

Article

Modeling Discretionary Lane-Changing Decisions: A Multi-Vehicle Information Enhanced Machine Learning Approach

Chenqiang Zhu ^{1,2} , Jiao Yao ^{1,*} and Ayihen Aernali ¹¹ Business School, University of Shanghai for Science and Technology, Shanghai 200093, China; cqzhu@usst.edu.cn (C.Z.); 2335070112@st.usst.edu.cn (A.A.)² Laboratory of Computation and Analytics of Complex Management Systems (CACMS), Tianjin University, Tianjin 300072, China

* Correspondence: yaojiao@usst.edu.cn

Abstract

Accurately predicting human lane-changing (LC) decisions is critical for enhancing the safety and efficiency of autonomous driving. Most existing machine learning-based LC decision models rely on immediate neighboring vehicle interaction features, which may fail to capture drivers' consideration of long-term traffic conditions in the target lane. Using discretionary LC trajectory data from the US101 dataset, this paper first qualitatively identifies key latent variables influencing LC decisions, then quantitatively ranks these factors using feature importance analysis, and finally constructs a prediction model based on ensemble learning. The analysis reveals that drivers consider not only neighboring vehicles but also multi-vehicle information further ahead, particularly the average speed and average spacing of multiple preceding vehicles. Feature importance ranking shows that safety-related features, especially the spacing with the following vehicle in the target lane (d_{Lag} , 0.187), rank significantly higher than benefit-related features such as the average speed of the target lane ($\bar{v}_T$, 0.091), suggesting that safety considerations play a dominant role in the observed LC decisions. Among five imbalanced processing methods, SMOTE+Tomek achieves the best balance (F1 = 0.68). When the Full Feature Set is used, the KNN model achieves the best performance (F1 = 0.79, AUC = 0.97) among six baseline models. This study contributes to the understanding of LC behavior and provides insights that could inform future development of LC prediction models for autonomous vehicles.

Keywords: lane-changing decision; multi-vehicle information; imbalanced data processing; machine learning



Academic Editor: George
Angelos Papadopoulos

Received: 30 April 2026

Revised: 19 June 2026

Accepted: 22 June 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the rise of a new wave of technological revolution represented by mobile internet, big data, and cloud computing, autonomous driving technology is advancing rapidly and has become the latest development direction in intelligent transportation systems and intelligent vehicle engineering. According to estimates by the U.S. Department of Transportation, full market penetration of autonomous driving technology is difficult to achieve in the short term. For at least the next 20 years, mixed traffic flows composed of both autonomous vehicles and human-driven vehicles will persist. Lane-changing (LC) behavior is a common maneuver in traffic and significantly impacts traffic flow efficiency, safety, and energy consumption [1,2]. Due to the numerous factors influencing LCs, such behavior is often

complex. For autonomous vehicles to safely coexist with human-driven vehicles, the greatest challenge lies in understanding and predicting human driving behavior. Furthermore, modeling approaches that align with the logical relationships inherent in the real world are more conducive to addressing driving behavior modeling problems [3].

To deeply understand and predict LC intentions, researchers have developed various prediction models, which can be broadly categorized into rule-based LC decision models and learning-based LC decision models [4]. Rule-based models involve manually presetting a series of clear, fixed “IF-THEN” rules to simulate a driver’s decision-making process regarding whether to change lanes in specific traffic scenarios. For example, the MOBIL (Minimize Overall Braking Induced by Lane Change) model [5] uses acceleration as a variable to establish safety and incentive criteria, incorporating a politeness factor to describe driver LC decisions in a rule-based format. In recent years, many scholars have combined cellular automata, game theory, Markov process, utility theory, and observed phenomena or patterns to construct new rules describing LC decisions and build corresponding models [6–12]. Such models offer strong interpretability, but the quality of the rules depends entirely on the expert knowledge of the modelers. Subsequently, researchers developed learning-based LC decision models, which utilize traditional machine learning or deep learning to automatically learn patterns of LC decisions from large amounts of driving data, thereby improving predictive capability. Traditional machine learning-based models encompass single-learner models (such as support vector machines, decision trees, and Bayesian classifiers [13–16]) and ensemble learning models (such as random forest (RF) and XGBoost [17–21]). These models are characterized by a simple structure, fast training speed, and strong interpretability. However, they often suffer from limited expressive capability when dealing with complex nonlinear problems and are prone to overfitting or underfitting. Deep learning-based models encompass architectures such as recurrent neural networks (RNNs), long short-term memory networks (LSTM), gated recurrent units (GRUs), convolutional neural networks (CNNs), and Transformers [22–27]. These deep learning models can automatically extract high-level spatio-temporal features from large-scale trajectory data and excel at capturing complex nonlinear relationships and long-term dependencies, achieving the highest prediction accuracy when sufficient data are available; nevertheless, they require substantial amounts of labeled data and computational resources, suffer from long training times, and are often criticized as “black box” models due to their poor interpretability.

Compared with rule-based models, traditional machine learning-based models are more powerful and flexible, as they can automatically learn complex, nonlinear decision boundaries from data, and possess stronger generalization capability. Compared with deep learning-based models, traditional machine learning-based models are more transparent and efficient, require less data, and offer stronger interpretability of the decision-making process. This is particularly crucial for the safety-critical domain of autonomous driving and is more conducive to understanding real-world LC behavior. Therefore, this paper selects traditional machine learning-based LC models as the subsequent modeling approach.

A systematic comparison of representative LC decision prediction studies is presented in Table 1, summarizing their methods, datasets, key features, evaluation metrics, and main findings.

Table 1. Summary of representative LC decision prediction studies.

	Method	Dataset	Key Features	Evaluation Metrics
Kesting et al. [5]	MOBIL (rule-based)	N/A (theoretical model; validated via simulation)	Speed, spacing, acceleration	N/A (rule formulation; validated via LC rate and spatiotemporal patterns)
Wang et al. [28]	Statistical analysis	NGSIM US101	Lateral speed, position	LC duration distribution

Table 1. *Cont.*

	Method	Dataset	Key Features	Evaluation Metrics
Jin et al. [12]	Rule-based validation	NGSIM	Speed, spacing	Hit rate
Ali et al. [17]	Random Forest, XGBoost	Naturalistic data	34 features (vehicle dynamics)	Accuracy, F1, AUC
Li and Ma [18]	GBDT	NGSIM	Speed, spacing, TTC	Accuracy, F1
Sun et al. [19]	Random Forest	NGSIM	Speed, acceleration, spacing	Accuracy, Recall
Sun et al. [20]	BO-XGBoost	NGSIM + highD	15 features	F1, AUC
Khelfa et al. [15]	Benchmark (RF, XGB, etc.)	highD	Kinematic features	F1, AUC

Despite the advantages of traditional machine learning-based models, a common challenge remains: most studies directly employ predefined feature engineering methods to extract input features, focusing primarily on immediate neighboring vehicles. However, discretionary LC decisions may involve consideration of traffic conditions beyond immediate neighbors, such as the average speed and spacing of multiple preceding vehicles. This paper addresses this gap by introducing multi-vehicle information factors. The main contributions of this paper are as follows:

- Identification of multi-vehicle information factors. Beyond conventional physical factors (such as speed, spacing, and speed difference), this paper identifies average speed and average spacing of multiple preceding vehicles as key factors influencing discretionary LC decisions. Statistical analysis confirms that drivers consider these multi-vehicle information factors even when immediate neighboring conditions appear favorable.
- Safety considerations dominate feature importance in LC decisions. In the dataset analyzed, feature importance reveals that drivers consider both safety and benefit when making LC decisions, with safety taking precedence. Safety-related features, particularly the spacing with the following vehicle in the target lane and the safety margin, rank significantly higher than benefit-related features such as the average speed of the target lane. This finding is consistent with a two-stage decision logic where safety conditions are evaluated prior to benefit considerations.
- Systematic evaluation of imbalanced processing and model selection. Based on the US101 dataset used in this paper, a comprehensive comparison of five imbalanced processing methods identifies SMOTE+Tomek as the optimal approach. Among six baseline models, KNN achieves the best performance ($F1 = 0.79$, $AUC = 0.97$). Ablation experiments further quantify the contribution of each multi-vehicle expectation feature.

The remainder of this paper is organized as follows. Section 2 describes the trajectory data and preprocessing steps. Section 3 presents the imbalance analysis and comparison of five imbalanced processing methods. Section 4 analyzes the factors influencing LC decisions, including feature importance ranking. Section 5 details the construction and evaluation of the LC decision prediction model. Section 6 concludes the paper and discusses the implications, limitations, and future work.

2. LC Trajectory Data

2.1. Data Preprocessing

This study utilizes the vehicle trajectory dataset from the US101 segment, publicly released by the NGSIM (Next Generation Simulation) project. The US101 segment is located in Los Angeles, California, adjacent to Lankershim Boulevard (as shown in Figure 1). Data

collection was conducted on 15 June 2005, from 7:50 a.m. to 8:35 a.m., during which three 15 min segments of vehicle trajectory data were collected. The dataset includes inherent vehicle attributes such as vehicle type, length, and width, as well as motion attributes including position, speed, and acceleration recorded at 0.1 s intervals.

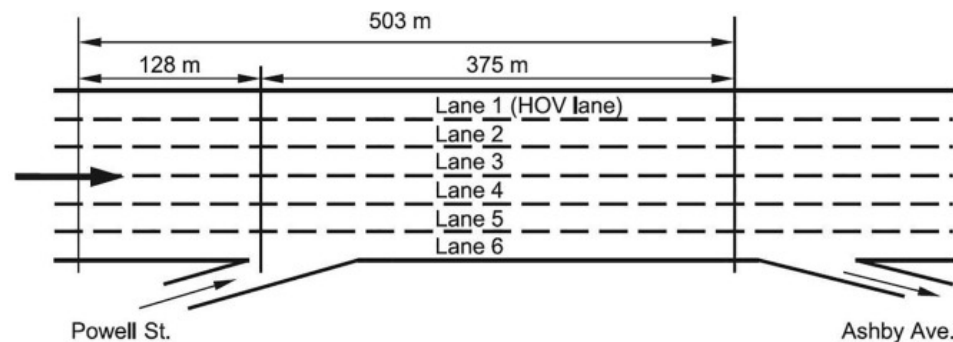


Figure 1. Road sections for collecting US101 data [29,30]. Arrows indicate the direction of traffic flow.

As the data were manually extracted frame by frame using video processing software, significant errors exist in the raw data. Therefore, data preprocessing is necessary. The preprocessing includes data smoothing and the selection of single discretionary LC trajectories for analysis.

To obtain more accurate vehicle trajectory data, the symmetric Exponential Moving Average (sEMA) method was employed to smooth the trajectory data, as shown in Equation (1):

$$\begin{cases} x'_m(k) = \sum_{n=k-D}^{k+D} x_m(k) e^{-|n-k|/\Delta} / \sum_{n=k-D}^{k+D} e^{-|n-k|/\Delta} \\ D = \max\{3\Delta, k-1, N_m-k\} \end{cases}, \quad (1)$$

where $x'_m(k)$ represents the smoothed position of LC vehicle m at time k ; $x_m(k)$ represents the original measured position of LC vehicle m at time k ; D denotes the smoothing window for boundary data; Δ denotes the smoothing window for intermediate data, $\Delta = T/d_t = 10T$; when x represents position data, $T = 0.5$ s; when x represents speed data, $T = 1$ s; and N_m represents the total number of frames in which vehicle m appears within the detected road segment [31].

The sEMA smoothing method reduces high-frequency noise while preserving the overall trajectory trend. Taking the trajectory data of Vehicle 31 in the US101 dataset during the time period from 7:50 to 8:05 as an example, the lateral position and velocity before and after smoothing are shown in Figure 2. The raw lateral position data exhibit noticeable frame-to-frame fluctuations due to manual video tracking errors. After applying sEMA, the smoothed trajectory eliminates these unrealistic oscillations while maintaining the key characteristics of the LC, including the start point, end point, and overall duration. This smoothing is essential for reliable LC detection and accurate calculation of vehicle kinematics.

Figure 1 shows the data collection segment of US101, where Lanes 1–5 are the mainline lanes, Lane 6 is an auxiliary lane. Based on the traffic characteristics of the target segment, LC maneuvers occurring in mainline Lanes 1–5 are generally regarded as discretionary LC trajectories [32]. Second, trajectories involving multiple LCs (e.g., consecutive LCs) were excluded, leaving only single LC trajectories that take place within mainline Lanes 1–5. Finally, LC data that are too short may contain incomplete information. To ensure data authenticity, for trajectories focusing on the LC vehicle, at least 10 s of data before and after the LC is required; if the available data duration is less than 10 s, the trajectory is removed.

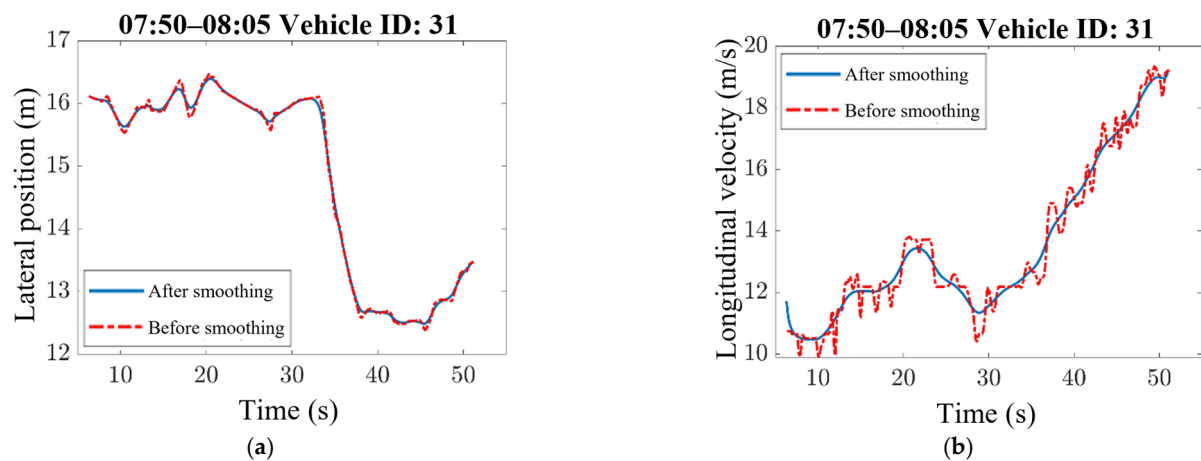


Figure 2. Trajectory before and after smoothing: (a) lateral position; (b) longitudinal velocity.

Since this study introduces multi-vehicle information features (average speed and average spacing of five preceding vehicles), samples were further filtered to ensure data quality. Specifically, a sample was excluded if (1) fewer than five vehicles were present ahead of the lane-changing vehicle (LCV) in the current lane or target lane, or (2) any of the five preceding vehicles was a non-standard vehicle (e.g., heavy trucks, motorcycles).

After applying the above criteria, trajectory data from 6101 LCVs totaling 4.028 million time steps were smoothed, resulting in 237 qualified LC trajectories with 16,500 time steps. The detailed statistics are presented in Table 2.

Table 2. Statistics of LC trajectories selected from US101 dataset.

	Number of Trajectories	Mean Trajectory Length (s)	Range of Trajectory Length (s)
Rightward LC	73	70.48	[39.0–116.2]
Leftward LC	164	69.49	[27.3–122.2]
Total	237	69.79	[27.3–122.2]

2.2. Identification of LC Start and End Points

The identification of LC decision points is closely related to the definition of the LC process. In determining the start and end points of a LC, both the vehicle's lateral speed and lateral position constraints were considered.

First, the lateral speed constraint was considered. To eliminate the influence of minor lateral displacements and errors, consistent with the approach in Reference [28], the vehicle's lateral speed was calculated at 0.5 s intervals:

$$\bar{v}(t) = \frac{x_i(t) - x_i(t - 0.5)}{0.5}, \quad (2)$$

Taking a rightward LC as an example, starting from the time point T when the lane number changes, the first time point along the forward direction of the time axis where the lateral velocity exceeds a threshold v_{latel}^{cri} is identified as the LC start point T_{start} . Similarly, the first time point along the backward direction of the time axis where the lateral velocity exceeds the threshold v_{latel}^{cri} is identified as the LC end point T_{end} . The time interval between T_{start} and T_{end} is taken as the LC duration $T_{duration}^{LC}$. In Reference [28], the authors used a threshold of $v_{latel}^{cri} = 0.2$ m/s. Given that the data in Reference [28] were not smoothed, whereas the data in this paper have been smoothed, the threshold is set to $= 0.1$ m/s. The LC durations obtained with this threshold are comparable to those reported in other studies

(see the analysis of LC duration for details), thereby demonstrating the reasonableness of this threshold selection.

Meanwhile, to address the issue of erroneous identification of LC start and end points caused by vehicle fluctuations near lane boundaries during the LC process, a lateral position constraint is introduced in the identification process, i.e., a vehicle will only start and end an LC when it is within the normal car-following lateral range. Lateral velocity changes occurring outside this normal car-following lateral range are considered temporary interruptions during the LC.

The method for determining the normal car-following lateral range of vehicles proposed in this paper draws on the concept of the outer fence in a box-and-whisker plot. In a box-and-whisker plot, to identify extreme outliers, the outer fence of the data is calculated, and data points lying outside the outer fence are considered extreme outliers. Similarly, when a vehicle is following another vehicle in each lane, its lateral position generally falls within a certain range. If it exceeds this range, it is regarded as an extreme outlier point. The specific steps are as follows: First, the upper quartile (Q1) and lower quartile (Q3) of the lateral positions of all vehicles during car-following on each lane are calculated, and the interquartile range (IQR) is obtained as $IQR = Q3 - Q1$. Then, the two outer fences, $Q1 - 1.5 \times IQR$ and $Q3 + 1.5 \times IQR$, are computed, which serve as the lateral position constraints for the vehicle's LC start point.

Figure 3 shows an example of vehicle lateral trajectories and the selected LC start and end points. The gray area in the figure represents the determined normal car-following range for the lane, the red circles indicate LC start points, and the black triangles indicate LC end points. Panels (a) and (b) illustrate rightward and leftward LCs, respectively. The trajectories in the figure exhibit driving near lane boundaries during the LC process. The method proposed in this paper can effectively avoid such issues.

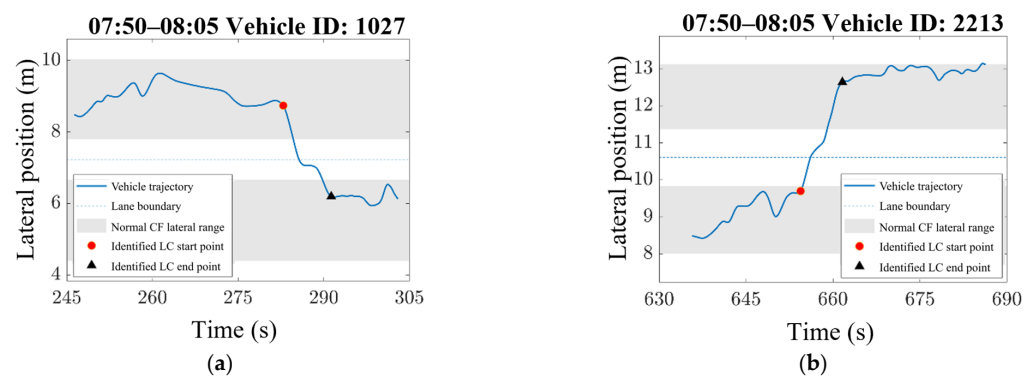


Figure 3. Identification of LC start and end points. (a) Rightward LC. (b) Leftward LC.

2.3. Analysis of LC Duration

Based on the above principles, the LC durations of the 237 LC trajectories were statistically analyzed, and the results are shown in Table 3. The vehicle LC duration fluctuates between 3 and 14 s, which is consistent with the LC duration ranges reported in other studies [28,33]. The mean duration of rightward LCs is slightly longer than that of leftward LCs.

Table 3. Statistical of duration of LC.

	Number of Trajectories	Mean LC Duration (s)	Range of LC Duration (s)
Rightward LC	73	6.41	[3.3–11.2]
Leftward LC	164	6.37	[3.3–13.7]
Total	237	6.38	[3.3–13.7]

Table 4 presents the results of the Kolmogorov–Smirnov test examining whether LC duration follows a log-normal distribution. In the table, μ_{Lognor} and σ_{Lognor} represent the calibrated mean and standard deviation of the logarithm of LC duration, respectively. The null hypothesis of the Kolmogorov–Smirnov test is that LC duration follows a log-normal distribution with a log-mean of μ_{Lognor} and a log-standard deviation of σ_{Lognor} . KS_STAT is the value of the test statistic, and KS_CV is the critical value for accepting or rejecting the hypothesis. As shown in the table, $\text{KS_STAT} < \text{KS_CV}$; therefore, the null hypothesis cannot be rejected, indicating that LC duration follows a log-normal distribution, with the probability that the null hypothesis holds being 0.723. Figure 4 presents the probability histogram of LC duration along with the fitted distribution curve. It can be seen that the two are in close agreement.

Table 4. Kolmogorov–Smirnov test results for LC duration.

μ_{Lognor}	σ_{Lognor}	KS_STAT	KS_CV	p
1.82	0.25	0.045	0.089	0.723

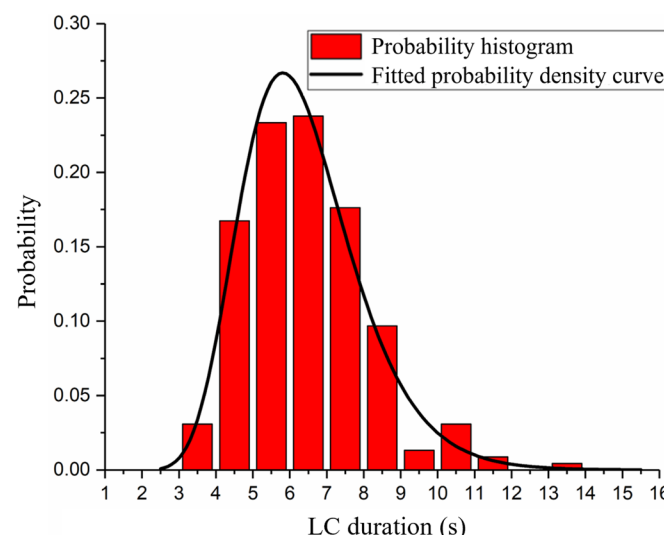


Figure 4. Distribution of LC duration.

This study focuses on maneuver recognition—identifying whether a vehicle is currently changing lanes—rather than early intention prediction. The primary objective is to identify factors influencing LC decisions (e.g., safety vs. benefit, multi-vehicle information), which requires temporally accurate correspondence between features and the LC event. Labeling the [LC start, LC end] interval ensures this alignment. Furthermore, as demonstrated by Ali et al. [30], LC maneuvers can be aborted during execution when drivers perceive unsafe conditions. Therefore, for a successfully completed LC, the traffic conditions throughout the entire period must have remained acceptable to the driver, justifying the labeling of the full interval for maneuver recognition purposes.

3. Imbalanced Data Processing for LC Decisions

3.1. Imbalance Analysis of LC Decision Samples

In LC decision modeling, the dataset typically exhibits a natural class imbalance. Specifically, during a vehicle’s driving process, the majority of time steps correspond to “lane-keeping” (non-LC) decisions, while LC decisions occur relatively infrequently. This imbalance poses a significant challenge for machine learning models, which tend to be

biased toward the majority class, resulting in poor prediction performance for the minority (LC) class—precisely the class of greatest interest for safety-critical applications.

In this paper, the original dataset (after preprocessing and LC timing identification as described in Section 2) consists of a total of 143,692 samples. Considering that the original data were sampled at a frequency of 10 Hz (i.e., vehicle states were recorded every 0.1 s), the changes in vehicle states within such a short time interval are relatively small. Therefore, the data were resampled at 1 s intervals.

During the LC process, if the vehicle finds that the LC conditions are no longer satisfied at a certain moment, it will abort the maneuver [30]. This paper focuses exclusively on successfully completed LCs. Therefore, for a successfully completed LC, the traffic conditions throughout the entire period from LC start to LC end must have continuously met the driver's requirements for performing the maneuver. Consequently, within the time period from the identified LC start point to the LC end point, the vehicle is considered to be in a state that consistently reflects the driver's LC intention. The LC decision label within this period is marked as "1" (indicating a LC event), while all other time instants are marked as "0" (indicating a non-LC event).

Based on the above processing, a total of 14,364 data samples are used for the LC decision analysis in this paper. Among these, non-LC samples account for approximately 94.09% (13,515 samples), while LC samples account for only 5.91% (849 samples). To quantitatively assess the degree of imbalance, the Imbalance Ratio (IR) is calculated as follows:

$$IR = \frac{N_{\text{majority}}}{N_{\text{minority}}}, \quad (3)$$

where N_{majority} is the number of non-LC samples and N_{minority} is the number of LC samples. In our dataset, the IR value is 15.92, indicating a severe imbalance that necessitates appropriate processing before model training. Such a severe imbalance can lead to prediction models that achieve high overall accuracy but fail to reliably predict LC events, which is unacceptable for autonomous driving decision-making systems where missing an LC event could have serious safety consequences.

3.2. Imbalanced Data Processing Methods

3.2.1. Overview of Imbalanced Data Processing Methods

In classification tasks with imbalanced class distributions, various techniques have been developed to mitigate the bias toward the majority class. These methods can be broadly categorized into three types: oversampling methods, which increase the number of minority class samples by either duplicating existing samples or generating synthetic ones; undersampling methods, which reduce the number of majority class samples by random or strategic selection; and hybrid methods, which combine both oversampling and undersampling strategies to achieve a balanced dataset while mitigating the limitations of each individual approach.

In this study, five representative methods are selected to address the class imbalance in LC decision data. These include three oversampling methods (Random Oversampling (ROS), synthetic minority oversampling technique (SMOTE), Adaptive Synthetic Sampling (ADASYN)), one undersampling method (Random Undersampling (RUS)), one hybrid method (SMOTE+Tomek), and a baseline with no processing (none). Table 5 summarizes these methods along with their respective categories.

Table 5. Imbalanced data processing methods considered in this paper.

Method	Category	Description
None	Baseline	No processing
ROS	Oversampling	Randomly duplicates minority class samples
SMOTE	Oversampling	Generates synthetic samples by interpolation
ADASYN	Oversampling	Generates more synthetic samples near boundaries
RUS	Undersampling	Randomly removes majority class samples
SMOTE+Tomek	Hybrid	Oversampling followed by boundary cleaning

3.2.2. Random Oversampling (ROS)

ROS balances the dataset by randomly duplicating existing samples from the minority class (LC) until the class distribution reaches a desired ratio, typically 1:1 [34]. The number of samples to be duplicated is determined as follows:

$$N_{\text{duplicate}} = N_{\text{majority}} - N_{\text{minority}}, \quad (4)$$

where $N_{\text{duplicate}}$ is the increased sample number of minority class (LC). This method is straightforward to implement and ensures that no information from the minority class is lost. However, it may lead to overfitting, as the duplicated samples do not provide new information to the model.

3.2.3. SMOTE

Proposed by Chawla et al. [35], SMOTE is a powerful method to address the imbalance by generating synthetic samples for the minority class rather than simply duplicating existing ones. For each minority class sample, SMOTE identifies its k nearest neighbors in the feature space and creates new synthetic samples along the line segments connecting the sample to its neighbors. The synthetic sample is generated as follows:

$$x_{\text{new}} = x_i + \lambda(\hat{x}_i - x_i), \quad (5)$$

where x_{new} is the new sample, x_i is a minority sample, $\hat{x}_i$ is one of its k nearest neighbors, λ is a random number belonging to $[0, 1]$. SMOTE helps to expand the decision region for the minority class and mitigates the overfitting problem associated with simple oversampling.

3.2.4. ADASYN

Proposed by He et al. [36], ADASYN is an adaptive extension of SMOTE that generates more synthetic samples for minority class samples that are harder to learn. The core idea is to use a density distribution as a criterion to automatically determine the number of synthetic samples to be generated for each minority sample. The weight for each minority sample x_i is calculated based on the proportion of majority samples among its k nearest neighbors:

$$r_i = \Delta_i / k, \quad (6)$$

where Δ_i is the number of majority samples among the k nearest neighbors of x_i , k is a positive integer, taken as 5 in this paper. The normalized weight $\hat{r}_i = r_i / \sum r_i$ determines how many synthetic samples are generated for x_i . Samples with more majority neighbors (i.e., harder examples) receive higher weights and thus more synthetic samples. ADASYN focuses on generating samples near the class boundary, which is particularly beneficial for LC decision modeling where the boundary between “lane-keeping” and “LC” is often ambiguous.

3.2.5. Random Undersampling (RUS)

RUS balances the dataset by randomly removing samples from the majority class (non-LC) to match the number of minority class (LC) samples. The number of samples to be removed is:

$$N_{\text{remove}} = N_{\text{majority}} - N_{\text{minority}}, \quad (7)$$

where N_{remove} is the removed sample number of majority class (non-LC). This method significantly reduces training time and storage requirements. However, it may discard potentially informative majority samples, leading to the loss of valuable information.

3.2.6. SMOTE+Tomek

SMOTE+Tomek is a hybrid method that combines oversampling and undersampling strategies [37]. First, SMOTE is applied to generate synthetic minority class samples, which helps to expand the decision region for the minority class. Then, Tomek Links are identified and removed to clean the overlapping boundary between classes. Tomek Links are pairs of samples from different classes that are each other's nearest neighbors. Removing these pairs helps to create a clearer decision boundary and reduces noise in the dataset. This hybrid approach leverages the strengths of both methods: SMOTE enriches the minority class, while Tomek Links eliminate ambiguous samples near the class boundary.

3.2.7. Workflow of Data Splitting and Imbalance Processing

To ensure rigorous model evaluation, the following workflow was strictly followed:

1. Data split: The dataset was first split into training (80%) and testing (20%) sets. The specific splitting strategy (random split or trajectory-level split) is described in the respective sections.
2. Feature standardization: Standard Scaler was fitted on the training set and applied to both training and testing sets.
3. Imbalance processing: Imbalance processing method was applied only to the training set to balance the class distribution. The testing set remained untouched to preserve the original class distribution and provide an unbiased evaluation of model generalization.
4. Model training and evaluation: Models were trained on the processed training set and evaluated on the original testing set.

This workflow ensures that no information from the testing set influences the training process.

3.3. Comparative Analysis of Imbalanced Data Processing Methods

To determine the most effective imbalanced data processing method for LC decision modeling described in this paper, a comparative study is conducted using XGBoost as the base classifier. The baseline and five methods described in Section 3.2 are evaluated using the same experimental setup. The goal is to select the optimal method that provides the best trade-off between identifying LC events (minority class) and maintaining overall prediction accuracy.

3.3.1. Experimental Setup for Imbalance Processing

All experiments are conducted on the dataset described in Section 2, with the same training–testing split ratio of 80:20. To ensure a fair comparison across different imbalanced data processing methods, the XGBoost classifier was configured with hyperparameters optimized individually for each resampled dataset. Specifically, for each resampling method, a grid search was performed with three-fold cross-validation using F1-score as the evaluation metric over the following parameter space: learning rate $\in \{0.05, 0.1, 0.15\}$, maximum tree depth $\in \{4, 6, 8\}$, number of estimators $\in \{100, 150, 200\}$, subsample $\in \{0.8, 0.9, 1.0\}$, colsam-

ple_bytree $\in \{0.8, 0.9, 1.0\}$, and min_child_weight $\in \{1, 3, 5\}$. The optimal hyperparameters identified for each resampling method were then used to train the final XGBoost model. This approach ensures that each imbalanced processing technique achieves its best possible performance, allowing for a fair comparison of their intrinsic effectiveness.

The performance of each method is evaluated using multiple metrics, with particular emphasis on the minority class (LC = 1), as failing to predict an LC event has more severe safety implications than a false alarm. The following metrics are reported:

- Precision: The proportion of predicted LC events that are actual LC events.

$$\text{Precision} = TP / (TP + FP), \quad (8)$$

where TP is the number of true positives (LC events correctly predicted as LC), FP is the number of false positives (non-LC events incorrectly predicted as LC).

- Recall (Sensitivity): The proportion of actual LC events that are correctly identified.

$$\text{Recall} = \text{Sensitivity} = TP / (TP + FN), \quad (9)$$

where FN is the number of false negatives (LC events incorrectly predicted as non-LC).

- Accuracy: The proportion of all predictions that are correct.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN), \quad (10)$$

where TN is the number of true negatives (non-LC events correctly predicted as non-LC).

- F1-Score: The harmonic mean of Precision and Recall, providing a single balanced measure.

$$F1 - \text{Score} = 2 * (\text{Recall} \times \text{Precision}) / (\text{Recall} + \text{Precision}), \quad (11)$$

- AUC (Area Under the ROC Curve): A measure of the model's ability to distinguish between the two classes. A higher AUC indicates better discriminative performance between the two classes.

3.3.2. Comparison of Results

Table 6 and Figures 5 and 6 present the performance comparison of the baseline and five imbalanced data processing methods with XGBoost. Table 6 lists the specific values of five evaluation metrics (F1-Score, Recall, Precision, Accuracy, and AUC) for each method. Figure 5 presents the comparison with performance of each method on three core metrics (F1-Score, Recall, and Precision), facilitating observation of each method's strengths and weaknesses across different metrics. Figure 6 plots Recall on the x -axis and Precision on the y -axis, with F1-Score represented on bubble colors. This enables clear evaluation of the trade-off between missed detections (Recall) and false alarms (Precision). Methods closer to the top-right corner (high Recall, high Precision) and closer to the diagonal line (Recall = Precision) achieve better overall performance.

Table 6. Performance comparison of imbalanced data processing methods.

Method	F1-Score	Recall	Precision	Accuracy	AUC
SMOTE+Tomek	0.68	0.79	0.59	0.90	0.93
ROS	0.67	0.73	0.63	0.91	0.93
None	0.67	0.56	0.85	0.93	0.93
SMOTE	0.67	0.78	0.58	0.90	0.93
ADASYN	0.64	0.81	0.54	0.89	0.93
RUS	0.55	0.85	0.41	0.82	0.92

Note: The best performance for each metric is underlined.

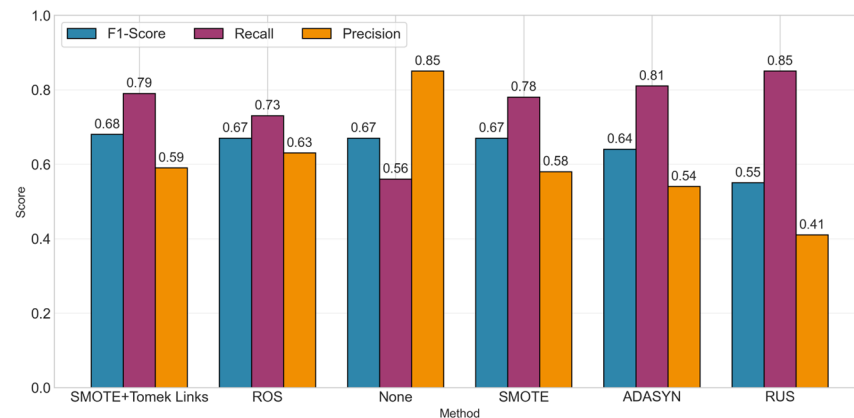


Figure 5. Comparison of F1-Score, Recall, and Precision across imbalanced processing methods.

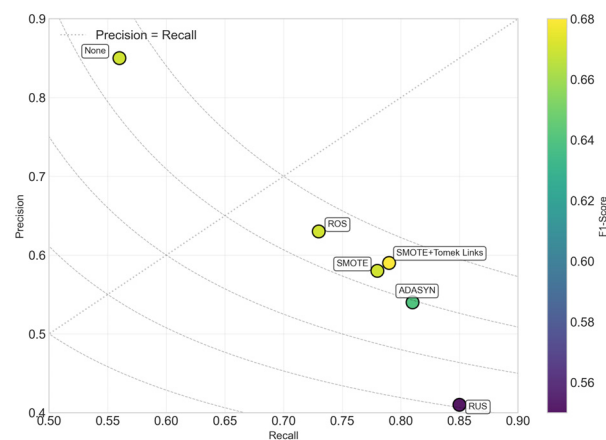


Figure 6. Recall–Precision scatter plot.

It can be seen that SMOTE+Tomek achieves the highest F1-Score (0.68), slightly outperforming other methods, and strikes the best balance between Recall (0.79) and Precision (0.59). In contrast, the baseline with no processing achieves the highest Precision (0.85) but the lowest Recall (0.56), meaning that nearly half of the LC events are missed, which fails to meet safety requirements. RUS achieves the highest Recall (0.85) but the lowest Precision (0.41), resulting in an excessively high false positive rate that negatively impacts user experience. ADASYN yields a lower F1-Score (0.64) compared to SMOTE and ROS. Based on the above analysis, SMOTE+Tomek is selected as the optimal imbalanced data processing method for subsequent feature importance analysis and LC decision modeling.

4. Analysis of Factors Influencing LC Decisions

4.1. Selection of LC Features

This paper utilizes discretionary LC data from the US101 dataset. As shown in Figure 7, a vehicle engages in interactions with various surrounding vehicles while driving, and these interactions can be characterized by a range of factors. Based on the LC influencing factors commonly considered in the previous literature, 13 attributes, as shown in Table 7, were extracted for each LC trajectory. In this paper, subscript C denotes the current lane and subscript T denotes the target lane. These 13 factors are categorized into three categories: physical factors, multi-vehicle expectation factors, and safety factors.

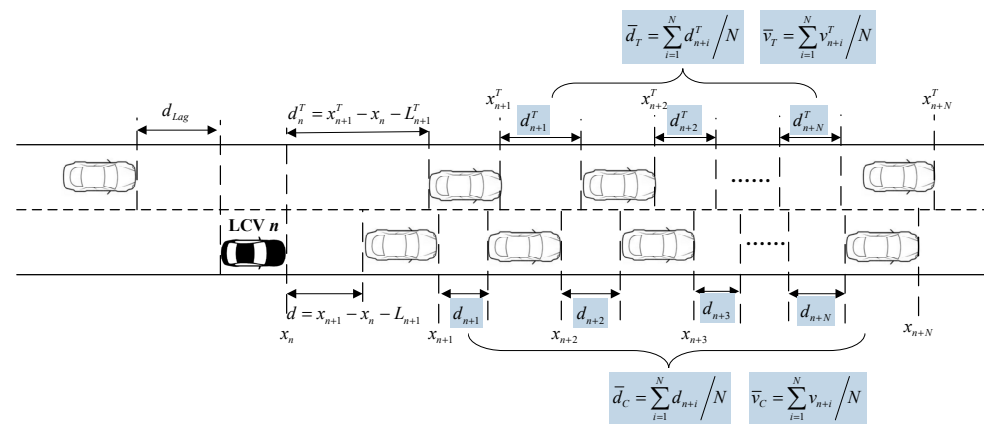


Figure 7. Factors involved in the LC process.

Table 7. LC influencing factors commonly considered in the literature.

No.	Category	Variable	Variable Description
1	Physical Factors	Δv	$v_{n+1} - v_n$, where v_{n+1} is the preceding vehicle in the current lane
2		Δv^T	$v_{n+1}^T - v_n$, where v_{n+1}^T is the preceding vehicle in the target lane
3		Δv^{Lag}	$v_n - v_{n-1}^T$, where v_{n-1}^T is the following vehicle in the target lane
4		d	$x_{n+1} - x_n - L_{n+1}$, where L_{n+1} is the length of the preceding vehicle in the current lane
5		d^T	$x_{n+1}^T - x_n - L_{n+1}^T$, where L_{n+1}^T is the length of the preceding vehicle in the target lane
6		d_{Lag}	$x_n - x_{n-1}^T - L_n$, where L_n is the length of the LCV
7		v	Speed of the LCV
8	Multi-Vehicle Information Factors	$\bar{v}_C$	$(v_{n+1} + v_{n+2} + v_{n+3} + v_{n+4} + v_{n+5})/5$, the absolute average speed of the 5 vehicles ahead in the current lane
9		$\bar{v}_T$	$(v_{n+1}^T + v_{n+2}^T + v_{n+3}^T + v_{n+4}^T + v_{n+5}^T)/5$, the absolute average speed of the 5 vehicles ahead in the target lane
10		$\bar{d}_C$	$(d_{n+1} + d_{n+2} + d_{n+3} + d_{n+4} + d_{n+5})/5$, the absolute average spacing of the 5 vehicles ahead in the current lane
11		$\bar{d}_T$	$(d_{n+1}^T + d_{n+2}^T + d_{n+3}^T + d_{n+4}^T + d_{n+5}^T)/5$, the absolute average spacing of the 5 vehicles ahead in the target lane
12	Safety Factors	SM	SM of the LCV with current lane preceding vehicle
13		SM^T	SM of the following vehicle in the target lane

The physical factors include the speed (v) of the LCV itself, the speed difference (Δv) and spacing (d) between the LCV and the preceding vehicle in the current lane, the speed difference (Δv^T) and spacing (d^T) between the LCV and the preceding vehicle in the target lane, and the speed difference (Δv^{Lag}) and spacing (d_{Lag}) between the LCV and the following vehicle in the target lane.

The multi-vehicle information factors include the absolute average speed of five preceding vehicles in the current lane ($\bar{v}_C$) and the target lane ($\bar{v}_T$), and the absolute average spacing of five preceding vehicles in the current lane ($\bar{d}_C$) and the target lane ($\bar{d}_T$). These absolute averages are used directly as input features.

The choice of five preceding vehicles is not arbitrary but is theoretically grounded in traffic flow stability analysis. According to the multi-anticipative IDM model [38], linear stability analysis reveals a pattern of diminishing returns: as the number of look-ahead vehicles increases from 1 to 3, the stable region of traffic flow expands significantly; however, the marginal benefit gradually saturates, and once the number reaches five, further improvements in string stability become negligible. Moreover, as noted by Treiber et al. [39], for typical driver reaction times of 1.0–1.5 s, considering up to five preceding vehicles is sufficient to fully compensate for the instability caused by human reaction delays. Given these theoretical and empirical considerations, selecting five vehicles strikes

an optimal balance between capturing sufficient multi-vehicle information and maintaining model parsimony. Therefore, the absolute average speed and spacing of the five preceding vehicles in both the current and target lanes are adopted as multi-vehicle expectation factors in this study.

It is worth noting that our conclusions regarding multi-vehicle information are primarily applicable to the five-vehicle setting adopted in this study. We did not systematically vary the number of preceding vehicles. Therefore, whether the same findings would hold for a different number remains an open question, and future work could perform sensitivity analyses to explore the optimal trade-off between information richness and model parsimony.

Due to the continuously dynamic nature of the vehicle and its surrounding vehicles during driving, and considering the need for a continuous indicator, the safety margin (SM) is selected as the safety factor. The safety margin is a threshold that protects the driver from hazards [40]. Generally, $SM \geq 0$ is considered safe, while $SM < 0$ is considered unsafe. This paper adopts the definition of SM as shown in Equation (12):

$$SM_n(t) = 1 - \frac{0.15v_n(t) + \frac{(v_n(t))^2}{1.5g} - \frac{(v_{n+1}(t))^2}{1.5g}}{d_n(t)}, \quad (12)$$

where $v_n(t)$ and $v_{n+1}(t)$ are the speeds of the subject vehicle and its preceding vehicle, respectively, $d_n(t)$ is the distance between the subject vehicle and its preceding vehicle, and g is the gravitational acceleration. This paper considers the SM of the LCV (SM) as well as the SM of the following vehicle in the target lane (SM^T).

4.2. Statistical Analysis of Results

Currently, the factors considered in LC decision-making are largely focused on the physical factors of neighboring vehicles, such as the spacing and speed between the LCV and the preceding/following vehicles in adjacent lanes. The core concept underlying these rules is that discretionary LC decisions are based on the instantaneous states of neighboring vehicles. That is, the LCV can immediately obtain the benefits of the LC at the next time step, immediately after executing the LC. This section will explore the influence of physical factors on LC decisions using empirical data, and further analyze the importance of multi-vehicle information factors in LC decision-making.

Table 8 presents the explanatory power of physical factors regarding LC decisions, measured as the proportion of cases where specific conditions are satisfied at the moment of LC execution. The results indicate that at the moment of LC execution, the speed of the preceding vehicle in the target lane exceeds that in the current lane in 63.02% of cases, suggesting that the speed of the immediate preceding vehicle is indeed an important influencing factor. Furthermore, the average speed of preceding vehicles in the target lane exceeds that in the current lane in 59.25% of cases, indicating that drivers consider not only the immediate preceding vehicle (i.e., instantaneous benefit, where traffic conditions would improve immediately after changing lanes) but also the long-term average speed of the entire lane (i.e., potential benefit, representing future traffic conditions in that lane) when making decisions.

Additionally, the average spacing of preceding vehicles in the target lane exceeds that in the current lane in 57.71% of cases, whereas the spacing of the immediate preceding vehicle in the target lane exceeds that in the current lane in only 42.17% of cases. This suggests that seeking more space is another primary motivation for LCs, although its driving force is slightly weaker than that of seeking higher speed. Moreover, unlike speed—where drivers place greater emphasis on immediate benefits—for spacing, drivers appear to value the long-term potential trend of the entire lane more highly.

Table 8. Impact of physical factors on LC decisions.

Considered Factor	Speed Gain (Lead)	Speed Gain (Avg.)	Space Gain (Avg.)	Space Gain (Lead)
Meaning	$v_{n+1}^T > v_{n+1}$	$\bar{v}_C < \bar{v}_T$	$\bar{d}_C < \bar{d}_T$	$d < d^T$
Proportion of LCs Explained	63.02%	59.25%	57.71%	42.17%

Figure 8 illustrates a scenario where a vehicle changes lanes despite the preceding vehicle in the current lane having both greater speed and spacing compared to the preceding vehicle in the target lane. The first row of subplots presents the vehicle's physical parameters, including velocity, average velocity, spacing, and average spacing. In these plots, the blue line represents the LCV, the red line represents the preceding vehicle in current lane, and the green line represents the preceding vehicle in adjacent target lane. Solid lines indicate the immediate preceding vehicle, while dashed lines represent the average conditions of multiple vehicles ahead. The black vertical dashed line marks the moment the LCV crosses the lane line, and the light blue vertical line indicates the LC execution time.

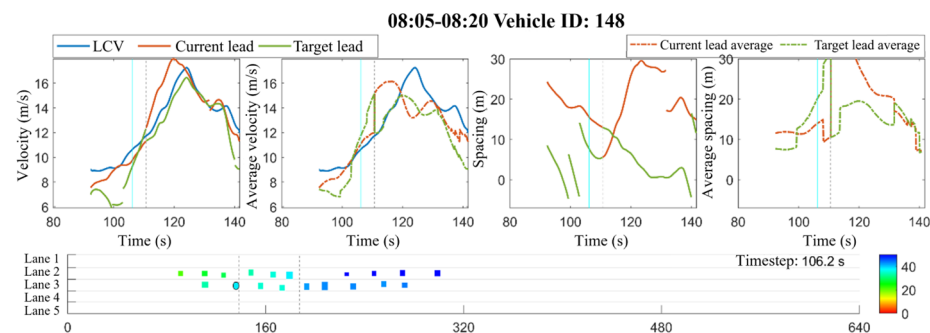


Figure 8. Example of considering multi-vehicle information in LC decision-making. Blue/red/green lines represent the LCV, preceding vehicle in current lane, and preceding vehicle in target lane, respectively. Solid/dashed lines distinguish immediate vehicles from multi-vehicle averages.

The second row of subplots depicts the actual traffic situation at the LC execution moment. In these plots, circular points represent the LCV, square points represent surrounding vehicles, with point sizes proportional to actual vehicle dimensions and different colors indicating different speeds.

As shown in the figure, although the spacing and velocity of the immediate preceding vehicle in the target lane are not superior to those in the current lane, the average headway distance and average speed of multiple vehicles ahead in the target lane are indeed better. This further demonstrates that when deciding whether to change lanes, drivers consider not only the speed and spacing of the immediate preceding vehicle but also multi-vehicle information. Even when the immediate neighboring conditions in the current lane appear favorable, drivers may still initiate an LC if they anticipate deteriorating conditions in the current lane or better conditions in the target lane in the future. This observation confirms that the average spacing and average velocity of vehicles ahead in the target lane are indeed variables worth considering.

4.3. Importance Ranking of LC Influencing Factors

As concluded in Section 3.3, among the five imbalanced data processing methods evaluated with XGBoost, SMOTE+Tomek achieved the best overall performance, with the highest F1-Score (0.68) and the best balance between Recall (0.79) and Precision (0.59).

Therefore, the dataset processed by SMOTE+Tomek is adopted for subsequent feature importance analysis.

Figure 9 presents the feature importance ranking and cumulative contribution rate curve obtained from the XGBoost model trained on the balanced dataset. The bar chart shows the importance score of each feature (left y -axis), with features arranged by their contribution to LC decision prediction. The line chart represents the cumulative contribution rate of the top n features (right y -axis), and the dashed line indicates the 80% and 90% cumulative contribution rate reference line.

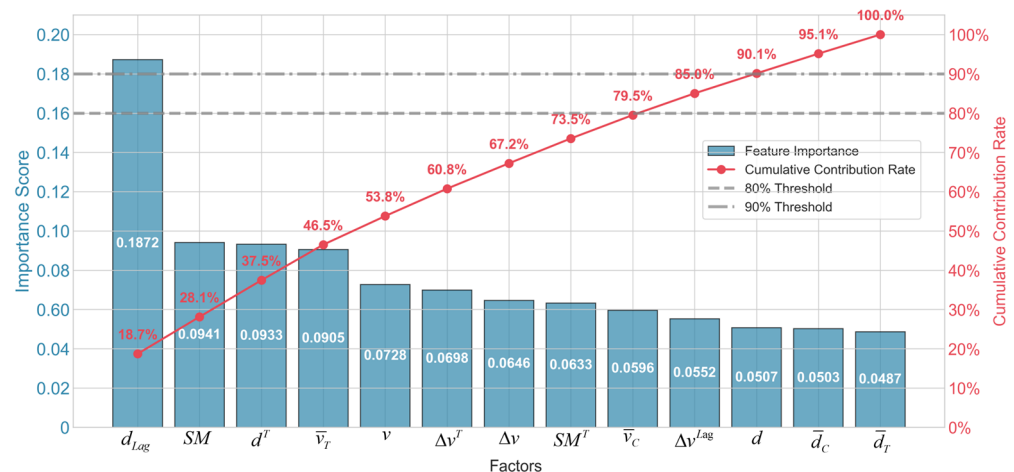


Figure 9. Importance ranking of influencing factors.

As shown in Figure 9, the top four features (d_{Lag} , SM, d^T , $\bar{v}_T$) achieve a cumulative contribution rate of 46.5%. The top nine features reach 79.6% (approaching 80%), and the top 11 features reach 90.1% (exceeding 90%). Notably, the importance scores of the bottom four features (Δv^{Lag} , d , $\bar{d}_C$, $\bar{d}_T$) are very close (0.055, 0.051, 0.050, 0.049), exhibiting a “long-tail” distribution—although each individual feature contributes little, the cumulative contribution of multiple low-importance features may not be negligible. Whether these low-importance features can be removed without significantly affecting model performance requires further validation through experiments in Section 5.

Based on the feature importance ranking and cumulative contribution rate analysis presented above, the following main conclusions can be drawn:

- The spacing with the following vehicle in the target lane is the most critical factor in this balanced dataset. The feature d_{Lag} (spacing between the LCV and the following vehicle in the target lane) achieves the highest importance score (0.187), far exceeding all other features (approximately 2.4 times the average importance). This result suggests that, in the driving behavior reflected by this dataset, drivers prioritize the availability of sufficient space in the target lane’s rear gap when making LC decisions.
- Following vehicles in the target lane matter more than preceding vehicles in this balanced dataset. When aggregating features related to following vehicles ($d_{Lag} + \Delta v^{Lag}$) and preceding vehicles ($d^T + \Delta v^T$) in the target lane, the total importance of following-vehicle features (0.242) is substantially higher than that of preceding-vehicle features (0.163). This suggests that, in the LC decisions reflected by this dataset, drivers pay more attention to “risks from the rear” than to “gains from the front” in the target lane.
- Average speed of the target lane contributes most significantly among multi-vehicle information features in this dataset. Among the four multi-vehicle information features, $\bar{v}_T$ (average speed of preceding vehicles in the target lane, 0.091) shows notably higher importance than $\bar{v}_C$ (0.060), $\bar{d}_C$ (0.050), and $\bar{d}_T$ (0.049). This result indicates that,

under the data conditions of this paper, drivers are more concerned with the average traffic efficiency of the target lane (measured by average speed), while average spacing information contributes relatively little.

- A dual-layer pattern emerges from the feature importance ranking: safety-related features dominate the top positions, followed by benefit-related features. In the feature importance ranking of this paper, a clear dual-layer structure can be observed as shown in Table 9:

Table 9. Dual-layer pattern in feature importance ranking.

Layer	Logic	Key Features	Core Demand
Layer 1	Safety Threshold	d_{Lag}, SM	LC safely
Layer 2	Benefit Incentive	$d_T, \bar{v}_T, \Delta v^T$	Better after LC

Safety Threshold: In this dataset, the spacing with the following vehicle in the target lane (d_{Lag}) and the LCV's own safety margin with preceding vehicle in the target lane (SM) rank at the top of the importance list, indicating that safety conditions are a prerequisite for LC decisions.

Benefit Incentive: After safety conditions are satisfied, the spacing (d_T) and speed difference (Δv^T) with the preceding vehicle in the target lane, together with the average speed of the target lane ($\bar{v}_T$), reflect the spatial and speed benefits obtainable after LC, further influencing the decision outcome.

To further validate the feature importance ranking and provide interpretability at the individual prediction level, SHAP (SHapley Additive exPlanations) analysis was conducted on the XGBoost model. Figure 10 presents the SHAP summary plot, where features are ranked by their mean absolute SHAP values. The SHAP results confirm the dominant role of safety-related features: d_{Lag} (spacing with the following vehicle in the target lane) exhibits the highest SHAP value, followed by d^T (spacing with the preceding vehicle in the target lane). This ranking is consistent with the gain-based feature importance shown in Figure 9, providing complementary evidence that safety-related features are the strongest predictors of LC decisions in this dataset. Notably, $\bar{v}_C$ (average speed of the current lane) and $\bar{v}_T$ (average speed of the target lane) show moderate contributions, confirming that multi-vehicle information factors have predictive value, albeit secondary to safety considerations. However, we note that SHAP values reflect the model's learned patterns and do not necessarily imply causal driver behavior.

The dominance of safety-related features is consistent with risk-aware driving behavior modeling. Shao et al. [41] proposed a safety potential field model for autonomous vehicle longitudinal control, demonstrating that risk-aware control strategies—where vehicles maintain larger following distances under higher perceived risk—improve both safety and traffic efficiency. From this perspective, the high importance of d_{Lag} (spacing with the following vehicle in the target lane) can be interpreted as drivers responding to a risk potential field that intensifies with proximity to surrounding vehicles. This alignment between our SHAP analysis (where d_{Lag} exhibits the highest mean absolute SHAP value) and the risk potential field theory provides a theoretical foundation for understanding why safety-related factors dominate LC decisions: drivers are fundamentally minimizing collision risk, particularly rear-end collisions in the target lane.

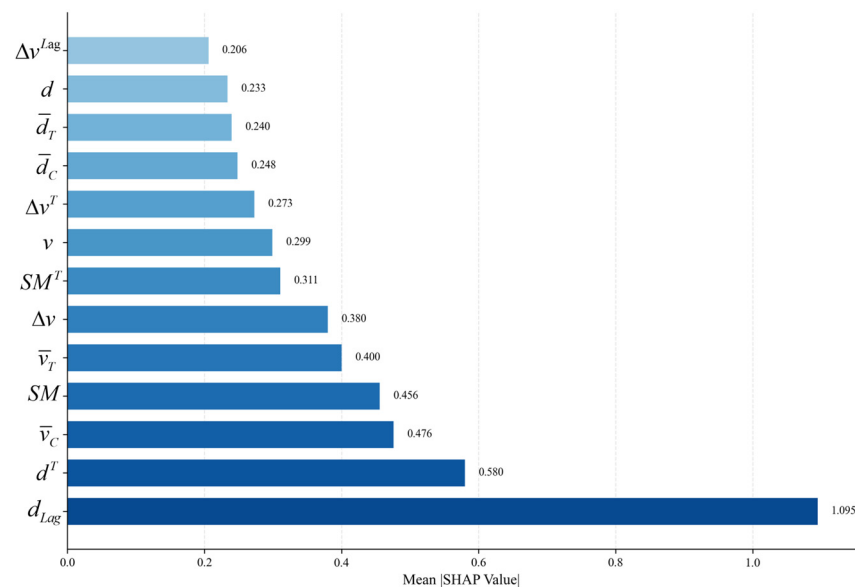


Figure 10. SHAP summary plot of feature contributions for the XGBoost model.

5. LC Decision Modeling Considering Multi-Vehicle Information

5.1. Experimental Setup for Model Comparison

5.1.1. Dataset and Evaluation Metrics

Following the data preprocessing procedures described in Section 2 and the imbalance processing evaluation conducted in Section 3, the dataset processed by SMOTE+Tomek—identified as the optimal imbalanced data processing method—is adopted for all experiments in this section. Consistent with the workflow described in Section 3.2.7, SMOTE+Tomek was applied only to the training set after data splitting. The final dataset consists of balanced samples between LC and non-LC classes in the training set only. The dataset is divided into training and testing sets using an 80/20 stratified split, ensuring consistent class distribution across both subsets.

The same evaluation metrics described in Section 3.3.1 (Precision, Recall, F1-Score, Accuracy, and AUC) are used to assess model performance.

All experiments were implemented in Python 3.9 using the following libraries: scikit-learn (version 1.2.2), XGBoost (1.7.5), imbalanced-learn (0.10.1), and SHAP (0.41.0). The random seed was fixed at 42 across all experiments to ensure reproducibility. The hardware environment consisted of an Intel Core i7-12700K CPU with 32 GB RAM.

5.1.2. Baseline Models

As discussed in the introduction, traditional machine learning-based models can be broadly categorized into single-learner models and ensemble learning models. To provide a comprehensive evaluation of different modeling paradigms for LC decision prediction, six representative models are selected from both categories.

From the ensemble learning category, four models are included:

- XGBoost and GradientBoosting (both from the software libraries listed in Section 5.1.1) represent gradient boosting methods, which build models sequentially by correcting previous errors. They are widely adopted in traffic behavior modeling due to their high predictive accuracy and ability to handle complex feature interactions.
- Random Forest (RF) represents bagging-based ensemble methods, which build multiple decision trees in parallel and aggregate their predictions. It offers robustness against overfitting and provides feature importance analysis.

- AdaBoost represents adaptive boosting methods that iteratively focus on hard-to-classify samples, providing a contrast to gradient-based boosting approaches.

From the single-learner category, two models are included:

- Logistic Regression (LR) serves as a linear model baseline with strong interpretability, helping to assess whether nonlinear relationships in the data provide substantial performance gains over a linear decision boundary.
- K-Nearest Neighbors (KNNs) represents instance-based learning, which makes no assumptions about data distribution and offers insight into the local structure of the feature space.

5.1.3. Hyperparameter Optimization

For each of the six baseline models, hyperparameters are optimized using grid search with three-fold cross-validation on the training set. F1-Score is used as the optimization metric to balance Precision and Recall. Table 10 summarizes the search spaces for each model. The search spaces are designed to cover typical ranges reported in the literature while balancing computational efficiency. After grid search, the optimal hyperparameters for each model are selected based on the highest cross-validated F1-Score, as shown in Table 11. These optimal configurations are then used to train the final models on the full training set.

Table 10. Hyperparameter search spaces for baseline models.

Model	Parameter	Search Space	Default Value
XGBoost	number of estimators	{100, 150, 200}	100
	maximum tree depth	{4, 6, 8}	6
	learning rate	{0.05, 0.1, 0.15}	0.1
	subsample	{0.8, 0.9, 1.0}	1.0
	colsample_bytree	{0.8, 0.9, 1.0}	0.9
	min_child_weight	{1, 3, 5}	3
GradientBoosting	number of estimators	{100, 150, 200}	100
	maximum tree depth	{3, 5, 7}	3
	learning rate	{0.05, 0.1, 0.15}	0.1
RF	number of estimators	{100, 150, 200}	100
	maximum tree depth	{5, 7, 10}	5
	min_samples_split	{2, 5, 10}	2
	min_samples_leaf	{1, 2, 4}	1
AdaBoost	number of estimators	{50, 100, 150}	50
	learning rate	{0.5, 1.0, 1.5}	1.0
LR	C	{0.1, 1, 10, 100}	1.0
	penalty	{‘l2’}	‘l2’
	solver	{‘lbfgs’, ‘liblinear’}	‘lbfgs’
	max_iter	{1000, 2000}	1000
KNN	number of neighbors	{3, 5, 7, 9, 11}	5
	weights	{‘uniform’, ‘distance’}	‘uniform’
	metric	{‘euclidean’, ‘manhattan’}	‘euclidean’

Table 11. Optimal hyperparameters for each baseline model identified by grid search.

Model	Selected Hyperparameters
XGBoost	number of estimators = 150; maximum tree depth = 6; learning rate = 0.1; subsample = 0.8; colsample_bytree = 0.9; min_child_weight = 3
GradientBoosting	number of estimators = 150; maximum tree depth = 7; learning rate = 0.1

Table 11. Cont.

Model	Selected Hyperparameters
RF	number of estimators = 200; maximum tree depth = 10; min_samples_split = 2; min_samples_leaf = 1
AdaBoost	number of estimators = 150; learning rate = 1.0
LR	C = 0.1; penalty = 'l2'; solver = 'liblinear'; max_iter = 1000
KNN	number of neighbors = 3; weights = 'distance'; metric = 'manhattan'

5.2. Comparison of Feature Set Configurations

Based on the cumulative contribution rate analysis in Section 4.3, the importance scores of the bottom four features (Δv^{Lag} , d , $\bar{d}_C$, $\bar{d}_T$) are very close (ranging from 0.049 to 0.055), exhibiting a “long-tail” distribution. While the top nine features achieve a cumulative contribution rate of 79.6% and the top 11 features reach 90.1%, it remains unclear whether removing these low-importance features would significantly affect model performance. To investigate this issue, three feature set configurations were compared:

- Full Feature Set (13 features): All features
- 11-Feature Set: Excludes the two lowest-importance features ($\bar{d}_C$ and $\bar{d}_T$)

9-Feature Set: Excludes the four lowest-importance features (Δv^{Lag} , d , $\bar{d}_C$, $\bar{d}_T$)

Both configurations are evaluated using XGBoost under the same experimental conditions. Table 12 and Figure 11 present the performance comparison, and Figure 12 shows the corresponding ROC curves.

Table 12. Performance comparison of feature set configurations.

Feature Set	No. of Features	F1-Score	Recall	Precision	Accuracy	AUC
Full Feature Set	13	0.68	0.79	0.59	0.90	0.93
11-Feature Set	11	0.63	0.77	0.53	0.88	0.92
9-Feature Set	9	0.60	0.76	0.50	0.87	0.91

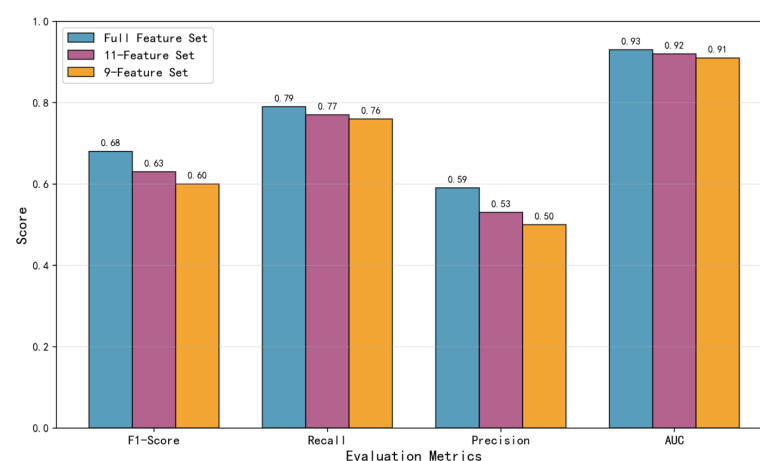


Figure 11. Performance comparison bar chart.

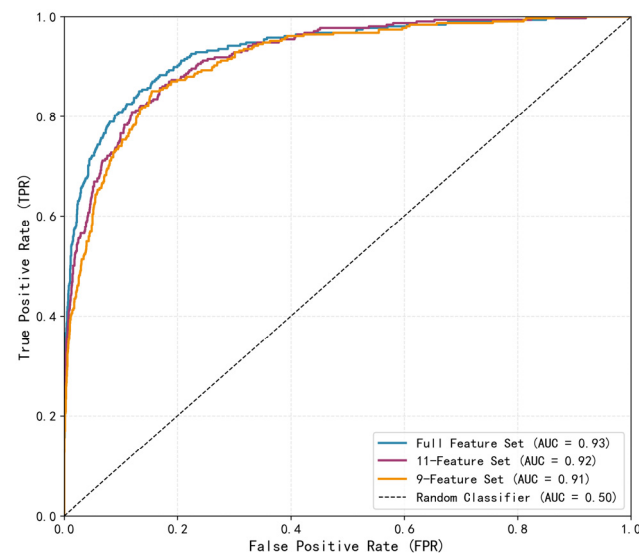


Figure 12. ROC curve comparisons of different feature sets.

As shown in Table 12 and Figures 11 and 12, the Full Feature Set consistently outperforms the reduced feature sets across all evaluation metrics. The F1-Score decreases from 0.68 (Full Feature Set) to 0.63 (11-Feature Set) and further to 0.60 (9-Feature Set), with similar declines observed in Recall, Precision, Accuracy, and AUC. These results indicate that removing the low-importance features leads to noticeable performance degradation. Therefore, to maximize prediction performance, the Full Feature Set (13 features) is adopted for all subsequent experiments in this study.

5.3. Comparison of Different Baseline Models

Using the Full Feature Set (13 features) identified in Section 5.2, six baseline models are compared to identify the most suitable model for LC decision prediction. All models are evaluated under the same experimental conditions. Table 13 presents the performance comparison.

Table 13. Performance comparison of different baseline models.

Model	F1-Score	Recall	Precision	Accuracy	AUC
KNN	0.79	0.90	0.70	0.94	0.97
GradientBoosting	0.68	0.77	0.61	0.91	0.94
XGBoost	0.68	0.79	0.59	0.90	0.93
RF	0.58	0.79	0.45	0.85	0.91
ADA	0.48	0.76	0.35	0.79	0.86
LR	0.44	0.72	0.32	0.77	0.81

As shown in Table 13, KNN achieves the highest F1-Score (0.79) and AUC (0.97), significantly outperforming all other models. GradientBoosting and XGBoost show comparable performance ($F1 = 0.68$), while RF (0.58), AdaBoost (0.48), and LR (0.44) show relatively modest performance.

Since KNN is sensitive to feature scales, all variables were standardized using StandardScaler prior to model training. The scaler was fitted on the training set and applied to both training and testing sets to ensure no data leakage. Additionally, to address KNN's sensitivity to class distribution and local sample density, SMOTE+Tomek was applied exclusively to the training set to balance the classes, and the 'weights = 'distance'' parameter was used to give greater influence to closer neighbors.

To evaluate KNN's performance on the original (unbalanced) test set, the confusion matrix and class-wise metrics are reported in Tables 14 and 15 below.

Table 14. Confusion matrix of KNN on the original test set.

Actual \ Predicted	Predicted	
	Lane-Keeping	Lane-Changing
Lane-Keeping	1969	119
Lane-Changing	30	278

Table 15. Class-wise performance metrics of KNN on the original test set.

Class	F1-Score	Recall	Precision	Support
Lane-Keeping	0.96	0.94	0.98	2088
Lane-Changing	0.79	0.90	0.70	308
accuracy		0.94		2396
macro avg	0.88	0.92	0.84	2396
weighted avg	0.94	0.94	0.95	2396

As shown in Table 14, the KNN classifier correctly identified 1969 out of 2088 lane-keeping instances (94.3%) and 278 out of 308 LC instances (90.3%), resulting in an overall accuracy of 93.8%. The misclassification pattern reveals that 119 lane-keeping samples were falsely flagged as LC (false positives), while only 30 LC samples were missed (false negatives).

Table 15 presents the class-wise performance metrics. For the minority LC class, KNN achieves a recall of 0.90, indicating that 90% of actual LC are successfully detected—a critical property for safety-oriented autonomous driving systems. The precision of 0.70 implies that 70% of predicted LC are correct, corresponding to a false alarm rate of 30%. The macro-average F1-score of 0.88, which equally weights both classes, confirms that the model maintains balanced performance despite the substantial class imbalance (LC accounts for only 13% of test samples). These results demonstrate that KNN, combined with appropriate feature normalization and imbalance processing (SMOTE+Tomek applied exclusively to the training set), generalizes effectively to the original imbalanced test data.

Several observations can be made from these results. First, KNN's strong performance suggests that instance-based learning is particularly well-suited for LC decision prediction in this dataset. Unlike tree-based models that partition the feature space globally, KNN makes local decisions based on neighboring samples. This is advantageous for LC prediction, where the decision boundary between lane-keeping and LC may be irregular and context-dependent. KNN makes no prior assumptions about data distribution and can effectively capture local patterns in the feature space, which may explain its superior performance. Second, GradientBoosting and XGBoost show nearly identical performance. Both are boosting-based ensemble methods that build models sequentially by correcting previous errors. Third, low performance of LR ($F1 = 0.44$) suggests that a linear decision boundary is insufficient for capturing the complexity of LC decisions. The superior performance of KNN and the tree-based models confirms that nonlinear relationships play an important role in this task. Based on these results, KNN is selected as the primary model for subsequent analysis due to its superior predictive performance.

5.4. Contribution Analysis of Multi-Vehicle Expectation Factors

To further validate the contribution of multi-vehicle information factors, several feature configurations are compared using KNN as the base classifier:

- Model A: physical factors + Safety Factors
- Model B: physical factors + Safety Factors + $\bar{v}_T$

- Model C: physical factors + Safety Factors + $\bar{v}_C$
- Model D: physical factors + Safety Factors + $\bar{d}_C$
- Model E: physical factors + Safety Factors + $\bar{d}_T$
- Model F: physical factors + Safety Factors + all multi-vehicle Information Factors

Model A serves as the baseline, including only physical and safety factors (nine features). Models B, C, D, and E add individual multi-vehicle information features to the baseline. Model F includes all four expectation features (13 features). Table 16 presents the performance comparison.

Table 16. Contribution analysis of multi-vehicle information factors.

Model	Added Feature(s)	F1-Score	Recall	Precision	Accuracy	AUC
Model A	Baseline (Physical + Safety)	0.70	0.84	0.60	0.91	0.91
Model B	$\bar{v}_T$ added	0.76	0.87	0.67	0.93	0.94
Model C	$\bar{v}_C$ added	0.74	0.86	0.65	0.92	0.93
Model D	$\bar{d}_C$ added	0.74	0.86	0.64	0.92	0.93
Model E	$\bar{d}_T$ added	0.72	0.85	0.63	0.92	0.93
Model F	All four multi-vehicle information added	0.79	0.90	0.70	0.94	0.97

As shown in Table 16, different multi-vehicle expectation features contribute differently, but all improve prediction performance. Each single-feature addition (Models B–E) achieves higher F1 scores than the baseline Model A (0.70), with improvements ranging from 0.02 to 0.06. This indicates that all four expectation features—including speed and spacing information from both current and target lanes—provide valuable predictive information for LC decisions. Second, the average velocity of the target lane ($\bar{v}_T$) is the most valuable individual expectation feature. Model B, which adds $\bar{v}_T$, achieves the highest F1 score (0.76) among all single-feature addition models, which suggests that the traffic efficiency of the target lane (measured by average speed) is the primary consideration for drivers when making LC decisions.

5.5. Generalization Assessment: Trajectory-Level Split

The preceding experiments used random sample-level splitting, where individual time steps from the same vehicle trajectory could appear in both training and testing sets. While this approach maximizes sample size for model comparison, it may lead to optimistic performance estimates due to potential data leakage. To assess the generalization capability of models to unseen vehicles and address this methodological concern, an additional experiment using trajectory-level splitting was also conducted in this section.

In this evaluation protocol, all time steps belonging to the same vehicle trajectory (identified by the combination of vehicle ID and time period) were assigned exclusively to either the training set (80% of trajectories) or the testing set (20% of trajectories). This ensures that the model is evaluated on entire maneuvers from vehicles it has never seen during training, providing a more realistic assessment of generalization performance.

Table 17 presents the performance of all six baseline models under this trajectory-level split, with SMOTE+Tomek applied for imbalance processing and hyperparameters optimized via grid search. Table 18 summarizes the performance differences between random splitting and trajectory-level splitting for direct comparison.

As shown in Table 17, GradientBoosting achieves the best overall performance under trajectory-level splitting (F1 = 0.47, AUC = 0.83), followed by RandomForest (F1 = 0.45) and XGBoost (F1 = 0.45). Compared with the random splitting results reported in Section 5.3, several observations can be made.

Table 17. Performance comparison of baseline models under trajectory-level split.

Model	F1-Score	Recall	Precision	Accuracy	AUC
GradientBoosting	0.47	0.58	0.39	0.86	0.83
RF	0.45	0.55	0.38	0.86	0.83
XGBoost	0.45	0.48	0.42	0.87	0.83
ADA	0.42	0.56	0.34	0.84	0.82
LR	0.38	0.47	0.31	0.83	0.77
KNN	0.38	0.39	0.37	0.86	0.72

Table 18. Performance comparison between random splitting and trajectory-level splitting.

Model	Random Split (F1)	Trajectory-Level Split (F1)	Difference
KNN	0.79	0.38	−0.41
XGBoost	0.68	0.45	−0.23
GradientBoosting	0.68	0.47	−0.21
RF	0.58	0.45	−0.13
ADA	0.48	0.42	−0.06
LR	0.44	0.38	−0.06

First, the performance gap reveals substantial driver heterogeneity. Under random splitting, KNN achieved an F1-score of 0.79, while under trajectory-level splitting, the best-performing model (GradientBoosting) achieved an F1-score of only 0.468—a relative decrease of approximately 41%. This finding aligns with recent safety modeling studies on driver takeover behavior. Shao et al. [42] demonstrated that aggressive drivers respond faster to takeover requests but exhibit reduced post-takeover stability, while cautious drivers show the opposite pattern. This parallel suggests that LC decisions, like takeover responses, are highly driver-specific and may require personalized modeling approaches for accurate prediction across diverse driver populations.

Second, the relative ranking of models changes under trajectory-level splitting. GradientBoosting and RandomForest outperform XGBoost and KNN, suggesting that tree-based ensemble methods may generalize better to unseen drivers than instance-based methods like KNN, which are more sensitive to vehicle-specific feature distributions.

Together, these findings confirm that LC behavior exhibits significant driver heterogeneity, and that tree-based ensemble methods are more suitable than KNN for generalization to unseen drivers.

From a practical deployment perspective, the trajectory-level split results offer two implications. First, random sample-level splitting risks over-optimistic performance estimates. In real-world deployment, models encounter entirely new vehicles unseen during training. Trajectory-level splitting, which assigns all time steps of a given vehicle exclusively to either training or test set, simulates this scenario more faithfully and provides a more realistic estimate of generalization performance. Second, the substantial performance drop under trajectory-level splitting suggests that driver heterogeneity may be a relevant factor. This finding motivates future investigation into whether personalized or driver-adaptive modeling approaches could further enhance prediction accuracy for individual drivers. These findings advocate for trajectory-level data partitioning in model evaluation and adaptive LC prediction systems.

6. Conclusions

This paper conducted a comprehensive analysis of factors influencing discretionary LC decisions using the US101 trajectory dataset. The main findings are presented below, distinguishing between results directly supported by the data and broader implications that require further validation.

Findings directly supported by the dataset:

1. Multi-vehicle information contributes to prediction accuracy. Adding the average speed and average spacing of five preceding vehicles to the baseline feature set improved KNN's F1-score from 0.70 to 0.79.
2. Safety-related features rank higher than benefit-related features in feature importance. In the XGBoost ranking, d_{Lag} (spacing with the following vehicle in the target lane, 0.187) ranks highest, while benefit-related features, such as $\bar{v}_T$ (0.091), rank lower. SHAP analysis shows a consistent pattern.
3. SMOTE+Tomek showed the best performance among the tested imbalance processing methods in the random split setting. Under random splitting, SMOTE+Tomek achieved the highest F1-score (0.68) and the most balanced trade-off between recall (0.79) and precision (0.59) among the five methods evaluated.
4. KNN achieves the best performance under random splitting (F1 = 0.79, AUC = 0.97). However, under trajectory-level splitting (evaluating on unseen vehicles), the best F1-score drops to 0.47, revealing a substantial performance gap. The results indicate that tree-based ensemble methods generalize better to unseen drivers than instance-based methods such as KNN.

Broader implications requiring further validation:

1. The observed feature importance hierarchy (safety features received the highest importance scores, followed by benefit features) is broadly consistent with a two-stage decision logic in which safety could be considered before potential gains. However, feature importance indicates correlation, not causation; controlled experiments would be needed to confirm causal driver behavior.
2. The performance gap observed between random and trajectory-level splitting suggests that driver heterogeneity could be a factor influencing LC behavior. The specific driver characteristics (e.g., driving style, risk perception) that might relate to this heterogeneity merit further investigation.

Several limitations of this study should be acknowledged:

1. Imbalance between leftward and rightward LCs. The dataset contains 73 rightward (31%) and 164 leftward (69%) LC trajectories, reflecting the actual traffic composition of the US101 segment (an auxiliary lane on the right makes leftward LCs more frequent). Future work should validate findings on datasets with more balanced direction distributions.
2. Single dataset and road segment. This study is based exclusively on the US101 dataset collected on a single freeway segment during morning peak hours. Quantitative results should be interpreted within this context, and future work should validate qualitative conclusions on other datasets (e.g., high D).
3. Our labeling strategy ([LC start, LC end]) corresponds to maneuver recognition rather than early intention prediction. Future work should investigate pre-maneuver labeling strategies (e.g., labeling the 2–3 s before LC start) to enable early warning applications.
4. This paper focused exclusively on successfully completed LCs. Aborted s may involve different behavioral patterns and thresholds. Thus, the generalizability of our findings to aborted maneuvers is not guaranteed. Future work could extend the analysis to aborted events to reveal potential differences in decision-making mechanisms.
5. This study is based on offline classification, not real-time vehicle control or field validation. The reported metrics reflect retrospective accuracy, not real-time performance. Therefore, the findings should be interpreted as contributions to LC behavior modeling rather than direct evidence of autonomous driving reliability. Future work should test the approach in real-time simulation or closed-loop environments.

Author Contributions: Conceptualization, J.Y. and C.Z.; methodology, J.Y. and C.Z.; software, C.Z.; validation, C.Z. and A.A.; formal analysis, J.Y. and C.Z.; investigation, J.Y. and C.Z.; resources, C.Z.; data curation, C.Z. and A.A.; writing—original draft preparation, C.Z.; writing—review and editing, J.Y.; visualization, C.Z. and A.A.; supervision, J.Y.; project administration, J.Y.; funding acquisition, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shanghai Planning Office of Philosophy and Social Sciences (2023EGL005).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Acknowledgments: The authors gratefully acknowledge the financial support from the Shanghai Planning Office of Philosophy and Social Science, and the Laboratory of CACMS (Tianjin University).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LC	Lane-Changing
LCV	Lane-Changing Vehicle
SM	Safety Margin
ROS	Random Oversampling
SMOTE	Synthetic Minority Oversampling Technique
ADASYN	Adaptive Synthetic Sampling
RUS	Random Undersampling
RF	Random Forest
LR	Logistic Regression
KNN	K-Nearest Neighbors

References

- Li, X.; Sun, J.-Q. Studies of vehicle lane-changing dynamics and its effect on traffic efficiency, safety and environmental impact. *Phys. A Stat. Mech. Its Appl.* **2017**, *467*, 41–58. [\[CrossRef\]](#)
- Ma, C.; Li, D. A review of vehicle lane change research. *Phys. A Stat. Mech. Its Appl.* **2023**, *626*, 129060. [\[CrossRef\]](#)
- Tian, J.; Zhu, C.; Chen, D.; Jiang, R.; Wang, G.; Gao, Z. Car following behavioral stochasticity analysis and modeling: Perspective from wave travel time. *Transp. Res. Part B Methodol.* **2021**, *143*, 160–176. [\[CrossRef\]](#)
- Chen, Y.; Dong, C.; Lyu, K.; Shi, X.; Han, G.; Wang, H. A review of car-following and lane-changing models under heterogeneous environments. *Phys. A Stat. Mech. Its Appl.* **2024**, *654*, 130127. [\[CrossRef\]](#)
- Kesting, A.; Treiber, M.; Helbing, D. General Lane-Changing Model MOBIL for Car-Following Models. *Transp. Res. Rec. J. Transp. Res. Board* **2007**, *1999*, 86–94. [\[CrossRef\]](#)
- Li, M.; Yu, X.; Zou, Y. A study of mixed-traffic lane change decision for connected and autonomous vehicles considering driving styles. *Phys. A Stat. Mech. Its Appl.* **2026**, *686*, 131331. [\[CrossRef\]](#)
- Zhang, H.; Wang, Y.; Li, D.; Li, J.; Cao, Y.; Cheng, Y.; Ranjitkar, P. How does lane status influence drivers' lane-change decisions?—An analysis based on naturalistic driving data. *Transp. Res. Part F Traffic Psychol. Behav.* **2026**, *117*, 103453. [\[CrossRef\]](#)
- Pan, J.; Shen, Y.; He, C.; Shi, J. TGLD: A trust-aware game-theoretic lane-changing decision framework for automated vehicles in heterogeneous traffic. *Accid. Anal. Prev.* **2026**, *227*, 108365. [\[CrossRef\]](#) [\[PubMed\]](#)
- He, J.; Hu, Y.; Zhang, W.; Zheng, Z.; Lu, W.; Wang, T. Enhancing Traffic Safety and Efficiency with GOLC: A Global Optimal Lane-Changing Model Integrating Real-Time Impact Prediction. *Technologies* **2025**, *13*, 410. [\[CrossRef\]](#)
- Li, W.; Yang, C.; Zhou, X.; Liu, W.; Zheng, G. Situation-Aware Causal Inference-Driven Vehicle Lane-Changing Decision-Making. *Appl. Sci.* **2025**, *15*, 8864. [\[CrossRef\]](#)
- Singh, K.; Li, B. Estimation of Traffic Densities for Multilane Roadways Using a Markov Model Approach. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4369–4376. [\[CrossRef\]](#)
- Jin, C.-J.; Knoop, V.L.; Li, D.; Meng, L.-Y.; Wang, H. Discretionary lane-changing behavior: Empirical validation for one realistic rule-based model. *Transp. A Transp. Sci.* **2019**, *15*, 244–262. [\[CrossRef\]](#)

13. Liu, X.; Hang, P.; Wang, Y.; Sun, J. A cooperative decision-making method for CAVs from the perspective of opinion dynamics. *Transp. Res. Part C Emerg. Technol.* **2026**, *182*, 105412. [\[CrossRef\]](#)
14. Wang, J.; Wang, H.; Fei, M.; Zhou, G. Vehicle Lane Changing Game Model Based on Improved SVM Algorithm. *World Electr. Veh. J.* **2024**, *15*, 505. [\[CrossRef\]](#)
15. Khelfa, B.; Ba, I.; Tordeux, A. Predicting highway lane-changing maneuvers: A benchmark analysis of machine and ensemble learning algorithms. *Phys. A Stat. Mech. Its Appl.* **2023**, *612*, 128471. [\[CrossRef\]](#)
16. Mechernene, A.; Judalet, V.; Chaibet, A.; Boukhni, M. Detection and Risk Analysis with Lane-Changing Decision Algorithms for Autonomous Vehicles. *Sensors* **2022**, *22*, 8148. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Ali, Y.; Hussain, F.; Bliemer, M.C.J.; Zheng, Z.; Haque, M.M. Predicting and explaining lane-changing behaviour using machine learning: A comparative study. *Transp. Res. Part C Emerg. Technol.* **2022**, *145*, 103931. [\[CrossRef\]](#)
18. Li, D.; Ma, C. Research on lane change prediction model based on GBDT. *Phys. A Stat. Mech. Its Appl.* **2022**, *608*, 128290. [\[CrossRef\]](#)
19. Sun, Q.; Wang, C.; Fu, R.; Guo, Y.; Yuan, W.; Li, Z. Lane change strategy analysis and recognition for intelligent driving systems based on random forest. *Expert Syst. Appl.* **2021**, *186*, 115781. [\[CrossRef\]](#)
20. Sun, H.; Cheng, Q.; Wang, P.; Huang, Y.; Liu, Z. Lane change decision prediction: An efficient BO-XGB modelling approach with SHAP analysis. *Transp. A Transp. Sci.* **2026**, *22*, 2372020. [\[CrossRef\]](#)
21. Ren, G.; Zhang, Y.; Liu, H.; Zhang, K.; Hu, Y. A New Lane-Changing Model with Consideration of Driving Style. *Int. J. Intell. Transp. Syst. Res.* **2019**, *17*, 181–189. [\[CrossRef\]](#)
22. Liu, H.; Wang, T.; Li, W.; Ye, X.; Yuan, Q. Lane-change intention recognition considering oncoming traffic: Novel insights revealed by advances in deep learning. *Accid. Anal. Prev.* **2024**, *198*, 107476. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Guo, H.; Keyvan-Ekbatani, M.; Xie, K. Lane change detection and prediction using real-world connected vehicle data. *Transp. Res. Part C Emerg. Technol.* **2022**, *142*, 103785. [\[CrossRef\]](#)
24. Xie, D.-F.; Fang, Z.-Z.; Jia, B.; He, Z. A data-driven lane-changing model based on deep learning. *Transp. Res. Part C Emerg. Technol.* **2019**, *106*, 41–60. [\[CrossRef\]](#)
25. Guo, H.; Keyvan-Ekbatani, M.; Xie, K. Modeling coupled driving behavior during lane change: A multi-agent Transformer reinforcement learning approach. *Transp. Res. Part C Emerg. Technol.* **2024**, *165*, 104703. [\[CrossRef\]](#)
26. Li, G.; Yang, Y.; Li, S.; Qu, X.; Lyu, N.; Li, S.E. Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness. *Transp. Res. Part C Emerg. Technol.* **2022**, *134*, 103452. [\[CrossRef\]](#)
27. Chen, Z.; Zhu, K. Risk-rule filtered LSTM-dueling DQN for autonomous lane change decision. *Traffic Inj. Prev.* **2026**, in press. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Wang, Q.; Li, Z.; Li, L. Investigation of Discretionary Lane-Change Characteristics Using Next-Generation Simulation Data Sets. *J. Intell. Transp. Syst.* **2014**, *18*, 246–253. [\[CrossRef\]](#)
29. Cambridge Systematic. *NGSIM US-101 Data Analysis*; Cambridge Systematic: Oakland, CA, USA, 2005.
30. Ali, Y.; Zheng, Z.; Mazharul Haque, M.; Yildirimoglu, M.; Washington, S. Detecting, analysing, and modelling failed lane-changing attempts in traditional and connected environments. *Anal. Methods Accid. Res.* **2020**, *28*, 100138. [\[CrossRef\]](#)
31. Thiemann, C.; Treiber, M.; Kesting, A. Estimating Acceleration and Lane-Changing Dynamics from Next Generation Simulation Trajectory Data. *Transp. Res. Rec. J. Transp. Res. Board* **2008**, *2088*, 90–101. [\[CrossRef\]](#)
32. Zhu, C.; Zhong, S.; Ma, S. Two-lane lattice hydrodynamic model considering the empirical lane-changing rate. *Commun. Nonlinear Sci. Numer. Simul.* **2019**, *73*, 229–243. [\[CrossRef\]](#)
33. Toledo, T.; Zohar, D. Modeling Duration of Lane Changes. *Transp. Res. Rec.* **2007**, *1999*, 71–78. [\[CrossRef\]](#)
34. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [\[CrossRef\]](#)
35. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [\[CrossRef\]](#)
36. He, H.; Bai, Y.; Garcia, E.A.; Li, S.J.I. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*; IEEE: New York, NY, USA, 2008.
37. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [\[CrossRef\]](#)
38. Chen, X.-Q.; Xie, W.-J.; Shi, J.; Shi, Q.-X. Perturbation and Stability Analysis of the Multi-Anticipative Intelligent Driver Model. *Int. J. Mod. Phys. C* **2010**, *21*, 647–668. [\[CrossRef\]](#)
39. Treiber, M.; Kesting, A.; Helbing, D. Delays, inaccuracies and anticipation in microscopic traffic models. *Phys. A Stat. Mech. Its Appl.* **2006**, *360*, 71–88. [\[CrossRef\]](#)
40. Lu, G.; Cheng, B.; Lin, Q.; Wang, Y. Quantitative indicator of homeostatic risk perception in car following. *Saf. Sci.* **2012**, *50*, 1898–1905. [\[CrossRef\]](#)

41. Shao, Y.; Han, Z.; Shi, X.; Zhang, Y.; Ye, Z. Risk-informed longitudinal control in autonomous vehicles: A safety potential field modeling approach. *Phys. A Stat. Mech. Its Appl.* **2024**, *633*, 129419. [[CrossRef](#)]
42. Shao, Y.; Xu, Y.; Zhang, Y.; Ye, Z. From prediction to prevention: Safety modeling of driver takeover time with mental workload, risk perception, and driving style in ramp scenarios. *Accid. Anal. Prev.* **2026**, *233*, 108565. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.