



Article

Vehicle Trajectory-Prediction Method Based on Driver Behavior-Classification and Informer Models

Jianyu Su ¹, Muyang Li ¹, Langqian Zhu ¹, Sijia Zhang ^{1,2,3,*} and Mingjian Liu ^{1,2,3}

¹ College of Information Engineering, Dalian Ocean University, Dalian 116023, China

² Key Laboratory of Environment Controlled Aquaculture, Ministry of Education, Dalian Ocean University, Dalian 116023, China

³ Dalian Key Laboratory of Smart Fisheries, Dalian 116023, China

* Correspondence: zhangsijia@dlou.edu.cn; Tel.: +86-13889616966

Abstract: In order to improve the accuracy of vehicle trajectories and ensure driving safety, and considering the differences in driver behavior and the impact of these differences on vehicle trajectories, a vehicle trajectory-prediction method (DBC-Informer) based on the categorization of driver behavior is proposed: firstly, the characteristic driver feature data are extracted through data preprocessing; secondly, descriptive statistical data are obtained through the classification of the driver's behavior into categories; finally, based on the Informer model, a two-layer driver category trajectory-prediction network architecture is established, which inputs the vehicle trajectories of different driving types into independent prediction sub-networks, respectively, to realize the accurate prediction of vehicle trajectories. The experimental results show that the MAE and MSE values of trajectory prediction of the DBC-Informer model in different time domains are much smaller than those of other comparative models, and the improvement of accuracy is more obvious in the long-term domain trajectory-prediction task scenario, and the increase in prediction error of the DBC-Informer model is significantly reduced after the prediction time exceeds 1 s. The on-line behavioral categorization is achieved by comparing different categorization models; it reaches 98% in classification accuracy and, according to the results of ablation experiments, the addition of the driver behavior-classification method to the prediction model improves the accuracy of prediction in longitudinal and lateral motion by 56% and 61%, respectively, which verifies the effectiveness of the driver behavior-classification method. It can be seen that the DBC-Informer model can more accurately portray the effects of different driving behaviors on vehicle trajectories and improve the accuracy of vehicle trajectory prediction, which provides important data support for vehicle warning systems.

Keywords: traffic engineering; trajectory prediction; driver behavior classification; Informer model



Citation: Su, J.; Li, M.; Zhu, L.; Zhang, S.; Liu, M. Vehicle Trajectory-Prediction Method Based on Driver Behavior-Classification and Informer Models. *World Electr. Veh. J.* **2024**, *15*, 237. <https://doi.org/10.3390/wevj15060237>

Academic Editor: Joeri Van Mierlo

Received: 2 May 2024

Revised: 26 May 2024

Accepted: 27 May 2024

Published: 29 May 2024



Copyright: © 2024 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) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, automated driving technology has made significant progress and is seen as an important component of future transportation systems [1,2]. As the level of autonomous driving increases, accurate vehicle trajectory-prediction and optimized urban intersection control methods are indispensable to achieve safe and efficient autonomous driving when the fifth level of self-driving cars is reached [3]. In mixed traffic, although self-driving cars can realize intersection-control methods by communicating with each other and centrally planning their driving [4], precise vehicle trajectory-prediction technology is the prerequisite and foundation for their efficient and accurate operation. However, how to accomplish accurate prediction of vehicle trajectory is still an urgent problem to be solved because the future trajectory of the vehicle is affected by a variety of complex factors, such as driver behavior, road conditions and traffic environment.

In response to the above needs, scholars at home and abroad have carried out relevant research on vehicle trajectory prediction. According to a summary of the research results of

scholars at home and abroad, the methods of vehicle trajectory prediction can be divided into three categories: (1) based on physical models; (2) based on statistical models; (3) based on machine learning prediction methods.

Physical model-based vehicle trajectory-prediction methods build trajectory-prediction models based on the physical motion principles of vehicles, such as speed, acceleration, steering, etc., and take into account the mechanical characteristics of the vehicle, such as friction, inertia, etc. Ju C et al. [5] proposed a multilayered architecture of an interaction-aware Kalman neural network, which aims to address the interaction effects faced by self-driving cars in complex traffic environment challenges. Wang H et al. [6] integrated dynamics models into an improved Model Predictive Control (MPC) framework. Jiang Y [7] et al. integrated the driver's intention, maneuvering behavior, and vehicle dynamics to propose a probabilistic vehicle trajectory-prediction method based on a Dynamic Bayesian Network (DBN) model. Akhtar S et al. [8] used multiple models to describe the target's motion behavior and achieved estimation and prediction of the target state by interacting with each other as well as smooth variable structure control. The above methods usually perform more accurately in short-term prediction because the physical models are able to take into account the kinematics and dynamics of the vehicle. However, their predictive ability may be limited when facing complex and uncertain traffic scenarios, e.g., the complexity and diversity of environmental factors, and the intention and behaviors of traffic participants are often subject to a certain degree of uncertainty, which is a challenge for physical models. As a result, in the long term trajectory prediction may be highly biased.

Statistical model-based approaches to vehicle trajectory prediction use regression methods to establish relationships between vehicle trajectories and variables such as time and space, as well as statistical learning or probability distribution modeling to estimate the likely position and speed of a vehicle at future moments. Li J et al. [9] proposed a conditional generative neural system for probabilistic trajectory prediction to approximate data distributions, which allows for realistic, feasible and diverse future trajectory hypotheses. Cui H et al. [10] proposed a method using deep convolutional networks to predict multiple possible trajectories of a participant while estimating their probabilities. Ivanovic B et al. [11] proposed a deep generative model-based interactive vehicle trajectory-prediction method by learning interactive relationships and multimodal trajectory representations in traffic scenarios, in which future trajectories are used to generate multimodal probability distributions. Ma Y et al. [12] proposed a real-time traffic prediction algorithm TrafficPredict based on long and short term memory, which uses an instance layer to learn the movement and interaction of instances and has a category layer to learn the similarity of instances belonging to the same type. However, statistical model-based approaches for vehicle trajectory prediction require high data quality, which may affect the prediction if the input data are noisy or missing, and may have limitations for complex, nonlinear trajectory-prediction problems. Meanwhile, the prediction accuracy of statistical models may not be as good as other methods, such as deep learning, when facing complex traffic environments or unexpected events.

Most of the machine learning-based vehicle trajectory prediction utilizes deep neural networks for modeling and prediction of trajectory data, such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), etc. Deo N et al. [13] extracted features from trajectories of neighboring vehicles for predicting the future trajectories of vehicles by combining convolutional social pooling with LSTMs. Chen X et al. [14] proposed a novel spatio-temporal dynamic attention network for vehicle trajectory prediction that comprehensively captures spatio-temporal and social patterns in a hierarchical manner. Chandra R et al. [15] used a novel hybrid LSTM-CNN network to model the interactions between different road agents. Liu Yicheng et al. [16] used Transformer as the main neural network structure and incorporated an attention mechanism for the multimodal motion prediction task. Messaoud K et al. [17] combined an attention mechanism with an LSTM model to capture important information in the input sequences more efficiently. Although there have been studies proposing trajectory-prediction methods based on interaction

perception, these methods tend to ignore the influence of driver behavior on vehicle trajectory; in fact, the driver's behavior largely determines the vehicle's trajectory, including acceleration, deceleration and cornering behaviors. Meanwhile, these neural network models produce the problem of gradient explosion when the amount of input data is large, while the Informer model has better long-sequence modeling ability and higher computational efficiency.

In order to solve the above problems, this study proposes a vehicle trajectory-prediction method based on Driver Behavior Classification and the Informer algorithm (Driver Behavior Classification and Informer, DBC-Informer), to achieve accurate prediction of vehicle trajectory and to ensure driving safety.

The main contributions of this study are as follows:

A new vehicle trajectory-prediction model, DBC-Informer, which integrates driving behavior classification and the Informer algorithm, is proposed.

A real-time driving behavior-classification method based on historical driving data is proposed.

The historical trajectory data of different driving styles are analyzed through deep learning to mine the potential laws and patterns to achieve accurate trajectory prediction.

This paper is organized as follows:

- Section 2 defines the problem and explains the specific methodology;
- Section 3 describes the experimental setup and dataset;
- Section 4 presents the experimental results and discussion;
- Section 5 summarizes the conclusions of the study and future work.

2. Materials and Methods

The vehicle trajectory-prediction problem refers to predicting the trajectory of a vehicle at a future moment based on given historical trajectory data and environmental information. This problem is particularly important in vehicle warning systems because accurate trajectory prediction can significantly improve driving safety and driving efficiency. Traditional trajectory-prediction methods focus on the interaction between vehicles, but ignore the individual behavioral differences of drivers. However, driver behavior (e.g., accelerating, decelerating, turning, etc.) largely determines the vehicle trajectory. Therefore, a trajectory-prediction method that can take driving behavior into account is needed to improve the accuracy and reliability of prediction.

In order to achieve accurate prediction of vehicle trajectory and thus ensure the safety of vehicle driving, this study proposes a vehicle trajectory-prediction method considering different drivers' driving behaviors. The overall architecture of the model is shown in Figure 1. The method fully considers the influence of different drivers' driving behaviors on the prediction of vehicle trajectories: firstly, based on the historical characteristic data of vehicle driving, it classifies the driving styles of drivers online in real time; secondly, through the in-depth learning and analysis of the historical trajectory characteristic data of drivers with different driving styles, it mines out the potential laws and patterns in these trajectory data to complete the accurate prediction of vehicle trajectories within the foreseeable time. Secondly, through in-depth learning and analysis of the historical trajectory characteristic data of drivers with different driving styles, the potential laws and patterns in these trajectory data are mined to complete the accurate prediction of vehicle trajectories in the foreseeable time.

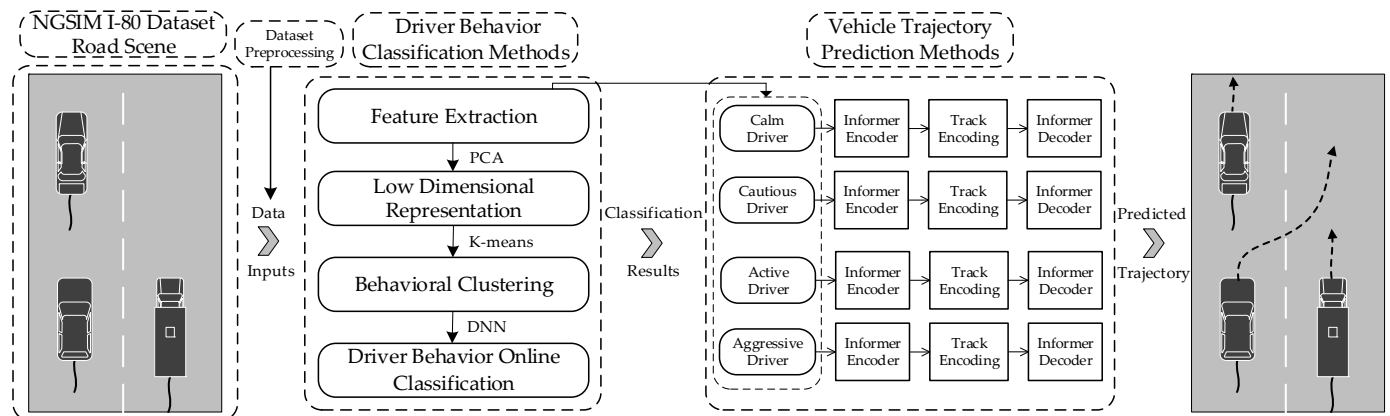


Figure 1. DBC-Informer model architecture.

2.1. Driver Behavior-Classification Methods

In order to more accurately classify drivers' driving behaviors, features that can highlight the differences in drivers' personalities are refined based on observable partial vehicle trajectory data, and at the same time, in order to simplify the data processing and retain the core feature information, the data features are downsampled using the Principal Component Analysis (PCA) method. Subsequently, the simplified data distribution was clustered using the K-means algorithm, so as to categorize drivers' driving behaviors into four different categories. In addition, a deep neural network (DNN) is introduced to receive the driver behavior class labels from the K-means algorithm, and online behavior labels are applied to the driver behaviors in the test set by observing the driver behavioral data in order to achieve efficient segmentation of the driver behaviors and to provide data support for improving the accuracy of trajectory prediction.

2.1.1. Vehicle Trajectory Data Feature Selection

After completing the preprocessing of the vehicle historical trajectory data, reference [18,19] selected a series of important and descriptive driving behavior characteristics data, as shown in Table 1. These descriptive statistics provide detailed information about vehicle driving behavior. By analyzing these data, the motion patterns and driving behaviors of vehicles can be better understood, and the subtle differences in driving behaviors can be more accurately captured, which in turn provides a basis for the precise classification of driver categories.

Table 1. Characteristic data.

Characteristic Data	Notation	Unit
Average lateral velocity	$ \bar{v}_x $	$\frac{m}{s}$
Maximum lateral speed	$ v_x^{max} $	$\frac{m}{s}$
Standard deviation of transverse velocity	$\sigma(v_x)$	$\frac{m}{s}$
Average longitudinal speed	$\bar{v}_y$	$\frac{m}{s}$
Maximum longitudinal speed	v_y^{max}	$\frac{m}{s}$
Standard deviation of longitudinal velocity	$\sigma(v_y)$	$\frac{m}{s}$
Average lateral acceleration	$ \bar{a}_x $	$\frac{m}{s^2}$
Average longitudinal acceleration	$\bar{a}_y$	$\frac{m}{s^2}$
Average speed relative to traffic flow	v_y^{flow}	$\frac{m}{s}$
Average speed relative to the vehicle in front	v_y^{prec}	$\frac{m}{s}$

2.1.2. Low-Dimensional Representation

In the process of model prediction estimation, if all the relevant indicators in Table 1 are used as model inputs, it will inevitably increase the model complexity and reduce the generalization ability of the model to better capture the underlying structure and

characteristics of the data. Therefore, the Principal Component Analysis (PCA) method is used to project each driver characteristic datum in Table 1 into a low-dimensional space for model data processing complexity.

(1) Indicator correlation analysis

In PCA, the existence of correlation between variables is the first condition for principal component analysis. Therefore, the Pearson correlation coefficient was used in this study to conduct correlation analysis of the above characteristic data, and the correlation heat map is shown in Figure 2, which shows a positive and strong correlation between the average longitudinal speed and the average speed relative to the traffic flow, the average longitudinal acceleration and the maximum longitudinal speed, with the correlation coefficients of 0.87, 0.7, 0.57; concerning the maximum transverse speed and the average longitudinal speed, relative to the average speed of the traffic flow there is a moderate degree of negative correlation between the correlation coefficients -0.04 and -0.02 . In addition, it can be seen that the rest of the characteristics of the data between the two have a certain degree of correlation. The correlation analysis can provide the basis for the subsequent data dimensionality reduction.

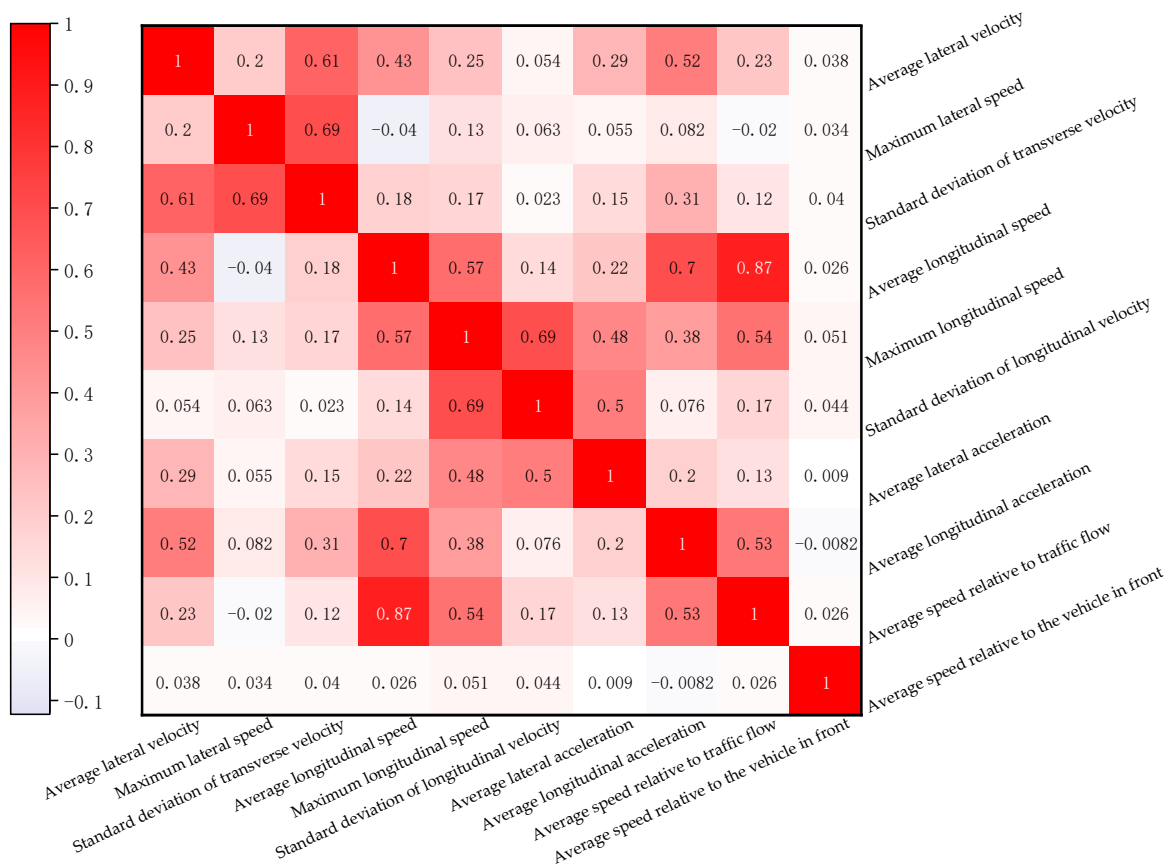


Figure 2. Heat map of feature data correlation.

(2) Data dimensionality reduction

The data are downsampled using the Principal Component Analysis (PCA) algorithm. PCA downscaling not only reduces computational complexity, but also reduces redundant information in the data, which helps the model capture key features more efficiently, thus improving classification accuracy. The process includes calculating the variance contribution rate and cumulative variance contribution rate of the feature data. Retaining the principal components with the top two variance contribution rates can reduce the input dimensions of the model while preserving as many features of the original data as possible.

According to Experiment 3.3, it can be concluded that the total variance of principal component 1 named longitudinal speed and principal component 2 named transverse speed are much larger than the other principal components, and their cumulative variance contribution rate can account for 60% of the total variance, so the behavioral clustering is chosen to be carried out in the dimensional space defined by principal component 1 and principal component 2, and the closer the distance between the two points in this space, the more similar the data of these two points are; i.e., their driving behaviors are more similar.

2.1.3. Behavioral Clustering

After completing the extraction of the main features affecting driving behavior, the K-means clustering method is used to classify drivers, where each cluster represents a unique driver type. In order to determine the optimal number of clusters for the K-means clustering algorithm, firstly, a relatively broad range of values for the optimal number of clusters is determined via the elbow method; secondly, the range is evaluated in a refined way using the profile coefficient method, so as to accurately determine the optimal number of clusters.

(1) Calculate the Within Cluster-Sum of Squared errors (WCSS) for different values of k based on the elbow method to select the value that minimizes the error, as shown in Figure 3, when $k = 2$, continuing to increase the number of clusters does not appear to significantly reduce the WCSS values, and at this point $k = 2$ is considered the optimal number of clusters. However, when $k = 2$, i.e., obtaining two categories of drivers with different driving styles, is not sufficient to capture the diversity of driving behaviors, as aggressive drivers may represent a very small portion of the total number of drivers in the dataset, but due to the low values, these data types may be categorized as outliers and ultimately merged with the other driver categories, generating inaccurate categorization. Therefore, the following constraints were imposed on the selection of k : $k > 2$.

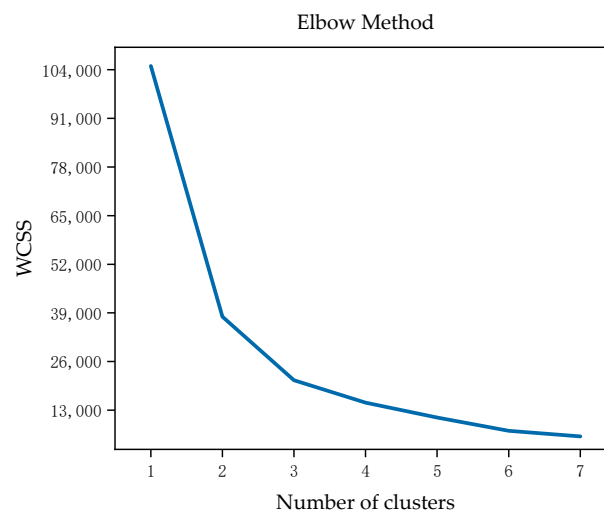


Figure 3. Elbow method index.

(2) On top of the classification results obtained via the elbow method, the contour coefficient method is used to further determine the optimal number of clusters. The optimal number of clusters can be selected by calculating the contour values at different values, as shown in Figure 4. Considering the constraints of the elbow method on the values and ignoring the first peak at $k = 2$, the optimal number of clusters can be obtained as $k = 4$, i.e., four types of driver types with different driving styles are finally obtained.

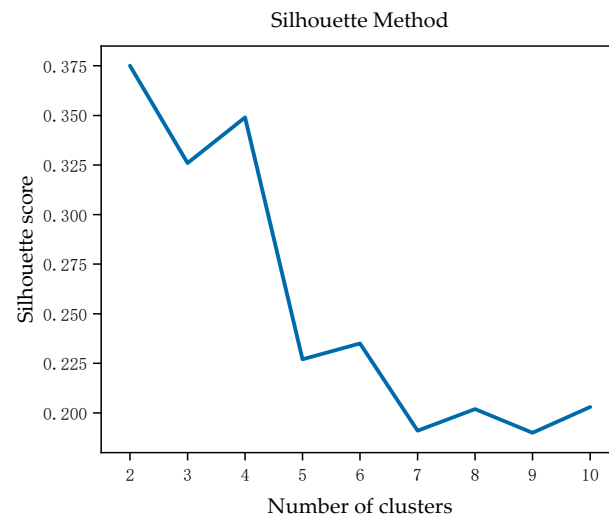


Figure 4. Silhouette coefficient method index.

2.1.4. Driver Behavior Online Classification

In order to accurately classify the drivers in the test set into the four driver categories in Section 2.1.3, the driver behavior class labels from the K-means algorithm and the descriptive statistics from the training set are received in this section, and a deep neural network (DNN) is used to classify the drivers' online behaviors; the DNN network classified in this study is made up of two hidden layers (with 256 hidden units each), the ReLU activation function and SoftMax output layer to complete the real-time accurate driver category classification. The ReLU activation function and SoftMax output layer are used to accomplish real-time accurate driver class classification.

2.2. Vehicle Trajectory-Prediction Methods

Vehicle trajectory prediction is essentially a time series prediction problem. The Informer model not only significantly improves the accuracy of the time series prediction problem by introducing the sparse attention mechanism, but also effectively reduces the demand for computational resources. Therefore, this study designs a two-level trajectory-prediction method based on Informer for the trajectory-prediction problem of different driver categories: firstly, at the vertical level, four driver categories are given in Section 2.1.3, and for each of them, after integrating the information of the state of the surrounding vehicles and their behavioral labels, the vehicle trajectory data with different driving behaviors are inputted into the corresponding independent horizontal prediction methods, and then the trajectory data are input into the corresponding independent horizontal prediction sub-networks in order to realize the accurate prediction of vehicle trajectories for each category; secondly, at the vertical level, the predicted trajectory information output from these four prediction sub-networks is integrated so as to arrive at a comprehensive and accurate vehicle trajectory-prediction result that covers different types of drivers.

2.2.1. Horizontal Hierarchical Network-Prediction Structure

(1) Encoder

The main task in the encoder section of the Informer model for projected trajectory prediction is to encode the input historical vehicle trajectory data into a fixed-length vector that contains key trajectory feature information. The input vehicle data sequence is:

$$L = \{l_1, l_2, \dots, l_t\} \quad (1)$$

In Equation (1), t is the number of time steps of the history trajectory, and for each t , l_t represents the state representation at time step t .

In order for the prediction model to learn the feature representation of the data more efficiently, firstly, the vehicle data sequence L are fed into an embedding layer, which converts the original discrete input data into continuous data, the l_t into continuous data $l_{embed,t}$ and combines them into dense vector representations, L_{embed} . These dense vector representations are usually better able to capture the semantic relationships and syntactic structures between the data, i.e., each l_t is embedded as $l_{embed,t}$, which converts a sequence of vehicle data L converted into a continuous representation of dense vectors:

$$L_{embed} = \{l_{embed,1}, l_{embed,2}, \dots, l_{embed,t}\} \quad (2)$$

Afterwards, this vector is combined with positional coding to help the predictive model better understand the order and positional relationships of the sequences.

Next, it is encoded using the Informer-specific sparse self-attention mechanism, which reduces the complexity of the self-attention computation by introducing probabilistic sparsity to identify weights that can be ignored, i.e., those vehicle historical location points that have little impact on the prediction results. The KL sparsity formula is used to measure the distribution of the attentional weights of the vector p difference between it and the uniform distribution. Specifically, each location of $l_{embed,t}$ will be considered as a query vector, and for the i -th first query vector Q_i , its attentional weight can be regarded as a probability distribution $p(k_j|q_i)$. When some values have a significant effect on the attention weights, the distribution p will deviate from the uniform distribution. In contrast, if the effect of certain values on the attention weights is not significant, the distribution p approaches a uniform distribution and these values can be considered redundant. The probability distribution of the weights is calculated as follows:

$$p(k_j|q_i) = \frac{k(q_i, k_j)}{\sum_1 k(q_i, k_l)} \quad (3)$$

$$q(k_j|q_i) = \frac{1}{L_k} \quad (4)$$

Equation (3) is the attention probability distribution, where $k(q_i, k_i)$ denotes the asymmetric exponential kernel function $\exp\left(\frac{q_i k_j^T}{\sqrt{d}}\right)$. Equation (4) is the uniform distribution, where L_k is the length of Q .

The sparsity measure can be obtained by substituting Equations (3) and (4) into the KL dispersion formula; it is used to measure the degree of deviation between the actual distribution of attentional weights p and the uniform distribution, which is calculated as follows:

$$M(q_i, K) = \ln \sum_{l=1}^{L_k} \exp\left(\frac{q_i k_l^T}{\sqrt{d}}\right) - \frac{1}{L_k} \sum_{j=1}^{L_k} \frac{q_i k_j^T}{\sqrt{d}} \quad (5)$$

If the i -th Q of $M(q_i, K)$ takes a larger value, it means that its corresponding attentional weight is more widely distributed and may cover the part that has more influence in the dot product pairs. Therefore, it is possible to perform a test on all Q of the $M(q_i, K)$ values taken and select a number of the top-ranked Q s as the input matrix of the SoftMax function, and select the number of $u = \text{cln}L_Q$. L_Q denotes the Q time step of the matrix, and c is a constant sampling factor.

The arithmetic process of the sparse attention mechanism can be expressed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (6)$$

In Equation (6), Q is a query vector, and K and V represent the keys and values of the vehicle history states for matching and weighting the history information, respectively.

In order to capture long-term dependencies more efficiently, the Informer model in this study uses a multi-layer encoder structure, where each layer contains the self-attention mechanism described above, i.e., multi-head self-attention, and contains a feed-forward neural network (FFN) to enhance the model's expressive and nonlinear modeling capabilities. The output of the multi-head self-attention is the splicing of the individual heads:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O \quad (7)$$

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (8)$$

In Equation (8), W^O is the output linear transformation weight matrix.

The feedforward network nonlinearly transforms the vectors at each position through two fully connected layers. The feedforward neural is computed as:

$$\text{FFN}(Z) = \text{ReLU}(ZW_1 + b_1)W_2 + b_2 \quad (9)$$

In Equation (9), the Z is the output of the multi-head self-attention, W_1, W_2 is the weight matrix of a fully connected layer and b_1, b_2 are the bias terms.

Finally, the inputs and outputs of each sublayer are residually concatenated and then layer normalized to output the encoded sequence, called the context vector I_t ; this context vector synthesizes information from past moments and will be used as the output of the encoder and will also be used as the input to the decoder, providing the necessary information about the historical state of the vehicle for the next vehicle trajectory-prediction task.

(2) Decoder

The decoder part of the Informer model is responsible for generating future trajectory sequences in vehicle trajectory prediction. Specifically, it utilizes the rich historical information provided by the encoder to predict the next trajectory points.

First, the decoder will generate the target sequence:

$$Y = \{y_1, y_2, \dots, y_T\} \quad (10)$$

This is converted to a continuous representation after an embedding layer:

$$Y_{\text{embed}} = \{y_{\text{embed},1}, y_{\text{embed},2}, \dots, y_{\text{embed},T}\} \quad (11)$$

Positional coding is introduced to preserve the positional information of the target sequence, and the positional coding is computed in the same way as the encoder part.

Next, at each time step, the decoder receives the joint context vector from the encoder I_t as inputs and will process these inputs using a self-attention mechanism similar to that in the encoder; i.e., the context vector I_t is linearly transformed to obtain the hidden state h_t . The following is an example of an encoder–decoder mechanism. Second, the relationship between the input sequence and the target sequence is understood through the encoder–decoder attention mechanism, where the input is the target representation of the decoder Y_{embed} ; the encoder–decoder attention is formulated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (12)$$

In Equation (12), the Q is the input hidden state h_t and K and V are the outputs of the encoder I_t .

Similar to the encoder, in order to predict future states more accurately, the decoder captures the dependencies within the decoder sequence by means of multi-head self-attention, which contains multiple identical hierarchies, each of which produces a prediction that is used as an input to the next layer. The outputs of the multi-head self-attention are spliced together, projected by a linear transformation, and the vectors at each position

are nonlinearly transformed by a feed-forward network, which is computed in the same way as the encoder part of the position feed-forward network. The inputs and outputs of each sublayer of the decoder are summed and then layer normalized to form the final encoder output.

The final output of the encoder is a vector of real numbers, which is converted from floating point numbers to probability indices by a linear transformation, and the SoftMax function converts it to the final output probability of the trajectory distribution $P(O|I)$, and the final vehicle trajectory-prediction result is obtained. The calculation formula is:

$$P(O|I_t) = \sum_i (O|m_i, I_t) P(m_i|I_t) \quad (13)$$

In Equation (13), O is the output quantity, P_θ is the binary Gaussian function and θ is the parameter of the binary Gaussian function.

2.2.2. Vertical Hierarchy

At the longitudinal level, as shown in Figure 5, in order to fully consider the impact of vehicle driving styles on future trajectories, the trajectories of vehicles with different driving style types are input into four separate horizontal networks described above according to the four types of driver style obtained in previous chapters so as to achieve accurate prediction of vehicle trajectories for different types of drivers. Each transversal network specializes in processing data for a specific driver style type to accurately capture its unique driving behaviors and habits.

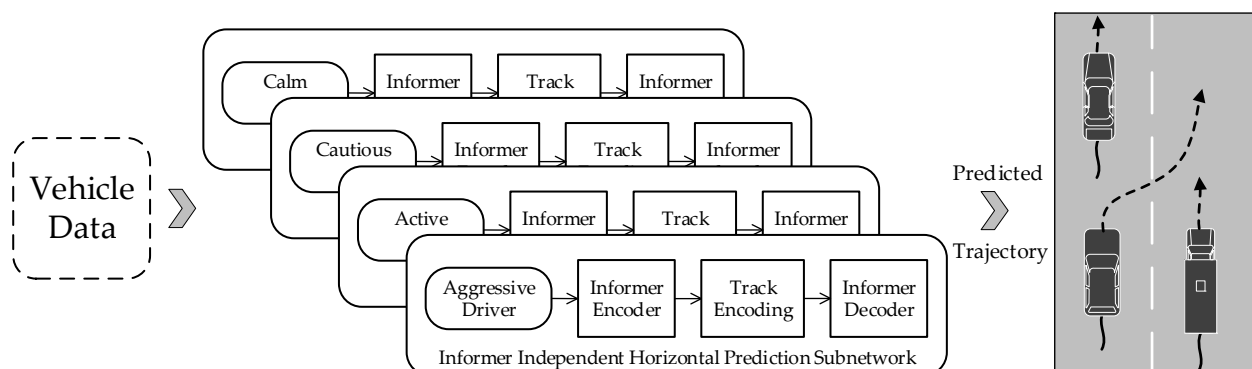


Figure 5. Vertical hierarchical network architecture.

3. Results

3.1. Introduction to Data

This study uses the NGSIM public dataset published by the U.S. Federal Highway Administration to train and test the model. The I-80 section in the dataset, located in Emeryville, California, is approximately 500 m in length and consists of six freeway lanes containing a total of 2355 trajectories, which provide the precise position, speed and acceleration of each vehicle in the study area at 0.1 s intervals.

3.2. Experimental Environment and Assessment Indicators

For the training and testing phases of the model, the simulation environment and hyperparameter settings are shown in Table 2.

In the experimental evaluation part, the performance of the model was evaluated using three evaluation indexes, MAE, MSE and RMSE, with the following formulas.

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (14)$$

$$MSE = \frac{1}{T} \sum_{t=1}^T (x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2 \quad (15)$$

$$RMSE = \frac{\sum_{t=1}^T \sqrt{(\hat{x}_t - x_t)^2 + (\hat{y}_t - y_t)^2}}{T} \quad (16)$$

The experiments in this study are divided into three categories, namely, the validation of the driver behavior-classification method, the validation of the vehicle trajectory-prediction method and the ablation experiment. The performance of the trajectory-prediction model is evaluated by using the evaluation indexes mentioned above, and the experimental effects are analyzed according to the predicted trajectory effect diagrams.

Table 2. Experimental environment.

Simulation Environment	Specific Setting
Operating System	Windows 10
Graphics Card Configuration	NVIDIA 3090 Ti
Programming Framework	Pytorch 1.12.0
Programming Language	Python3.8
Encoder Dimension	64
Batch Size	16
Parameter Optimizer	Adam Optimizer
Learning Rate	0.001

3.3. Validation of Driver Behavior-Classification Methods

After preprocessing and feature extraction of the data, the obtained feature data are applied to the dimensionality reduction implemented via Principal Component Analysis (PCA), as shown in Table 3. According to the data in the table, it is obvious that we keep the first two principal components because their cumulative variance explained rate reaches 60%; i.e., these two principal components are able to express 60% of the information of all the data, and the effect of principal component analysis is better.

Table 3. Variance contribution rate after PCA dimensionality reduction.

Principal Component Number	Eigenvalue	Variance Contribution/%	Cumulative Variance Contribution/%
1	3.603	36.027	36.027
2	2.390	23.895	59.923
3	1.095	10.953	70.876
4	0.884	8.845	79.721
5	0.708	7.082	86.803
6	0.497	4.968	91.771
7	0.418	4.181	95.953
8	0.181	1.811	97.764
9	0.152	1.517	99.281
10	0.072	0.719	100.000

In order to better interpret and understand the underlying dimensions or factors represented by each principal component, the original component matrix after the rotation operation, as shown in Table 4, can be more easily assigned a meaningful name to each principal component, which enables a clearer description of what they represent.

Based on the rotated component matrix, it is considered that the variables with loading coefficients ≤ 0.4 do not have a correspondence with this component, then the first principal component is related to the average longitudinal speed, average speed relative to traffic flow, average longitudinal acceleration, standard deviation of longitudinal speed, maximum longitudinal speed and average speed relative to the vehicle in front of the vehicle, and

the second principal component is related to the average transversal acceleration, standard deviation of transversal speed, maximum transversal speed and maximum transverse velocity. Therefore, it is possible to name the first principal component as longitudinal velocity and the second principal component as transverse velocity.

Table 4. Rotated component matrix.

Name	Factor Loading Factor	
	Principal Component 1	Principal Component 2
Average longitudinal speed	0.974	0.163
Average speed relative to traffic flow	0.870	0.163
Average longitudinal acceleration	0.796	0.053
Standard deviation of longitudinal velocity	0.913	0.007
Maximum longitudinal speed	0.772	0.368
Average lateral acceleration	0.095	0.757
Standard deviation of transverse velocity	0.153	0.932
Maximum lateral speed	−0.137	0.820
Maximum lateral speed	0.071	0.628
Average speed relative to the vehicle in front	0.467	0.038

The low-dimension representation obtained from Principal Component Analysis (PCA), as shown in Figure 6, observes a linear relationship between principal component 1 and principal component 2. In this low-dimension space, the closer the distance between two points, the more similar their data are, i.e., the more similar their driving behaviors are.

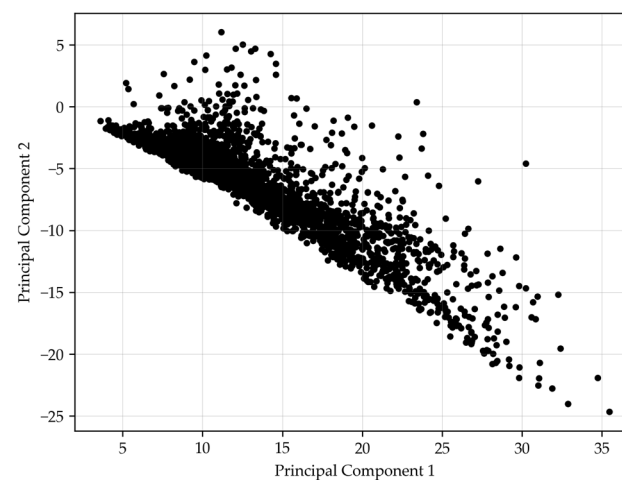


Figure 6. The low-dimensional space formed by principal components 1 and 2.

In the above low-dimensional space, with $k = 4$ as the number of clusters, the K-means clustering algorithm is used for behavioral clustering, as shown in Figure 7. According to the definition of principal component 1 and principal component 2, it can be seen that the longitudinal speed of cluster one to cluster four gradually increases, and it can be judged from the change of transverse speed that the intention of changing lanes of cluster one to cluster four also gradually increases. Therefore, the driving styles of cluster 1 to cluster 4 are calm, cautious, active and aggressive, respectively.

For the online behavior-classification part, we trained Support Vector Machines (SVM) model, Logistic Regression (LR) model, Decision Tree (DT) model, k-nearest Neighbor Classification (KNN) model and Deep Neural Network (DNN) model. The experiments set the learning rate to 0.001, and the evaluation index is cross-entropy loss, and we use the Adam optimizer to train the above networks 200 times. After several experiments to train the above model and several tests, we came up with the results as shown in the

table. According to the results in the Table 5, we can see that the logistic regression model and the k-neighborhood model were trained very quickly and both of them showed better results in terms of classification accuracy (84% and 87% accuracy, respectively); the support vector machine model and the decision tree model improved both in terms of accuracy and training time, and the deep neural network model took a longer time to train but reached 98% in terms of classification accuracy. In order to have a better performance on the prediction work, this study prioritized accuracy over training time, so the deep neural network model was chosen for classification.

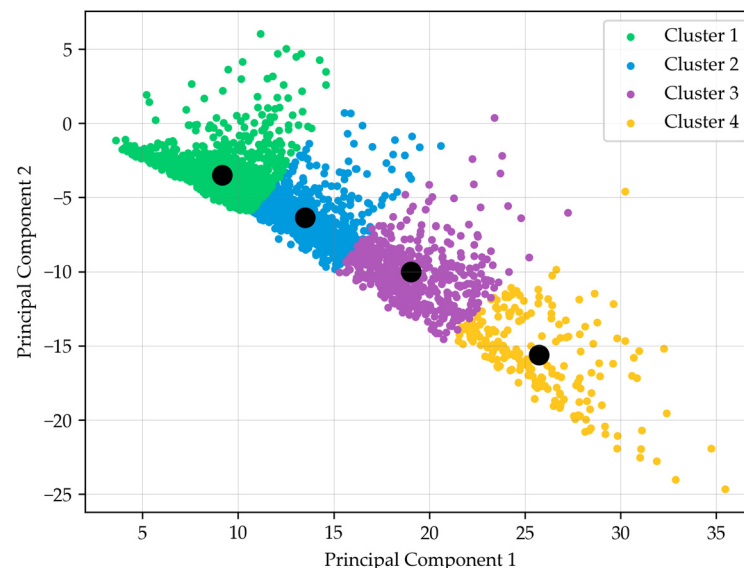


Figure 7. Four clusters formed using the K-means clustering algorithm.

Table 5. Comparison of online action classification models.

Name	Accuracy (%)	Time (s)
SVM	95.341	0.8
LR	84.359	0.08
DT	91.348	0.5
KNN	86.855	0.03
DNN	98.170	1.3

3.4. Validation of Vehicle Trajectory-Prediction Methods

The experiment used 80% of the dataset as a training set, 10% as a testing set and another 10% as a validation set.

On observation of the positive loss value curve of the model, as shown in Figure 8, it can be found that when the model training epoch reaches about 50 times, the model has gradually converged to the fitting state, and the loss value decreases from 0.95 to about 0.30. When the trained model is used in the validation set, the loss value of the validation set decreases from 0.39 to about 0.3 when the number of iterations reaches about 10 times, and then the decreasing trend of the loss value tends to be flat, and at this time the model already has a good convergence, indicating that the model training is effective.

In order to verify the actual performance of trajectory prediction, this paper compares other trajectory-prediction models to verify the accuracy of the trajectory-prediction models in this study. In this paper, the performance of various trajectory-prediction models is evaluated using Mean Absolute Error (MAE) and Mean Square Error (MSE), and the following trajectory models are mainly used for comparative analysis: the Social-LSTM [20] model, CNN-LSTM [21] model, Social-GAN [22] model, STGAT [23] model. The comparative results of trajectory-prediction model displacement prediction are shown in Tables 6 and 7.

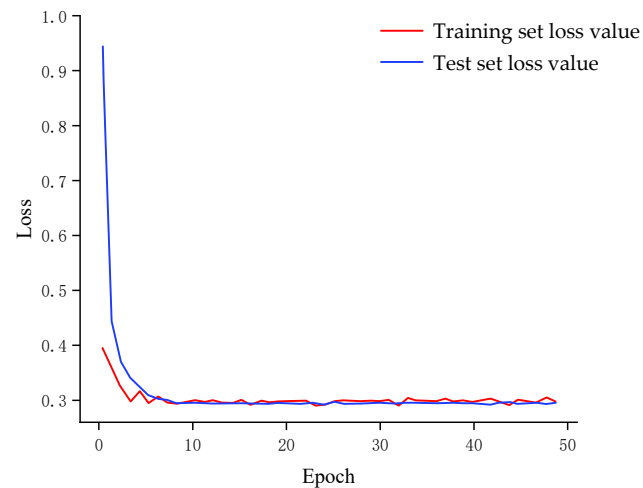


Figure 8. Model positive loss curve.

Table 6. Mean Absolute Error (MAE) of displacement trajectory.

Name	Predicted Time Domain/s				
	1 s	2 s	3 s	4 s	5 s
Social-LSTM	1.02	1.75	3.38	4.84	5.68
CNN-LSTM	0.68	1.77	2.54	3.48	4.76
Social-GAN	0.64	1.79	2.72	3.65	4.84
STGAT	0.62	1.69	2.51	3.69	4.61
DBC-Informer	0.59	1.13	1.82	2.63	3.51

Table 7. Mean Square Error (MSE) of displacement trajectory.

Name	Predicted Time Domain/s				
	1 s	2 s	3 s	4 s	5 s
Social-LSTM	1.51	2.98	3.71	5.22	6.14
CNN-LSTM	0.81	1.86	2.51	3.53	4.77
Social-GAN	0.77	2.03	2.69	3.81	4.70
STGAT	1.08	2.52	2.87	3.53	4.84
DBC-Informer	0.47	1.45	2.23	3.12	3.45

As can be seen from Figures 9 and 10, the DBC-Informer prediction model possesses higher accuracy for trajectory prediction for long-term series.

Among them, the Social-LSTM model has larger MAE and MSE values than other trajectory-prediction models, and its model usually needs to specify a fixed time window size to capture historical trajectories, and this fixed window size may not be able to adequately adapt to different scenarios and different speeds of moving behaviors, which leads to the poor performance of the model in some cases. Therefore, classifying driving behaviors enables the model to fully learn the patterns and laws in the data to achieve good performance and improve the accuracy of trajectory prediction. The MAE and MSE values of the CNN-LSTM model, the Social-GAN model and the STGAT model are also larger than those of the DBC-Informer model, which suggests that the DBC-Informer model can better extract the features of vehicle history data and has better performance in predicting future trajectories over a long period of time.

The actual trajectory-prediction results of the model are shown in Figure 11.

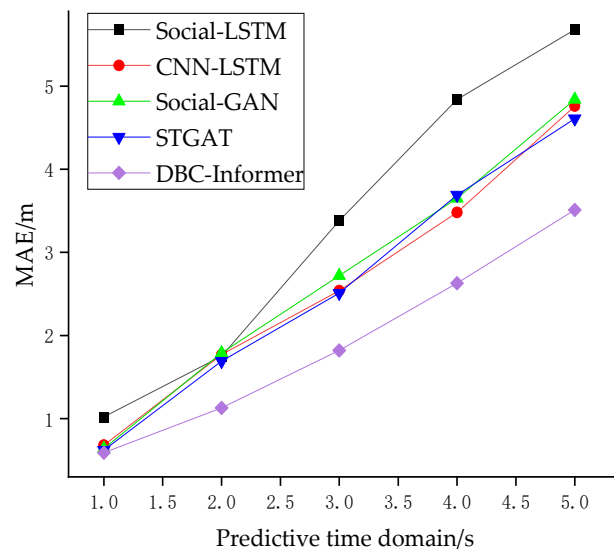


Figure 9. MAE comparison chart.

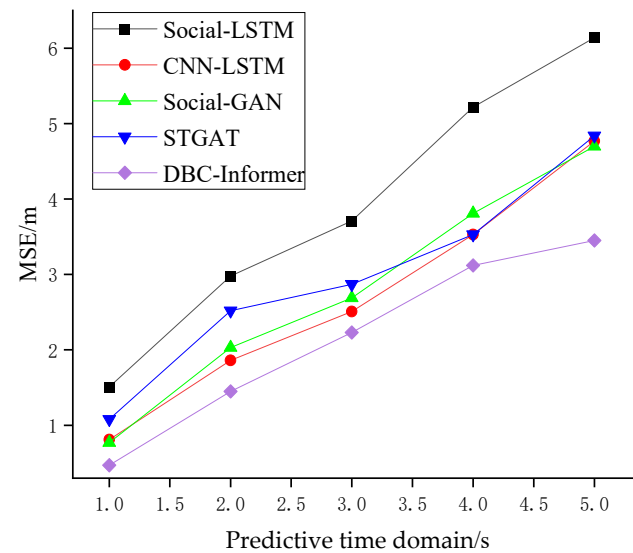


Figure 10. MSE comparison chart.

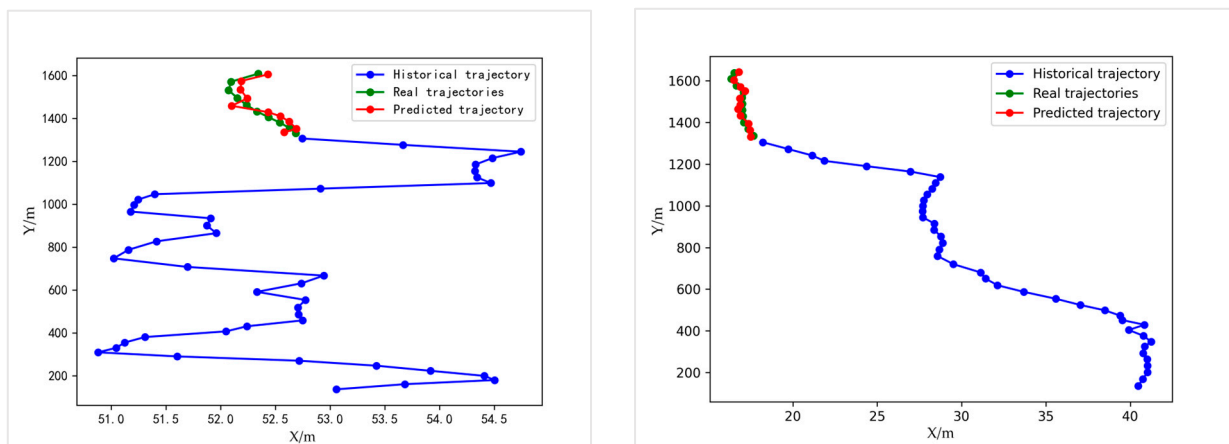


Figure 11. The actual trajectory-prediction effect of the model.

3.5. Ablation Experiments

In order to verify the impact of the driver behavior-classification method on trajectory prediction and its effectiveness, the trajectory-prediction errors for the next 1 s for the target vehicle are listed using the root mean square error (RMSE) to compare the DBC-Informer model with and without driver behavior classification, as shown in Table 8.

Table 8. Comparison of ablation experiments.

Name	Informer	DBC-Informer
Longitudinal motion	0.88	0.39
Lateral motion	0.21	0.08

It can be observed that the driver behavior-classification method is estimated to improve the accuracy of prediction by 56% in longitudinal motion and 61% in lateral motion, which indicates that the driver behavior-classification method in this paper has an impact on trajectory prediction and proves its effectiveness.

4. Discussion

The DBC-Informer model significantly improves vehicle trajectory prediction by incorporating driver behavior categorization. It achieves notable reductions in Mean Absolute Error (MAE) and Mean Squared Error (MSE) compared to other models, especially in long-term predictions, highlighting the importance of considering diverse driving behaviors. The model's ability to capture dynamic driver behavior reduces prediction errors beyond 1 s into the future, enhancing vehicle warning systems.

Achieving a high classification accuracy of 98% for real-time behavioral categorization demonstrates the feasibility of using driver behavior as a predictive feature. The model improves prediction accuracy for both longitudinal and lateral motions by 56% and 61%, respectively, underscoring the value of integrating driver behavior into trajectory-prediction frameworks.

These findings emphasize the need for personalized driving assistance systems that adapt to individual driver behaviors. Future research should refine behavior categorization algorithms, integrate additional contextual information and develop robust real-time prediction frameworks. The DBC-Informer model represents a significant advancement in vehicle trajectory prediction, highlighting the potential for personalized driving assistance systems to enhance safety and efficiency in diverse road environments.

5. Conclusions

For the vehicle trajectory-prediction problem, considering the impact of different drivers' driving behaviors on vehicle trajectory, the DBC-Informer model is proposed to accurately predict the future trajectory of the vehicle. The DBC-Informer model adopts a multilayered network architecture, which has the advantage of capturing long-term relevance features, and also learns the driving style features in the historical trajectory.

In the driver behavior-classification method, the Principal Component Analysis (PCA) method is used to reduce the dimensionality of the data, and the simplified data distribution is clustered using the K-means algorithm to classify the online behaviors of the driver behaviors in the test set, which compares the classification accuracy of different classification models, and reaches 98% in the accuracy of classification. The results of the ablation experiments show that adding a driver behavior-classification method to the prediction model can improve the accuracy of prediction on both longitudinal and lateral motion.

The DBC-Informer model was trained and tested using the NGSIM dataset in the experiments, and the experimental results showed that the prediction accuracy of the DBC-Informer model was much higher than other models, and at the same time, the prediction accuracy enhancement of the DBC-Informer model was more obvious in the task of trajectory prediction in the long-term domain.

While our model shows significant improvements in trajectory-prediction accuracy, it has several limitations. One major shortcoming is the need for extensive computational resources during the training phase. Additionally, the model's performance may degrade in scenarios with highly dynamic and unpredictable driver behaviors that were not well-represented in the training data.

In the future, the team's subsequent research can be based on improving the driver behavior-classification method and quantifying the labels more accurately through the algorithm; studying how to effectively fuse different sensor data into the trajectory-prediction model to improve the efficiency of the model's use of multimodal information; and exploring a multi-task learning framework to simultaneously handle the driver behavior-classification and vehicle trajectory-prediction tasks in order to improve the model's overall performance and generalization ability.

Author Contributions: Conceptualization, J.S., S.Z. and M.L. (Mingjian Liu); methodology, J.S., S.Z. and M.L. (Mingjian Liu); software, J.S.; validation, S.Z. and M.L. (Mingjian Liu); formal analysis, J.S., S.Z. and M.L. (Mingjian Liu); investigation, J.S., M.L. (Muyang Li) and L.Z. resources, J.S., S.Z. and M.L. (Mingjian Liu); data curation, J.S., M.L. (Muyang Li) and L.Z.; writing—original draft preparation, J.S.; writing—review and editing, J.S., S.Z. and M.L. (Mingjian Liu); supervision, S.Z. and M.L. (Mingjian Liu); project administration, J.S., S.Z. and M.L. (Mingjian Liu); funding acquisition, S.Z. and M.L. (Mingjian Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the National Natural Science Foundation of China (61802046); Liaoning Provincial Education Science “14th Five-Year Plan” 2021 General Project (JG21DB076), 2022 General Project (JG22DB096).

Data Availability Statement: Data are contained within the article.

Acknowledgments: We sincerely thank all the individuals and organizations that contributed to this study. Special thanks to the National Natural Science Foundation of China for their support throughout this project. We would also like to thank the reviewers for their helpful feedback, which greatly enhanced the quality of this work.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

Acronyms	Full Name
DBC-Informer	Driver Behavior Classification and Informer
PCA	Principal Component Analysis
DNN	Deep Neural Network
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Square Error
SVM	Support Vector Machine
LR	Logistic Regression
DT	Decision Tree
KNN	k-Nearest Neighbor
LSTM	Long Short-Term Memory
Social-LSTM	Social Long Short-Term Memory
CNN-LSTM	Convolutional Neural Network-Long Short-Term Memory
Social-GAN	Social Generative Adversarial Network
STGAT	Spatial-Temporal Graph Attention Network
WCSS	Within Cluster-Sum of Squared

References

1. Huang, Y.; Du, J.; Yang, Z.; Zhou, Z.; Zhang, L.; Chen, H. A Survey on Trajectory-Prediction Methods for Autonomous Driving. *IEEE Trans. Intell. Veh.* **2022**, *7*, 652–674. [[CrossRef](#)]

2. Cao, D.; Wang, X.; Li, L.; Lv, C.; Na, X.; Xing, Y.; Li, X.; Li, Y.; Chen, Y.; Wang, F.-Y. Future Directions of Intelligent Vehicles: Potentials, Possibilities, and Perspectives. *IEEE Trans. Intell. Veh.* **2022**, *7*, 7–10. [\[CrossRef\]](#)
3. Wiseman, Y. Autonomous Vehicles. In *Encyclopedia of Information Science and Technology*, 5th ed.; IGI Global: Hershey, PA, USA, 2020; Volume 1, pp. 1–11.
4. Liu, M.J.; Zhu, Y.; Zhang, S.; Sun, H. Autonomous intersection control method based on maximum clique theory. *J. Zheng-Zhou Univ. (Eng. Sci.)* **2024**, *45*, 114–120.
5. Ju, C.; Wang, Z.; Long, C.; Zhang, X.; Chang, D.E. Interaction-aware Kalman Neural Networks for Trajectory Prediction. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1793–1800.
6. Wang, H.; Liu, B.; Ping, X.; An, Q. Path Tracking Control for Autonomous Vehicles Based on an Improved MPC. *IEEE Access* **2019**, *7*, 161064–161073. [\[CrossRef\]](#)
7. Jiang, Y.; Zhu, B.; Yang, S.; Zhao, J.; Deng, W. Vehicle Trajectory Prediction Considering Driver Uncertainty and Vehicle Dynamics Based on Dynamic Bayesian Network. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 689–703. [\[CrossRef\]](#)
8. Akhtar, S.; Habibi, S. The Interacting Multiple Model Smooth Variable Structure Filter for Trajectory Prediction. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 9217–9239. [\[CrossRef\]](#)
9. Li, J.; Ma, H.; Tomizuka, M. Conditional Generative Neural System for Probabilistic Trajectory Prediction. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 6150–6156.
10. Cui, H.; Radosavljevic, V.; Chou, F.; Lin, T.-H.; Nguyen, T.; Huang, T.-K.; Schneider, J.; Djuric, N. Multimodal Trajectory Predictions for Autonomous Driving using Deep Convolutional Networks. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 2090–2096.
11. Ivanovic, B.; Leung, K.; Schmerling, E.; Pavone, M. Multimodal Deep Generative Models for Trajectory Prediction: A Conditional Variational Autoencoder Approach. *IEEE Robot. Autom. Lett.* **2021**, *6*, 295–302. [\[CrossRef\]](#)
12. Ma, Y.; Zhu, X.; Zhang, S.; Yang, R.; Wang, W.; Manocha, D. TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence/Thirty-First Innovative Applications of Artificial Intelligence Conference/Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, Oral, Honolulu, HI, USA, 27 January–1 February 2019; pp. 6120–6127.
13. Deo, N.; Trivedi, M.M. Convolutional Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 1549–1557.
14. Chen, X.; Zhang, H.; Zhao, F.; Hu, Y.; Tan, C.; Yang, J. Intention-Aware Vehicle Trajectory Prediction Based on Spatial-Temporal Dynamic Attention Network for Internet of Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 19471–19483. [\[CrossRef\]](#)
15. Chandra, R.; Bhattacharya, U.; Bera, A. TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic Using Weighted Interactions. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8475–8484.
16. Liu, Y.; Zhang, J.; Fang, L.; Jiang, Q.; Zhou, B. Multimodal Motion Prediction with Stacked Transformers. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7573–7582.
17. Messaoud, K.; Yahiaoui, I.; Verroust-Blondet, A.; Nashashibi, F. Attention Based Vehicle Trajectory Prediction. *IEEE Trans. Intell. Veh.* **2021**, *6*, 175–185. [\[CrossRef\]](#)
18. Shahverdy, M.; Fathy, M.; Berangi, R.; Sabokrou, M. Driver behavior detection and classification using deep convolutional neural networks. *Expert Syst. Appl.* **2020**, *149*, 113240. [\[CrossRef\]](#)
19. Bouhsissin, S.; Sael, N.; Benabbou, F. Driver Behavior Classification: A Systematic Literature Review. *IEEE Access* **2023**, *11*, 14128–14153. [\[CrossRef\]](#)
20. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; Savarese, S. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 961–971.
21. Sainath, T.N.; Vinyals, O.; Senior, A.; Sak, H. Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 19–24 April 2015; pp. 4580–4584.
22. Gupta, A.; Johnson, J.; Li, F.; Savarese, S.; Alahi, A. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2255–2264.
23. Kong, X.; Xing, W.; Wei, X.; Bao, P.; Zhang, J.; Lu, W. STGAT: Spatial-Temporal Graph Attention Networks for Traffic Flow Forecasting. *IEEE Access* **2020**, *8*, 134363–134372. [\[CrossRef\]](#)

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.