Physica A 632 (2023) 129332

Contents lists available at ScienceDirect

Physica A: Statistical Mechanics and its Applications

journal homepage: www.elsevier.com/locate/physa

A unified modeling framework for lane change intention recognition and vehicle status prediction

Renteng Yuan [a], Mohamed Abdel-Aty [b], Xin Gu [c], Ou Zheng [b], Qiaojun Xiang [a,*]

[a] Jiangsu Key Laboratory of Urban ITS, School of Transportation, Southeast University, Nanjing, Jiangsu 210000, PR China
[b] Department of Civil, Environmental and Construction Engineering, University of Central Florida, 12800 Pegasus Dr #211, Orlando, FL 32816, USA
[c] Beijing Key Laboratory of Traffic Engineering, Beijing University of Technology, Beijing 100124, PR China

ARTICLE INFO

Keywords:
Vehicle trajectory
Lane-change intention recognition
Driving status prediction
Multi-task TCN model
Attention Mechanism

ABSTRACT

Accurately detecting and predicting Lane Change (LC) processes of human-driven vehicles can help autonomous vehicles better understand their surrounding environment, recognize potential safety hazards, and improve traffic safety. This paper focuses on LC processes, first developing a Temporal Convolutional Network (TCN) with an attention mechanism (ATM) model to recognize LC intention. Then, considering the intrinsic relationship among output variables, the Multi-Task Learning (MTL) framework is employed to simultaneously predict multiple LC vehicle status indicators. Furthermore, a unified modeling framework for LC intention recognition and driving status prediction (LC-IR-SP) is developed. The results indicate that the classification accuracy of LC intention was improved from 95.83% to 98.20% when incorporating the ATM into the TCN model. For LC vehicle status prediction issues, Pearson's correlation coefficient indicates that metrics extracted from LC processes show stronger correlation than those extracted from Lane-keeping processes. Consequently, three multi-tasking learning models are constructed based on the MTL framework. The results indicate that the MTL with Long Short-Term Memory (MTL-LSTM) model outperforms the MTL with TCN (MTL-TCN) and MTL with TCN-ATM (MTL-TCN-ATM) models. Compared to the corresponding single-task model, the MTL-LSTM model demonstrates an average decrease of 26.04% in MAE and 25.19% in RMSE. The LC-IR-SP model developed holds great potential in enhancing autonomous vehicles' perception and prediction capabilities, such as identifying LC behaviors, calculating real-time traffic conflict indices, and improving vehicle control strategies.

1. Introduction

It can be expected that, for an extended period of time, vehicles with varying levels of automation will coexist on the roads [2,5]. During the transition period, assisting intelligent driving vehicles to understand and predict changes in the behavior of human-driven vehicles is particularly critical for driving decisions. LC is a common driving behavior that leads to two-dimensional spatial (longitudinal and lateral) interaction between vehicles. The LC process consists of a series of continuous, complex maneuvering actions that significantly impact road traffic efficiency and safety [10,11]; [28]; [83]. Accurately identifying and predicting lane change processes can help intelligent driving vehicles anticipate potential safety risks and execute appropriate response strategies.

* Corresponding author.
E-mail address: xqj@seu.edu.cn (Q. Xiang).

https://doi.org/10.1016/j.physa.2023.129332
Received 8 July 2023; Received in revised form 15 September 2023; Accepted 23 October 2023
Available online 31 October 2023

The LC behavior is a time-varying, continuous maneuvering process [1,65]. LC intention recognition has been a challenging problem in traffic engineering since it is hard to observe directly. There are two types of information that are typically utilized to identify LC intentions: vehicle dynamics indicators and driver characteristic indicators. Vehicle dynamics indicators include steering wheel angle, steering velocity, lateral velocity, turn signal, and brake pedal position [26,36,41,44]. In addition to the difficulty of obtaining certain information directly from human-driven vehicles (e.g., steering wheel angle, steering velocity, etc.), the reliability of the data obtained is also difficult to guarantee. For example, turn signal usage is reported to be between 44% and 40% in the US and China, respectively [41,61]. The driver characteristic indicators consist of head movement, eye movement, body gestures, and even electroencephalography [16]; [21]; [27]; [49]; [50]; [52]; [55]; [73]. Such information can only be gathered through sensors or driving simulation experiments. Inevitably, experimental settings constrain these investigations, such as potential concerns with low data quality, high cost, and small sample size, making it difficult to generalize and apply the research findings.

With the advancement of technology, traffic system monitors and road users can obtain massive, real-time, individualized, and high-precision vehicle trajectory data. Lane change trajectory prediction has attracted a lot of attention over the past few years [32, 53]; [78]. However, vehicle status indicators are more frequently utilized than vehicle trajectory information in practical engineering applications, such as real-time risk assessment, driving decisions, and vehicle control [30]; [59]; [84]. The research on driving status prediction can be classified into two categories: speed prediction [6,82,88,89] and steering angle prediction [22]; [29]; [33]. Previous modeling frameworks have required separate prediction models for each metric to predict the driving status, resulting in significant training time and potential conflicts between the prediction results of different metrics. In fact, these driving status indicators are interrelated, especially for vehicles performing lane-changing behavior [9,80]. Considering the correlation among indicators, developing a multi-task prediction model to predict multiple indicators simultaneously is necessary to reduce model training time and improve prediction performance [60].

To our knowledge, no study has been conducted specifically to focus on LC vehicle status indicator prediction. In this paper, the vehicle status was characterized using six variables, including the longitudinal velocity (v_x), lateral velocity (v_y), longitudinal acceleration (a_x), lateral acceleration (a_y), vehicle heading (θ), and yawRate ($\Delta\theta$). This paper, using vehicle trajectory data, aims to build a unified approach to LC intention recognition (LC-IR) and LC vehicle status prediction (LC-SP). The contribution of this paper is threefold.

- Firstly, a new unified modeling framework for Lane Change Intention Recognition and Status Prediction based on vehicle trajectory data is proposed. A new vehicle trajectory dataset (CitySim Dataset) is employed to develop the LC-IR-SP model. As far as we know, this is the first study to combine lane change intention recognition and status prediction.
- Secondly, to effectively capture crucial temporal features, this study integrates the attention mechanism into TCN networks, resulting in developing a novel TCN-ATM model specifically designed for LC intention recognition. The incorporation of the attention mechanism enhances the model's capacity to selectively focus on and extract pertinent temporal information.
- Thirdly, considering the inherent interdependencies among outcome variables, this study constructs three multi-task learning models (MTL-LSTM, MTL-TCN, MTL-TCN-ATM) for predicting driving status variables. To our knowledge, no studies simultaneously considered the intrinsic relationship between outcome factors to predict driving status indicators.

The rest of this paper is structured as follows. Section 2 presents a brief literature review. The data collection and post-processing are described in Section 3. In Section 4, a new unified modeling framework for Lane Change Intention Recognition and Status Prediction is proposed. The experimental results and discussion are included in Section 5. Section 6 draws out the conclusions of this study.

Table 1
A summary of the representative research for LC intention recognition.

Study	Data	Method	Number of Samples	Advance time	Accuracy (%)
([77]	Image	CNN	637	–	73.97
([31]	Image	GoogleNet & LSTM	714	3.76 s	74.46
([45]		Vision-cloud	-– 2(Pts)	–	79.2
([25]	Simulator	AT-BiLSTM	-– 2(5Pts)	3 s	93.33
([26]		BN	–(1 Pt)	–	95.4
([92]	Naturalistic	HMM	–(58 Pts)	–	83.22
([36]		SVM	139(6 Pts)	1.3	80
([73]		EBiLSTM	201(3 Pts)	0.5 s	96.1
([41]		HMM	642(50 Pts)	0.5 s	90.3
([20]		LSTM	814(6 Pts)	–	88.26
([17]		RVM	903(8 Pts)	3 s	88.51
([91]	Trajectory	NN	Above 1000	–	73.33
([72]		LSTM	—	2.5 s	92.40
([51]		Logit	Above 1000	–	66.41
([66]		LSTM	Above 1000	2 s	86.21
([76]		HMM	3410	6 s	94.4
([1]		Extra trees classifier	Above 1000	2 s	82
([32]		SVM	351	3 s	85

Notes: –represents Not reported; Pt represent the participants

2. Related research

There are three kinds of methods, including dynamic or kinematic models [34]; [37]; [40]; [56]; [63]; [71]; [74], statistical models [26,36,49,51,54,70], machine learning methods [8,13,20,31]; [35]; [42]; [48], have been widely used for the LC intention recognition. The method based on dynamic or kinematic models detects the vehicle's motion by considering the kinematic relationship among parameters (e.g., position, velocity, acceleration) and the different forces (the longitudinal and lateral tire forces or the road banking angle) that affect the vehicle motions. As classical statistical methods, multinomial logit models and Bayesian theory are utilized to predict the lane change probability. To capture the inherent characteristics of time series, three machine learning methods, including Hidden Markov Model (HMM), Support Vector Machines (SVM), and LSTM, have been widely used. The commonly used models and their performance are listed in Table 1.

The summary of the literature in Table 1 reveals several valuable conclusions. First, vision-based LC behavior recognition methods exhibit lower classification accuracy than other methods. Second, because simulators and natural experiments are limited by the small number of experimental participants and high data homogeneity, ensuring the model's generalizability is challenging. Third, machine learning-based models have better classification accuracy compared to statistics-based models. Fourth, the LSTM model is widely used for lane change intention recognition and has made great progress in improving the accuracy of the LC behavior recognition but still has excellent potential to improve classification accuracy.

The LSTM approach has two limitations: the gradient vanishing problem and the inability to perform parallel computation [64]. To overcome the above two constraints, the TCN, first proposed by [3], has attracted considerable interest. TCN is designed for processing sequential data, such as time series or natural language [24,39,43]. With dilated causal convolution layers, the TCN effectively captures long-term dependencies over multiple time scales in the input sequences. Following that, the TCN has achieved significant promotion in both regression and classification tasks, involving forecasting carbon prices [39], predicting wind speed [18]; [38], and diagnosing power converter faults [81].

Although vehicle status indicators can be extracted from the predicted trajectories, the method is restricted by error accumulation and lags in trajectory prediction results. Changes in vehicle velocity and driving direction will cause changes in vehicle trajectory. Extracting status indicators from predicted vehicle trajectories necessitates an extended prediction time of vehicle trajectories. Research reported that minor positioning errors might significantly affect extraction indicators [23]. Hence, building independent prediction models for driving status indicators is essential to improve predictive performance. The dilemma encountered by traditional end-to-end models is that to predict multiple indicators, a given model may be repeatedly trained to predict different indicators with the same input parameters, leading to computational redundancy and higher costs. To address this issue, the MTL framework was proposed first by Rich Caruana [7] and involved training a model to learn multiple related tasks simultaneously. As a promising area in machine learning, MTL aims to improve the performance of multiple related learning tasks by leveraging useful information among them [87]. The tasks can be supervised, semi-supervised, or unsupervised, and the model is designed to take advantage of shared representations across the various tasks to improve performance. Deng et al. [14] employed the MTL framework for traffic prediction, achieving up to 18% and 30% improvement in short- and long-term predictions. Xu et al. [75], Zhang et al. [86], and Gao et al. [19,75,86] demonstrate the effectiveness of the MTL structure in travel time prediction areas. However, the ascendancy of MTL in driving status has not been tested. To fill the gap, this paper considers the MTL framework for LC-driving status prediction.

3. Data

The publicly available CitySim dataset [90] is used in this research. The CitySim dataset is a drone-based vehicle trajectory dataset extracted from 12 locations with a sampling frequency of 30 Hz. Notably, the dataset supplies the vehicle center, head, tail, and bounding box vertices locations, thereby enabling an intricate assessment of their movements. With six lanes in two directions, a sub-dataset Freeway-B collected in Asia [15,90] is chosen to verify the performance of our proposed model. A snapshot of the freeway-B segment is shown in Fig. 1.

The freeway-B dataset is collected simultaneously with two UAVs over a 2230-ft basic freeway segment. Totally 5623 vehicle trajectories are extracted from 60 min of drone videos. This study focuses on lane change processes. After data processing (explained in more detail in the next section), a total number of 1023 vehicle trajectories are extracted ultimately from the freeway-B dataset, including 545 lane-change (LC) vehicle trajectories (240 left lane change (LLC) vehicle trajectories and 305 right lane change (RLC)

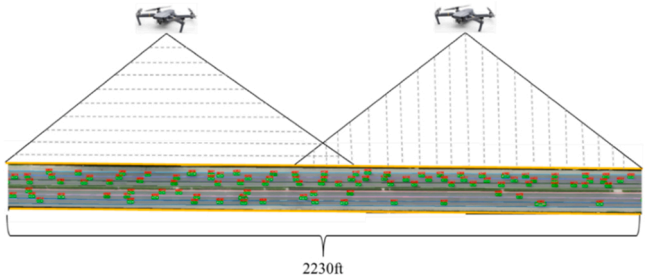


Fig. 1. A snapshot of the freeway-B segment.

vehicle trajectories) and 478 lane-keeping (LK) vehicle trajectories. Lane-keeping vehicle trajectories are randomly extracted.

3.1. Data processing

Four significant steps are further employed for data processing with extracted vehicle trajectory data.

(1) Removing abnormal data. The freeway-B dataset is collected from two stitched drone videos. The vehicle trajectory with the difference of adjacent frames greater than one is removed to avoid the effects of frame misalignment or skipping.
(2) Data smoothing. Minor positioning errors might significantly affect extraction indicators [23]. To reduce the negative effect of errors, a moving average (MA) method is used to smooth the trajectory, and the moving average filter is set to 0.5 s. A comparison of the original trajectory and processed trajectory is shown in Fig. 2.

(3) Indicator calculation. To accurately describe the vehicle driving status, six indicators are extracted from the two-dimensional (i.e. longitudinal and lateral) vehicle position coordinates, including the longitudinal velocity (v_x), lateral velocity (v_y), longitudinal acceleration(a_x), lateral acceleration (a_y), vehicle heading (θ), and yawRate (Δθ). Furthermore, a non-linear low-pass filter is employed to reduce the negative effect of measurement errors [12]. First, the vehicle speed at the *t-th* frame is calculated and can be formulated as.

$$v_n(t) = \frac{s(t+n) - s(t-n)}{2 \cdot nT} \tag{1}$$

Where t represents the current frame, T is a constant, representing 1/30 s in this research, n represents the time-step; $s(t-n)$ represents the vehicle's position in the frame $t-n$,where n takes different values, a vector $\{v_1(t), v_2(t), \ldots, v_N(t)\}$ (In this paper, n is set to 8) will be obtained. Thus, the vehicle velocity $v(t)$ at the *t-th* frame is calculated by taking the median of all N time steps. The lateral velocity (v_y) and longitudinal velocity (v_x) can be determined based on the change in the lateral and longitudinal positions of the vehicle, respectively. With the calculated velocity, acceleration can be obtained as.

$$a(t) = \frac{v(t+1) - v(t-1)}{2 \cdot T} \tag{2}$$

The lateral acceleration (a_y) and longitudinal acceleration (a_x) also can be determined based on the change of v_y and v_x, respectively. In addition, the vehicle heading can be calculated as,

$$\theta_n(t) = \arctan\left(\frac{y_H(t+n) - y_R(t-n)}{x_H(t+n) - x_R(t-n)}\right) \tag{3}$$

Where $\theta_n(t)$represents the vehicle heading at the frame t,$y_H(t+n)$ is the vehicle head point longitudinal position in the frame $t+n$, $x_R(t+n)$ denotes vehicle tail point horizontal position in frame $t+n$. yawRate is used to represent the rate of change of the vehicle's steering wheel angle [30]. It is calculated as,

$$\Delta\theta(t) = \frac{\theta(t+1) - \theta(t-1)}{2T} \tag{4}$$

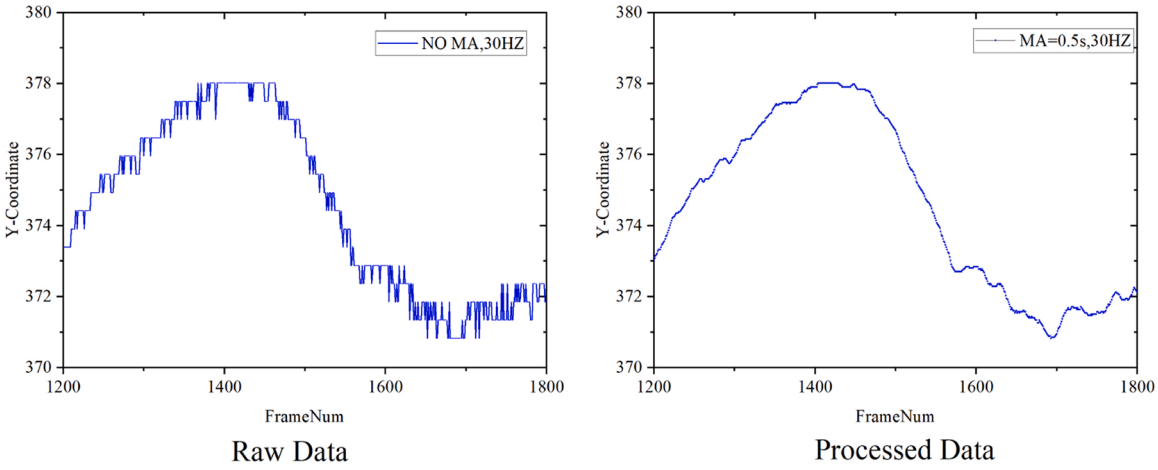


Fig. 2. Comparison of original trajectory and processed trajectory.

(4) Normalization. Variations in magnitude and units among different metrics can have an impact on the outcomes of data analysis. To mitigate this issue, it is necessary to standardize all the indicators.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{5}$$

3.2. Input indicator

The vehicle status is influenced by other vehicles in the driving environment [85]. To fully consider the impact of various factors, the input of the combined model consists of three parts: ego vehicle (E-vehicle) information, surrounding vehicle information, and relative position information. Surrounding vehicles include the closest preceding and following vehicles in the adjacent and the current lanes. The ego vehicle is the human-driven vehicle. This research aims to detect and predict the human-driven vehicle LC process. The six indicators (v_x, v_y, a_x, a_y, θ, $\Delta\theta$) were calculated for each vehicle. Limited by the video coverage, some trajectory fragments of surrounding vehicles were not recorded. A categorical variable (0 means it has recorded trajectory information; 1 means the trajectory information is missing) is added to each surrounding vehicle indicating this phenomenon. For instance, when the ego vehicle first appeared, the following vehicle (F-vehicle) was not yet in the drone videos. The following vehicle status variable (*F-val*) is set to 1. Relative position information (*dw*) is the headway distance between the E-vehicle and other vehicles, as shown in Fig. 3. If the corresponding vehicle is not recorded in drone video, the corresponding *dw* is set to 0. Ultimately, a total of 54 indicators are taken as input variables. More details can be obtained from Table 2.

4. Method

In this section, this paper first proposes a new modeling framework for Lane Change Intention Recognition and vehicle Status Prediction. Then, by incorporating an attention mechanism, a new novel TCN-ATM model is proposed. Furthermore, based on the MTL framework, several multitask learning models, MTL-TCN-ATM, MTL-TCN, and MTL-LSTM, are developed to predict LC vehicle status. Finally, the commonly used evaluation metrics are presented.

4.1. Modeling framework

Fig. 4 presents the framework of the proposed LC-IR -SP model, which consists of two core modules: the LC-IR module and the LC-SP module. The LC-IR module is a classification model used to recognize whether the vehicle produces LLC intention or RLC intention. When the LC-IR module detects that a vehicle generates a lane change intention, the LC-SP module will predict the LC vehicle driving status. The LC-SP module consists of separate multi-task learning and single-task learning models for sequence-to-sequence prediction. Multi-task learning models are employed to predict related variables. Unrelated variables were predicted separately using a single-task model.

(1) LC-IR module.LC intention is divided into three categories: lane keeping (LK), left lane changing (LLC), and right lane changing (RLC). The generation of LC intention is a complex process, and it is influenced by other vehicles in the driving environment [25,49]. As mentioned above, there are 54 indicators used as input variables in this research. Lane Change Intention Recognition can be conceptualized as a classification issue based on multivariate time-series data. The function of the LC-IR module is defined as,

$$L_t = \varphi(\ell \cdot S_{t-\Delta t:t}) \tag{6}$$

The output L_t represents the LC intention of the ego vehicle at time t, which is labeled as 1, 2, and 3 for LK, RLC, and

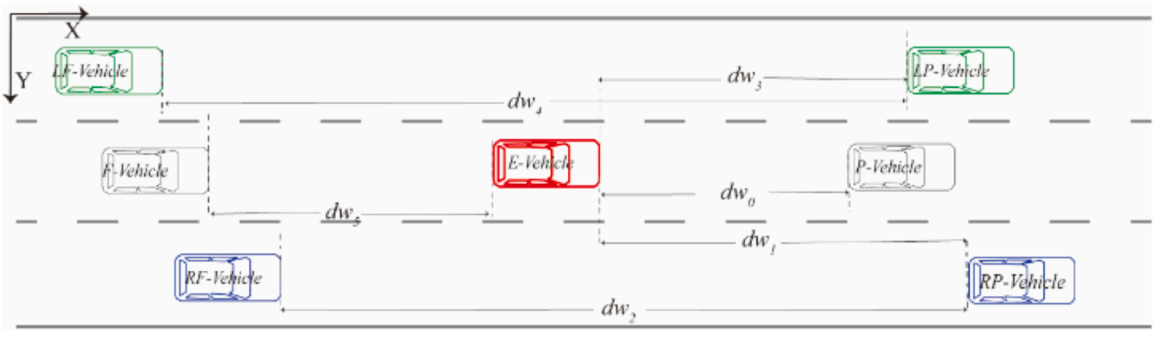


Fig. 3. The headway distance between the E-vehicle and surrounding vehicles.

Table 2
Input indicators of the model.

Inputs Variable	Variable descriptions
E-, P-, F-, LP-, LF-, RP-, RF-v_x	The longitudinal velocity of E-vehicle and surrounding vehicle (ft/ sec)
E-, P-, F-, LP-, LF-, RP-, RF-v_y	The lateral velocity of E-vehicle and surrounding vehicle (ft/ sec)
E-, P-, F-, LP-, LF-, RP-, RF-a_x	The longitudinal acceleration of E-vehicle and surrounding vehicle (ft/ sec^2)
E-, P-, F-, LP-, LF-, RP-, RF-a_y	The lateral acceleration of E-vehicle and surrounding vehicle (ft/ sec^2)
E-, P-, F-, LP-, LF -, RP-, RF-θ	The heading of E-vehicle and surrounding vehicle (degree)
E-, P-, F-, LP-, LF -, RP-, RF-$\Delta\theta$	The yawRate of E-vehicle and surrounding vehicle (degrees/sec)
dw_0, dw_1, dw_2, dw_3, dw_4, dw_5	Space headway between E-vehicle and surrounding vehicle (ft)
P-, F-, LP-, LF -, RP-, RF-val	0 means it has recorded trajectory information; 1 means the trajectory information is missing

Note: "E-" represents the ego vehicle; "P-" represents the closest preceding vehicle in the same lane; "F-" represents the closest following vehicle in the same lane; "LP-" represents the closest preceding vehicle in the adjacent left lane; "LF-" represents the closest following vehicle in the adjacent left lane; "RP-" represents the closest preceding vehicle in the adjacent right lane; "RF-" represents the closest following vehicle in the adjacent right lane.

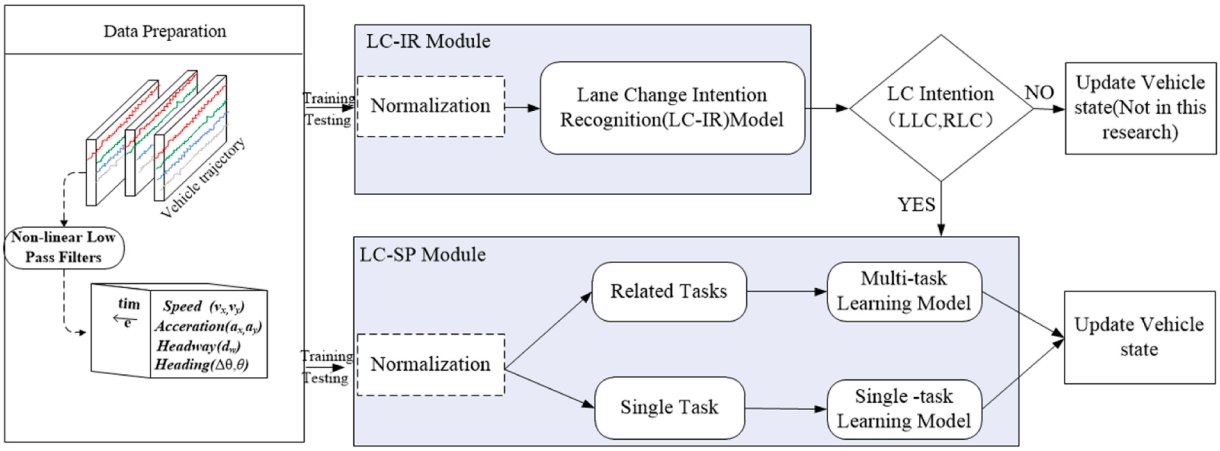


Fig. 4. Modeling framework based on deep learning.

LLC;$S_{t-\Delta t:t} = \{v_x, v_y, \ldots, \Delta\theta\}_{t-\Delta t:t}$represents the input variables described in Section III, the notation $t-\Delta t:t$ denotes a time-series of the indicator from time $t-\Delta t$ to time t; $\diagup$ denotes the parameter vector; φ() represents the mapping relationships.

(2) LC-SP module. The driving status is represented by six indicators: v_x, v_y, a_x, a_y, θ, and Δθ. Velocity (v_x, v_y) and heading (θ) can be regarded as macroscopic indicators that reflect the aggregated effects of prior driving behavior. Meanwhile, acceleration (a_x, a_y) and yawRate (Δθ), used as microscopic indicators, indicate the driving behavior that the driver is about to perform, reflecting changes in the throttle, brake pedal, and steering angle of the vehicle, respectively. Predicting LC vehicle driving status requires the simultaneous prediction of these six indicators (v_x, v_y, a_x, a_y, θ, and Δθ).

The LC process usually lasts for several seconds. With a 1 s interval (indicators take an average of 60 frames), lane-change vehicle status in the next 2 s is predicted in this study. For instance, taking the longitudinal velocity (v_x) and the lateral velocity (v_y) as an example, the function of the LC-SP module is defined as,

$$\begin{aligned}\left(v_{y,t+1}, v_{y,t+2}\right) &= g_1(\xi_1 \cdot R_{t-\Delta t:t}) \\ \left(v_{x,t+1}, v_{x,t+2}\right) &= g_2(\xi_2 \cdot R_{t-\Delta t:t})\end{aligned} \tag{7}$$

Where $v_{x,t+1}, v_{x,t+2}$ represent the longitudinal speed of the ego vehicle at time t + 1 and time t + 2, respectively; $v_{y,t+1}, v_{y,t+2}$ represent the lateral speed of the ego vehicle at time t + 1 and time t + 2, respectively. Compared with (6), the input $R_{t-\Delta t:t}$has an additional variable $L(t-\Delta t:t)$,which denotes the LC intention from time t −Δt to time t. $g_1()$ and $g_2()$ represent the mapping relationships. ξ_1 and ξ_2 denote the parameter vector. The expected six output variables (v_x, v_y, a_x, a_y, θ, Δθ) are simultaneously influenced by the same surrounding environment. By intelligently leveraging the inherent relationships between variables, it becomes possible to enhance prediction accuracy effectively.

4.2. Temporal convolutional networks

TCN consists of causal convolution and dilated convolution [3,24]. Causal convolutions are used to ensure the temporal

dependencies of the input data. Given an input sequence $x_0, x_1, \ldots, x_n$ and the corresponding output sequence $y_0, y_1, \ldots, y_n$, the causality constraint causal convolutions ensure that the output y_t at time t is only determined by the input sequence $x_0, x_1, \ldots, x_t$. The one-dimensional fully-convolutional network (1DFCN) architecture is employed to produce the same length output as the input [46]. The TCNs can be expressed as,

$$TCNs = 1DFCN + causalconvolutions \tag{8}$$

Using causal convolution, it is theoretically possible to generate TCN. However, the receptive field of causal convolution is constrained, making it difficult to capture the correlation between points in a long-term temporal sequence. Hence, dilated convolutions were added to causal convolutions, enabling an exponentially large receptive field. For a filter $f : \{1, 2, \ldots, k-1\}$, the dilated convolution operation F on the element s of a 1-D sequence $x \in R^n$ is formulated as,

$$F(s) = \left(x *_d f\right)(s) = \sum_{i=0}^{k-1} f(k) \cdot x_{s-d \cdot i} \tag{9}$$

Where d is the dilation parameter and is used to control the size of the interval, k is the filter size and represents the number of convolution kernels, * is the convolution operator, $s - d \bullet i$ accounts for the direction of the past. A dilated convolution will be backward to a full convolution when $d = 1$. The dilated causal convolution structure is depicted in Fig. 5.

As shown in Fig. 5, the kernel size is set to 2, and the depth of the causal convolution is 3. The convolution indicated that the output at time t is associated with the input points from time t-7 to time t. Residual blocks are used to address disappearance and gradient expansion in TCN. Utilizing techniques such as longer convolutional kernels and residual connections allows TCN to capture long-term dependencies. As shown in Fig. 6, the rectified linear unit (ReLU) is utilized as an activation function, and batch normalization is used as the convolutional filter. A 1×1 convolution is added in the residual block when the input and output data have different lengths.

By adjusting dilation parameters, the amount of information received by the TCN can be changed. The receptive field of the TCN can be calculated as,

$$R_{field} = 1 + (K - 1) \times N_{stack} \times \sum_i d_i \tag{10}$$

Where R_{field} represents the receptive field of the TCN, K is the filter size, N_{stack} represents the number of stacks, d_i represents the dilation parameter in the ith layer.

4.3. TCN with attention mechanism model

To prioritize important input elements and enhance model performance and generalization, this study introduces an attention mechanism into the TCN network, creating a novel TCN-ATM model. The attention mechanism can be understood as a straightforward weighted summation operation. The relevant equations are formulated as,

$$u_t = tanh(\omega * h_t + b) \tag{11}$$

$$a_t = softmax(u_t) \tag{12}$$

$$c = \sum_{t=1}^{n} a_t * h_t \tag{13}$$

Where h_t represents the extracted features by TCN layers at time t, ω is the parameter matrix at time t, a_t is the weight of h_t and could be calibrated based on the impact of each data feature on the output. c denotes the weighted sum of features h_t at time t. The structure of TCN-ATM is depicted in Fig. 7.

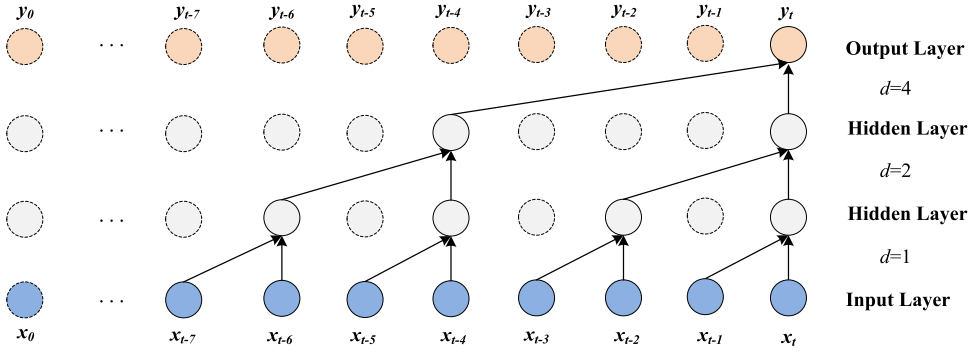


Fig. 5. A dilated causal convolution with dilation factors d = 1,2,4 and kernel size k = 2 [68].

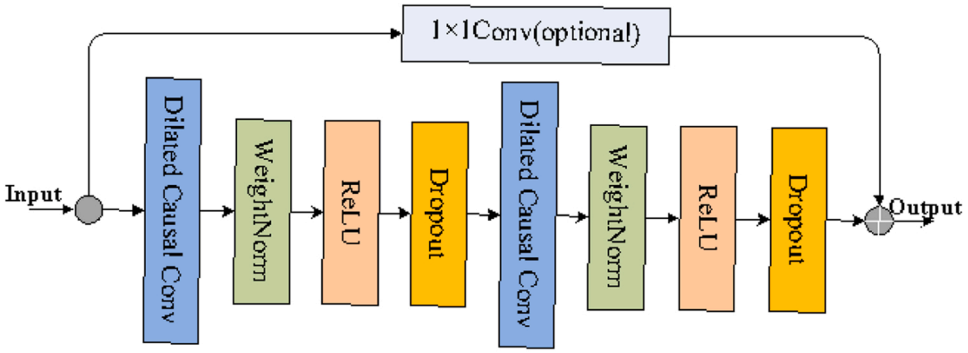


Fig. 6. TCN residual block.

The TCN-ATM model architecture consists of an input layer followed by a TCN layer and an attention layer. The TCN layer processes the input data while preserving the sequence information. An attention mechanism is then applied to the TCN output, capturing important features. Next, a global max pooling layer condenses the information into a fixed-length representation. Finally, a dense layer with softmax activation is added to produce class probabilities. This model architecture combines TCN and attention to effectively extract temporal patterns and make accurate predictions and classification tasks.

4.4. Multi-task prediction model

MTL can be viewed as a generalization of multi-label learning and multi-output regression [87] and has the advantages of improving data efficiency, generalization ability, regularization ability, and overall performance[14]. It is designed to leverage a shared representation at the bottom layer and simultaneously enables learning multiple related tasks. For the kth task, the output y_k in MTL can be expressed as:

$$y_k = h^k(f(x)) \tag{14}$$

Where the f function represents the shared-bottom network, h^k denotes the kth tower network, and x is the input variable vector. Compared to single-task learning, the multi-task learning framework can share information among different tasks, reduce training time, and improve the efficiency of data utilization [67]. Given three learning tasks, the comparison of single-task and multi-task learning architecture based on single-layer networks is shown in Fig. 8. Tasks 1, 2, and 3 are three related output variables that share common input indicators. Using a single-task learning model requires building three separate models, but in the multi-task learning framework, only one model is required to be constructed with three outputs.

The loss function is a critical component of multi-task learning. How to design the loss function for multi-task learning is crucial to determining the performance of the model. A common approach is to use a linear function to directly combine these loss functions, as shown,

$$Loss_{total} = \sum_i \omega_i Loss_i \tag{15}$$

Where $Loss_{total}$represents the cumulative loss of all tasks, $Loss_i$ is the loss of the *i-th* task, and ω_i is the weight of the *i-th* task. By adjusting

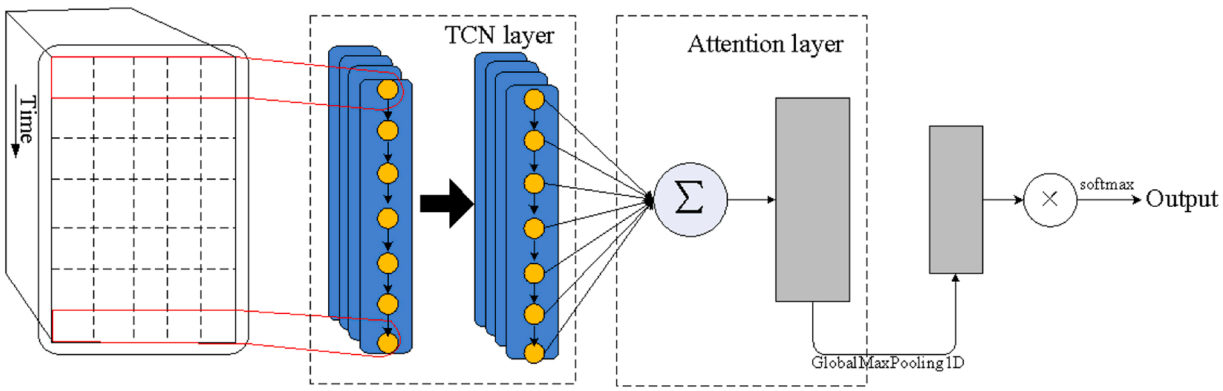


Fig. 7. The structure of the TCN-ATM model.

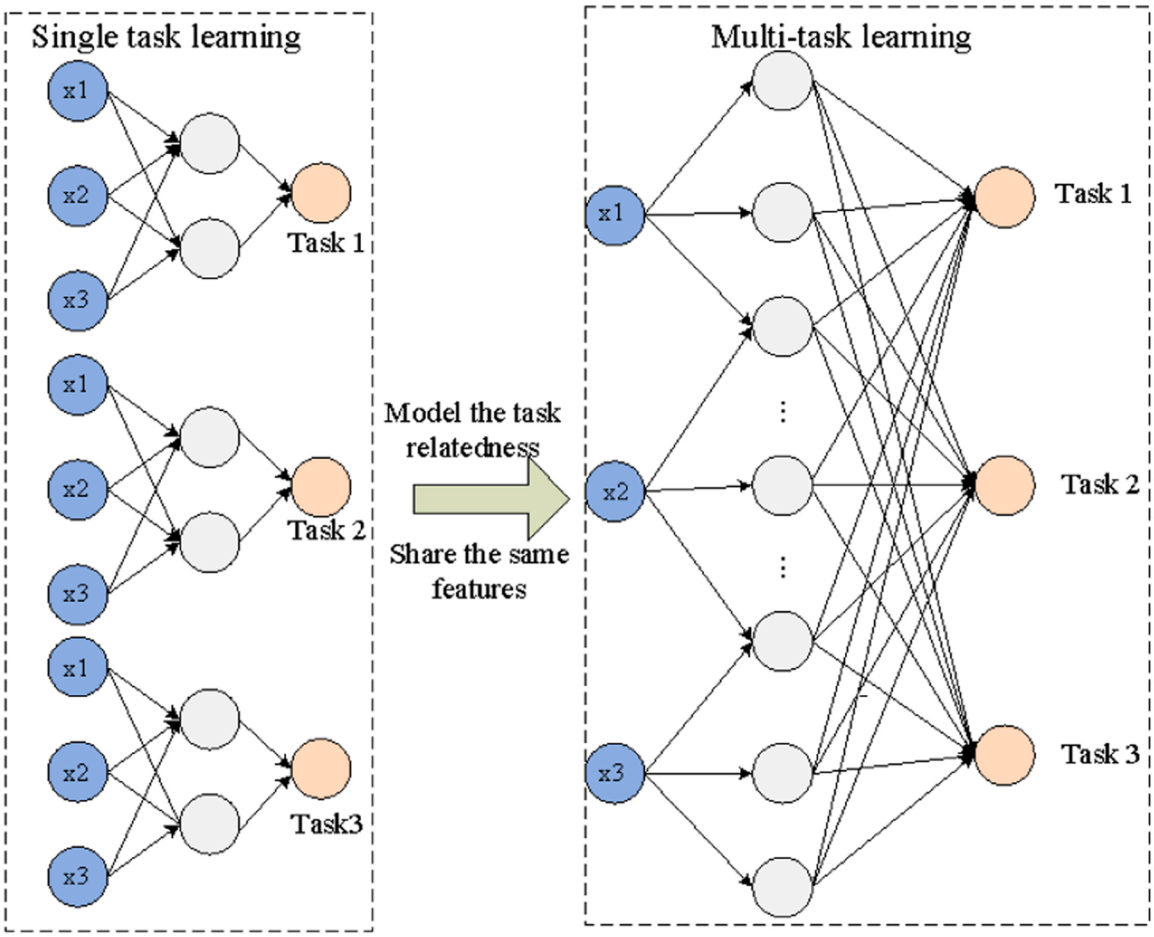


Fig. 8. The comparison of single-task and multi-task learning architecture.

ω_i, the model performance for the *i-th* task can be changed. For instance, if there is a main task in all the tasks, increasing the loss weight of the main task can improve the model performance for it. In this study, all tasks are considered equally important with assigned equal weights.

Based on the multi-task model framework, three multi-task models (MTL-LSTM, MTL-TCN, MTL-TCN-ATM) are first developed in

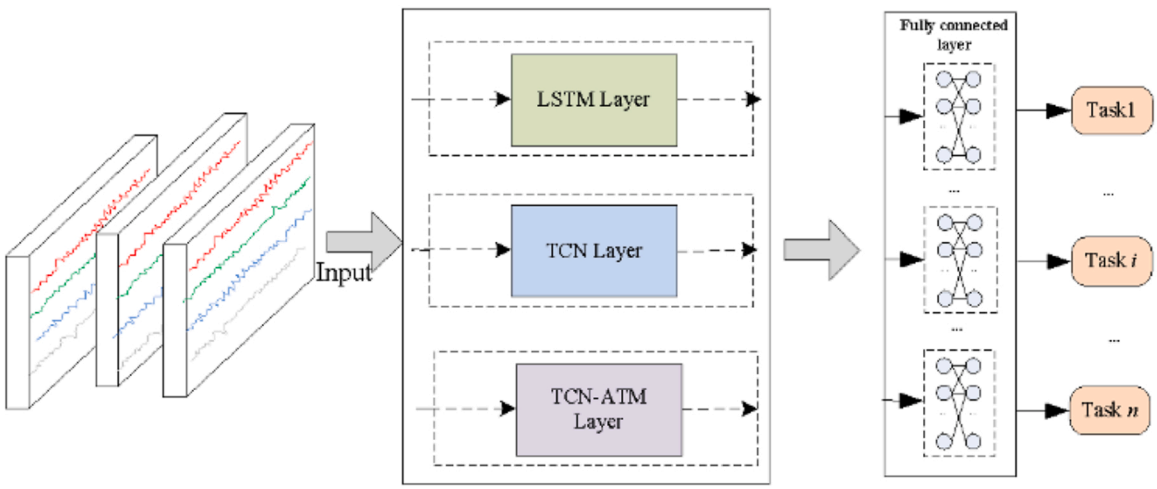


Fig. 9. The general model structure of the three multi-task learning models.

this study. The multi-task model consists of a feature processing layer and a fully connected layer. The feature processing layer is used to extract temporal features, which can be LSTM or TCN or TCN-ATM. The general model structure is shown in Fig. 9. In the LSTM layer, TCN layer, or TCN-ATM layer, multiple tasks benefit from shared parameters and features. This sharing allows the model to leverage common patterns and relationships across different tasks, leading to improved performance and efficiency. Then, fully connected layers are employed to output prediction results for each variable.

4.5. Evaluation indexes

The modeling framework proposed includes classification models and sequence prediction models. The performance of classification models is evaluated from two aspects. One is the overall performance of the classification, and the other is the recognition performance of each class [79]. The two indexes, precision and recall, are used to evaluate the detection performance of each class. The accuracy index measures the overall performance of the model. The three indexes can be calculated as follow,

$$Accuracy = \frac{T}{T+F} \tag{16}$$

$$Precision = \frac{TP}{TP+FP} \tag{17}$$

$$Rcall = \frac{TP}{TP+FN} \tag{18}$$

Where T represents the number of correctly classified samples, F represents the number of incorrectly classified samples, TP is the number of correctly classified samples in a given class, FP is the number of incorrectly classified samples in a given class, FN denotes the number of incorrectly classified samples in a given class. The two indexes, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are employed to evaluate the performance of sequence prediction models. The definitions are as follows,

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y}_i| \tag{19}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2} \tag{20}$$

Where y_i is the observed value of the i-th output, N is the number of outputs, $\hat{y}_i$ represents the predicted value of y_i. The prediction model with lower MSE and RMSE values performs better.

5. Results

To testify the feasibility of our modeling framework, the LC-IR model and the LC-SP model are developed in this section, respectively. And we selected 545 LC vehicle trajectories and 478 LK vehicle trajectories for training and testing the lane change intention recognition model. However, only LC vehicle trajectories were used to train and test the LC vehicle status prediction model.

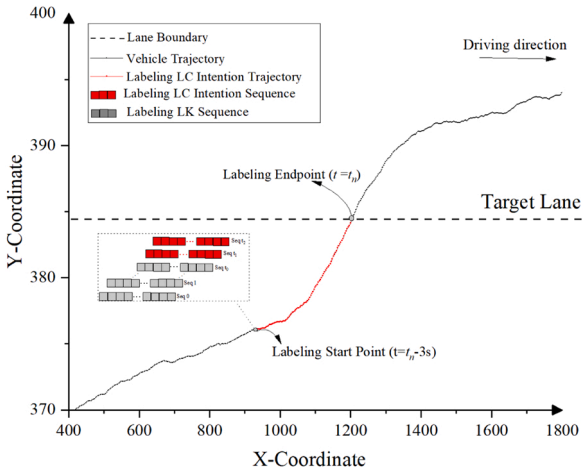


Fig. 10. Lane-change intention labeling process.

5.1. Lane change intention recognition

The vehicle lane change intentions are defined as some LC operational behavior produced before the lane change. To select an appropriate algorithm for classifying lane change intentions, this section compares the performance of four algorithms: LSTM, SVM, TCN, and TCN-ATM.

5.1.1. Lane-change intention labeling

In this study, the start time of the LC process is determined as the moment when the front boundary point of vehicles touches the lane boundary [84]. The annotation procedure determines the LC intention start time as 3 s forward the start time of LC processes[17]; [25]; [32]. The start point of LC processes is considered to be the end point of LC intention. A total of 24,092 frames are labeled as RLC points, while 19,792 frames are labeled as LLC points. Fig. 10 shows the detailed labeling processes. If the extracted sequence endpoint is located between the start time of the LC process and the LC intention start time (at least one RLC point or LLC point is included in the sequence), it is labeled as either LLC or RLC; Otherwise, it is labeled as LK.

5.1.2. Results of LC intention recognition models

The dataset is randomly split into a training dataset and a test dataset with a ratio of 8:2. For training the LC intention classification model, eighty percent of total data is applied, and twenty percent of samples are used for testing the classification performance. The parameter setting will affect the performance of the model. To obtain optimal parameter settings, some sensitivity experiments are performed on four models, using the control variable method. The parameters are selected based on the metrics of classification accuracy and training time. The final model used should minimize the training time of the model (reduce the complexity of the model) without compromising the accuracy of the model. With an equal number of samples, all experiments are conducted using the same device. As an example, the impact of the number of epochs was evaluated with maintaining the same input durations (input time duration = 5 s). Fig. 11 depicted the results of the experiment. It is evident that when the number of epochs is set to a value greater than 50, the loss function does not exhibit significant changes. Hence, the epoch is set to 50. Finally, the kernel size is set to 2, the Batch size is set to 128, the Loss function is set to *categorical cross entropy*, and the Adam optimizer was employed. The rate used in the dropout layer is 0.3. The size of dilated convolution interval is set to $\{2^1,2^2,2^3,\ldots,2^n\}$, which depends on the input time series length. The number of filters is set to 64. The number of stacks of residual blocks is set to 1.

In order to investigate the effect of input sequence length on the classification results, we evaluated the TCN-ATM model with different input durations. A total of 12 input lengths ranging from 30 frames (1 s) to 180 frames (6 s) were extracted at 15-frame intervals. Eight models such as SVM, Random Forest (RF), ExtraTrees (ET), Convolutional Neural Network (CNN), LSTM, Gated Recursive Unit (GRU), TCN, and TCN-LSTM are used as benchmark models. Among them, SVM, RF and ET models are three traditional machine learning models, CNN, LSTM and GRU are three commonly used deep learning models, and TCN-LSTM is developed based on TCN.

Fig. 12 illustrates the overall accuracy comparison results of the nine models. It can be observed that, except for the CNN, each model has good classification performance (above 80%), even though the eight models are slightly different among different durations. The classification performance of SVM, GRU, CNN, and LSTM is more sensitive to the length of the input time series than the other five models. With the same input data time scale, the TCN-ATM model, in most cases, outperforms the other eight models. The best classification accuracy was achieved for five models (LSTM, CNN, TCN, TCN-LSTM, and TCN-ATM) when the input length was set to five seconds. Despite not attaining optimal accuracy using a 5-second input time length, the SVM, RF and ET al.gorithm exhibited a marginal variance in classification accuracy (less than 0.5%) compared to the optimal accuracy. Hence, a time duration T = 150 frames (5 s) was chosen as input sequence lengths. Finally, 22160 RLC sequences and 15410 LLC sequences were extracted. To

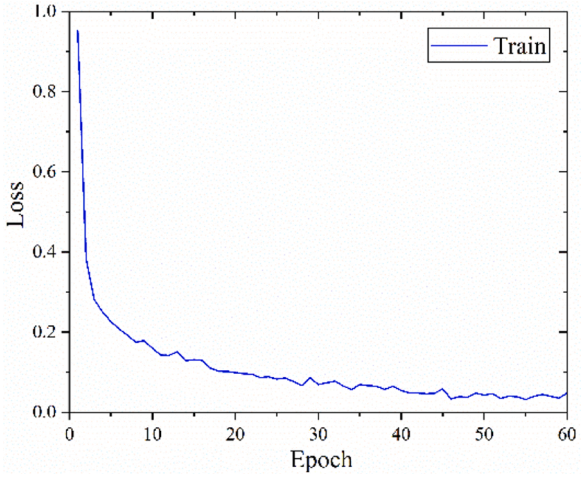


Fig. 11. loss function.

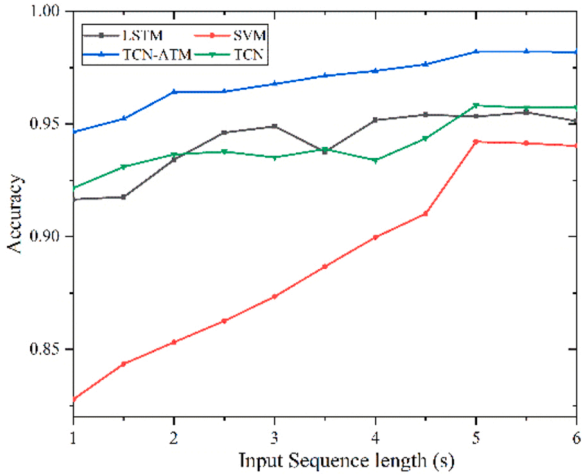


Fig. 12. Accuracy comparison of nine different models.

maintain data balance, 18000 LK sequences are randomly extracted from the raw dataset. Using the training dataset, the ten-fold cross-validated method is employed for model training and evaluation.

Fig. 13 illustrates the outcomes of a Ten-fold cross-validation analysis conducted on the TCN-ATM model. The average accuracy for TCN-ATM algorithms is 0.982, with a standard deviation of 0.00001. The result indicated that TCN-ATM algorithms demonstrate high accuracy and consistency in their performance. With an input length of 150 frames, Fig. 14 illustrates the confusion matrix for the four models using the validation set.

Errors in classifying LC intentions can be categorized into three categories: the misidentification of LK as LC (Type I), the misclassification of LC as LK (Type II), and the misidentification of LLC and RLC from each other (Type III). Fig. 14 shows that the proposed TCN-ATM algorithm reduces the impact of Type II and Type III errors compared to the other models. Type I errors significantly impact the accuracy of all four models. This error could originate from two sources. One is that the model correctly identifies the behavior of a failed lane change. The other could be attributed to the variations in individual lane change behaviors among drivers [4,62]. The LK process is influenced by factors such as driving style and driving ability, which can exceed the cognitive capabilities of the model, resulting in misjudgment. To provide a comprehensive assessment of classification performance, in addition to accuracy, other evaluation metrics such as precision, recall, and training time were evaluated through the confusion matrixes. The comparison results are displayed in Table 3.

Table 3 presents the classification performance of TCN, TCN-LSTM, and TCN-ATM with an accuracy of 95.83%, 96.67%, and 98.20%, respectively. SVM, ET, and RF demonstrate overall performance rates of 94.21%, 96.67%, and 94.39%, respectively. Meanwhile, LSTM, GRU, and CNN exhibit overall performance levels of 95.33%, 92.19%, and 78.71%, correspondingly. The results indicate that models developed based on TCN exhibit superior classification performance when compared to traditional machine learning algorithms such as SVM, RF, and ET, as well as other deep learning models like LSTM, GRU, and CNN. For instance, compared to LSTM, GRU, and CNN, the classification performance is improved using TCN by 2.87%,6.01%, and 19.49%. The results indicate that

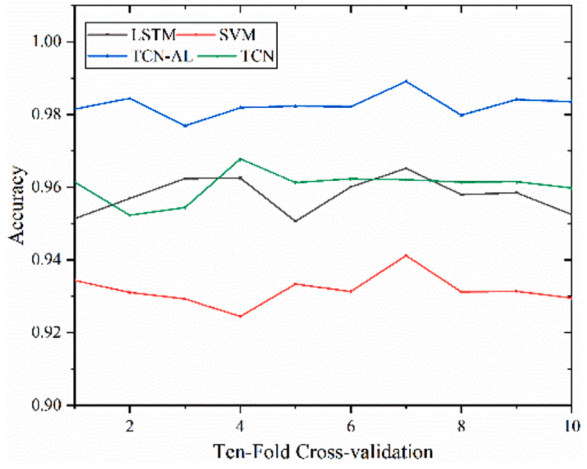


Fig. 13. Ten-fold Cross-validation for the TCN-ATM model.

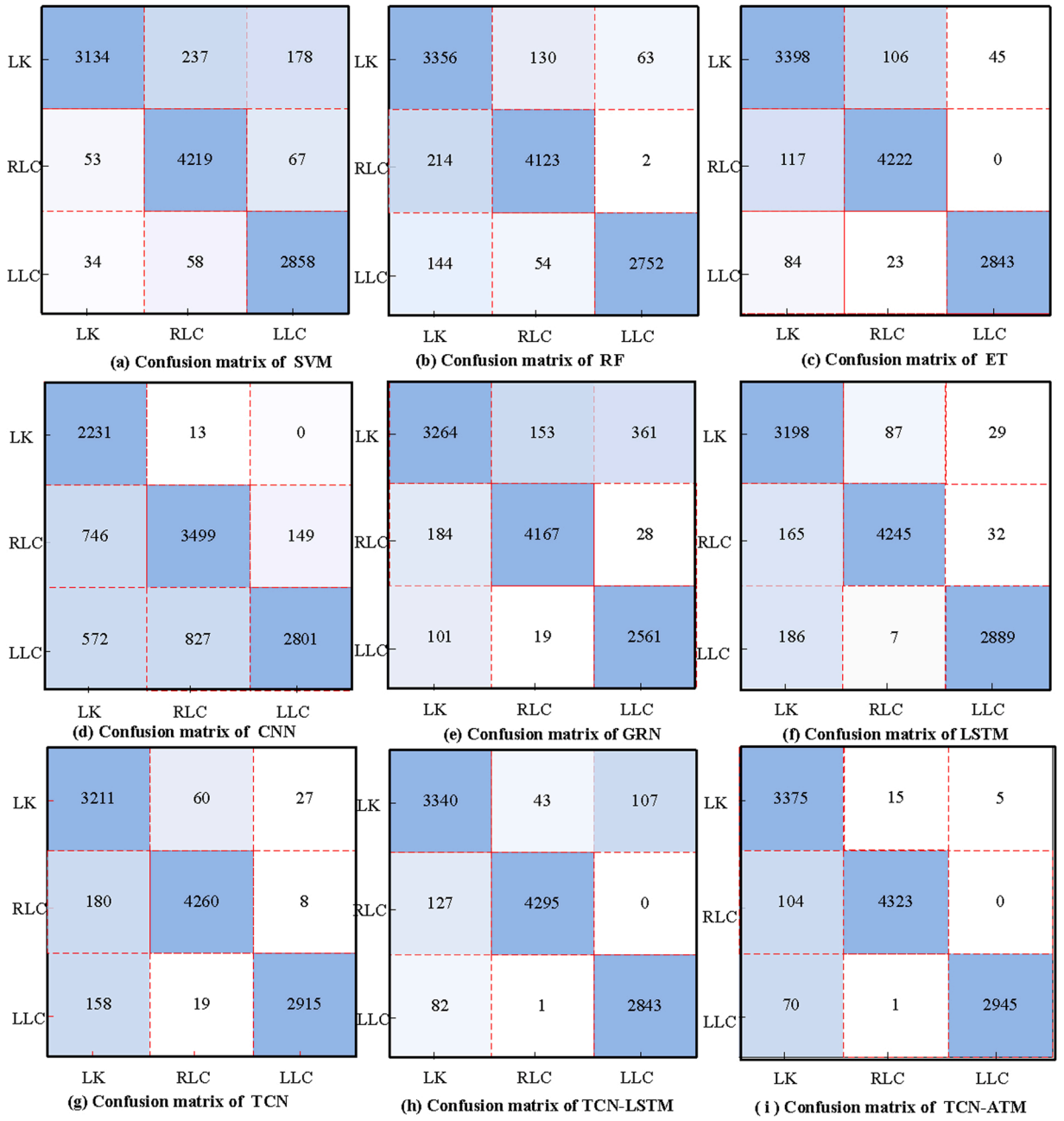


Fig. 14. Confusion matrix of classification models.

the TCN effectively captures long-term dependencies over multiple time scales in the input sequences. Notably, the TCN-ATM model achieves the highest classification accuracy, exhibiting improvements of 2.37% and 1.53% compared to the TCN and TCN-LSTM models, respectively. The results indicate that incorporating attention mechanisms into TCN to focus on relevant information can improve classification performance. On the other hand, when considered individually, the maximum deviation in classification precision is 8.92% for SVM, 8.34% for ET, 7.33% for RF, 9.22% for GRU, 7.73% for LSTM, 8.29% for TCN, and 4.74% for TCN-ATM. The maximum difference in recall index for each model is 1.87% for TCN-ATM. The results indicate that the TCN-ATM model provides more balanced results than other models. In summary, the proposed TCN-ATM model provides a promising solution for LC intention classification tasks, as it outperforms other models regarding classification accuracy.

Table 3
Evaluation results of nine different models.

Model	Type	Precision	Recall	Accuracy	Model	Type	Precision	Recall	Accuracy
SVM	LK	88.31%	97.29%	94.21%	ET	LK	94.41%	95.74%	96.53%
	RLC	97.23%	93.46%			RLC	97.03%	97.30%	
	LLC	96.88%	92.10%			LLC	98.44%	96.37%	
RF	LK	90.36%	94.56%	94.39%	LSTM	LK	90.10%	96.21%	95.33%
	RLC	95.72%	95.02%			RLC	97.83%	95.78%	
	LLC	97.69%	93.28%			LLC	97.79%	93.73%	
GRU	LK	91.96%	86.39%	92.19%	CNN	LK	62.86%	99.42%	78.71%
	RLC	96.03%	95.15%			RLC	80.64%	79.63%	
	LLC	86.81%	95.52%			LLC	94.94%	66.69%	
TCN	LK	90.47%	97.36%	95.83%	TCN-LSTM	LK	94.19%	95.70%	96.67%
	RLC	98.17%	95.77%			RLC	98.99%	97.13%	
	LLC	98.81%	94.28%			LLC	96.37%	97.16%	
TCN-ATM	LK	95.09%	99.41%	98.20%					
	RLC	99.63%	97.65%						
	LLC	99.83%	97.64%						

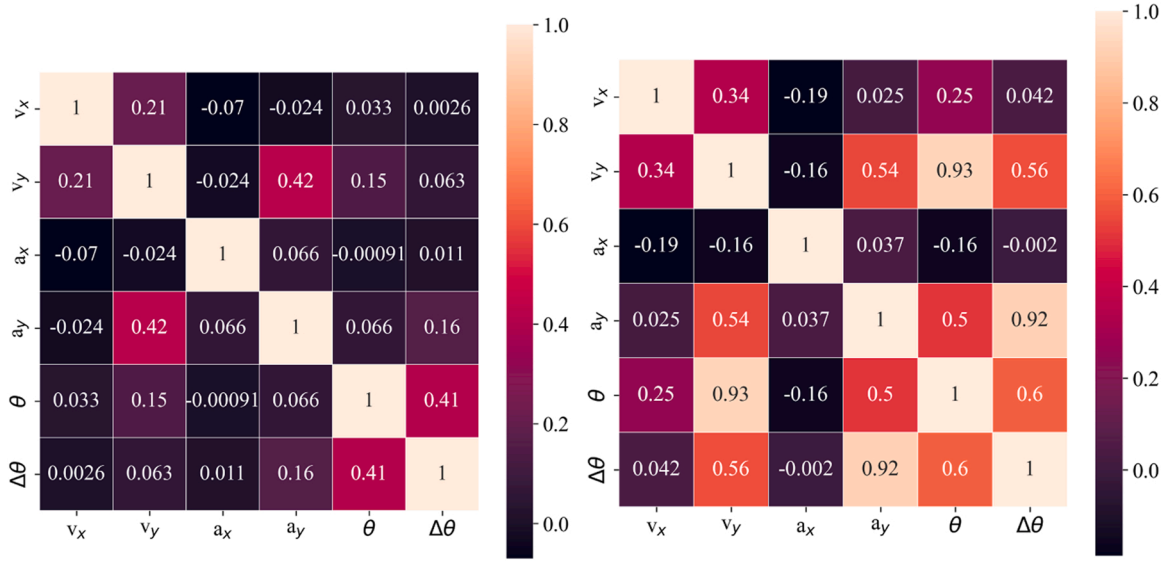
5.2. Lane change status prediction

Lane change vehicle status involves six indicators: v_x, v_y, a_x, a_y, θ, Δθ. Predicting lane change status requires the simultaneous prediction of these six indicators. This section first uses the Pearson coefficient to investigate the relationship between those output indicators. Then three proposed multi-tasking learning models are used to capture the intrinsic relationship among these indicators. Since SVM, RF, and ET al.gorithms are commonly used for classification learning, LSTM outperforms GRU and CNN algorithmic models in classification tasks. Therefore, only the LSTM, TCN, and TCN-ATM algorithms are considered in this section.

5.2.1. Correlation analysis

MTL framework involves jointly learning multiple output indicators. The underlying assumption behind this approach is that all output indicators are related. Typically, the relationship between tasks could significantly affect the predictive quality of multi-task models [47]. Hence, the Pearson coefficient is employed to investigate whether there is an association between the variables. It can be expressed as,

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{21}$$



(a) Indicators Extracted from LK Sequences (b) Indicators Extracted from LC Sequences

Fig. 15. Pearson coefficient heat map.

Where x_i represents the ith value of the indicator x, $\overline{x}$ and $\overline{y}$ represent the average value of indicator x and y, r represents the Pearson coefficient and takes values in the range [− 1,1]. The larger the absolute value of r, the stronger the correlation. In this research, only the indicators with an absolute value of Pearson coefficient greater than 0.2 are considered to be related[57]; [58]. There are two types of vehicle information used to calculate the Pearson coefficient separately: LK vehicles and LC vehicles. The LC vehicle information used is the driving intention labeled segment defined in Fig. 11. To mitigate the effect of sample imbalance on the results, 200 samples from each LK vehicle trajectory are extracted randomly.

The Pearson coefficient heat map is shown in Fig. 15. It can be found that the Pearson coefficient between v_x and v_y, a_y and v_y, θ and Δθ, which are greater than 0.2 in both types of sequences. The indicators extracted from the LC sequences exhibit a stronger correlation than those extracted from LK sequences. No significant correlation was found between lateral acceleration (a_y) and other indicators. Furthermore, the main discrepancies are observed between the heading-related indicators (θ and Δθ) and the velocity-related indicators (v_x, v_y, and a_y). For lane-changing processes, strong correlations were observed between v_x and θ (0.25), v_y and Δθ (0.56), a_y and θ (0. 92), and v_y and θ (0.93), indicating a close relationship between these variables. In contrast, no significant relationship was found in LK processes between heading-related indicators (θ and Δθ) and velocity-related indicators (v_x, v_y, and a_y). The result could be explained by the fact that drivers have to adjust their driving direction and velocity during a lane change to achieve the desired purpose. However, during the lane-keeping phase without a specific task, the changes in heading and speed are random and separate from each other.

5.2.2. Results of LC status prediction models

With a focus on the purpose of the study, only sequences labeled as RLC and LLC are utilized in this section. Eighty percent of the extracted samples are used to train the model, while the remaining samples are used for validating the performance. The input sequence length is set to 150 frames(5 s). With a 1 s interval (indicators take an average of 60 frames), lane-change vehicle status in the next 2 s is predicted. Several experiments are performed to obtain optimal parameter settings. The batch size is set to 64, and training epochs are set to 30. The loss function is mean_squared_error, and the optimizer is Adam. For TCN and TCN-ATM layer, the number of filters is specified as 64. For the LSTM layer, the number of neurons in the hidden layer is set to 64, and the depth of LSTM is set to 2.

The output indicators consist of six variables: v_x, v_y, a_x, a_y, θ, Δθ. Among these variables, v_x, v_y, a_y, θ, *and* Δθ are related output variables, and a_x is not correlated with other variables. In practice, three models, including LSTM, TCN, and TCN-ATM, could be utilized separately to address such sequence-to-sequence prediction issues. Given the expected output variables are simultaneously influenced by the same surrounding environment, three multi-task models (MTL-LSTM, MTL-TCN, MTL-TCN-ATM) are trained to improve prediction accuracy in this study. Three single-task learning models are used for comparison. The prediction results are listed in Table 4.

From a single-task prediction perspective, the TCN-ATM model demonstrates superior performance compared to LSTM across various metrics (v_x, v_y, a_x, Δθ, θ). Additionally, it exhibits a significant reduction in Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) when compared to the TCN model. Notably, the TCN-ATM model excels in predicting the longitudinal acceleration a_x, with an RMSE value of 1.235 ft/s^2. These results highlight the effectiveness of incorporating the attention mechanism into the TCN model, thereby enhancing its performance in single-task learning. Consequently, the TCN-ATM model can be considered a practical and reliable option for single-index forecasting tasks.

From a multi-task prediction perspective, the MTL-LSTM model outperforms the MTL-TCN and MTL-TCN-ATM models for indicators v_x, v_y, a_y, and θ. The MTL-TCN-ATM model demonstrates optimal prediction results for the Δθ indicator while performing poorly for other indicators. With an RMSE value of 2.510 degree/s and an MAE value of 2.042 degree/s, the MTL-LSTM model shows considerable space for improvement in terms of the Δθ indicator. To compare the performance of single-task learning models with multi-task learning models, the improvement ratio is defined as follows:

$$p_i = 1 - \frac{m_i}{s_i} \tag{22}$$

Where m_i represents the evaluation index (RMSE, MAE) value of *i-th* task using multi-task model, s_i represents the evaluation index (RMSE, MAE) value of *i-th* task using the corresponding single-task models, p_i is evaluation index improvement ration of task i using MTL model comparing to single-task model. A positive value of p_i indicates that the MTL model outperforms the corresponding single-task model in predicting task i, while a negative value indicates the opposite. Table 5 presents the result of the improvement in prediction performance.

As is evident in Table 5, the proposed MTL-LSTM, and MTL-TCN over five indicators provide markedly increased performance compared to the corresponding single-task model. Specifically, the MTL-LSTM model demonstrates an average decrease of 26.04% in MAE and 25.19% in RMSE, while the MTL-TCN model exhibits an average reduction of 25.2% in MAE and 24.49% in RMSE. The performance improvement resulting from considering the relationship between output variables may be critical to accurately predicting driving status. However, the performance of the MTL-TCN-ATM model is much lower than that of the TCN-ATM model. The decrease in performance resulting from introducing attention mechanisms in multi-task learning can be attributed to issues such as task competition and conflicts, optimization challenges, and feature conflicts [69]. Different tasks may require attention to different features or information. When incorporating attention mechanisms, it becomes necessary to address feature conflicts among tasks to ensure that the attention mechanism can properly focus on and capture the relevant features for each task. In addition, when attention becomes excessively focused on one task, the important features of other tasks may be neglected, resulting in performance degradation.

Table 4
Model result comparison.

Model	Metrics	Task					
		v_x	v_y	a_x	a_y	$\Delta\theta$	θ
LSTM	MAE	2.817	0.572	1.256	0.692	2.375	1.420
	RMSE	3.926	0.704	1.662	0.937	3.049	1.845
MTL-LSTM	MAE	**1.288**	**0.502**	–	**0.632**	2.042	**0.838**
	RMSE	**1.712**	**0.684**	–	**0.866**	2.510	**1.080**
TCN	MAE	2.977	0.596	1.064	0.669	8.159	1.894
	RMSE	4.134	0.793	1.402	0.922	10.60	2.408
MTL-TCN	MAE	1.982	0.547	–	0.648	2.945	1.566
	RMSE	2.534	0.751	–	0.916	3.875	2.062
TCN-ATM	MAE	1.749	0.561	**0.975**	0.875	**0.560**	1.002
	RMSE	2.080	0.693	**1.235**	1.188	**0.799**	1.293
MTL-TCN-ATM	MAE	17.18	1.751	–	0.869	0.601	1.775
	RMSE	19.88	2.081	–	1.183	0.858	1.464

Table 5
Performance improvement rate of prediction (%).

Model	Index	v_x	v_y	a_y	$\Delta\theta$	θ
MTL-LSTM vs LSTM	MAE	54.28	12.24	8.67	14.02	40.99
	RSME	56.39	2.84	7.58	17.68	41.46
MTL-TCN vs TCN	MAE	33.42	8.22	3.14	63.90	17.32
	RSME	38.70	5.30	0.65	63.44	14.37
MTL-TCN-ATM vs TCN-ATM	MAE	-882.2	-212.12	0.69	-7.32	-77.15
	RSME	-855.77	-200.29	0.42	-7.38	-13.23

6. Conclusion

LC behavior is a fundamental driving operation that largely affects traffic efficiency and safety. Accurately detecting and predicting LC processes of the surrounding vehicles can help autonomous vehicles better understand their surrounding environment, recognize potential safety hazards, and improve traffic safety. In this paper, the LC vehicle status was characterized using six variables, including the longitudinal velocity (v_x), lateral velocity (v_y), longitudinal acceleration (a_x), lateral acceleration (a_y), vehicle heading (θ), and yawRate ($\Delta\theta$). Using vehicle trajectory data, this paper developed a unified modeling framework for lane-change intention recognition (LC-IR) and lane-change status prediction (LC-SP). To accurately identify LC intention, a novel TCN-ATM model was first utilized in this research. Considering the intrinsic relationship between outcome factors, three MTL models (MTL-LSTM, MTL-TCN, and MTL-TCN-ATM) were constructed to predict LC vehicle status. A total number of 1023 vehicle trajectories was first extracted from the CitySim dataset to validate the reliability of the proposed model. The Pearson coefficient was conducted to investigate the relationship between the output variables. Both training time and classification accuracy were utilized as metrics to evaluate the performance of the model.

For the LC intention recognition issues, this study conducted a comprehensive comparison of SVM, RF, ET, LSTM, GRU, CNN, TCN, TCN-LSTM, and TCN-ATM models. The Ten-fold cross-validated method was employed to ensure robustness in model training and evaluation. With an input length of 150 frames, the proposed TCN-ATM model achieves an impressive overall classification performance of 98.20%. Compared to the other models, the results demonstrate that the TCN-ATM model reduces the impact of Type I and Type III errors, demonstrating a higher accuracy. For the LC driving status prediction issue, six metrics are extracted from the vehicle trajectory to characterize the driving status in this paper. The Pearson coefficient was employed to investigate the relationship between six output indicators. The result indicated a close relationship between the heading-related indicators (θ and $\Delta\theta$) and the velocity-related indicators (v_x, v_y, and a_y). To capture the intrinsic relationship of output indicators, this research developed three multi-task models: MTL-LSTM, MTL-TCN, and MTL-TCN-ATM. The results showed that the proposed TCN-ATM models could be considered a practical and reliable option for single-index forecasting tasks. The MTL-LSTM model outperforms the MTL-TCN and MTL-TCN-ATM models for indicators v_x, v_y, a_y, and θ. With an average reduction of 26.04% and 25.19% in the MAE and RMSE, respectively. The proposed MTL-LSTM over five indicators provides markedly increased performance compared to the corresponding single-task model.

The research shows that the novel TCN-ATM model outperforms LSTM, SVM, and TCN models in lane change intention recognition. Considering the correlation of related indicators could improve the prediction accuracy and training efficiency of the model. According to the obtained index v_x, v_y, a_y, a_x, θ, and $\Delta\theta$, the real-time traffic conflict index can be calculated [11]. According to the index a_y, a_x, θ, *and* $\Delta\theta$, it can be determined whether the driver has taken the avoidance behavior. The developed model holds great potential in enhancing autonomous vehicles' perception and prediction capabilities and improving vehicle control strategies. This study also has some study limitations. In the multi-task learning model, we use the same weights for the loss function of each task. To eliminate the effect of magnitude on the prediction results, all input and output vectors are normalized. In the future, the prediction accuracy can be further improved by using the adaptive loss function. For instance, if there is a main task in all the tasks, increasing the loss weight of the main task could improve the model performance.