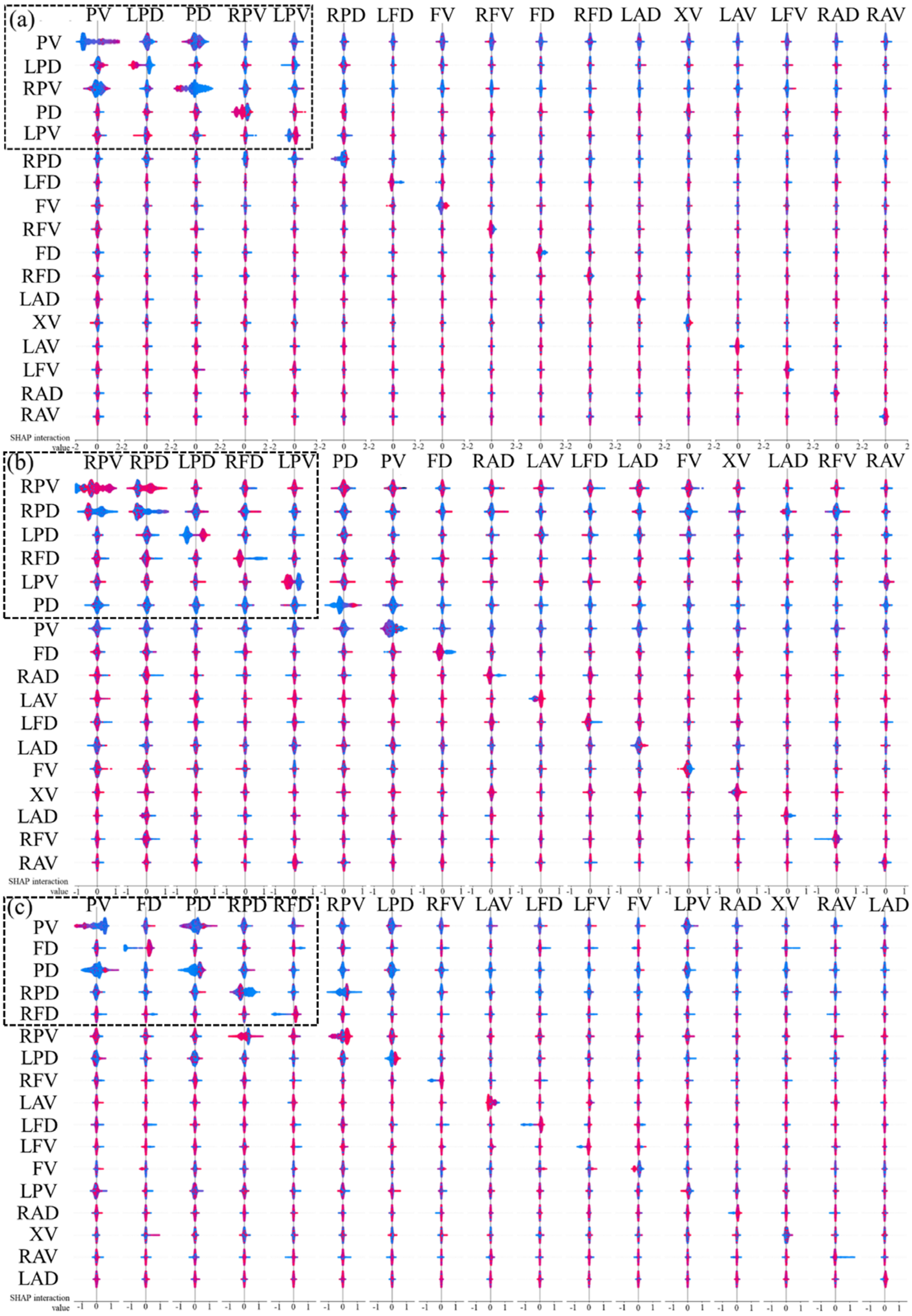


Fig. 12. SHAP analysis of BO-XGB model: (a) Summary plot, (b) right lane change vehicles, (c) left lane change vehicles, (d) straight keeping vehicles (Fig. 12(b), (c), and (d) can be understood from the following: (1) the vertical positions of these input features indicate their feature importance, (2) the horizontal position of one dot represents its influence on LCD in this instance, (3) the color of one dot denotes the value of that input feature ranging from high (red) to low (blue), and (4) the density of dots expresses the distribution of inputs in the dataset).

Figs. 12(b), (c), and (d) show the contribution of each feature to the LCD model in the classes "right," "left," and "straight," respectively. Each dot in these figures represents a SHAP value for its instance of one input feature. Based on the information in Fig. 12, we can observe a rough correlation between the input features and the output decisions. There is no more important feature for a right-lane-change vehicle than the velocity difference from its preceding vehicle. The faster it is than the preceding vehicle, the stronger its desire to change lanes. Next are the relative distances between it and the left-preceding vehicle, and between it and the preceding vehicle. The smaller these distances, the greater the probability that the subject vehicle will change lanes to the right. The fourth important feature is the relative velocities of the right and left preceding vehicles. The greater the speed of the right-preceding vehicle compared to the subject vehicle and the smaller the speed difference between the left-preceding vehicle and the subject vehicle, the greater the possibility of the subject vehicle shifting to the right. For left-lane-change vehicles, a smaller right-preceding vehicle velocity and relative distance and a larger left-side lane-change gap are essential factors for the subject vehicle to switch left. The velocity of a straight vehicle is far lower than its preceding vehicle's, and larger distances between it and its preceding and following vehicles have a larger impact on its decision to stay straight.

To investigate the effects of the different features and their combinations on the decision output, the SHAP interaction values for the different features were examined, as shown in Fig. 13. Figs. 13(a), (b), and (c) correspond to Figs. 12(b), (c), and (d), which indicate they relate to the classes "right," "left," and "straight," respectively. Taking Fig. 13(a) (class "right") as an example, the figure reveals the contribution of the interaction combination between features to the model output. The graphs in the dashed box in Fig. 13 were enlarged in Fig. 14 to show the five most influential features results of Fig. 13. The subgraphs in Fig. 14 on the diagonal reveal the primary influence of the feature, whereas the subgraphs on the non-diagonal line indicate the two-by-two interaction of the feature with other features. Each dot represents a sample. The redder (bluer) dot represents the larger (smaller) feature value for this sample. The lateral position of the dot indicates its effect on the output. We focused on diagonal subplots to reveal the influence of a single feature on the predicted output. As shown in Fig. 14, features such as the velocity difference between the subject vehicle and its preceding vehicle, as well as the velocity difference between it and its left-preceding vehicle, can positively affect the model's decision to predict a left LCD. In contrast, the distance difference between the subject vehicle and its left-preceding vehicle, as well as the distance difference between it and its preceding vehicle can harm the model's prediction of a left LCD. In Figs. 13(a), (b), and (c), the sum of all the SHAP interaction values for a given feature row is the SHAP value for that feature, as shown in Figs. 12(b), (c), and (d).



RPV: rightPrecedingV
LPV: leftPrecedingV
RPD: rightPrecedingD
LPD: leftPrecedingD
XV:xVelocity

PV: precedingV
FV: followingV
PD: precedingD
FD: followingD

RAV: rightAlongsideV
LAV: leftAlongsideV
RAD: rightAlongsideD
LAD: leftAlongsideD

RFV: rightFollowingV
LFV: leftFollowingV
RFV: rightFollowingD
LFD: leftFollowingD

Fig. 13. SHAP interaction value analysis of features in BO-XGBoost model for the LCD: (a) right lane change vehicles, (b) left lane change vehicles, (c) straight keeping vehicles. (These figures are similar in form and principle to Fig. 12.)

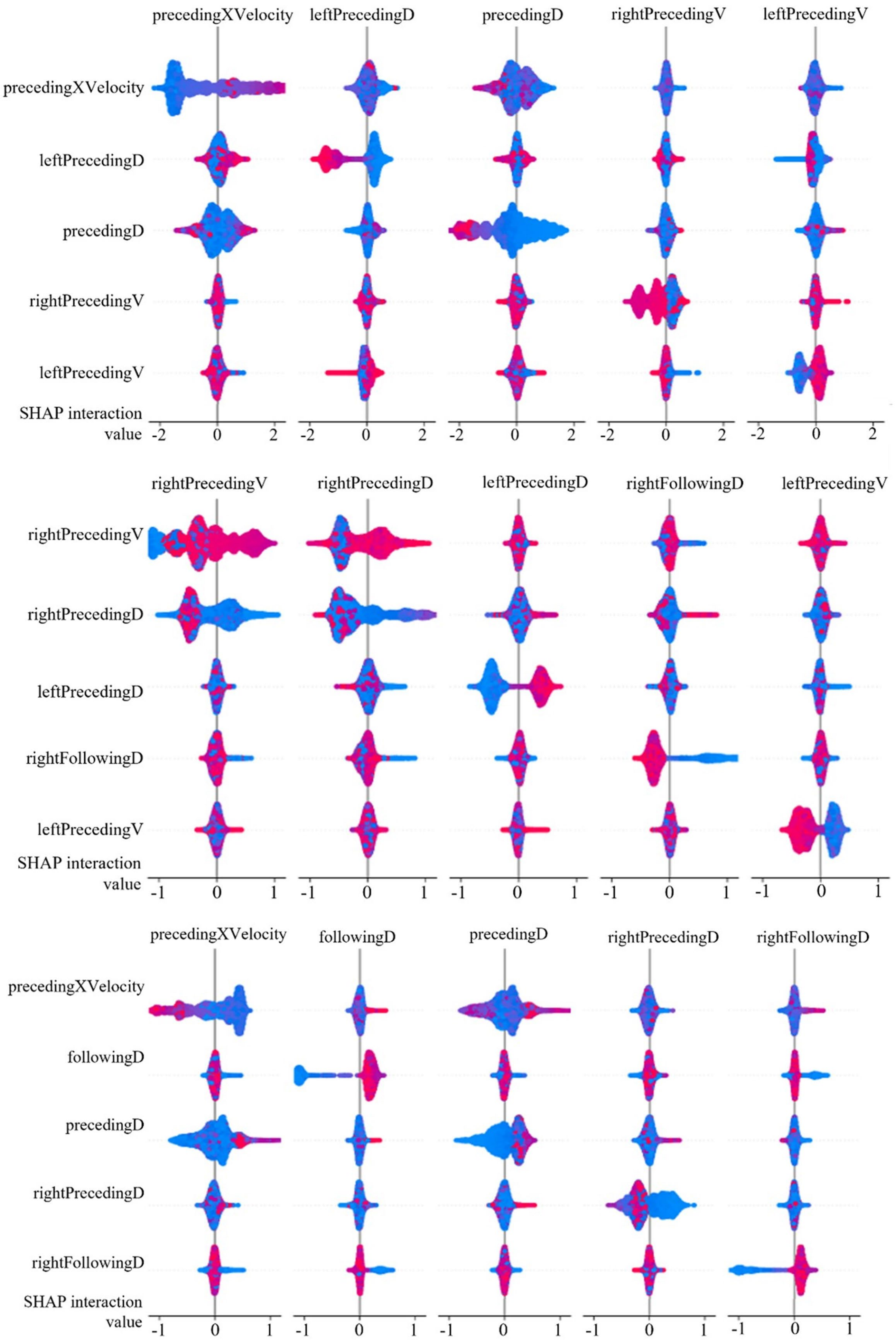


Fig. 14. Dashed box in Fig. 13.

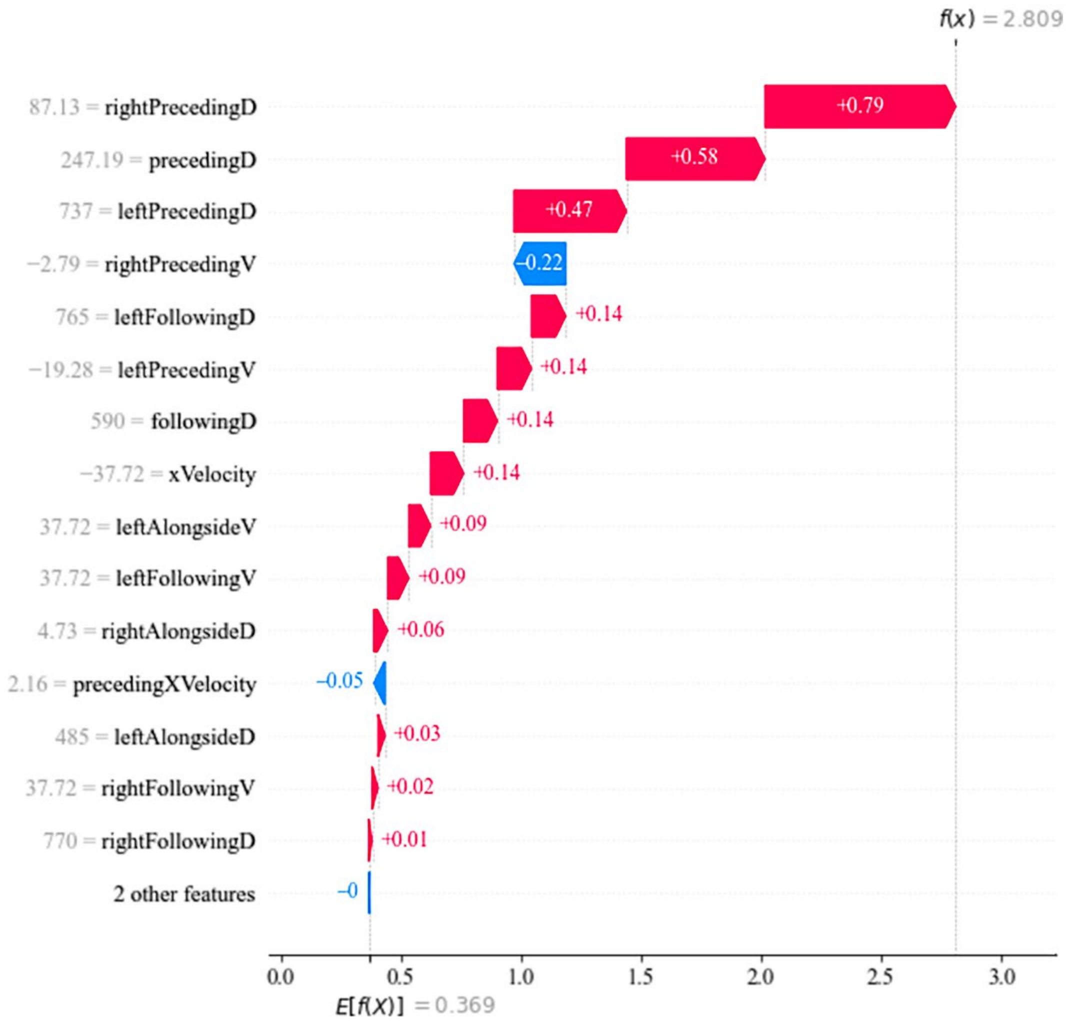


Fig. 15. Model prediction explanation for a specific sample (sample index is 370).

5.4.2 Sample explanation

Fig. 15 shows the BO-XGB model prediction for LCD with a specific sample by SHAP (here, the sample index is 370). In this figure, the horizontal and vertical axes represent the SHAP values and features, respectively. Blue indicates that the feature weakens the prediction (arrow to the left, SHAP value decreases), and red indicates that the feature positively enhances the prediction (arrow over the right, SHAP value increases). $E[f(x)]$ represents the baseline value of SHAP (i.e., the base value in Equation (12)), which is the mean value of the model prediction as well. In this sample, the relative distance (87.13 m) between the subject vehicle and the right-preceding vehicle contributed the maximum to the LCD prediction results (0.79). The second highest positive contribution variable was the relative distance (247.19 m) between the subject vehicle and its preceding vehicle (0.58 m). Using the above method, the specific feature contribution to each LCD was obtained using the BO-XGB model. Based on this, we can understand the model's predictions and practically assess the validity of the predicted results.

5.5 Additional test

To validate the effectiveness of our experimental results, we further tested the proposed method on the CQSkyEyeX dataset (http://www.cqskyeyex.com/index.html) in Chongqing, China. The CQSkyEyeX dataset comprises high-resolution highway traffic operation videos captured by drones, employing advanced computer vision techniques to extract vehicle trajectories. It overcomes challenges such as image shaking, vehicle misclassification, and missing road information (Zhang et al., 2023). This dataset includes 650 minutes of measurement data from eight locations on Chinese highways. After data preprocessing and class balancing, the data of location 3, as shown in Fig. 16, in this dataset were input into the model for testing, with the results depicted in Fig. 17. As shown in Fig. 17, in the right lane-change, keep straight, and left lane-change test data of the CQSkyEyeX dataset, 2,660, 2,660, and 2,660 groups of vehicles made correct decisions with accuracies of 99.85%, 98.50%, and 100.00%, respectively. Similar to Figs. 7-9 and 12, the convergence process, the sampling point order diagram, the hyperparameters auto-tuning and corresponding performance, and the SHAP analysis of BO-XGB model on the CQSkyEyeX dataset are summarized in Figs. 18-21. The results show that our method can also achieve superior performance for the CQSkyEyeX dataset.

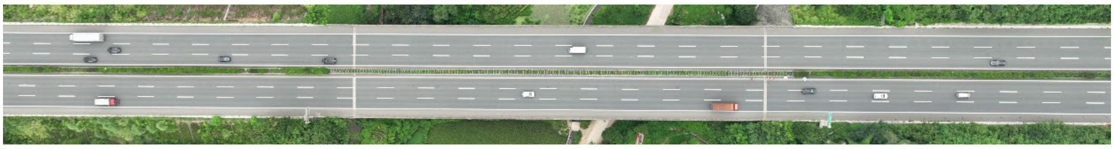


Fig. 16. Top view of location 3

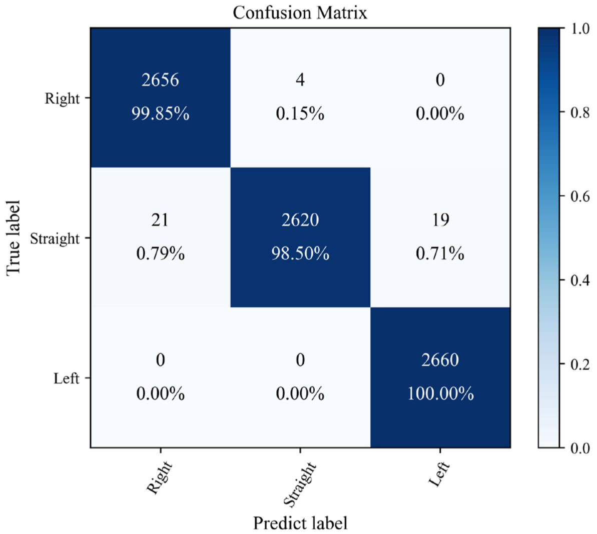


Fig. 17. Lane change decision results on the CQSkyEyeX dataset.

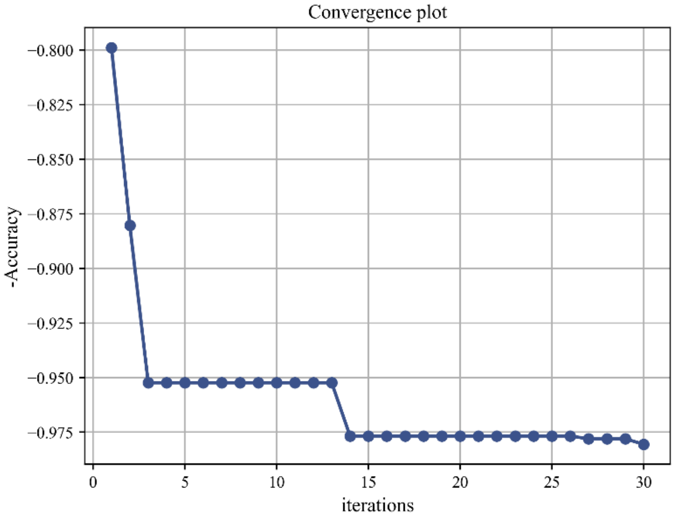


Fig. 18. Convergence plot of BO process for 30 iterations on the CQSkyEyeX dataset.

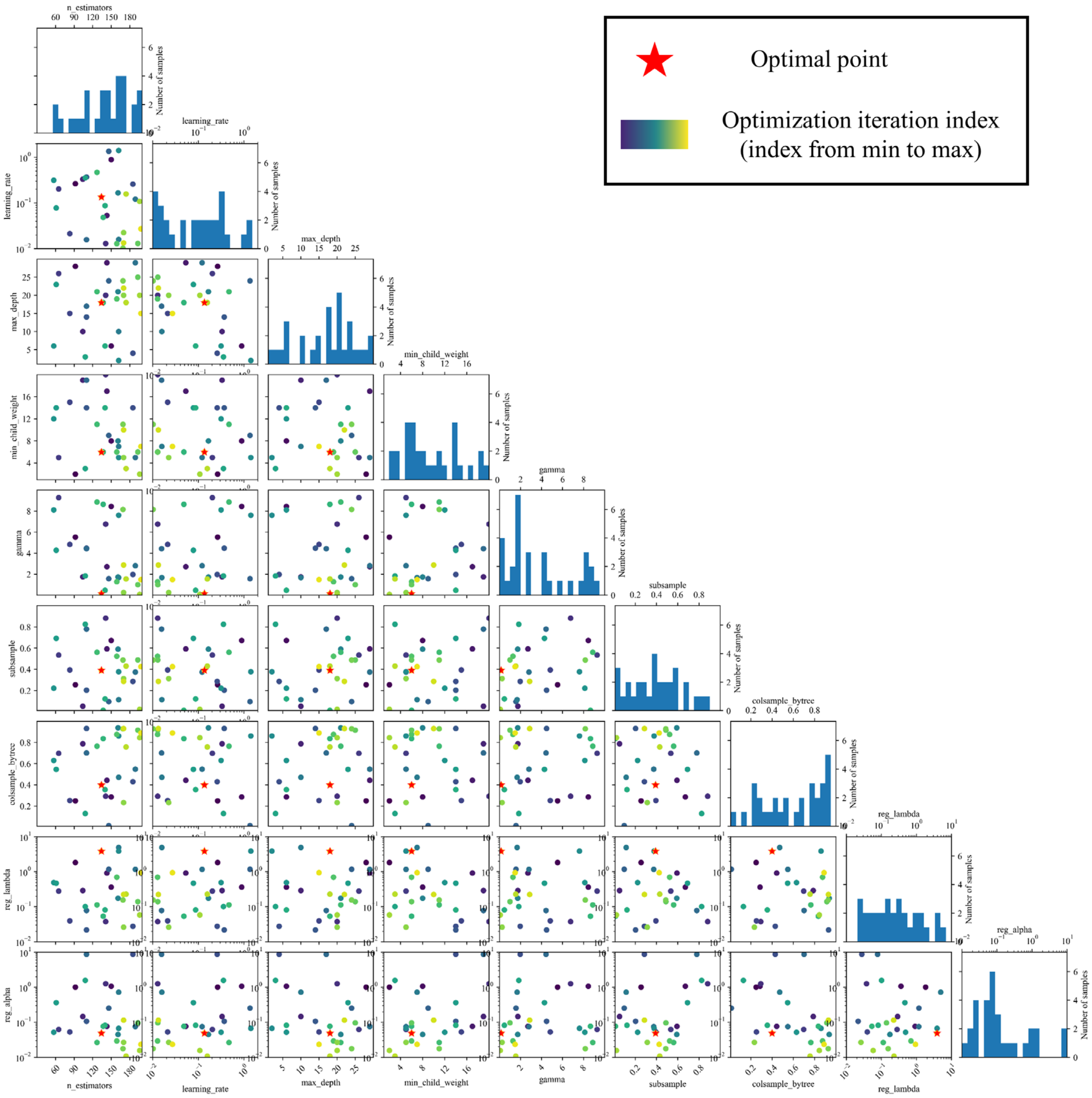


Fig. 19. Sampling point order diagram for Bayesian optimization process on the CQSkyEyeX dataset.

6 Conclusion

In this study, a BO-XGB model for LCD prediction using an exact LCD point extraction method and the SHAP method was proposed to assist drivers in making accurate LCDs. XGBoost was the core model used to learn the relationship between input and output. BO was used to tune the various hyperparameters of XGBoost. The exact LCD point extraction method was used to extract the precise LCD point by considering the changes in the vehicle's velocity and the trajectory trend. If only the change in vehicle velocity is considered, ignoring some avoidance behaviors, the extraction of the initial LCD points may extract some delayed LCD points. In this study, the precise input of LCDs and hyperparameter optimization is probably the reasons for the high accuracy of the model (the proposed model achieved an accuracy of 99.14% and 99.45% when it was applied to LCD samples extracted from the highD dataset and CQSkyEyeX dataset, respectively).

Furthermore, the SHAP value method was used to explain the model prediction results for the whole sample set and a specific sample, respectively. It improved the model's interpretability, validated the model's credibility, and enhanced our understanding of the prediction results. A corresponding visual analysis provided extra

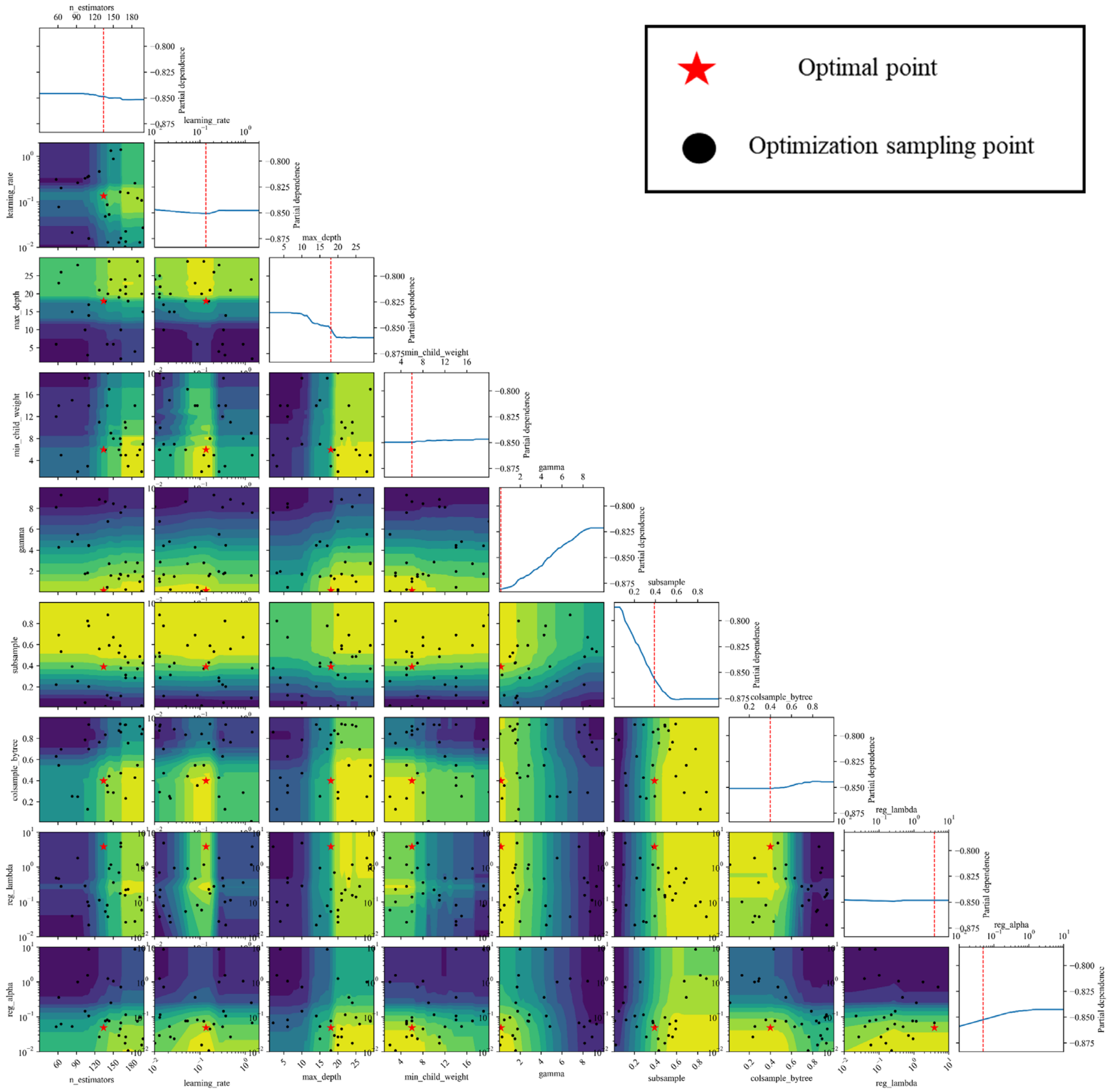


Fig. 20. Hyperparameters auto-tuning and corresponding performance on the CQSkyEyeX dataset.

information regarding the contribution of the features. In the highD dataset, the model explanation results show that vehicles in front of a subject vehicle have a greater influence on the LCD than vehicles behind or on the side. In addition, we can quantitatively determine the contribution of the features to a specific sample. This indicates which features significantly affect the LCD output in this sample and provides a way to analyze the causes of anomalous samples.

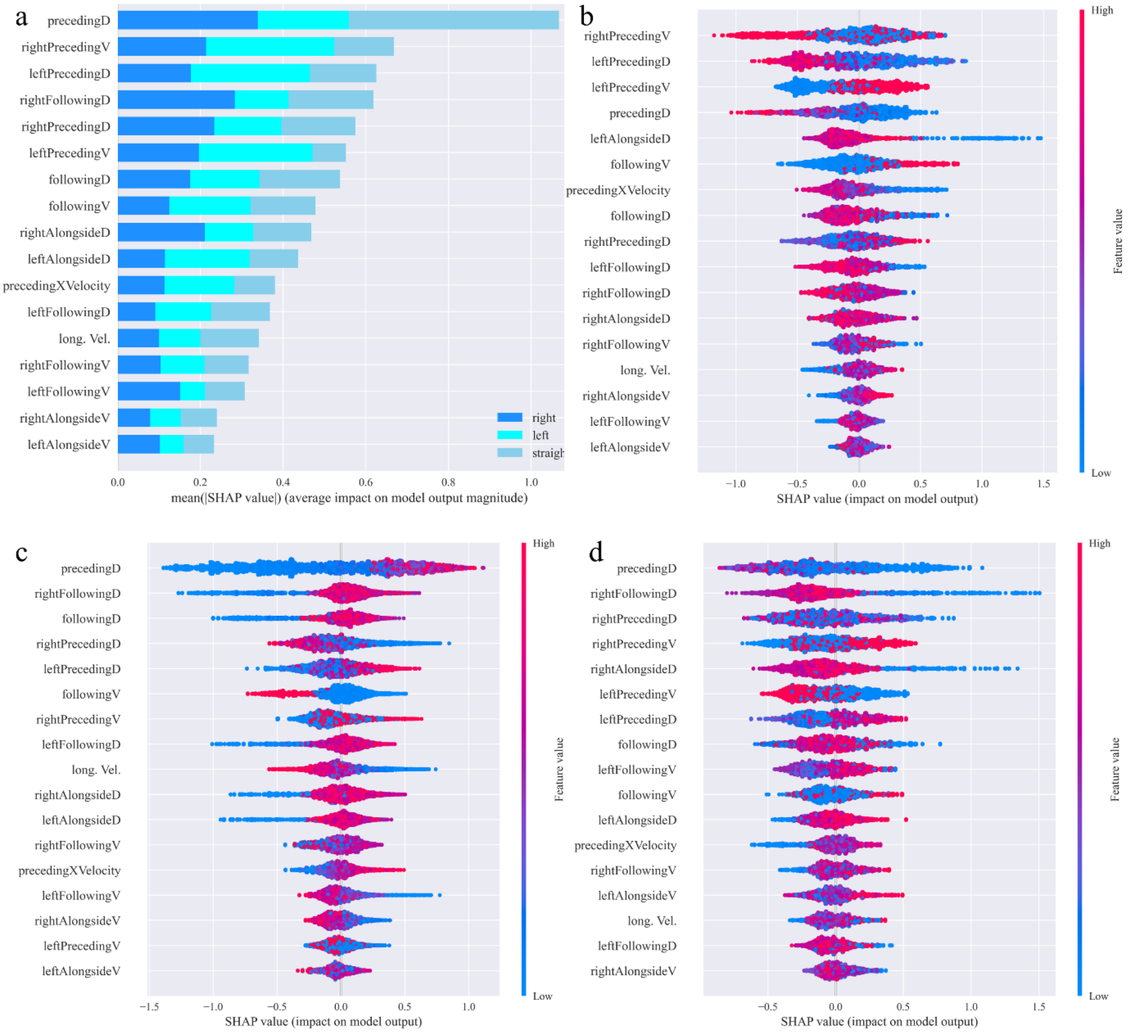


Fig. 21. SHAP analysis of BO-XGB model on the CQSkyEyeX dataset: (a) Summary plot, (b) right lane change vehicles, (c) left lane change vehicles, (d) straight keeping vehicles (Figs. 21(b), (c), and (d) can be understood from the following: (1) the vertical positions of these input features indicate their feature importance, (2) the horizontal position of one dot represents its influence on LCD in this instance, (3) the color of one dot denotes the value of that input feature ranging from high (red) to low (blue), and (4) the density of dots expresses the distribution of inputs in the dataset).

In summary, the contributions of this study can be summarized as follows:

- An accurate lane-changing decision point extraction method is designed to consider vehicle avoidance behavior, which has been neglected in existing studies.
- Bayesian parameter optimization is used to optimize multiple parameters of XGBoost model (BO-XGB).
- SHAP is used to analyze the principles of our proposed LCD decision model at both global and local levels. The method can be specifically refined to determine the effect of the features of a sample on the lane change decision.

However, the limitations of this study are as follows: First, specific traffic contexts may influence LCD, such as vehicle type, abnormal weather, and geometric road conditions. These factors can be included in our modeling framework when additional trajectory data are available. Second, the prediction of LCDs was insufficient for assisted driving. The prediction of lane-change trajectories and the analysis of lane-change risks have not been considered. Third, this model considers only the effect of vehicle-to-vehicle interactions on LCDs, ignoring the effect of the driver's features and habits. The accuracy and applicability of this model may be enhanced when used in conjunction with computer vision techniques for gathering and evaluating the driver's facial information. Further research in this area could be conducted.

Acknowledgment

This research is supported by the Key Programme of the National Natural Science Foundation of China (No. 52131203), the Natural Science Foundation of Jiangsu Province (No. BK20232019), the Jiangsu Provincial Scientific Research Center of Applied Mathematics (No. BK20233002) and the SEU Innovation Capability Enhancement Plan for Doctoral Students (No. CXJH_SEU 24199).

References

Ali, Y., Hussain, F., Bliemer, M.C.J., Zheng, Z., Haque, M.M., 2022. Predicting and explaining lane-changing behaviour using machine learning: A comparative study. Transp. Res. Part C Emerg. Technol. 145, 103931.

Ali, Y., Zheng, Z., Bliemer, M.C.J., 2023. Calibrating lane-changing models: Two data-related issues and a general method to extract appropriate data. Transp. Res. Part C Emerg. Technol. 152.

Ali, Y., Zheng, Z., Mazharul Haque, M., Yildirimoglu, M., Washington, S., 2020. Detecting, analysing, and modelling failed lane-changing attempts in traditional and connected environments. Anal. Methods Accid. Res. 28.

Ashfaq, F., Ghoniem, R.M., Jhanjhi, N.Z., Khan, N.A., Algarni, A.D., 2023. Using

Dual Attention BiLSTM to Predict Vehicle Lane Changing Maneuvers on Highway Dataset. Systems 11.

Atagoziev, M., Güran Schmidt, E., Schmidt, K.W., 2023. Lane change scheduling for connected and autonomous vehicles. Transp. Res. Part C Emerg. Technol. 147.

Ayoub, J., Du, N., Yang, X.J., Zhou, F., 2022. Predicting Driver Takeover Time in Conditionally Automated Driving. IEEE Trans. Intell. Transp. Syst. 23, 9580–9589.

Ayoub, J., Yang, X.J., Zhou, F., 2021a. Combat COVID-19 infodemic using explainable natural language processing models. Inf. Process. Manag. 58, 102569.

Ayoub, J., Yang, X.J., Zhou, F., 2021b. Modeling dispositional and initial learned trust in automated vehicles with predictability and explainability. Transp. Res. Part F Traffic Psychol. Behav. 77, 102–116.

Ben-Akiva, M.E., Gao, S., Wei, Z., Wen, Y., 2012. A dynamic traffic assignment model for highly congested urban networks. Transp. Res. Part C Emerg. Technol. 24, 62–82.

Cai, Q., Abdel-Aty, M., Zheng, O., Wu, Y., 2022. Applying machine learning and google street view to explore effects of drivers' visual environment on traffic safety. Transp. Res. Part C Emerg. Technol. 135, 103541.

Chauhan, P., Kanagaraj, V., Asaithambi, G., 2022. Understanding the mechanism of lane changing process and dynamics using microscopic traffic data. Phys. A Stat. Mech. its Appl. 593, 126981.

Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794.

Chen, T., Shi, X., Wong, Y.D., 2019. Key feature selection and risk prediction for lane-changing behaviors based on vehicles' trajectory data. Accid. Anal. Prev. 129.

Cheng, Q., Lin, Y., Zhou, X. (Simon), Liu, Z., 2024a. Analytical formulation for explaining the variations in traffic states: A fundamental diagram modeling perspective with stochastic parameters. Eur. J. Oper. Res. 312.

Cheng, Q., Liu, Z., Lu, J., List, G., Liu, P., Zhou, X.S., 2024b. Using frequency domain analysis to elucidate travel time reliability along congested freeway corridors. Transp. Res. Part B Methodol. 184, 102961.

Das, A., Khan, M.N., Ahmed, M.M., 2020. Detecting lane change maneuvers using SHRP2 naturalistic driving data: A comparative study machine learning techniques. Accid. Anal. Prev. 142, 105578.

Fakhrmoosavi, F., Kamjoo, E., Kavianipour, M., Zockaie, A., Talebpour, A., Mittal, A., 2022. A stochastic framework using Bayesian optimization algorithm to assess the network-level societal impacts of connected and autonomous vehicles. Transp. Res. Part C Emerg. Technol. 139, 103663.

Fei, C., Wang, B., Zhuang, Y., Zhang, Z., Hao, J., Zhang, H., Ji, X., Liu, W., 2020. Triple-GAIL: A multi-modal imitation learning framework with generative adversarial nets, in: IJCAI International Joint Conference on Artificial Intelligence. pp. 2929–2935.

Frazier, P.I., 2018. A Tutorial on Bayesian Optimization.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F., 2012. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.

Gao, J., Murphey, Y.L., Yi, J., Zhu, H., 2020. A data-driven lane-changing behavior detection system based on sequence learning. Transp. B.

Gao, J., Yi, J., Murphey, Y.L., 2022. Joint learning of video images and physiological signals for lane-changing behavior prediction. Transp. A Transp. Sci. 18.

Gao, K., Li, X., Chen, B., Hu, L., Liu, J., Du, R., Li, Y., 2023. Dual Transformer Based Prediction for Lane Change Intentions and Trajectories in Mixed Traffic Environment. IEEE Trans. Intell. Transp. Syst. 24.

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

Gu, X., Han, Y., Yu, J., 2020. A novel lane-changing decision model for autonomous vehicles based on deep autoencoder network and XGBoost. IEEE Access 8, 9846–9863.

Guimaraes, M., Soares, C., Ventura, R., 2022. Decision Support Models for Predicting and Explaining Airport Passenger Connectivity From Data. IEEE Trans. Intell. Transp. Syst. 23, 16005–16015.

Guo, H., Keyvan-Ekbatani, M., Xie, K., 2022. Lane change detection and prediction using real-world connected vehicle data. Transp. Res. Part C Emerg. Technol. 142.

Guo, Y., Zhang, H., Wang, C., Sun, Q., Li, W., 2021. Driver lane change intention recognition in the connected environment. Phys. A Stat. Mech. its Appl. 575, 126057.

Halati, A., Lieu, H., Walker, S., 1997. CORSIM- corridor traffic simulation model, in: Proceedings of the Conference on Traffic Congestion and Traffic Safety in the 21st Century. pp. 570–576.

He, L., Yu, B., Chen, Y., Bao, S., Gao, K., Kong, Y., 2023. An interpretable prediction model of illegal running into the opposite lane on curve sections of two-lane rural roads from drivers' visual perceptions. Accid. Anal. Prev. 186.

He, X., Yang, H., Hu, Z., Lv, C., 2023. Robust Lane Change Decision Making for Autonomous Vehicles: An Observation Adversarial Reinforcement Learning Approach. IEEE Trans. Intell. Veh. 8.

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

Hornberger, P., Cramer, S., Lange, A., 2018. Evaluation of Driver Input Variations for Partially Automated Lane Changes, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. pp. 1023–1028.

Hou, Y., Edara, P., Sun, C., 2015. Situation assessment and decision making for lane change assistance using ensemble learning methods. Expert Syst. Appl. 42.

Huang, H., Mao, J., Liu, R., Lu, W., Tang, T., Liu, L., 2024. MTLMetro: A Deep Multi-Task Learning Model for Metro Passenger Demands Prediction. IEEE Trans. Intell. Transp. Syst.

Ji, A., Levinson, D., 2020. A review of game theory models of lane changing. Transp. A Transp. Sci. 16.

Jin, C.J., Knoop, V.L., Li, D., Meng, L.Y., Wang, H., 2019. Discretionary lane-changing behavior: empirical validation for one realistic rule-based model. Transp. A Transp. Sci. 15.

Jones, D.R., 2001. A Taxonomy of Global Optimization Methods Based on Response Surfaces. J. Glob. Optim. 21, 345–383.

Ke, J., Zheng, H., Yang, H., Chen, X. (Michael), 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. Transp. Res. Part C Emerg. Technol. 85, 591–608.

Kesting, A., Treiber, M., Helbing, D., 2007. General Lane-Changing Model MOBIL for Car-Following Models. Transp. Res. Rec. J. Transp. Res. Board 1999, 86–94.

Kong, X., Zhang, Y., Eisele, W.L., Xiao, X., 2022. Using an Interpretable Machine Learning Framework to Understand the Relationship of Mobility and Reliability Indices on Truck Drivers' Route Choices. IEEE Trans. Intell. Transp. Syst. 23, 13419–13428.

Krajewski, R., Bock, J., Kloeker, L., Eckstein, L., 2018. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC.

Li, G., Yang, Y., Li, S., Qu, X., Lyu, N., Li, S.E., 2022. Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness. Transp. Res. Part C Emerg. Technol. 134.

Li, K., Wang, X., Xu, Y., Wang, J., 2016. Lane changing intention recognition based on speech recognition models. Transp. Res. Part C Emerg. Technol. 69, 497–514.

Li, Linheng, Gan, J., Zhou, K., Qu, X., Ran, B., 2020. A novel lane-changing model of connected and automated vehicles: Using the safety potential field theory. Phys. A Stat. Mech. its Appl. 559, 125039.

Li, Li, Jiang, R., He, Z., Chen, X. (Michael), Zhou, X., 2020. Trajectory data-based traffic flow studies: A revisit. Transp. Res. Part C Emerg. Technol. 114, 225–240.

Li, S., Wei, C., Wang, Y., 2022. Combining Decision Making and Trajectory Planning for Lane Changing Using Deep Reinforcement Learning. IEEE Trans. Intell. Transp. Syst. 23, 16110–16136.

Li, X., Sun, J.Q., 2017. Studies of vehicle lane-changing dynamics and its effect on traffic efficiency, safety and environmental impact. Phys. A Stat. Mech. its Appl. 467, 41–58.

Li, Z.N., Huang, X.H., Mu, T., Wang, J., 2022. Attention-Based Lane Change and Crash Risk Prediction Model in Highways. IEEE Trans. Intell. Transp. Syst. 23.

Liu, S., 2019. Research on Lane-changing Trajectory Model Based on Deep Learning. Changsha University of Science & Technology.

Liu, Yang, Bansal, P., Daziano, R., Samaranayake, S., 2019. A framework to integrate mode choice in the design of mobility-on-demand systems. Transp. Res. Part C Emerg. Technol. 105, 648–665.

Liu, Yonggang, Wang, X., Li, L., Cheng, S., Chen, Z., 2019. A Novel Lane Change Decision-Making Model of Autonomous Vehicle Based on Support Vector Machine. IEEE Access 7, 26543–26550.

Liu, Z., Gu, X., Yang, H., Wang, L., Chen, Y., Wang, D., 2022. Novel YOLOv3 Model With Structure and Hyperparameter Optimization for Detection of Pavement Concealed Cracks in GPR Images. IEEE Trans. Intell. Transp. Syst. 1–11.

Mo, P., Liu, Z., Tan, Z., Yi, W., Liu, P., 2024. Subsidy Allocation Problem with Bus Frequency Setting Game: A Trilevel Formulation and Exact Algorithm. Transp. Sci. 58, 639–663.

Mo, P., Yao, Y., D'Ariano, A., Liu, Z., 2023. The vehicle routing problem with underground logistics: Formulation and algorithm. Transp. Res. Part E Logist. Transp. Rev. 179.

Mohammadi, R., He, Q., Ghofrani, F., Pathak, A., Aref, A., 2019. Exploring the impact of foot-by-foot track geometry on the occurrence of rail defects. Transp. Res. Part C Emerg. Technol. 102, 153–172.

Monteil, J., Nantes, A., Billot, R., Sau, J., El Faouzi, N., 2014. Microscopic cooperative traffic flow: calibration and simulation based on a next generation simulation dataset. IET Intell. Transp. Syst. 8, 519–525.

Moridpour, S., Sarvi, M., Rose, G., 2010. Lane changing models: a critical review. Transp. Lett. 2, 157–173.

Nie, J., Zhang, J., Wan, X., Ding, W., Ran, B., 2016. Modeling of decision-making behavior for discretionary lane-changing execution, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. pp. 707–712.

Oseni, A., Moustafa, N., Creech, G., Sohrabi, N., Strelzoff, A., Tari, Z., Linkov, I., 2022. An Explainable Deep Learning Framework for Resilient Intrusion Detection in IoT-Enabled Transportation Networks. IEEE Trans. Intell. Transp. Syst. 1–15.

Parsa, A.B., Movahedi, A., Taghipour, H., Derrible, S., Mohammadian, A., 2020. Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. Accid. Anal. Prev. 136, 105405.

Peng, J., Zhang, S., Zhou, Y., Li, Z., 2022. An Integrated Model for Autonomous Speed and Lane Change Decision-Making Based on Deep Reinforcement Learning. IEEE Trans. Intell. Transp. Syst. 1–13.

Rahman, M., Chowdhury, M., Xie, Y., He, Y., 2013. Review of Microscopic Lane-Changing Models and Future Research Opportunities. IEEE Trans. Intell. Transp. Syst. 14, 1942–1956.

Sagi, O., Rokach, L., 2018. Ensemble learning: A survey. WIREs Data Min. Knowl. Discov. 8.

Schomakers, E.-M., Lotz, V., Glawe, F., Ziefle, M., 2023. The effect of design and behaviour of automated micro-vehicles for urban delivery on other road users' perceptions. Multimodal Transp. 2.

Seeger, M., 2004. Gaussian processes for machine learning. Int. J. Neural Syst.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N., 2016. Taking the human out of the loop: A review of Bayesian optimization. Proc. IEEE.

Shangguan, Q., Fu, T., Wang, J., Fang, S., Fu, L., 2022. A proactive lane-changing risk prediction framework considering driving intention recognition and different lane-changing patterns. Accid. Anal. Prev. 164, 106500.

Shapley, L.S., 2016. 17. A Value for n-Person Games, in: Contributions to the Theory of Games (AM-28), Volume II. Princeton University Press, pp. 307–318.

Shi, X., Wong, Y.D., Li, M.Z.F., Palanisamy, C., Chai, C., 2019. A feature learning approach based on XGBoost for driving assessment and risk prediction. Accid. Anal. Prev. 129.

Simon Zhou, X., Cheng, Q., Wu, X., Li, P., Belezamo, B., Lu, J., Abbasi, M., 2022. A meso-to-macro cross-resolution performance approach for connecting polynomial arrival queue model to volume-delay function with inflow demand-to-capacity ratio. Multimodal Transp. 1.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical Bayesian optimization of machine learning algorithms, in: Advances in Neural Information Processing Systems. pp. 2951–2959.

Song, Y., Wang, Y.D., Hu, X., Liu, J., 2022. An Efficient and Explainable Ensemble Learning Model for Asphalt Pavement Condition Prediction Based on LTPP Dataset. IEEE Trans. Intell. Transp. Syst. 1–10.

Sun, D., Elefteriadou, L., 2014. A driver behavior-based lane-changing model for urban arterial streets. Transp. Sci. 48, 184–205.

Sun, H., Huang, Y., 2024. Automatic deflection measurement for outdoor steel structure based on digital image correlation and three-stage multi-scale clustering algorithm. Autom. Constr. 163, 105416.

Sun, Q., Wang, C., Fu, R., Guo, Y., Yuan, W., Li, Z., 2021. Lane change strategy analysis and recognition for intelligent driving systems based on random forest. Expert Syst. Appl. 186.

Tajdari, F., Ghaffari, A., Khodayari, A., Kamali, A., Zhilakzadeh, N., Ebrahimi, N., 2019. Fuzzy control of anticipation and evaluation behaviour in real traffic flow, in: ICRoM 2019 - 7th International Conference on Robotics and Mechatronics. pp. 248–253.

Tang, J., Liu, F., Zhang, W., Ke, R., Zou, Y., 2018. Lane-changes prediction based on adaptive fuzzy neural network. Expert Syst. Appl. 91.

Tang, J., Yu, S., Liu, F., Chen, X., Huang, H., 2019. A hierarchical prediction model for lane-changes based on combination of fuzzy C-means and adaptive neural network. Expert Syst. Appl. 130.

Tang, T., Gu, Z., Yang, Y., Sun, H., Chen, S., Chen, Y., 2024. A data-driven framework for natural feature profile of public transport ridership: Insights from Suzhou and Lianyungang, China. Transp. Res. Part A Policy Pract. 183.

Tang, T., Liu, R., Choudhury, C., Fonzone, A., Wang, Y., 2023. Predicting Hourly Boarding Demand of Bus Passengers Using Imbalanced Records From Smart-Cards: A Deep Learning Approach. IEEE Trans. Intell. Transp. Syst. 24.

Toledo, T., Koutsopoulos, H.N., Ben-Akiva, M., 2007. Integrated driving behavior modeling. Transp. Res. Part C Emerg. Technol. 15, 96–112.

Toledo, T., Koutsopoulos, H.N., Ben-Akiva, M.E., 2003. Modeling Integrated Lane-Changing Behavior, in: Transportation Research Record. pp. 30–38.

Venthuruthiyil, S.P., Chunchu, M., 2022. Interrupted and uninterrupted lane changes: a microscopic outlook of lane-changing dynamics. Transp. A Transp. Sci. 18.

Wang, J., Zhang, Z., Lu, G., 2021. A Bayesian inference based adaptive lane change prediction model. Transp. Res. Part C Emerg. Technol. 132, 103363.

Wang, K., Guo, B., Yang, H., Li, M., Zhang, F., Wang, P., 2022. A semi-supervised co-training model for predicting passenger flow change in expanding subways. Expert Syst. Appl. 209.

Wang, Z., Huang, H., Tang, J., Lee, J., Meng, X., 2022. Driving angle prediction of lane changes based on extremely randomized decision trees considering the harmonic potential field method. Transp. A Transp. Sci. 18.

Wang, Z., Liu, Z., Cheng, Q., Gu, Z., 2024. Integrated self-consistent macro-micro traffic flow modeling and calibration framework based on trajectory data. Transp. Res. Part C Emerg. Technol. 158.

Wang, Z., Shi, Y., Tong, W., Gu, Z., Cheng, Q., 2023. Car-Following Models for Human-Driven Vehicles and Autonomous Vehicles: A Systematic Review. J. Transp. Eng. Part A Syst. 149.

Whelan, P.M. (Patrick M., Hodgson, M.J., 1978. Essential principles of physics 499.

Wu, C., Pozdnukhov, A., Bayen, A.M., 2020. Block Simplex Signal Recovery: Methods, Trade-Offs, and an Application to Routing. IEEE Trans. Intell. Transp. Syst. 21, 1547–1559.

Xie, D.F., Fang, Z.Z., Jia, B., He, Z., 2019. A data-driven lane-changing model based on deep learning. Transp. Res. Part C Emerg. Technol. 106, 41–60.

Xing, Y., Lv, C., Wang, H., Cao, D., Velenis, E., 2020. An ensemble deep learning approach for driver lane change intention inference. Transp. Res. Part C Emerg. Technol. 115, 102615.

Xue, Q., Xing, Y., Lu, J., 2022. An integrated lane change prediction model incorporating traffic context based on trajectory data. Transp. Res. Part C Emerg. Technol. 141, 103738.

Yan, R., Wu, S., Jin, Y., Cao, J., Wang, S., 2022. Efficient and explainable ship selection planning in port state control. Transp. Res. Part C Emerg. Technol. 145, 103924.

Yang, Q., Koutsopoulos, H.N., 1996. A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transp. Res. Part C Emerg. Technol. 4, 113–129.

Yin, R., Liu, Z., Zheng, N., 2022. A Simulation-Based Model for Continuous Network Design Problem Using Bayesian Optimization. IEEE Trans. Intell. Transp. Syst. 1–16.

Zhang, H., Gao, S., Guo, Y., 2024a. Driver Lane-Changing Intention Recognition Based on Stacking Ensemble Learning in the Connected Environment: A Driving Simulator Study. IEEE Trans. Intell. Transp. Syst. 25.

Zhang, H., Tan, X., Fan, M., Pan, C., Zheng, Z., Luo, S., Xu, J., 2023. Accurate Detection and Tracking of Small-Scale Vehicles in High-Altitude Unmanned Aerial Vehicle Bird-View Imagery. J. Adv. Transp. 2023.

Zhang, H., Wu, F., Guo, D., Gao, S., 2024b. What Are the Differences in Driver Lane-Changing Intention Models Recognition Performance Between Connected and Non-Connected Environments. IEEE Trans. Intell. Transp. Syst.

Zhang, Y., Owen, L.E., Clark, J.E., 1998. Multiregime approach for microscopic traffic simulation. Transp. Res. Rec.

Zhang, Yi, Shi, X., Zhang, S., Abraham, A., 2022. A XGBoost-Based Lane Change Prediction on Time Series Data Using Feature Engineering for Autopilot Vehicles. IEEE Trans. Intell. Transp. Syst. 1–14.

Zhang, Y., Xu, Q., Wang, J., Wu, K., Zheng, Z., Lu, K., 2023. A Learning-Based Discretionary Lane-Change Decision-Making Model With Driving Style Awareness. IEEE Trans. Intell. Transp. Syst. 24.

Zhang, Yue, Zou, Y., Tang, J., Liang, J., 2022. Long-term prediction for high-resolution lane-changing data using temporal convolution network. Transp. B 10.

Zhao, C., Li, Z., Li, L., Wu, X., Wang, F.Y., 2021. A negotiation-based right-of-way assignment strategy to ensure traffic safety and efficiency in lane changes. IET Intell. Transp. Syst. 15, 1345–1358.

Zheng, Z., 2014. Recent developments and research needs in modeling lane changing. Transp. Res. Part B Methodol. 60, 16–32.

Zheng, Z., Ahn, S., Monsere, C.M., 2010. Impact of traffic oscillations on freeway crash occurrences. Accid. Anal. Prev. 42, 626–636.

Zhou, F., Yang, X.J., De Winter, J.C.F., 2022. Using Eye-Tracking Data to Predict Situation Awareness in Real Time during Takeover Transitions in Conditionally Automated Driving. IEEE Trans. Intell. Transp. Syst. 23, 2284–2295.

Zhu, J., Tasic, I., Qu, X., 2022. Flow-level coordination of connected and autonomous vehicles in multilane freeway ramp merging areas. Multimodal Transp. 1.

Appendix A. Algorithm 1 for extracting the exact LCD point

Algorithm 1: A novel method for extracting the exact LCD point

Input: Vehicle x-axis speeds $\mathbf{V}_x$ and y-axis speeds $\mathbf{V}_y$, the vector of frames $\mathbf{T}$, probes b_0 and b_1, lane *ids* $\mathbf{l}$, x-axis positions $\mathbf{P}$ of a lane change vehicle

Output: Frame of the exact lane change decision point t'

```
Initialize $t_0$, $t_1$ to store the frame of the real lane change decision point;
$\mathbf{T} = (t_1, t_2, \cdots, t_k)$;
For probe $b_0 \in \mathrm{T}$ do                      # iterate overall frame
    If $\mathbf{l}_{b_0} - \mathbf{l}_{b_{0+1}} = 0$ then
        $b_0 = b_0 + 1$;
    Else
        Break;
End For
For probe $b_1 \in (t_1, t_2, \cdots, b_0)$ do          # iterate overall $(t_1, t_2, \cdots, b_0)$
    If $v_y^{b_0} \cdot v_y^{b_1} > 0$ then
        $b_1 = b_1 - 1$;
    Else
        Break;
End For
$\mathbf{P} = (p_1, p_2, \cdots, p_k)$
For probe $b_0 \in (t_1, t_2, \cdots, b_1)$ do          # iterate overall $(t_1, t_2, \cdots, b_1)$
    For $t \in (b_0, \cdots, b_1)$ do
        Find max $|p_t - p_{b_0}|$
    End For
End For
If $|p_t - p_{b_0}| > \min\{\max\{V_y \cdot \Delta t\}, 50\}$
    $t' = b_0$;
Else
    $t' = b_1$;
Return $t'$
```

Note: Probes b_0 and b_1 have no specific practical meaning, but are simply ever-changing timestamps used for loop traversal in the algorithm.

Appendix B. Data matching and Quantification of vehicle driving environment

Data matching

Most existing vehicle trajectory datasets (such as NGSIM and highD) do not contain detailed and important information (such as velocity, position, and acceleration) of surrounding vehicles (Sun et al., 2021). However, we require the trajectory data of surrounding vehicles during the lane change process.

Table 7. Selected fields from the highD dataset before data matching in the present model

No.	Required raw field name	No.	Required raw field name
1	frame	7	leftPrecedingId
2	id	8	leftAlongsideId
3	xVelocity	9	leftFollowingId
4	x	10	rightPrecedingId
5	precedingId	11	rightAlongsideId
6	followingId	12	rightFollowingId

For the vehicle driving environment shown in Table 7, the vehicle *id* was considered as the search object, and the frame rate was used as the time axis for data matching. Then, we retrieve the driving environment information of the lane-change vehicle through the *id* of the vehicles around the lane-change vehicle and the same frame. The matching process can be expressed by the following equation:

$$\mathbf{q}_{i,t}^{env} = \left[\mathbf{V}_{i,t}, \mathbf{P}_{i,t}\right], \tag{14}$$

$$\mathbf{V}_{i,t} = \left\{v_{\mathbf{s},t} \mid \mathbf{s} = VID_{i,t}^{a} \ \left(a = 1,2,\cdots,8\right)\right\}, \tag{15}$$

$$\mathbf{P}_{i,t} = \left\{p_{\mathbf{s},t} \mid \mathbf{s} = VID_{i,\mathrm{t}}^{a} \ \left(a = 1,2,\cdots,8\right)\right\}, \tag{16}$$

where $\mathbf{q}_{i,t}^{env}$ denotes the environment vector of vehicle i at frame t; $\mathbf{V}_{i,t}$ and $\mathbf{P}_{i,t}$ are the velocities along the x-axis and the coordinates of the x-axis of vehicles $\mathbf{s}$ at frame t around vehicle i as shown in Fig. 1, respectively; $VID_{i,\mathrm{t}}^{a}$ represents the *id* of its surrounding vehicle related to the subject vehicle i at frame t marked with a, where $a = 1,2,\cdots,8$ means the surrounding vehicles shown in Fig. 1. The field results obtained after data matching are presented in Table 8.

Table 8. Fields after data matching

No.	Name of the matched data field	No.	Name of the matched data field
1	frame	11	leftAlongsideV
2	id	12	rightAlongsideV
3	xVelocity	13	precedingX
4	x	14	leftPrecedingX
5	precedingXVelocity	15	rightPrecedingX
6	leftPrecedingV	16	leftFollowingX
7	rightPrecedingV	17	rightFollowingX
8	leftFollowingV	18	followingX
9	rightFollowingV	19	leftAlongsideX
10	followingV	20	rightAlongsideX

Quantification of vehicle driving environment

This section quantifies the vehicle driving environment from the trajectory dataset $\mathbf{Q}$, and introduces the detailed source of the model input information $\mathbf{X}_i$ in Equation (2).

In the data matching process, vehicles may not exist in the subject vehicle's left, front, right, or other directions. The matching result in this situation corresponds to the null value of NaN. To solve this problem, we discussed the situation separately; particularly, it was set to zero if it was a velocity variable and to positive infinity if it was a distance variable. As indicated in Equation (2), the inputs to the LCD model can be summarized as follows:

1. Velocity of the subject (lane change) vehicle,

2. Relative distances between the subject vehicle and its surrounding vehicles, and

3. Relative velocities between the subject vehicle and its surrounding vehicles.

Thus, for vehicle i, the position is transformed into relative distance, as shown in Equation (17).

$$d_i^a = \begin{cases} \left|p_i^a - p_i^0\right| & \text{if } \ VID_i^a \neq 0\text{, there is vehicle } a \text{ arround vehicle } i \\ \infty & \text{if } \ VID_i^a = 0\text{, there is no vehicle } a \text{ arround vehicle } i \end{cases}, \tag{17}$$

$$\mathbf{D}_i = \left[d_i^1, d_i^2, \cdots, d_i^8\right], \tag{18}$$

where $a = 1 \sim 8$ represents its surrounding vehicles; d_i^a indicates the relative distance between the subject vehicle and its surrounding vehicles; VID_i^a is the *id* of subject vehicle *i*'s surrounding vehicle *a*, with a value of NaN meaning there is no vehicle around the subject vehicle i; p_i^a refers to its x-coordinate in this frame; p_i^0 is the x-coordinate of vehicle i; and $\mathbf{D_i}$ denotes the vector of relative distances between vehicle *i* and its surrounding vehicles.

The transformation of relative velocity is similar to Equation (17):

$$\mu_i^a = \begin{cases} v_i^a - v_i^0, \text{ if } VID_i^a \neq 0 \text{ and } v_i^0 < 0 \\ v_i^0 - v_i^a, \text{ if } VID_i^a \neq 0 \text{ and } v_i^0 > 0, \\ 0 \quad , \qquad \text{if } VID_i^a = 0 \end{cases} \tag{19}$$

$$\mathbf{U_i} = \left[\mu_i^1, \mu_i^2, \cdots, \mu_i^8\right], \tag{20}$$

where μ_i^a indicates the relative velocity of the subject vehicle *i* and its surrounding vehicle *a*, v_i^a denotes the velocity of the surrounding vehicle *a* along the x-axis in this frame, and v_i^0 represents the velocity of the subject vehicle i along the x-axis. The data fields obtained after the data processing are listed in Table 9.

Table 9. Final input feature fields of the model after data processing

No.	Processed fields	No.	Processed fields
1	frame	11	leftAlongsideV
2	id	12	rightAlongsideV
3	xVelocity	13	precedingD
4	x	14	leftPrecedingD
5	precedingXVelocity	15	rightPrecedingD
6	leftPrecedingV	16	leftFollowingD
7	rightPrecedingV	17	rightFollowingD
8	leftFollowingV	18	followingD
9	rightFollowingV	19	leftAlongsideD
10	followingV	20	rightAlongsideD