IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 March 2024, accepted 11 April 2024, date of publication 19 April 2024, date of current version 26 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3391170

RESEARCH ARTICLE

Driving Behavior Primitive Classification Using CNN-Based Fusion Models

XIAOTONG CUI, XIANSHENG LI, XUELIAN ZHENG, AND YUANYUAN REN
Transportation College, Jilin University, Changchun 130000, China

Corresponding author: Xuelian Zheng (zhengxuelian@jlu.edu.cn)

This work was supported by the National Key R&D Program of China (2023YFC3009600).

ABSTRACT Driving behavior primitives play a crucial role in semantic explanation of driving behaviors. Although much work has been done on exacting driving behavior primitives from naturalistic driving data, few studies was published on primitive classification. Driving behavior primitives are typically described by multi-dimensional variables with varying durations, which leads to the inefficiency of the traditional classification methods. There hence, a CNN-based fusion model for primitive classification is proposed in this paper. Primitive feature matrix is constructed using statistical methods for the four basic and the four constructed variables, which serves as the input. A 1D-CNN is employed to extract global information of the total eight variables in the feature matrix, while a 2D-CNN is used to extract the local information. The 1D-CNN and the 2D-CNN are fused in parallel using a new fusion method to combine different types of information, and two models, namely the FC-before fusion model and the FC-after fusion model, are acquired. Compared with the classical methods, the empirical results demonstrate that CNN-based fusion model can recognize driving behavior primitives more accurately. Specifically, the FC-after fusion model achieves an accuracy of 91.12% and a macro F1-score of 90.88%, while the accuracy and macro F1-score of the FC-before fusion model are 93.47% and 92.57%, respectively.

INDEX TERMS Driving behavior analysis, driving behavior primitive classification, CNN-based fusion model, primitive feature matrix, information fusion.

I. INTRODUCTION

Driving behavior refers to a series of driving maneuvers performed in response to external factors. The semantic analysis of driving behavior is valuable for understanding the relationship between driving behavior and traffic environment [1], enhancing the human-like decision of intelligent vehicles [2], as well as promoting the development of intelligent transportation systems [3]. Generally, the semantic explanation involves driving behavior classification [4], modeling [5], prediction [6], and analysis of driving styles [7].

In recent years, semantic analysis of driving behaviors using driving behavior primitives has become a hot topic due to its high efficiency [8], [9], [10]. Driving behavior primitives are the smallest data segments with clear physical meanings. The total number of primitive clusters is limited. By utilizing driving behavior primitives, researchers could analyze driving behavior at a micro level. For instance, Li et al. identified the fine-grained driving style through coupling the intensity and frequency features of primitives [11]. Higgs et al. developed corresponding car-following models for different car-following primitives, which significantly improved the accuracy of driving behavior modeling [12]. Furthermore, in the field of intelligent driving, a partially observable MDP (POMDP) model built based on primitives is used to achieve more efficient and reliable decision-makings [13].

Currently, driving behavior primitives were typically extracted through offline methods, including the descriptive variable selection, driving data segmentation, and primitive clustering [14]. The clustered and defined primitives were then used for semantic analysis. However, online semantic analysis is more valuable in road safety guarantee and intelligent vehicle design. For example, it is important to recognize driving styles in real time to alert drivers and reduce driving risks [15]. The shared control system of intelligent vehicles needs online classification of driving maneuvers to achieve

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

real-time and adaptive driving authority allocations [16]. Moreover, the online driving behavior identifications of surrounding vehicles contributes to the driving risk predictions and decision-makings of intelligent vehicles [17]. In order to achieve online semantic analysis of driving behaviors, the classification of driving behavior primitives is one of the key problems.

Driving behavior primitives are descripted by multi-dimensional variables, and primitive durations are inconsistent. Obviously, the essence of primitive classification lies in constructing classifiers for multi-dimensional time series with different durations. The existing studies have primarily employed rule-based classifiers [11], traditional machine learning methods [18], and deep learning approaches to address this issue [19]. The rule-based classifier uses indictors such as convergency and accuracy to delete and fuse pre-set simple rules to obtain the final classification rules. Based on these rules, driving variables such as longitudinal acceleration and yaw angle were filtered, driving behaviors such as lane changes were successfully recognized [20]. However, efficient rule-based classifiers require a rich technical experience for researchers, and they have limited applicability. The traditional machine learning approaches use exacted high-quality driving features as input to achieve driving behavior identification, where the feature construction and extraction are key factors affecting the training performance. Li et al. constructed 24 statistical features, including average and maximum values, and utilized the sequential forward floating selection (SFFS) algorithm to extract the optimal feature subset. A recognition rate of 88% for lane changes were achieved by hidden Markov models (HMM) [21]. By comparing different feature extraction methods, it was found that features obtained by principal components analysis (PCA) and stacked sparse auto-encoder (SSAE) could improve the recognition efficiency of Random Forest (RF) [22]. Meanwhile, Yang et al. combined FFA and LDA to transform the continuous and chaotic EEG into discrete but representative features. Then, five types of car-flowing behaviors were successfully identified with the assistance of K-nearest neighbors (KNN) [23]. Although the traditional machine learning methods could achieve good classification results, it places high demands on feature extraction.

In order to solve the above-mentioned problems, deep learning approaches have become more and more popular in various driving behavior recognition tasks. Basic neural networks, such as Convolutional Neural Networks (CNN) [24], [25] and Recursive Neural Network (RNN) [26], [27], are fused to construct a classifier, and then the classifier is trained to extract complex features and identify various driving behaviors. Peng et al. proposed a CNN-LSTM framework using environment, trajectory, and vision features to identify driving behaviors in the early stage [28]. Arefnezhad et al. compared the performance of CNN-LSTM and CNN-GRU, and utilized the optimal fusion model to achieve multi-level classification of driver drowsiness [29]. Although these fusion models can achieve good classification results, the diverse types of the basic neural networks lead to the complexity increasing of fusion models. Therefore, fusing the similar neural networks becomes another approach to building classifiers. Xie et al. fused multiple 2D-CNNs in the early, the middle and the late stage [30]. The constructed model could process the initial features of the input, which would contribute to the efficient classification of behaviors such as braking and turning. Zhang et al. used multi-channel CNNs to deeply excavate weighted data and obtain the advanced features [31]. Using these complex features, turning and other driving behaviors were precisely identified. In addition, a hybrid CNN framework, including ResNet50, Inception V3 and Xception, was applied to extracted useful features from driving images, and the accuracy of risky behavior detection was up to 96.74% [32]. The fusion model based on the basic neural networks can achieve advanced feature exaction and driving behaviors classification simultaneously.

However, existing fusion models are limited by the basic neural networks and fusion ways. As a result, driving data was merely processed from a single perspective, and the global, local and hybrid features cannot be extracted. Consequently, the certain coupling information between different variables are lost, which has a significant impact on the primitive classification. In addition, primitives with different durations lead to the inability of existing fusion models on driving behavior primitive classification.

In this paper, a CNN-based fusion model is proposed to achieve driving behavior primitive classification. Feature matrices constructed by statistical methods were used as input while primitives with inconsistent durations will complicate the structure of classification model and increase training time. 1D-CNN and 2D-CNN are fused in parallel using a new fusion method to obtain the FC-before fusion model and FC-after fusion model. The contributions of this paper can be summarized as follows:

(1) A CNN-based fusion model using a new fusion method is developed to achieve driving behavior primitive classification. The proposed model can extract variables coupling information from global and local perspectives. Compared with the existing models, the proposed model explores multi-view features deeply, and improves the classification effects significantly.

(2) A feature matrix is constructed to solve the problem of inconsistent primitive durations.

(3) Two CNN-based fusion models are acquired by setting the fusion position before and after the fully connected layer (FC), in order to explore the effect of fusion position on the model performance.

The paper is organized as follows: In Section II, the data sources and driving behavior primitive samples are described. In Section III, the existing problems for primitive classification are analyzed. Section IV describes the structure of CNN-based fusion model and its related parameter set-

tings, the classification results are presented and discussed in Section V. Finally, the paper is concluded in Section VI.

II. DRIVING DATA AND BEHAVIOR PRIMITIVE DESCRIPTION

A. DRIVING DATA

Driving data used in this study is sampled by experiments carried out in the RADS 8 DOF Panoramic Driving Simulation. The test route contains 11 curves and the total round trip is about 10.35 km. 16 drivers (10 males, 6 females; age range 28∼50 years old, average age = 29.8, standard deviation (SD) = 2.7; driving experience 0∼12 years, average = 7.6 years, SD = 3.3) are paid for their participation in this study, and all drivers are required to drive vehicles in similar conditions to minimize potential disturbance caused by external factors.

The data recorded by the RADS 8 DOF Panoramic Driving Simulation has a sampling rate of 60 Hz. The moving-average solution is carried out to smooth the original data. The statistical results of the processed data are shown in Fig. 1. The velocity (v) falls in the 0∼34 m/s range, longitudinal acceleration (a_x) falls in the $-7\text{m/s}^2 \sim 6\text{m/s}^2$ range, lateral acceleration (a_y) ranges from $-7\text{m/s}^2 \sim 9\ \text{m/s}^2$, and longitudinal jerk (j) ranges from $-6\text{m/s}^3 \sim 6\text{m/s}^3$.

B. DRIVING BEHAVIOR PRIMITIVES

The extraction of driving behavior primitives includes three parts: variables selection, data segmentation, and segments clustering (shown in Fig. 2). The basic variables (such as velocity) cannot reflect drivers' subjective expectations, as a result, some constructed variables are calculated to describe drivers' preferences on driving performances (such as rapidity). The multi-type variable space, composed by the basic and the constructed variables, is used as the input. Then, the Bayesian Model-based agglomerative Sequence Segmentation (BMASS) is employed to divide the whole driving data into independent segments [14]. A total of 2957 segments are acquired, and the number of segments for each driver are show in Fig.3. Driver 10 and driver 11 have more than 200 segments, while driver 2, driver 12 and driver 16 have less segments. The number of segments for other drivers ranges from 170 to 200.

Subsequently, the obtained segments are clustered by a new latent Dirichlet allocation method, namely VC-LDA [33]. Specifically, the VC-LDA method consists of two parts: driving data discretization and segments clustering. During the process of driving data discretization, the coupling relationship between variables in the multi-type variable space is considered. Finally, five primitive clusters are obtained, and their semantics are shown as follows:

Cluster 1: continuous high-speed driving with better comfort and rapidity performance

Cluster 2: turning behaviors with gentle acceleration and good fuel economy performance

Cluster 3: low-velocity driving with aggressive acceleration and deceleration, as well as poor rapidity, fuel economy and comfort performance

Cluster 4: driving with strength braking, and poor fuel economy, comfort, and rapidity performance

Cluster 5: slightly braking when driving at high-speed, with fine driving performance.

III. PROBLEMS EXISTED IN DRIVING BEHAVIOR PRIMITIVE CLASSIFICATION

A. DRIVING BEHAVIOR PRIMITIVES WITH DIFFERENT DURATIONS

As show in Fig. 4, the durations of driving behavior primitives are inconsistent across all samples or within the same primitive clusters. The classical algorithms, such as RF and CNNs, can only handle the data with equal lengths. Although algorithms such as LSTM and Transformer can directly utilize data with various durations to achieve classification, this approach will increase the difficulty of model training and reduce the recognition efficiency in practically applying.

Therefore, transforming the primitives of uneven lengths into equal-length vectors is quite vital for the primitive classifier. Statistical methods are applied to construct the primitive feature matrix, and the feature matrix serves as the input of the classifier.

B. CLASSIFICATION MODEL SELECTION

Two problems should be considered when constructing primitive classification models: (1) The input of the classifier is the primitive feature matrix, which has no temporal orders. Thus, the classifier should not be affected by the order of data points in the input data. (2) the classifier needs to have good performance and robustness to ensure the accuracy of primitive recognition.

For deep learning approaches, CNNs achieve the translation invariant through convolution and pooling layers, which is robust to the order of input data. Also, CNNs have good performance and generalization ability, which ensures the reliability and robustness of the classification result. There hence, CNNs are chosen to develop the primitive classifier in this paper.

C. COUPLING INFORMATION EXPRESSION AMONG MULTI-TYPE VARIABLES

The input of the classifier is a matrix consisting of m features of 8 variables (as shown in Fig. 5). Specifically, v_w, $a_{x,w}$, $a_{y,w}$, and j_w are the constructed variables of v, a_x, a_y, and j. The relationships among multi-type variables result in the various information carried by diverse feature combinations. For instance, the yellow box in Fig. 5 describes driver horizontal and vertical maneuvers, as well as the related driving performance. The red box just focuses on longitudinal acceleration at different speeds. Meanwhile, the blue box can only reflect the driving performance (such as fuel economy) and has no knowledge of vehicle driving conditions. Certainly,

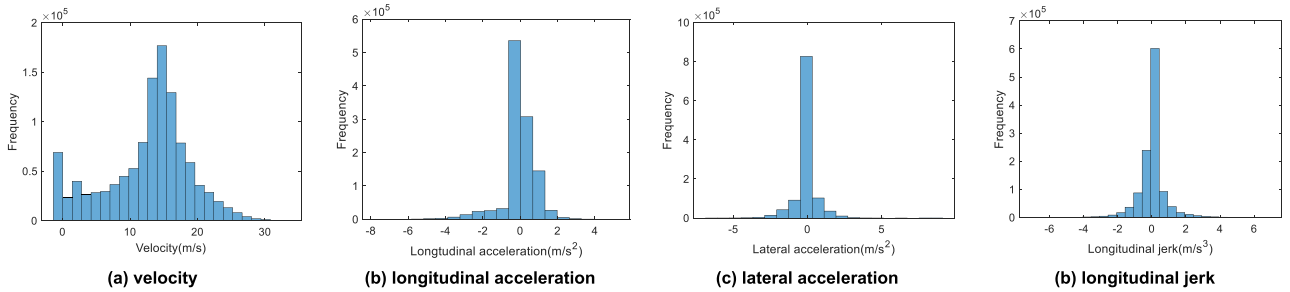


FIGURE 1. Statistical results of the processed driving data.

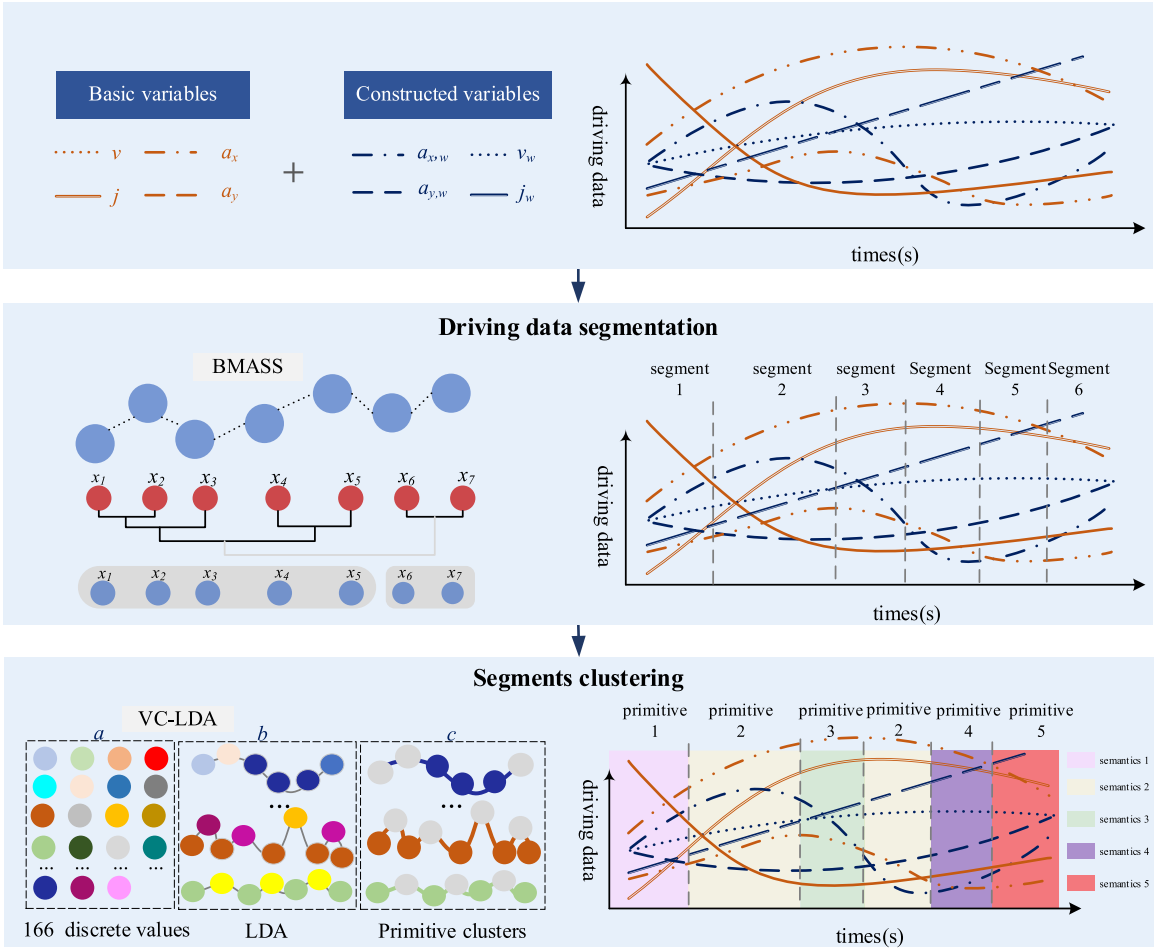


FIGURE 2. Flow diagram of driving behavior primitive extraction.

in order to accurately classify driving behavior primitives, it is crucial for the model to have the ability of comprehensively analyzing the relationships among multi-type variables.

CNNs usually include one-dimensional CNN (1D-CNN) and two-dimensional CNN (2D-CNN). 1D-CNN has been widely applied in natural language processing, which achieves language prediction and recognition by extracting global information from input data; 2D-CNN is usually used to process image data, and it pays more attention to local information. Obviously, 1D-CNN and 2D-CNN tend to extract the coupling information from a single perspective, leading to the loss of crucial features for accurate primitive classifications. So that, 1D-CNN and 2D-CNN are fused in parallel to obtain a CNN-based fusion model. By merging global and local information together, this CNN-based fusion model can automatedly explore deep features from multiple perspectives and significantly improve the efficiency of primitive classification.

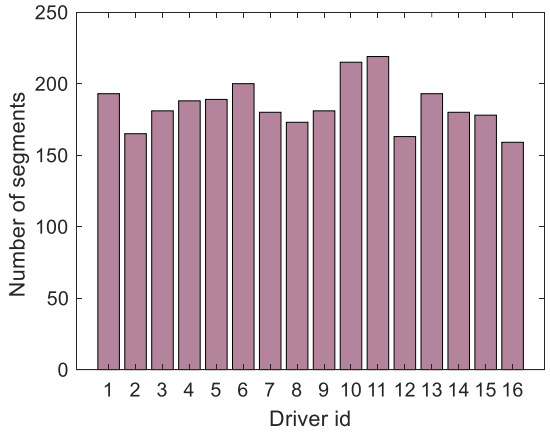


FIGURE 3. The number of segments for each driver.

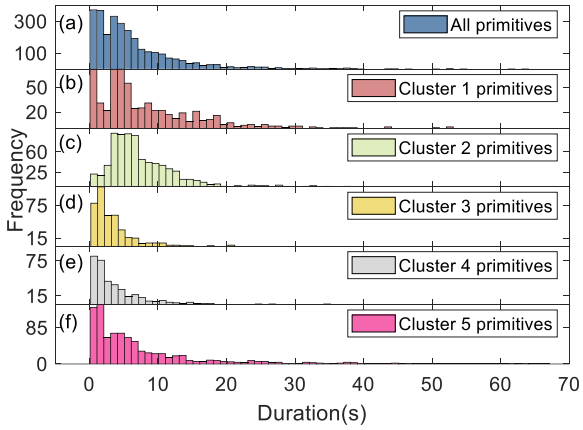


FIGURE 4. The distribution of primitive durations (a) the duration distribution for all primitives; (b)-(f) the duration distributions for primitive cluster 1-5.

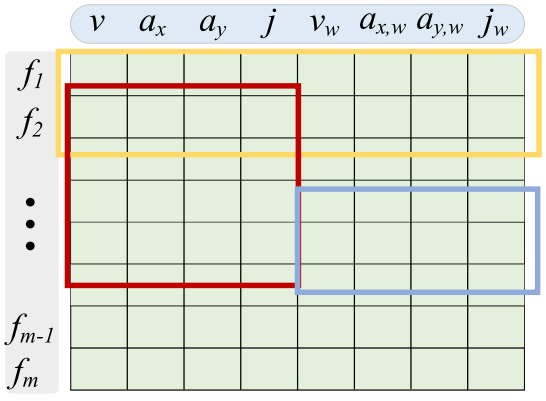


FIGURE 5. Schematic diagram of the coupling relationships between the 8 variables.

D. THE PROPOSED FRAMEWORK FOR DRIVING BEHAVIOR PRIMITIVE CLASSIFICATION

Based on the above analysis, a novel method for driving behavior primitive classification is proposed as shown in Fig. 6. The framework mainly consists of two parts: feature construction and CNN-based fusion model, which are presented in the following sections.

IV. DRIVING BEHAVIOR PRIMITIVE CLASSIFICATION

A. FEATURE CONSTRUCTION OF DRIVING BEHAVIOR PRIMITIVES

Primitive features are constructed according to the central tendency, dispersion, percentile values, extreme values, and information entropy. The variables, operations and descriptions used for feature construction are shown in Table 1.

The obtained features are reconstructed into matrix $\boldsymbol{A}$ as the input of CNN-based fusion models. The details of matrix $\boldsymbol{A}$ is show as follows,

$$\mathbf{A} = \begin{pmatrix} v.mean & \dots & j_w.mean \\ \vdots & \ddots & \vdots \\ v.ApEn & \cdots & j_w.ApEn \end{pmatrix}$$

B. CNN-BASED FUSION MODEL DEVELOPMENT

Feature matrix $\boldsymbol{A}$ is composed of the features of basic variables (including v, a_x, a_y and j) and constructed variables (including v_w, $a_{x,w}$, $a_{y,w}$ and j_w). There are different relationships among multi-type variables. It is necessary for the classifier to thoroughly analyze the coupling information between different variables in matrix $\boldsymbol{A}$. Based on that, a CNN-based fusion model is constructed to classify driving behavior primitives.

As shown in Fig. 7 and Fig. 8, the CNN-based fusion model consists of two parts: information extraction module and information fusion module. Firstly, the information extraction module utilizes one-dimensional (1D) hidden layers and two-dimensional (2D) hidden layers to separately extract the global and local coupling information of different variables. Then, the information fusion module fuses the extracted information through a new fusion method, which stacks the feature matrices output by the information extraction module after flattening them. In addition, the fusion stages are separately set before and after FC, and two fusion models are acquired. Finally, in terms of these models, primitives are accurately categorized.

1) INFORMATION EXTRACTION MODULE

a: 1D HIDDEN LAYERS

The 1D hidden layers are set up to extract the overall coupling information of eight variables. 1D hidden layers include the 1D convolutional layer (Conv_1d),1D max-pooling layer (Max-pooling_1d), and the related activation function called Rectified Linear Unit (ReLU). The convolutional kernel, denoted as ***W_1d***, is set to a size of $r1$. The convolution is operated by (1),

$$output_{Conv\_1d} = Conv\_1d(\mathbf{A}) = W\_1d \otimes \mathbf{A} = \sum_{i}^{r1} w\_1d \cdot x_i \tag{1}$$

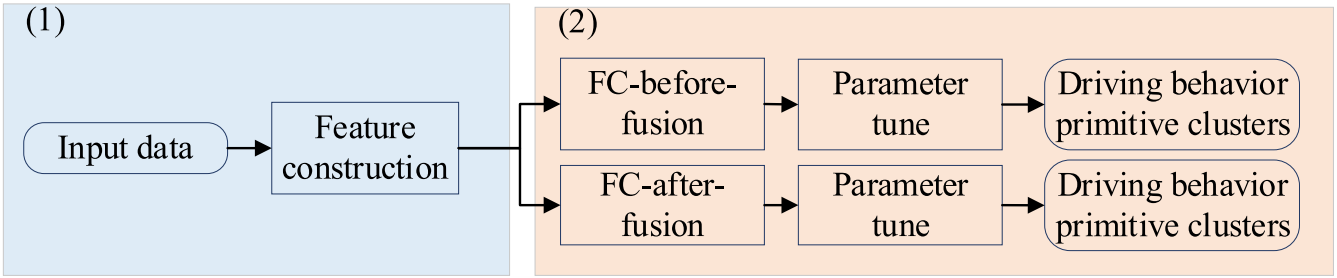


FIGURE 6. The proposed framework for driving behavior primitive classification.

TABLE 1. Variables, operations, and descriptions used for primitive feature construction.

Variables	Codes	Descriptions
Velocity	v	Time series data; velocity of vehicles
Longitudinal acceleration	a_x	Time series data; longitudinal acceleration of vehicles
Lateral acceleration	a_y	Time series data; lateral acceleration of vehicles
Jerk	j	Time series data; longitudinal jerk of vehicles
Constructed variable of velocity	v_w	Time series data; rapidity of vehicles
Constructed variable of longitudinal acceleration	a_{xw}	Time series data; fuel economy of vehicles
Constructed variable of lateral acceleration	a_{yw}	Time series data; safety of vehicles
Constructed variable of jerk	j_w	Time series data; comfort of vehicles
Operations	**Codes**	**Descriptions**
Central tendency	mean, median	Mean and median values
Dispersion	std, iqr	Standard deviation and quartile deviation
Percentile values	p05, p10, p25, p75, p90, p95	The 5th, 10th, 25th, 75th, 90th, and 95th percentiles
Extreme values	min, max	Minimum and maximum values
Information entropy	Shannon, ApEn	Shannon entropy and approximate entropy

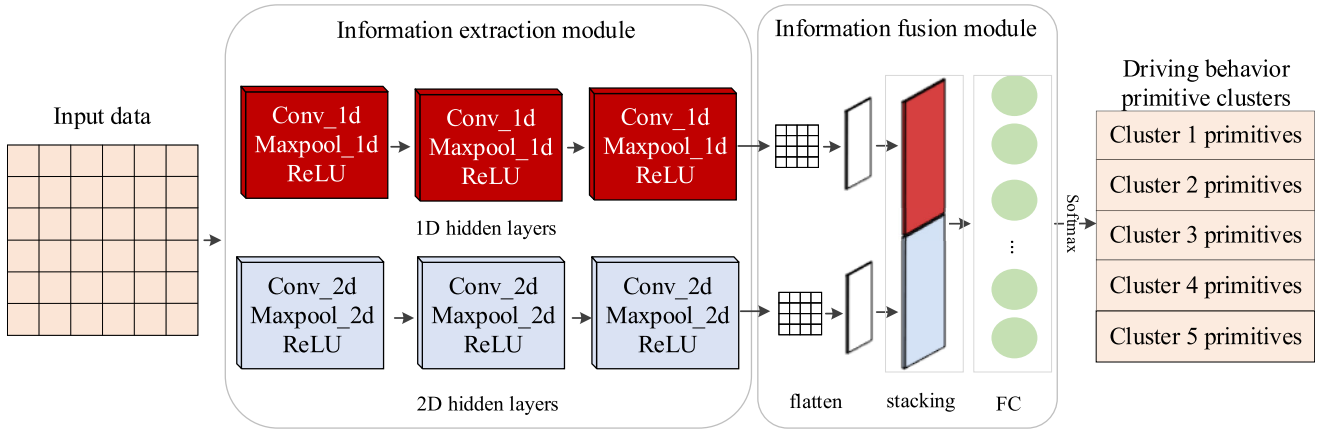


FIGURE 7. Flow diagram of the FC-before fusion model.

where, $\otimes$ is convolution, w_1d_i is the elements of $\boldsymbol{W\_1d}$, and x_i is the related features in $\boldsymbol{A}$.

Then, the max-pooling layer with the size of $c1$ is used to reduce the data dimensionality,

$$output_{Max-pooling\_1d} = Max - pooling\_1d(output_{Conv\_1d}) = \max_{j}^{c1}(output_{conv\_1d_j}) \quad (2)$$

Finally, the activation function named ReLU is applied to make the above output non-linear, and the deep features

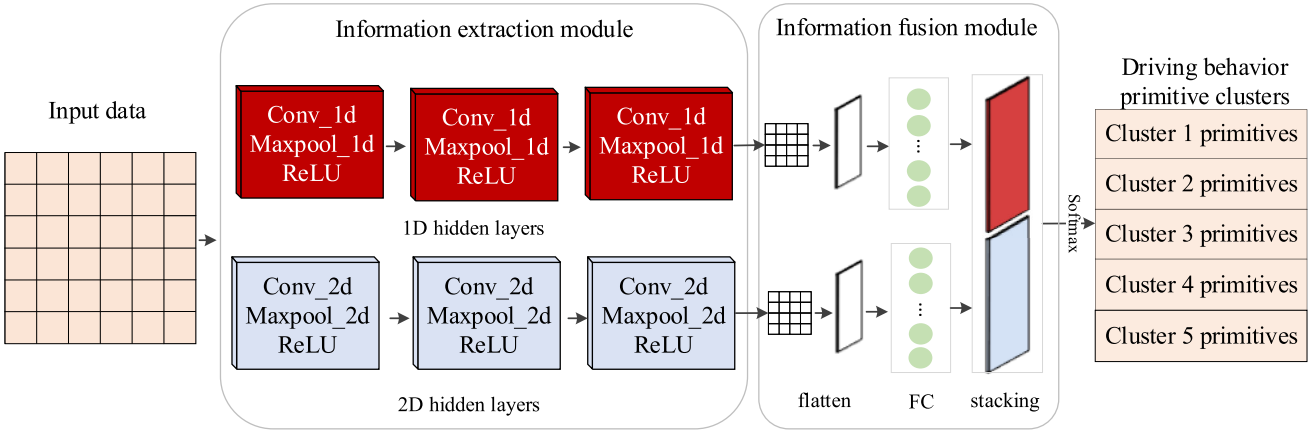


FIGURE 8. Flow diagram of the FC-after fusion model.

extracted by 1D hidden layers(y_1d) are obtained,

$$y\_1d = ReLu(output_{Max-pooling\_1d}) \tag{3}$$

b: 2D HIDDEN LAYERS

Besides the overall coupling information, there are different coupling information between various variable combinations. In order to understand the correlations between different variable combinations, the 2D hidden layers are employed to extract the advanced primitive features.

Similarly, 2D hidden layers include the 2D convolutional layer (Conv_2d), 2D max-pooling layer (Max-pooling_2d), and the related activation function called Rectified Linear Unit (ReLU). Specifically, the kernel size ($\boldsymbol{W\_2d}$) of Conv_2d is set as r2×s2, and the size of Max-pooling_2d is set as c2×b2. So, the operations of 2D hidden layer are,

$$\begin{aligned} output_{Conv\_2d} &= Conv\_2d(\mathbf{A}) = \mathbf{W\_2d} \otimes A \\ &= \sum_{i}^{r2\times s2} w\_2d_i \cdot x_i \end{aligned} \tag{4}$$

$$\begin{aligned} output_{Max-pooling\_2d} &= Max-pooling\_2d(output_{Conv\_2d}) \\ &= \max_{j}^{c2\times b2}(output_{Conv\_2d_j}) \end{aligned} \tag{5}$$

$$y\_2d = ReLu(output_{Max-pooling\_2d}) \tag{6}$$

where, w_2d_i is the elements of $\boldsymbol{W\_2d}$, y_2d is the deep features extracted by 2D hidden layers.

2) INFORMATION FUSION MODULE

In this module, the features extracted by the information extraction module are deeply fused, then the comprehensive and high-quality features are acquired to improve the model efficiency. Stacking after flattening is employed for fusion. During the fusion process, y_1d and y_2d are flattened into vectors named_1d' and y_2d'. These vectors are then stacked into a fusion vector ξ, which is applied for the primitive recognition.

The fully connected layer (FC) aims to integrate different extracted features. The y_1d and y_2d have different semantics, so the fusion stages are separately set before and after FC to get two different fusion models for obtaining different fusion information. These two models are separately named as FC-after fusion model and FC-before fusion model, as shown in Fig. 7 and Fig. 8.

To summarize, the fusion module can merge different features extracted by the 1D hidden layers and 2D hidden layers together, which extracts more comprehensive information from both global and local perspectives. In addition, two fusion models are developed to investigate the impacts of fusion stages on the model recognition performance.

TABLE 2. Pre-defined hyperparameters Of CNN-based fusion models.

Hyperparameter	Range
r_1	2; 3
c_1	1; 2
p	512
$c_2 \times b_2$	2×2
$r_2 \times s_2$	2×2; 3×2; 3×3
z_1d, z_2d	32, 64, 64; 32, 64, 128

C. HYPERPARAMETER TUNING FOR THE CNN-BASED FUSION MODEL

The hyperparameters of the fusion model include the parameters of Conv_1d, Conv_2d, Max-pooling_1d and Max-pooling_2d in the information extraction module, as well as the parameter of FC in the information fusion module. Firstly, the candidate values of hyperparameters are determined based on existing researches and experiences, as well as computational efficiency [29], [30]. Then, the optimal hyperparameter are generated according to the model performance under different hyperparameters.

Table 2 presents the candidate values for each hyperparameter. Hyper-opt searches in all possible combinations of these values to select the optimal one. The experimental results are shown in the next section.

r_1 is the kernel size of Conv_1d, $r_2 \times s_2$ is the kernel size of Conv_2d, c_1 is the size of Max-pooling_1d, $c_2 \times b_2$ is the size

TABLE 3. Hyperparameter values of the CNN-based fusion models.

Model id	r_1	c_1	$r_2 \times s_2$	$c_2 \times b_2$	z_1d, z_2d	p
M1	2	1	3×2	2×2	32, 64, 128	512
M2	2	1	2×2	2×2	32, 64, 128	512
M3	2	1	3×3	2×2	32, 64, 128	512
M4	3	2	3×2	2×2	32, 64, 128	512
M5	3	2	2×2	2×2	32, 64, 128	512
M6	3	2	3×3	2×2	32, 64, 128	512
M7	2	1	3×2	2×2	32, 64, 64	512
M8	2	1	2×2	2×2	32, 64, 64	512
M9	2	1	3×3	2×2	32, 64, 64	512
M10	3	2	3×2	2×2	32, 64, 64	512
M11	3	2	2×2	2×2	32, 64, 64	512
M12	3	2	3×3	2×2	32, 64, 64	512

of Max-pooling_2d, *z_1d* and *z_2d* are the number of kernels for Conv_1d and Conv_2d, and *p* is the number of neurons for FC. Meanwhile, the drop rate of FC is taken as 0.6 to avoid overfitting.

V. RESULTS AND DISCUSSION

A. EXPERIMENT DETAILS

The proposed CNN-based fusion model and existing models to be compared were implemented using Python programming, and were tested on the dataset mentioned in Section II. Specifically, the deep learning TensorFlow framework was adopted in Python programming.

B. RESULTS

Based on Table 2, the different combinations of the hyperparameter values are obtained, which are shown in Table 3. Based on this, the CNN-based fusion model is initialized and trained. Evaluation indicators including Accuracy and Macro F1-score are used to analyze the classification results of each model and determine the optimal fusion model.

The dataset has been split into the training, validation, and test set. The optimal model is selected according to the train and validation set, and performance of the model is validated by the test set.

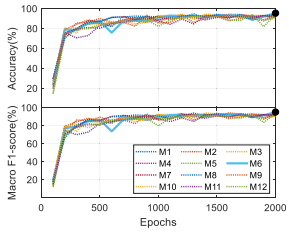


FIGURE 9. The training results of the FC-before fusion model.

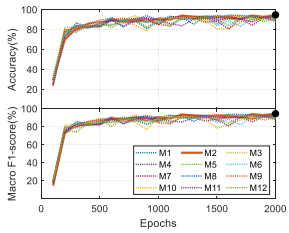


FIGURE 10. The training results of the FC-after fusion model.

TABLE 4. The classification performance of the optimal fusion model.

Fusion model	Accuracy	Macro F1-score
FC-after fusion model	91.12%	90.88%
FC-before fusion model	93.47%	92.57%

Fig. 9 and Fig. 10 respectively presents the Macro F1-score and Accuracy under diverse epochs for different fusion models. For FC-before fusion model, the highest Accuracy and Macro F1-score are obtained by M6 after 2000 training epochs, which are 95.50% and 95.32% respectively. Therefore, chose this model as the optimal FC-before fusion model. Similarly, the M2 for FC-after fusion model gets the best Accuracy and Macro F1-score, which are 94.82% and 94.55% respectively. Thus, M2 after 2000 training epochs is taken as the optimal model for FC-after fusion model.

Subsequently, the best fusion models are validated on the test sets to ensure their generalization ability. Table 4 shows the validation results. Compared with the training results, the two fusion models perform well on the test set, indicating the models have certain reliability and effectiveness. Furthermore, the performance of FC-before fusion model is slightly better than the FC-after fusion model.

C. DISCUSSION

To verify the ability of the CNN-based fusion model, three basic types of classification methods are also employed for comparative experiments. The compared methods include: (1) the ablation study of CNN-based fusion model:1D-CNN and 2D-CNN; (2) the existing classical methods: (a) the well-known traditional machine learning methods, which extracts features using PCA and classifies primitives by random forest (RF) [34]; (b) the other fusion models suitable for the primitive features: CNN-mid-fusion model and CNN-late-fusion model [30].

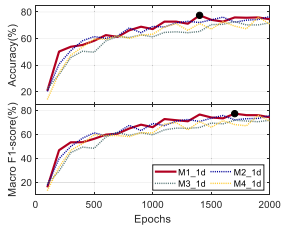


FIGURE 11. The training results of 1D-CNN.

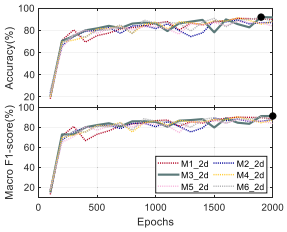


FIGURE 12. The training results of 2D-CNN.

TABLE 5. Hyperparameter value space Of 1D-CNN.

Model id	Hyperparameters			
	r_1	c1	z_1d，z_2d	p
M1_1d	2	1	32，64，128	512
M2_1d	2	1	32，64，64	512
M3_1d	3	2	32，64，128	512
M4_1d	3	2	32，64，64	512

TABLE 6. Hyperparameter value space Of 2D-CNN.

Model id	Hyperparameters			
	$r_2 \times s_2$	$c_2 \times b_2$	z_1d，z_2d	p
M1_2d	3×2	2×2	32，64，128	512
M2_2d	2×2	2×2	32，64，128	512
M3_1d	3×3	2×2	32，64，128	512
M4_2d	3×2	2×2	32，64，64	512
M5_2d	2×2	2×2	32，64，64	512
M6_2d	3×3	2×2	32，64，64	512

The hyperparameter values of 1D-CNN and 2D-CNN are also obtained by combining the candidate values for each hyperparameter, shown in Table 5 and Table 6. The training results are shown in Fig. 11 and Fig. 12. It is obvious that the optimal model of 1D-CNN is M1_1d and the best of 2D-CNN is M3_2d. These two models are chosen for comparisons.

For the traditional machine learning method including PCA and RF (PCA+RF), the explained variance ratio of PCA is set as 95% and the random trees is set as 100.

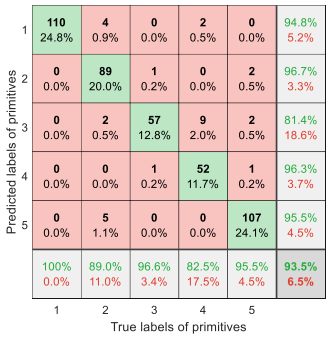
For the other fusion models, although many fusion models are applied to identify driving behaviors, some models are not suitable for the feature matrix $\boldsymbol{A}$ used in this paper [25], [26], [27], [28], [31]. CNN-mid-fusion model and CNN-late-fusion model, which are designed to identify driving behaviors based on features, are chosen to be the compared fusion models. The optimal hyperparameters of CNN-mid-fusion model and CNN-late-fusion model can be found in reference [30].

The results of the ablation study are shown in Table 7 and the performances of state-of-art methods are shown in Table 8, which are specifically represented by the Accuracy and Macro F1-score. The Accuracy and Macro F1-score of 1D-CNN and 2D-CNN are both lower than those of CNN-based fusion models. Obviously, the performances of the CNN-based fusion models are verified by the alation study. In addition, the Accuracy and Macro F1-score of PCA+RF are both lower than 80%, meaning a bad performance. Although the Accuracy and Macro F1-score of CNN-mid-fusion model have improved, they are still lower than evaluation indicators of CNN-based fusion models. The classification performances of the CNN-late-fusion model and the CNN-after fusion model are roughly equivalent. However, the Accuracy and Macro F1-score of FC-before fusion model achieve an 93.47% and 92.57% respectively, marking it as the top performer among all models.

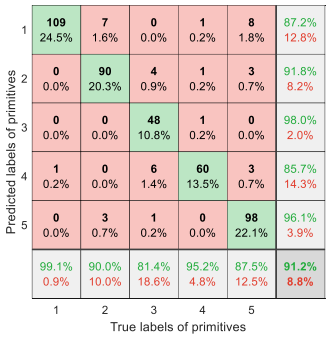
The computational complexity of different methods is also analyzed. The more complex the models are, the longer training times will be needed. Therefore, training times are chosen to measure the computational complexity. Table 9 shows the training times for different methods. Compared by other models, the PCA+RF and 2D-CNN have the shortest training times. The training times for 1D-CNN, FC-before fusion model, FC-after fusion model and CNN-mid-fusion ranges from 11s to 15s, indicating the similar computational complexity. The CNN-late-fusion has the longest training times, which means the highest level of computational complexity.

To sum up, 2D-CNN and PCA+RF have the minimize computational complexity, but their performances are not well. The CNN-late-fusion model achieve the same performance as the FC-after fusion model, but its training time is significantly longer than that of FC-after fusion model. While the proposed CNN-based fusion models, 1D-CNN, and CNN-mid-fusion model have the same computational complexity, the proposed CNN-based fusion models have much better performance than others. So that, we can draw the conclusion that the CNN-based fusion models proposed in this paper can improve the primitive classification results to some extent, and it reach a good balance between computational complexity and model performances. Specifically, the FC-before fusion model has the best recognition performance among all models.

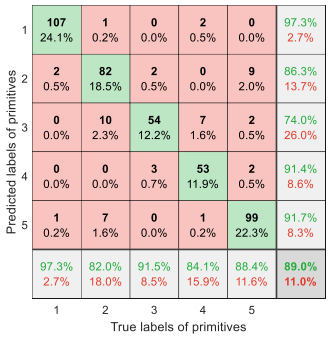
The confusion matrices of various methods are illustrated in Fig. 13 to further investigate their classification result. In matrices, the values on the diagonal represent the number of correctly identified samples, and their ratios of the whole samples. The green values in the right column reflect the Precision, the green values in the bottom row describe the Recall, and the green values in the bottom right corner represent the Accuracy.



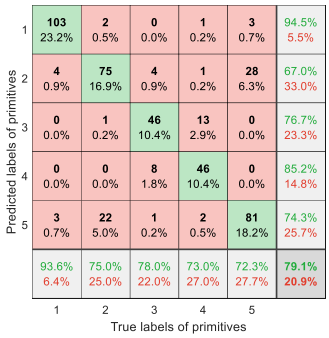
(a) the FC-before fusion model



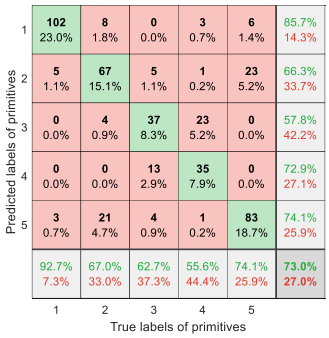
(b) the FC-after fusion model



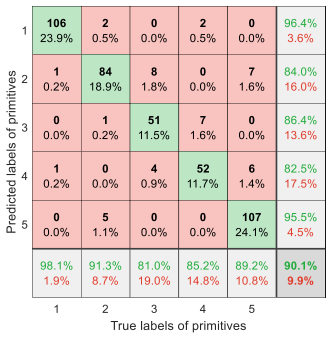
(c) 2D-CNN



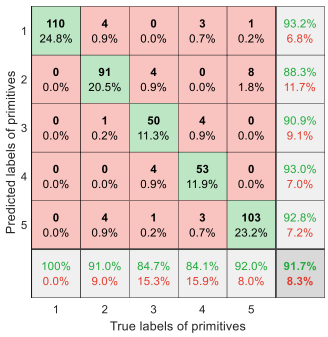
(d) RF



(e) 1D-CNN



(f) CNN-mid-fusion



(g) CNN-late-fusion

FIGURE 13. **Confusion matrices of different classification models.**

TABLE 7. **Ablation study of the CNN-based fusion model.**

Evaluation indicators	Classification methods			
	1D-CNN	2D-CNN	FC-before fusion model	FC-after fusion model
Accuracy	72.97%	88.96%	93.47%	91.12%
Macro F1-score	70.62%	88.16%	92.57%	90.88%

From the confused matrix, the 1D-CNN and RF both have lower Recall and Accuracy for primitive clusters except cluster 1. The velocity of cluster 1 primitives is higher, which indicates that analyzing the data from a global perspective can only obtain features with evident differences.

Although 2D-CNN has improved the Recall for each primitive category, the identification Accuracy of certain category is relatively lower. For example, the identification Precision of cluster 3 primitives is only 74%, which dues to the similarities of local variable features (such as features of the longitudinal acceleration) between different clusters (such as cluster 2, 3 and 4). In addition, the Recall and Precision of CNN-before fusion model, CNN-after fusion model, CNN-mid fusion model and CNN-late fusion model have all improved, but the Accuracy of CNN-before fusion model and CNN-after fusion model is higher than the that of CNN-mid-

TABLE 8. Comparison of classification performance for existing classical methods.

Evaluation indicators	Classification methods				
	FC-before fusion model	FC-after fusion model	PCA+RF	CNN-mid-fusion model	CNN-late-fusion model
Accuracy	93.47%	91.12%	79.05%	90.09%	91.67%
Macro F1-score	92.57%	90.88%	78.81%	88.89%	90.92%

TABLE 9. Comparison of training times for different classification methods.

Classification methods	1D-CNN	2D-CNN	FC-before fusion model	FC-after fusion model	PCA+RF	CNN-mid-fusion	CNN-late-fusion
Training times (s)	11.32	5.97	14.35	11.77	3.58	15.16	34.24

fusion. Moreover, the Accuracy of FC-before fusion model is better than that of the CNN-late fusion model.

As mentioned above, the need for fusing global and local information when classifying primitives is further verified. The two proposed fusion models, especially the FC-before fusion model, can achieve a better recognition performance.

VI. CONCLUSION

A CNN-based fusion model is proposed in this paper for driving behavior primitive classification. Firstly, primitive features were constructed by statistical methods to solve the issue of inconsistent durations among primitives. These features were reconstructed to be matrices as the classifier input. Secondly, the 1D-CNN and 2D-CNN were fused in parallel using a new fusion method. This model could simultaneously analyze the global and local features of input data, which deeply describes various relationships between multi-type variables. Further, the fusion stages were set before and after FC, and two fusion models were obtained. Based on the two fusion models, labels of driving behavior primitives were efficiently identified, and the proposed CNN-based fusion model were compared to the state-of-art models.

The driving behavior primitives are classified for the first time in this paper, which is important for the online semantic analysis of driving behaviors. The results shows that deep learning, especially the CNN-based fusion method, is very promising in driving behavior primitive classification. In addition, the numerical experiments verify the superiority and efficiency of the proposed method.

However, there are still some disadvantages in our work. The feature construction is chosen to address the problems of primitive having inconsistent durations, but this treatment has certain subjective limitations. In the future work, we will try to overcome this problem through more objective models. Moreover, the information fusion method used in this paper is fixed, so a more adaptive method for information fusion module will be considered. Last, the comparison with other methods is only conducted from the fusion model level, and the comparison with various information extraction or fusion module will also be included in the future.

REFERENCES

[1] W. Wang, W. Zhang, J. Zhu, and D. Zhao, "Understanding V2V driving scenarios through traffic primitives," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 610–619, Jan. 2022.

[2] T. Taniguchi, S. Nagasaka, K. Hitomi, K. Takenaka, and T. Bando, "Unsupervised hierarchical modeling of driving behavior and prediction of contextual changing points," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1746–1760, Aug. 2015.

[3] X. Chen, J. Sun, Z. Ma, J. Sun, and Z. Zheng, "Investigating the long- and short-term driving characteristics and incorporating them into car-following models," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102698.

[4] J. Gao, H. Zhu, and Y. L. Murphey, "Adaptive window size based deep neural network for driving maneuver prediction," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Hefei, China, Aug. 2020, pp. 87–92.

[5] W. Wang, J. Xi, and D. Zhao, "Learning and inferring a driver's braking action in car-following scenarios," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 3887–3899, May 2018.

[6] G. Weidl, A. L. Madsen, S. Wang, D. Kasper, and M. Karlsen, "Early and accurate recognition of highway traffic maneuvers considering real world application: A novel framework using Bayesian networks," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 3, pp. 146–158, Fall. 2018.

[7] M. M. Bejani and M. Ghatee, "A context aware system for driving style evaluation by an ensemble learning on smartphone sensors data," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 303–320, Apr. 2018.

[8] T. Bando, K. Takenaka, S. Nagasaka, and T. Taniguchi, "Generating contextual description from driving behavioral data," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Dearborn, MI, USA, Jun. 2014, pp. 183–189.

[9] T. Taniguchi, S. Nagasaka, K. Hitomi, N. P. Chandrasiri, and T. Bando, "Semiotic prediction of driving behavior using unsupervised double articulation analyzer," in *Proc. IEEE Intell. Vehicles Symp.*, Alcala de Henares, Spain, Jun. 2012, pp. 849–854.

[10] W. Wang, J. Xi, and D. Zhao, "Driving style analysis using primitive driving patterns with Bayesian nonparametric approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2986–2998, Aug. 2019.

[11] X.-S. Li, X.-T. Cui, Y.-Y. Ren, and X.-L. Zheng, "Unsupervised driving style analysis based on driving maneuver intensity," *IEEE Access*, vol. 10, pp. 48160–48178, 2022.

[12] B. Higgs and M. Abbas, "Segmentation and clustering of car-following behavior: Recognition of driving patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 81–90, Feb. 2015.

[13] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction," *Auton. Robot.*, vol. 41, no. 6, pp. 1367–1382, Jan. 2015.

[14] X. S. Li, X. T. Cui, X. L. Zheng, Y. Y. Ren, L. Shi, and J. F. Xi, "Extraction of driving behavior primitives based on multi-type variables space," *China J. Highway Transp.*, vol. 36, no. 7, pp. 223–235, Jul. 2023.

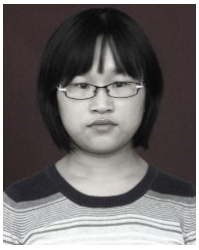
[15] E. Suzdaleva and I. Nagy, "An online estimation of driving style using data-dependent pointer model," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 23–36, Jan. 2018.

[16] B. H. Sun, "Research on personalized shared control considering driver's driving capability and style," Ph.D. dissertation, College Automot. Eng., JiLin Univ., Changchun, China, 2020.

[17] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH J.*, vol. 1, no. 1, p. 1, Jul. 2014.
[18] T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma, "Support vector machines classification with a very large-scale taxonomy," *ACM SIGKDD Explor. Newslett.*, vol. 7, no. 1, pp. 36–43, Jun. 2005.
[19] C. Ou and F. Karray, "Deep learning-based driving maneuver prediction system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1328–1340, Feb. 2020.
[20] M. M. Haque, S. Sarker, and M. A. A. Dewan, "Driving maneuver classification from time series data: A rule based machine learning approach," *Int. J. Speech Technol.*, vol. 52, no. 14, pp. 16900–16915, Nov. 2022.
[21] G. Li, S. E. Li, Y. Liao, W. Wang, B. Cheng, and F. Chen, "Lane change maneuver recognition via vehicle state and driver operation signals—Results from naturalistic driving data," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Seoul, South Korea, Jun. 2015, pp. 865–870.
[22] J. Xie, A. R. Hilal, and D. Kulic, "Driving maneuver classification: A comparison of feature extraction methods," *IEEE Sensors J.*, vol. 18, no. 12, pp. 4777–4784, Jun. 2018.
[23] L. Yang, R. Ma, H. M. Zhang, W. Guan, and S. Jiang, "Driving behavior recognition using EEG data from a simulated car-following experiment," *Accident Anal. Prevention*, vol. 116, pp. 30–40, Jul. 2018.
[24] A. Behera, Z. Wharton, A. Keidel, and B. Debnath, "Deep CNN, body pose, and body-object interaction features for drivers' activity monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2874–2881, Mar. 2022.
[25] M. M. Bejani and M. Ghatee, "Convolutional neural network with adaptive regularization to classify driving styles on smartphones," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 543–552, Feb. 2020.
[26] S. K. Kwon, J. H. Seo, J. Y. Yun, and K.-D. Kim, "Driving behavior classification and sharing system using CNN-LSTM approaches and V2X communication," *Appl. Sci.*, vol. 11, no. 21, p. 10420, Nov. 2021.
[27] S. Sarker, Md. M. Haque, and M. A. A. Dewan, "Driving maneuver classification using domain specific knowledge and transfer learning," *IEEE Access*, vol. 9, pp. 86590–86606, 2021.
[28] X. Peng, Y. L. Murphey, R. Liu, and Y. Li, "Driving maneuver early detection via sequence learning from vehicle signals and video images," *Pattern Recognit.*, vol. 103, Jul. 2020, Art. no. 107276.
[29] S. Arefnezhad, S. Samiee, A. Eichberger, M. Frühwirth, C. Kaufmann, and E. Klotz, "Applying deep neural networks for multi-level classification of driver drowsiness using vehicle-based measures," *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113778.
[30] J. Xie, K. Hu, G. Li, and Y. Guo, "CNN-based driving maneuver classification using multi-sliding window fusion," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114442.
[31] Y. Zhang, J. Li, Y. Guo, C. Xu, J. Bao, and Y. Song, "Vehicle driving behavior recognition based on multi-view convolutional neural network with joint data augmentation," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4223–4234, May 2019.
[32] C. Huang, X. Wang, J. Cao, S. Wang, and Y. Zhang, "HCF: A hybrid CNN framework for behavior detection of distracted drivers," *IEEE Access*, vol. 8, pp. 109335–109349, 2020.
[33] X. S. Li, X. T. Cui, X. L. Zheng, Y. Y. Ren, L. Zhao, J. Wang, and W. Y. Kang, "A driving event clustering method and system based on an LDA extended model," China Patent 202 310 068 157, Jul. 28, 2023.
[34] J. Xie and M. Zhu, "Maneuver-based driving behavior classification based on random forest," *IEEE Sensors Lett.*, vol. 3, no. 11, pp. 1–4, Nov. 2019.

XIAOTONG CUI was born in Zibo, Shandong, China, in 1996. She received the bachelor's degree in traffic engineering from Harbin Institute of Technology, Weihai, Shandong, in 2018. She is currently pursuing the Ph.D. degree in vehicle operation engineering with the School of Transportation, Jilin University, Changchun, China. Her research interests include the understanding and utilizing driving behavioral data and the driving behavior characteristics recognition.

XIANSHENG LI received the bachelor's degree in automobile application engineering from Jilin University, Changchun, China, in 1982, and the Ph.D. degree in vehicle operation engineering from the School of Transportation, Jilin University. His research interests include driving safety and reliability and transportation system resources optimization.



XUELIAN ZHENG was born in Shandong, in 1987. She received the B.S. degree in automobile engineering from Northeast Forestry University, in 2009, and the Ph.D. degree in vehicle operation engineering from Jilin University, in 2014. Currently, she is with the School of Transportation, Jilin University, as a Vice President. Her research interests include vehicle dynamics and control, driving behavior, and autonomous vehicle design and control.

YUANYUAN REN was born in Jilin City, Jilin Province, in 1982. She received the B.S. and M.S. degrees in traffic information engineering and control and the Ph.D. degree in vehicle operation engineering from Jilin University, in 2006, 2008, and 2011, respectively. Her research interests include driving stability and safety technology, intelligent analysis of driving behavior, and intelligent vehicle planning and control research.