Electronic Research Archive

ERA, 32(2): 979–1002.
DOI: 10.3934/era.2024048
Received: 19 October 2023
Revised: 06 December 2023
Accepted: 04 January 2024
Published: 19 January 2024

http://www.aimspress.com/journal/ERA

Research article

Traffic flow prediction with a multi-dimensional feature input: A new method based on attention mechanisms

Shaohu Zhang, Jianxiao Ma*, Boshuo Geng and Hanbin Wang

College of Automobile and Traffic Engineering, Nanjing Forestry University, Nanjing 210037, China

* **Correspondence**: Email: majx@njfu.edu.cn.

Abstract: Accurately predicting traffic flow is an essential component of intelligent transportation systems. The advancements in traffic data collection technology have broadened the range of features that affect and represent traffic flow variations. However, solely inputting gathered features into the model without analysis might overlook valuable information, hindering the improvement of predictive performance. Furthermore, intricate dynamic relationships among various feature inputs could constrain the model's potential for further enhancement in predictive accuracy. Consequently, extracting pertinent features from datasets and modeling their mutual influence is critical in attaining heightened precision in traffic flow predictions. First, we perform effective feature extraction by considering the temporal dimension and inherent operating rules of traffic flow, culminating in Multivariate Time Series (MTS) data used as input for the model. Then, an attention mechanism is proposed based on the MTS input data. This mechanism assists the model in selecting pertinent time series for multivariate forecasting, mitigating inter-feature influence, and achieving accurate predictions through the concentration on crucial information. Finally, empirical findings from real highway datasets illustrate the enhancement of predictive accuracy attributed to the proposed features within the model. In contrast to conventional machine learning or attention-based deep learning models, the proposed attention mechanism in this study demonstrates superior accuracy and stability in MTS-based traffic flow prediction tasks.

Keywords: intelligent transportation system; traffic flow modeling; time series analysis; deep learning; attention mechanism

1. Introduction

In recent years, the rapid development and application of traffic sensor technology have enabled the acquisition of large-scale traffic data of different types, ushering us into the era of big traffic data [1]. Based on this, Intelligent Transportation Systems (ITS) effectively control and plan urban traffic by collecting, processing, and utilizing large-scale traffic big data [2]. Traffic flow prediction, as a crucial component of ITS, plays an essential role in reducing traffic congestion, improving air quality and other factors.

There are many research findings on short-term traffic flow prediction, primarily categorized into parameter-based methods, traditional machine learning models, and deep learning models. Parameter-based methods and traditional machine learning models cannot effectively process and analyze large-scale data, limiting their prediction accuracy. In contrast, deep learning models have shown promising results in processing large-scale traffic data due to their powerful feature expression and modeling capabilities for complex problems. However, deep learning models have limitations that need to be addressed. On the one hand, traffic data is time-series data. while deep learning models can delve deeper into the temporal patterns of data, it is not rigorous to only process traffic data as time-series data. Traffic flow changes are affected by external factors such as holidays, traffic accidents, and more. Characterizing internal factors that affect traffic flow also helps the model learn the inherent patterns of traffic flow changes and further improve prediction accuracy. Therefore, it is necessary to identify related features and explore the deep relationships between features for improved prediction accuracy.

On the other hand, with the development of traffic sensor technology, collected data is usually in the form of multivariate time-series data (MTS), Where complex dynamic correlations between different variables may exist , proving crucial for traffic flow prediction but difficult to capture and analyze. For MTS prediction tasks, adequate consideration of multidimensional features can enhance the network's prediction ability. However, in current traffic flow prediction tasks, there is a lack of consideration for the multidimensional feature characteristics of MTS data.

In this article, we propose a solution to the traffic flow prediction problem, addressing the aforementioned limitations and requirements. The major contributions of this article are:

(I) Time features were generated by visualizing the data to analyze the impact of various external factors across different time dimensions on traffic flow. Additionally, traffic flow operation status features were created by combining traffic flow models to explore the inherent characteristics of traffic flow changes.

(II) Using different state variables composed of various features as input variables to investigate the interactions between features and their impact on traffic flow prediction.

(III) The proposal of an attention mechanism that weights relevant feature vectors instead of relevant time steps. The model's applicability was evaluated on four benchmark datasets and compared with several state-of-the-art models. The results demonstrated that the proposed model outperformed all four datasets regarding prediction accuracy.

The article is organized as follows: Section 2 provides an overview of the relevant research progress on traffic flow prediction. In Section 3, presents the dataset properties and explores the dataset features from two perspectives: The time dimension and the internal operation regularity of traffic flow. Sections 4 and 5 propose a method to use the attention mechanism based on the degree of influence of prediction results to weight features, while Section 6 discusses the experimental results. Finally, we conclude this article in Section 7.

2. Related work

Traffic flow prediction is a research field that dates back to the 1970s and plays a crucial role in Intelligent Transportation Systems (ITS). Early studies on traffic flow prediction encountered limitations in datasets and models, resulting in relatively singular input features. Bor et al. successfully forecasted traffic flow within high-speed networks by integrating traffic flow data with the fuzzy autoregressive (fuzzy-AR) model [3]. Williams et al. employed seasonal autoregressive integrated moving average processes to model univariate traffic condition data streams for predicting traffic flow [4]. Luis et al. predicted short-term passenger demand distribution within taxis through the utilization of taxi-installed sensor data [5].

With the development of traffic sensor technology, an increasing number of features are being utilized in traffic prediction. For example, Wu et al. combined multiple external data sources to analyze and predict taxi demand from various dimensions, including points of interest, weather, and vehicle collisions [6]. Trinh et al. combined traffic flow data, local weather, and traffic location connectivity graph data using a multivariate model to generate traffic flow predictions [7]. Similarly, Geng et al. forecast lane-changing behaviors by integrating road traffic conditions and vehicle motion parameters [8]. Aljuaydi et al. utilized an input dataset with five features (the flow rate, the speed, the density, road incidents, and rainfall) to predict traffic flow during road crashes and rainy conditions [9]. In addition, feature mining through data analysis and relevant prediction demands has also been a research focus. For instance, Chen et al. applied feature engineering to mine a large amount of traffic flow data [10]. An et al. utilized a fuzzy reasoning mechanism to generate traffic accident information from actual traffic data, which improved the prediction performance of the model [11]. Li, Wang, et al. employed time series decomposition to extract stable sub-sequences used for model prediction [12,13]. Zhang et al. developed a Peak Zoom network to capture the peak effect and enhance the prediction performance for crucial time steps in traffic flow prediction [14]. Mendez et al. employed CNN to extract valuable hidden features from input data for long-term traffic flow prediction tasks [15].

Despite the potential benefits of incorporating more features in deep learning models, the collinearity between different features and their dynamic interference may lead to worse prediction results. To address this issue, Du et al. used factor analysis to calculate the variance contribution of various features to the prediction results and selected features for traffic flow prediction [16]. Wang et al. introduced the MC-STGCN model for assessing the correlation among distinct features within spatiotemporal graphs [17]. Li et al. introduced a hybrid framework named HyDCNN based on position-aware Dilated CNN. This framework effectively mitigates the influence of linear dependencies among temporal data [18]. Zhao et al. proposed the Multivariate Constraint Sample Entropy (McSE) as a means to integrate multivariate constraint relationships, enhancing predictive accuracy [19]. Compared to other feature selection models, the attention mechanism has been widely used in deep learning models because of its simple structure and automatic weighting of different features [20–23]. For example, Fang et al. introduced the attention mechanism into LSTM, weighting features in the time dimension and achieving better prediction results [24]. Wan et al. developed a multivariate temporal convolutional attention network (MTCAN) by employing 1D dilated convolution to enhance the extraction of input features [25]. Shih et al. proposed a temporal pattern attention (TPA) mechanism, which utilized one-dimensional convolution to extract deep feature information from LSTM hidden layer states and applied an attention mechanism for feature weighting to complete MTS prediction tasks [26]. Geng et al. enhanced the representation of temporal correlations among external features by incorporating Feature-level attention into the graph attention mechanism module [27]. Cao et al. developed a Spectral Temporal Graph Neural Network (StemGNN) that utilizes attention mechanisms to establish dynamic correlations between features [28]. However, attention mechanisms often calculate the features processed by deep learning models, which may filter out high-value information during the deep learning model processing and fail to focus on the trends and dynamic interference between different features.

In summary, how to introduce more useful features that contribute to traffic flow prediction remains a topic worth discussing. For traffic flow prediction tasks that involve multi-dimensional feature input, it is also worth exploring how to reduce the mutual influence between different input features to improve prediction accuracy without significantly increasing model training costs.

3. Data analysis and feature processing

In this section, we describe the data preprocessing and feature extraction work. The section is divided into three parts: First, we analyzed and preprocessed the dataset to obtain the required data for feature creation, and filtered out raw features. Second, we visualized the traffic flow data to mine its time characteristics. Last, we analyzed the inherent variation characteristics of traffic flow, in combination with the traffic flow model, to obtain the traffic flow operating state features for subsequent traffic flow prediction.

3.1. Data preparation

The experimental dataset used in this study is the PEMS public dataset provided by the California Department of Transportation, which spans over a decade and contains various information affecting traffic volume. Four adjacent detection stations located on the I5 and I710 interstate highways were selected as the research objects. The data was collected at a time interval of 5 minutes, from January 1, 2019, to June 30, 2019, and the corresponding monitoring point locations are depicted in Figure 1. The dataset contains a small proportion of missing data, which was filled using the median method.

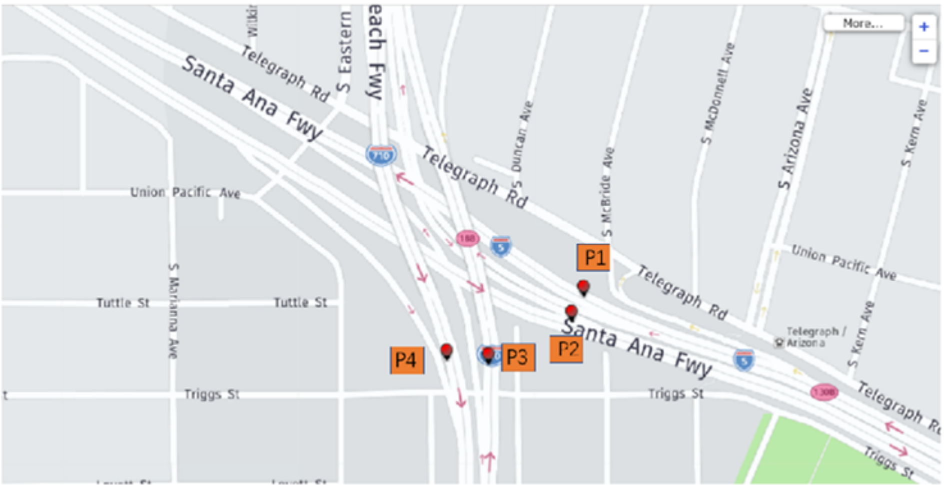


Figure 1. Corresponding location of detection station.

During the data collection process, four major aspects are considered: Time-related data, spatial location data, base station data, and traffic flow data. Time-related data includes information related to traffic flow's time, such as collection date and time. Spatial location data includes location information of stations, such as station location area and road grade. Base station data refers to the machine information of the base stations, such as the station number and length. Traffic flow data describes the traffic flow state, including traffic volume and occupancy rate. Table 1 provides a detailed description of the dataset, which is also divided into four parts for ease of introduction

Table 1. Dataset details.

Type	Attribute	Meanings
Time-related data	Year	Year of collection
	Month	Month of collection
	Day	Day of collection
	Time stamp	The time of the beginning of the summary interval. For example, a time of 08:00:00 indicates that the aggregate (s) contain measurements collected between 08:00:00 and 08:04:59.
Special location data	District	City area where the base station is located
	Lane Type	A string indicating the type of lane
Detection station data	Node	Unique station identifier
	Direction of Travel	N | S | E | W
	Station Length	Segment length covered by the station
Traffic flow data	Occupancy	Average occupancy over the 5-minute period expressed as a decimal number between 0 and 1
	Speed	Flow-weighted average speed over the 5-minute period
	Total Flow	Sum of flows over the 5-minute period

In the original dataset, valuable feature inputs are necessary to demonstrate the performance of non-parametric models more effectively. Spatial data represents the urban area where the base station is located and is used as quantitative data, but at the current research stage, it has limited impact. The base station data is primarily used for network node identification, and since the prediction target is based on a single base station, it is not processed in this paper. We consider constructing new features from two perspectives based on the existing features. First, time features are constructed based on the distribution characteristics of data in the time dimension. Second, to explore the inherent characteristics of traffic flow changes, traffic flow operation status is constructed by combining traffic flow models.

3.2. Time feature analysis

During the analysis of time features, the traffic flow data of the P1 station for two consecutive weeks was used as an example, as shown in Figure 2. Based on the daily trend analysis, there are local differences in traffic flow, but overall, it is associated with the time of day. Specifically, during the early morning period, traffic flow tends to decrease to a lower value, whereas it remains higher during the day.

From the analysis of the weekly trend, it is observed that traffic flow peaks in the morning and evening during weekdays, whereas this trend is not evident on weekends. To quantify the impact of holidays on traffic flow, we employ a normalization method. Assuming that the longest continuous holiday in a year lasts p days, the two-day weekend holiday can be quantified as $2/p$, and other holidays can be quantified as p_i/p, where p_i represents the length of the holiday. The specific expression is shown in Eq (1).

$$holiday = \begin{cases} 0, workday \\ 2/p, weekend \\ p_i/p, other\ holidays \end{cases} \tag{1}$$

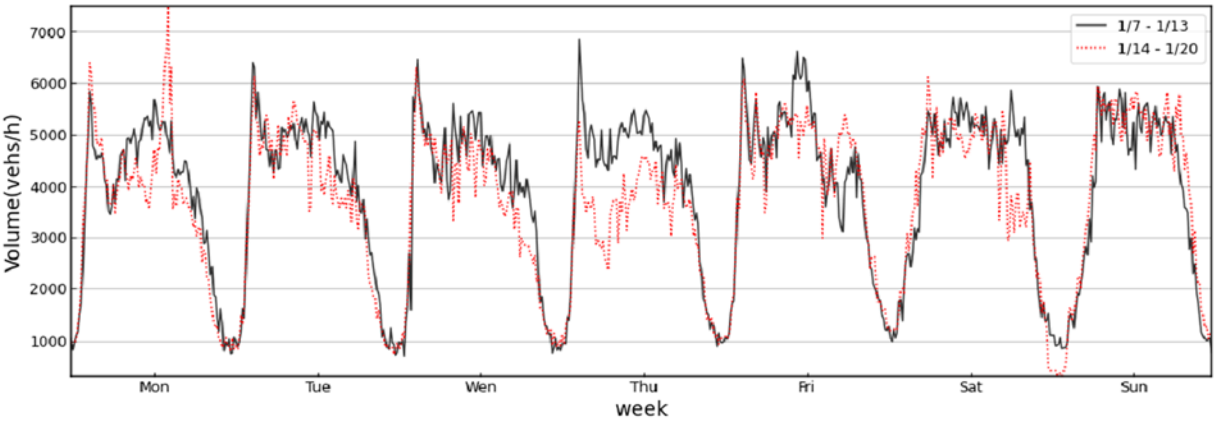


Figure 2. Two weeks traffic flow changes at the detection station.

3.3. Traffic flow operation state analysis

Traffic flow models aim to quantify the relationship between flow, speed, and traffic density to explain the basic rules and properties of actual traffic operations. To explore the inherent variability of traffic flow, selecting a suitable model is essential. We employ a traffic flow model presented in Table 2 to ensure the model's applicability.

Table 2. Traffic flow models.

Single-regime models	Functional form
Greenshields	$v = v_f \cdot (1 - k/k_{jam})$
Greenberg	$v = v_c \cdot ln(k_{jam}/k)$
Newell	$v = v_f \left[1 - exp\left(-\frac{\lambda}{v_f} \cdot \left(\frac{1}{k} - \frac{1}{k_{jam}}\right)\right)\right]$
5LP model	$v = v_b + \frac{v_f - v_b}{\{1+exp[(k-k_t)/\theta_1]\}^{\theta_2}}$
S3 model	$v = \frac{v_f}{\left[1+\left(\frac{k}{k_c}\right)^m\right]^{\frac{2}{m}}}, 1 \le m \le 8.53$

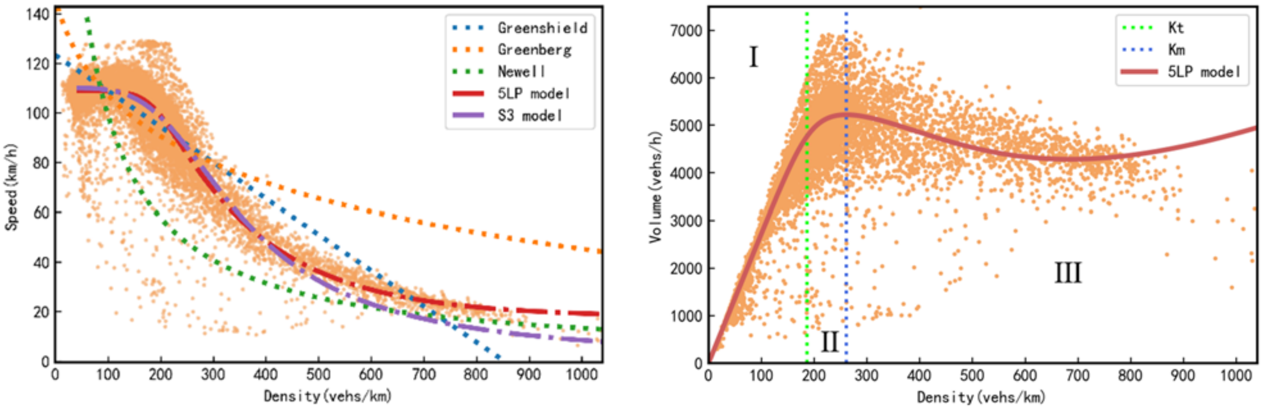
Note: Parameter explanation: v_f:free flow speed; k_{jam}:jam density; k_c: critical density; v_c: speed at critical density; k_t:transition density from free flow to synchronized flow; $v_b, \theta_1, \theta_2, m$: relevant coefficients in different models

In this paper, we utilize the least squares method to fit the parameters of various models. This mathematical optimization approach calculates the model parameters that best fit the data by minimizing the sum of the squared errors. As the dataset's raw data does not contain traffic density k, we use the classical Equation (2) to derive it. Figure 3(a) demonstrates the fitting results of the traffic flow model for P1. Table 3 lists the fitting results for various detection stations, where we define the evaluation index using Eq (3).

$$q = kv \tag{2}$$

$$MSE_v = \frac{1}{n}\sum_{i=1}^{n}(\hat{v}_i - v_i)^2 \tag{3}$$

here, n represents the total number of samples that need to be fitted. $\hat{v}_i$ represents the speed that the model fitted for the i-th sample, and v_i represents the corresponding true speed for the i-th sample.



(a) density-speed distribution

(b) density-volume distribution

Figure 3. Distribution of traffic, density and speed of P1 detection station.

Table 3. Fitting results of traffic flow models.

Traffic flow models	P1	P2	P3	P4
Greenshields	117.82	83.60	103.00	109.68
Greenberg	243.21	117.85	327.96	308.19
Newell	1652.70	2222.75	1277.96	1366.48
5LP model	78.15	69.04	34.67	40.28
S3	74.47	68.48	32.74	39.39

As presented in Table 3 and Figure 3(a), the S3 and 5LP models exhibit lower fitting errors than other models, and they are more closely aligned with the actual density-speed distribution at lower densities. While the S3 and 5LP models have comparable fitting errors overall, the discrepancy lies in their high-density interval fitting. To determine the optimal model, we conducted a statistical analysis of the fitting deviations of the models in various density ranges, as depicted in Figure 4. The fitting deviation is defined as Eq (4).

$$Err = \frac{1}{n}\sum\left|\frac{v_i - \widehat{v}_i}{v_i}\right| \tag{4}$$

here, n represents the total number of samples in the density range, $\hat{v}_i$ is the model-fitted speed in the density range, and v_i is the true speed in the density range.

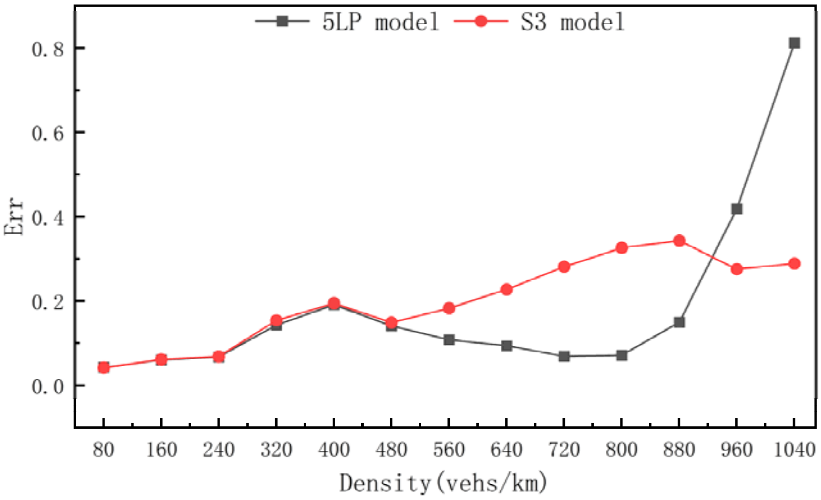


Figure 4. Distribution of fitting deviation for different density.

Based on Figure 4, the S3 model and 5LP models show similar fitting deviations for density intervals less than 480 $vehs/km$, while the S3 model exhibits more stable fitting deviations for density intervals above 480 $vehs/km$. When examining Figure 3(a),(b), it becomes evident that the sample size is extremely small when the traffic density exceeds 900 $vehs/km$. In addition, the 5LP model exhibits significantly lower fitting deviations than the S3 model for traffic densities ranging from 480 to 900 $vehs/km$. Therefore, we select the 5LP model for data set fitting considering the actual scenario.

Figure 3(b) shows that as the density increases, the flow monotonically increases while the dispersion between data points remains small until a certain density level is reached. At this point, the dispersion increases significantly, indicating a transition from a stable to an unstable traffic flow state. Thus, traffic density can be used to some extent to characterize changes in traffic flow states.

Critical nodes that exist during the traffic flow process can be obtained using the flow-density curve of the traffic flow model. The parameter k_t in the 5LP model is theoretically proven to be the density at which the transition from free flow to synchronized flow occurs [29]. The corresponding critical point of the maximum flow rate can be found from the curve, which is the inflection point of the traffic capacity. The density at this point is k_m, an important indicator for evaluating whether the traffic flow has entered a congested state. Therefore, we classify traffic flow into three operating states: Free flow, synchronized flow, and congested flow based on traffic flow density, which corresponds to regions I–III in Figure 3(b). Table 4 presents the transition density and critical density corresponding to different datasets.

Table 4. Density nodes corresponding to different datasets.

Density	P1	P2	P3	P4
k_t	186.88	179.92	174.00	190.24
k_m	261.12	283.04	279.96	236.40

As mentioned above, the input variables compiled in this paper are listed in Table 5:

Table 5. Model input variables.

Feature	Meaning	Range
Total Flow	Total flow of incoming and outgoing traffic of one specific station	q
Speed	Flow-weighted average speed of incoming and outgoing traffic of one specific station	v
Occupancy	Average occupancy of incoming and outgoing traffic of one specific station	[0,1]
Time section	Time section of one day	{0,1,2…95} (Take 15 min as a section)
Holiday	Mark the data if it is collected on holiday	Equation (1)
Traffic flow operation status	Classification according to traffic flow status	{0,1,2}

4. Preliminaries

This section will briefly introduce two important related models: the LSTM model and the attention mechanism. Through the reasonable use of both models, the TPA-LSTM+ model will be used to identify the degree of influence of different features on traffic flow and in traffic flow prediction tasks.

4.1. LSTM model

The LSTM model, a RNN model, incorporates a "cell state" C_t and utilizes three gate structures (input gate, output gate, and forget gate) to accumulate information. This design endows the LSTM model with the ability to capture long-term dependencies [30–33]. The detailed model structure is presented in Figure 5.

Assuming that there is a set of time series information $\{x_1, x_2, \dots, x_t\}$, where $x_i \in \mathbb{R}^n$, the function of the LSTM model is defined as shown in Eq (5). $h_t \in \mathbb{R}^m$ is used to represent the hidden state at time t.

$$h_t = LSTM(h_{t-1}, c_{t-1}, x_t) \tag{5}$$

The specific definition of the LSTM structure is shown below:
Input Gate:

$$i_t = sigmoid(W_{xi}x_t + W_{hi}h_{t-1}) \tag{6}$$

Forget Gate:

$$f_t = sigmoid(W_{xf}x_t + W_{hf}h_{t-1}) \tag{7}$$

Cell State:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xg}x_t + W_{hg}h_{t-1}) \tag{8}$$

Output Gate:

$$o_t = sigmoid(W_{xo}x_t + W_{ho}h_{t-1}) \tag{9}$$

Output hidden state and prediction results:

$$h_t = o_t \odot \tanh(c_t) \tag{10}$$

$$y_t = Linear(h_t) = W_{hl}h_t \tag{11}$$

where i_t, f_t, c_t, $o_t \in \mathbb{R}^m$, $y_t \in \mathbb{R}^w$, the learnable weight matrices W_{xi}, W_{xf}, W_{xo}, $W_{xg} \in \mathbb{R}^{m\times n}$, W_{hi}, W_{hf}, W_{ho}, $W_{hg} \in \mathbb{R}^{m\times m}$, $W_{hl} \in \mathbb{R}^{w\times m}$, $\odot$ represents element-wise multiplication. Since sigmoid and tanh are two different activation functions, the value ranges of i_t, f_t and $o_t$are (0,1), and the value range of $\tanh(c_t)$ is (-1, 1).

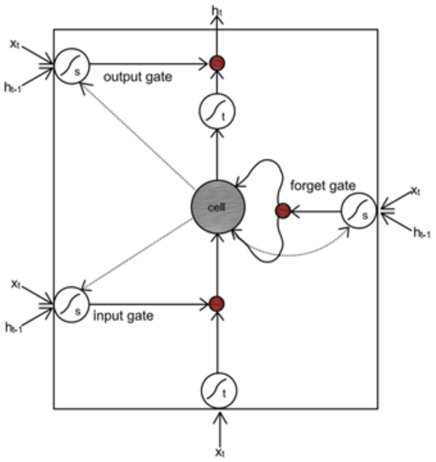


Figure 5. Basic Architecture of LSTM Model.

In summary, the LSTM model incorporates an input gate, a forget gate, and an output gate to regulate the degree of information accumulation and update the "cell sta" c_t. The updated "cell state"

c_t is combined with the output gate to produce the corresponding hidden state h_t at the next time step. The predicted valuey_t is obtained by a fully connected layer based on the output hidden state h_t. Compared to the RNN model, the LSTM model's gate structure and "cell state" can preserve long-term information and prevent issues like gradient explosion or vanishing during the gradient descent algorithm [30].

4.2. Attention mechanism

The attention mechanism was initially used in machine translation [34] to identify critical information by assigning weights to different parts of the input sequence, facilitating the model in making more precise predictions. In the LSTM model, the output hidden state is denoted as $H = (h_1, h_2, \ldots, h_{t-1})$, where $f(h_i, h_t) = h_i^T h_t$ represents the scoring function of the attention mechanism. The weighted sum of each hidden state h_i in H, denoted as v_t, is then used to represent the correlation between the previous state information and the current state h_t. Finally, the combination of v_t and h_t is utilized to predict the next time step.

$$\alpha_i = softmax(h_i, h_t) = \frac{\exp(f(h_i, h_t))}{\sum_{j=1}^{t-1} \exp(f(h_j, h_t))} \tag{12}$$

$$v_t = \sum_{i=1}^{t-1} \alpha_i h_i \tag{13}$$

5. The TPA-LSTM+ model

Previous research has focused on incorporating attention mechanisms into various model structures to enhance model performance for diverse tasks. In the LSTM model, the attention mechanism extends the scope of information consideration by applying a weighted sum to the hidden state H [35]. However, in MTS prediction, this attention mechanism is incapable of filtering out noisy features that could have an adverse impact on the prediction, and averaging over multiple output states makes it challenging to choose effective features for accurate prediction results.

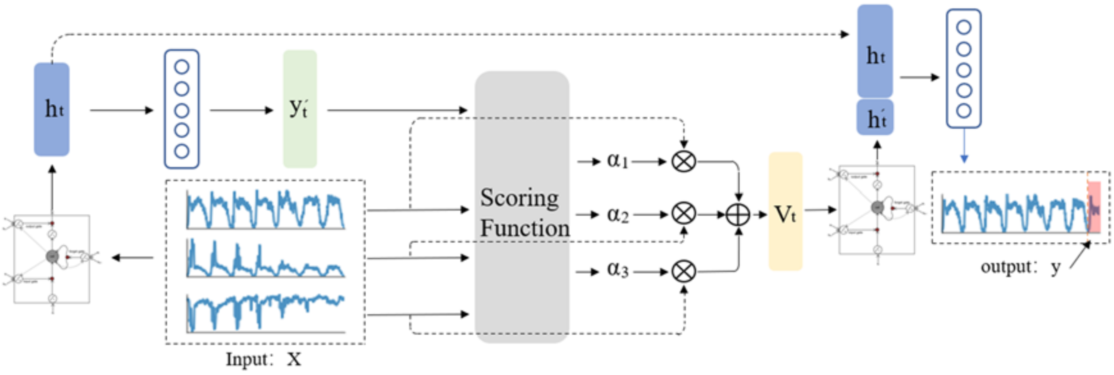


Figure 6. TPA-LSTM+ model architecture diagram.

The model structure proposed in this paper is illustrated in Figure 6, where the model employs an attention mechanism on the row vectors of the input data. The calculation of attention weights enables the model to select the variables that contribute to the prediction and obtain the context vector v_t, representing the weighted sum of the input row vectors.

5.1. problem formulation

In the task of predicting traffic flow in MTS, $X = \{x_{t-w}, x_{t-w+1}, \ldots, x_{t-1}\}$ represents the input data of traffic flow, where $x_i \in \mathbb{R}^n$ indicates n features acquired at time i, and w signifies the length of the time window. The corresponding predicted value, denoted as $\hat{y}_{t-1+\Delta}$, is compared to the ground-truth value, denoted as $y_{t-1+\Delta}$, where Δ represents a fixed horizon. Assuming the learning function is represented as $f_\theta(\cdot)$ and the loss function as $l(\cdot)$, the MTS traffic flow prediction task can be formulated as: $f_\theta^* = arg\ \ min\ \ l(f_\theta(X), y_{t-1+\Delta})$, where f_θ^* denotes the learning function with the optimal learned parameters.

5.2. Proposed attention mechanism

The attention mechanism proposed in this study consists of three parts:

1) Computing the relevant hidden layer state h_t of the input data X using the LSTM model. Since h_t includes the accumulated information of the input data from the previous w time steps and is utilized for the prediction component of the model, the output y_t' from the fully connected layer provides a comprehensive consideration of both the input data values and the prediction results, where $y_t' \in \mathbb{R}^w$.

$$h_t = LSTM(X) \tag{14}$$

$$y_t' = Linear(h_t) \tag{15}$$

2) The scoring function is defined as $\mathbb{R}^w \times \mathbb{R}^w \rightarrow \mathbb{R}$ and is utilized to compute the relevance between the row vector x_j of the input X and y_t', thereby evaluating the influence of features on the prediction results and the overall input data. The attention mechanism weight α_i is calculated according to Eq (17), while the context vector v_t is derived from the weighted row vector x_j, where $v_t \in \mathbb{R}^w$.

$$f(x_j, y_t') = (x_j)^{\mathrm{T}} y_t' \tag{16}$$

$$\alpha_i = softmax(x_j, y_t') \tag{17}$$

$$v_t = \sum_{j=1}^{n} \alpha_i x_j \tag{18}$$

Notably, when computing attention scores, we adopt the feature vectors $x_j \in \mathbb{R}^w$ to capture the significance of features fory_t', instead of utilizing the input vectors $x_i \in \mathbb{R}^n$ at each time step.

Specifically, we consider every row of the input matrix $X \in \mathbb{R}^n \times \mathbb{R}^w$ as the input for the scoring function, rather than each column.

After computing the vector v_t, which considers the impact of different features within the input window length w on the prediction result, it is important to note that the influence of the input on the prediction result can also change over time. Therefore, employing the LSTM model to extract the pertinent temporal information from v_t we acquire $h'_t \in \mathbb{R}$. Finally, h'_t is concatenated with h_t and used to make predictions in the fully connected layer.

$$h'_t = LSTM(v_t) \tag{19}$$

$$\hat{y}_{t-1+\Delta} = Linear(concat(h_t, h'_t)) \tag{20}$$

The TPA-LSTM+ model weights the different input features through an attention mechanism, where the weight of the feature that is more relevant to the prediction result will increase accordingly. By further extracting the weighted information of v_t, the model obtains critical information in both the feature and time dimensions. This resolves the limitation of attention mechanisms that can only select and weigh in the time dimension in the past.

6. Experimental results and analysis

6.1. Evaluation metric selection

Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are standard metrics used for evaluating the performance of prediction models. While MAE and RMSE measure the overall error between predicted and actual values, MAPE assesses the overall accuracy of the model's predictions. The definitions of these metrics are shown in Eqs (21)–(23) below.

$$\text{MAE}=\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{21}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{22}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{23}$$

here, n represents the total number of samples to be predicted, $\hat{y}_i$ is the predicted value of the i-th data, and y_i is the corresponding true value of the i-th data.

6.2. Experimental settings

In this study, we focus on short-term traffic flow volume prediction with a prediction time interval of 15 minutes. The time step length of the input model determines the amount of information used for prediction. For example, if the time step length is set to 3, the model uses the traffic flow data from the previous 45 minutes before the desired prediction time point t. We investigate the impact of different time step lengths on the model prediction results and the optimal input traffic flow sequence length by adjusting the time step length from 2 to 18, i.e., from 30 minutes to 4.5 hours.

In this experiment, the dataset was divided into two parts, with the first four months as the training set and the rest as the testing set. The Adam algorithm was used as the optimization algorithm for gradient descent, with an initial learning rate of 0.001. The hyperparameters optimized included the number of hidden layers and batch size. Following the study by LV [36], the number of hidden layers in the model should not exceed 4 for short-term traffic flow prediction tasks. Thus, the range for the hidden layers was set to $\{1, 2, 3, 4\}$. Batch size represents the number of training samples used in one gradient descent algorithm optimization process, and the range was set to $\{16, 32, 64, 128\}$. To reduce the impact of random factors on the model's prediction results, the prediction results were the average of five predictions using the same model and conditions. The experiment used early stopping to terminate the training process.

We compared the prediction results of our TPA-LSTM+ model with those of seven other widely-used advanced models: autoregressive model (AR), support vector regression model (SVR), random forest regression model (RF), StemGNN, LSTM, TPA-LSTM, and AM-LSTM. Whereas StemGNN calculates the feature correlations by employing a self-attention mechanism on the input features, TPA-LSTM can select input features, and AM-LSTM selects time features through an attention mechanism.

In the autoregressive model (AR), the model order p was set to 17. In the support vector machine regression (SVR) model, the kernel function used the radial basis function (RBF), and the kernel width was set to 3×10^{-2}, with a regression level set to 8. In the random forest regression (RF) model, the mean squared error (MSE) was used to measure regression tree quality. For the TPA-LSTM model, the number of 1D CNN filters was set to 6, and other hyperparameters were set according to [34].

In terms of hardware, the experiment was conducted on an Intel i5-7300 2.5GHz CPU and an NVIDIA GTX1050 GPU. The model was run on the open-source frameworks Python 3.8.12 and PyTorch 1.11.0.

6.3. Experimental results and analysis

In this section, we first compared the effect of different feature variables on the model prediction by constructing various state variables as input variables. Next, we experimentally compared the impact of different time steps on the model prediction to select appropriate hyperparameters. Finally, we compared the proposed TPA-LSTM+ model with the latest methods to demonstrate its superiority.

6.3.1. Analysis of the impact of different feature combinations on model prediction

In this section, we constructed five different state variables, A, B, C, D, and E, based on the feature factors obtained, in order to investigate the impact of different factor combinations on the prediction results of traffic flow volume. State variable A contains only flow factors, state variable B contains flow, speed, and occupancy factors obtained from the original dataset, state variable C includes time feature factors in addition to B, state variable D adds traffic flow operation status feature factors to B, and state variable E contains all feature factors.

To ensure a fair comparison of the prediction performance, we used commonly used LSTM prediction models to analyze the effect of different feature selections on the model prediction, with P1 traffic flow volume as the predicted object. The results are presented in Table 6.

Table 6. The impact of different state variables on prediction results.

	MAE	RMSE	MAPE (%)
StateA	291.96	385.08	11.06
StateB	288.32	379.72	10.49
StateC	274.80	360.76	10.56
StateD	284.32	376.20	10.15
StateE	275.88	361.12	10.18

Table 6 presented in the preceding section demonstrates that the overall prediction accuracy of the model tends to increase with an increase in the number of input factors. Notably, the prediction error and accuracy for state variable A, which has only a single factor input, is significantly worse than that for other state variables. Although the addition of speed and occupancy rate leads to slight improvements in the model's prediction error and accuracy, these improvements are insignificant. In contrast, the inclusion of time feature factors for state variable C reduces the mean absolute error (MAE) by 13.52 compared to state variable B, indicating that this addition is helpful for reducing the overall prediction error. Furthermore, the addition of traffic flow operation status feature factors for state variable D leads to a decrease in the mean absolute percentage error (MAPE) by 0.34 compared to state variable B, suggesting that the model is closer to the actual value in predicting results. However, the inclusion of all factors for state variable E does not lead to better results, which may be due to the collinearity and dynamic interference between different feature factors, leading to a decrease in the model's prediction performance.

To further validate the effect of the added traffic flow operation state feature on model prediction, we conducted statistics on the prediction results of different state variables and obtained the prediction result ratios within the range of 20 and 10% error, as shown in Figure 7.

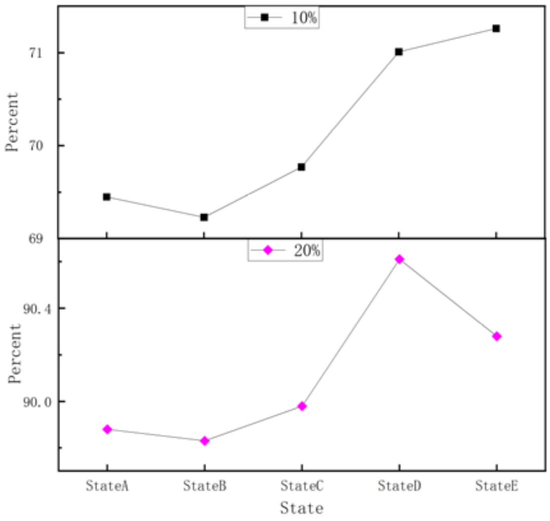


Figure 7. The influence of traffic flow operation status feature on prediction results.

Figure 7 indicates that state variables StateD and StateE, which include the traffic flow operating state feature, have a higher proportion of predicted results within the 10 and 20% error ranges compared to the state variables without this feature. This finding suggests that the added feature can improve the model's prediction accuracy.

6.3.2. Impact of different input sequence lengths on model prediction

The length of the input sequence in the LSTM model significantly impacts its prediction performance [33]. To explore how the input sequence length influences the attention mechanism and to determine the optimal performance of TPA-LSTM+, this study assessed the performance of LSTM, TPA-LSTM, and TPA-LSTM+ using nine different input sequence lengths. The results are presented in Figures 8–10, with the horizontal axis denoting the input length and the vertical axis representing the mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE), respectively. The colored lines in the figures represent the MAE, RMSE and MAPE of the three models for each input sequence length.

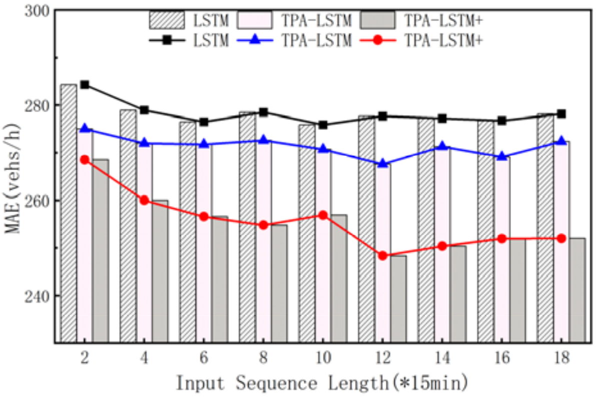


Figure 8. MAE under different time steps.

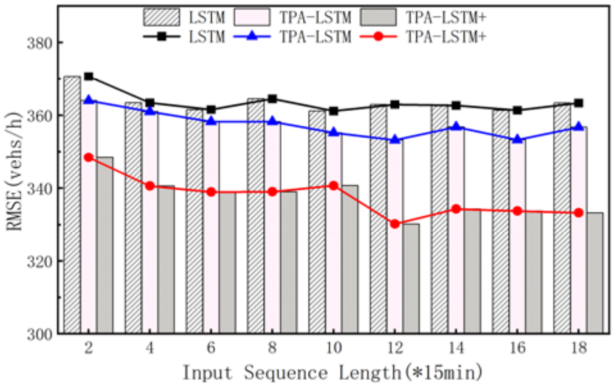


Figure9. RMAE under different time steps.

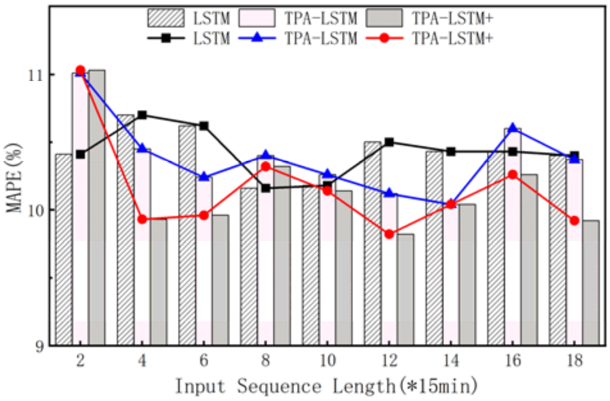


Figure 10. MAPE under different time steps.

The Figures above display that the prediction errors of the models decrease as the input time steps increase within a specific range; however, they increase when the time steps exceed this range. The impact of time steps on TPA-LSTM+ is the most significant. The TPA-LSTM+ and TPA-LSTM models have the minimum prediction errors when the time step is 12, with MAE values of 248.32 and 267.60 and RMSE values of 330.20 and 353.16, respectively, whereas the LSTM model has the lowest prediction errors when the time step is 10, with MAE and RMSE values of 275.88 and 361.12, respectively.

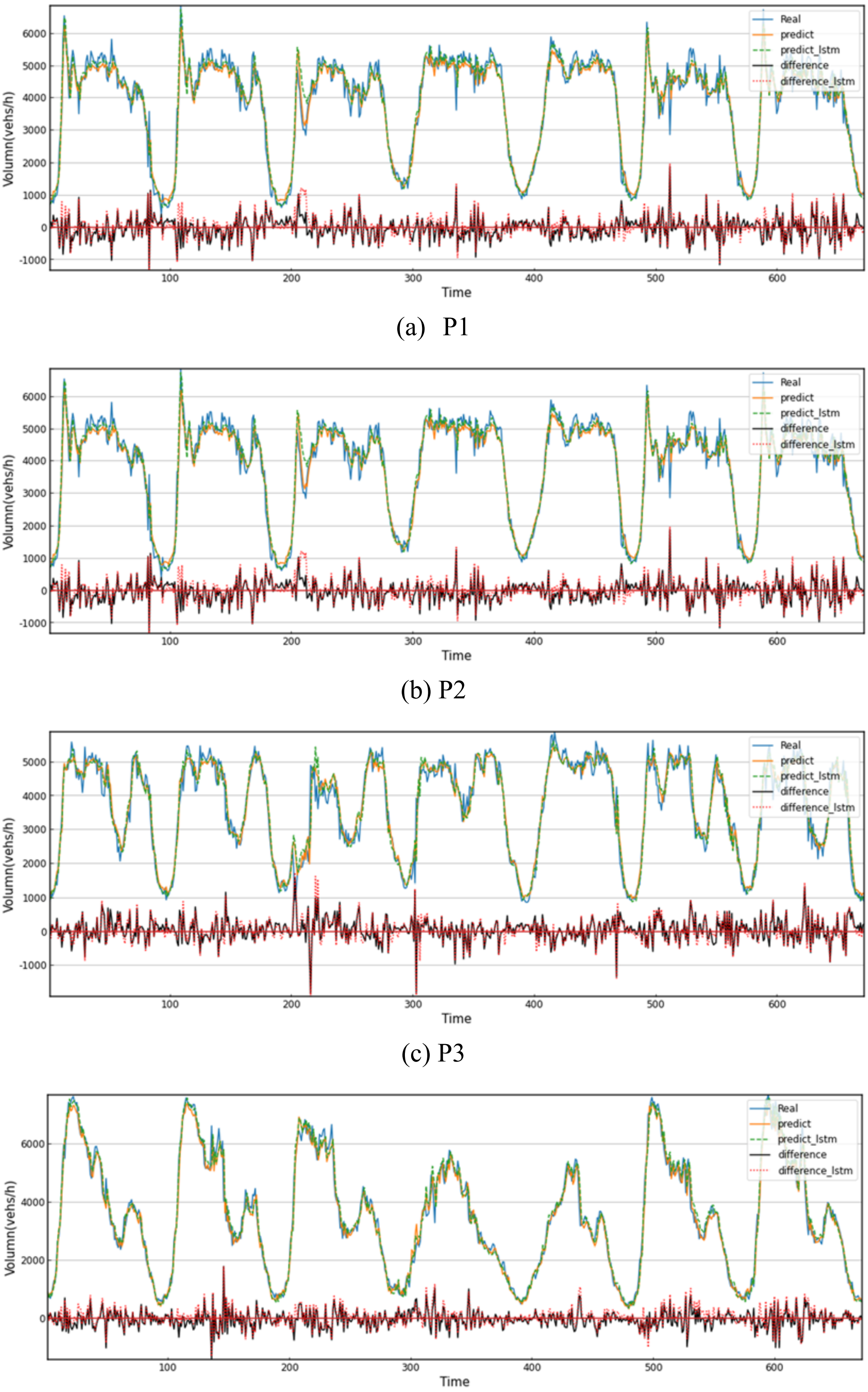
In contrast, the model prediction accuracy shows a downward trend within a specific range, even though there is some fluctuation with the increase in the time step length, as shown in Figure 10. TPA-LSTM+ achieved the lowest value of 9.82 at a time step length of 12. TPA-LSTM achieved the lowest value of 10.04 at a time step length of 14 and the second-best value of 10.12 at a time step length of 12. LSTM achieved the lowest value of 10.16 at a time step length of 8 and the second-best value of 10.18 at a time step length of 10.

By considering the results from Figures 8–10, it is apparent that TPA-LSTM+ is more affected by the time step length than other models, possibly due to the two-part input of the dataset, with the optimal input time step length being 12. As the difference in the optimal and second-best solutions in model prediction accuracy between TPA-LSTM and LSTM is not significant, the optimal input time step lengths for these models are set to 12 and 10, respectively.

6.3.3. Different model prediction results comparison

In the prediction experiment, we compared the proposed TPA-LSTM+ model with six other commonly used advanced prediction models across four datasets, namely P1, P2, P3 and P4, to validate its performance. The results of the comparison are presented in the table below.

Table 7 illustrates that TPA-LSTM+ performs better than traditional and other deep learning methods regarding overall prediction error and accuracy. Parametric methods have fixed structures and parameters, making them incapable of capturing the nonlinear relationships within traffic flows, resulting in poor performance. On the other hand, non-parametric and deep learning methods determine the parameters after training, making it challenging to modify them based on the input data's characteristics during testing. As traffic flow data is affected by various factors, achieving better performance is difficult. The embedding of the attention mechanism allows the model to capture input data changes and perform better.



(a) P1

(b) P2

(c) P3

(d) P4

Figure 11. TPA-LSTM+ model prediction results and error display.

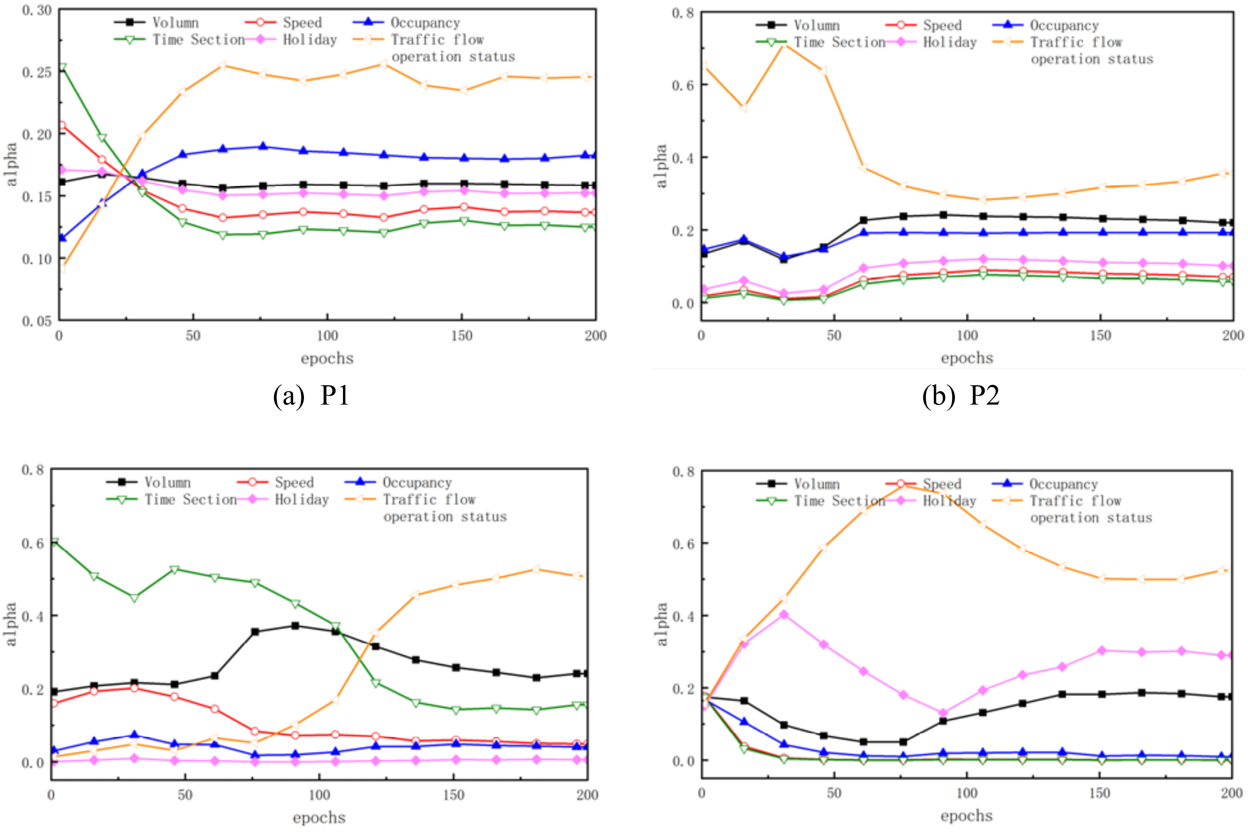
Table 7. Comparison of model prediction results.

Model	Criteria	P1	P2	P3	P4
AR	MAE	313.56	255.28	271.16	282.44
	RMSE	427.2	376.32	366.36	408.28
	MAPE	11.27	11.15	8.65	12.36
SVR	MAE	339.36	333.36	295.56	377.12
	RMSE	432.24	420.08	379.44	469.12
	MAPE	11.86	14.68	9.32	15.6
RF	MAE	283.88	245.12	284.32	297.56
	RMSE	373.4	339	363.28	370.64
	MAPE	10.39	8.86	8.83	10.91
StemGNN	MAE	270.90	252.67	258.20	285.00
	RMSE	368.80	347.27	350.28	367.32
	MAPE	10.24	10.17	10.23	10.56
LSTM	MAE	275.88	237.84	268.56	274.6
	RMSE	361.12	322.36	352.72	365.88
	MAPE	10.18	8.75	8.77	10.43
TPA-LSTM	MAE	267.60	231.68	263.4	264.16
	RMSE	353.16	315.16	344.52	359.12
	MAPE	10.12	8.27	8.12	**9.88**
AM-LSTM	MAE	263.08	231.64	258.16	259.64
	RMSE	348.68	317.56	339.96	354.36
	MAPE	9.97	8.78	8.06	10.1
TPA-LSTM+	MAE	**248.32**	**214.52**	**249.04**	**252.08**
	RMSE	**330.20**	**298.84**	**330.32**	**346.08**
	MAPE	**9.82**	**8.02**	**8.03**	9.94

Table 7 shows that TPA-LSTM+ outperforms TPA-LSTM and AM-LSTM in all datasets in terms of overall error indicators. TPA-LSTM+ performs worse than TPA-LSTM in dataset P4 in terms of overall accuracy indicators. However, for the other models, the MAPE values in dataset P4 are higher than in other datasets. Figure 11 shows that traffic flow changes more drastically in dataset P4 than in the other three datasets. The TPA attention mechanism employs one-dimensional CNN to extract deep features from the input data, making it perform better in dataset P4, where traffic flow changes are more volatile.

6.3.4. The TPA-LSTM+ model prediction results

Figure 12 shows the weight allocation of different feature factors during the training process of the TPA-LSTM+ model, where the horizontal axis represents the iteration times and the vertical axis represents the weight allocation of different feature factors. In the four datasets, the overall weight allocation gradually stabilizes with the increase in iteration times. For different datasets, the model assigns different weights to different factors, but the traffic flow operation status factor has the largest weight allocation in different datasets and has the greatest impact on the model prediction. The traffic flow volume factor has a relatively large weight allocation. It changes slightly during the iteration process, indicating that the model has already focused on this factor, which is essential for the prediction result.



(a) P1 (b) P2

(c) P3 (d) P4

Figure 12. TPA-LSTM+ model weight allocation.

The distribution of base stations in Figure 1 shows that the traffic flow external influence are minimal in P1 and P2 datasets located on the main road of the interstate highway, with minor differences in weight allocation. P3 and P4 datasets are located in the interwoven area, and the traffic flow changes are unstable and greatly affected by the inflow and outflow of vehicles. Therefore, there are different emphases in weight allocation. As shown in Figure 11(c), the traffic flow in the P3 dataset shows similar trends on weekends and weekdays, so the model assigns a lower weight to the holiday factor in weight allocation. In Figure 11(d), the traffic flow in the P4 dataset shows dissimilar trends on weekends and weekdays, and the changes are large, so the model increases its attention to the holiday factor. This indicates that the model can automatically weigh different features and calculate their impact according to the characteristics of the dataset, and it has certain practicality.

Figure 11 shows the fitting performance of the proposed TPA-LSTM+ model and the LSTM model used for comparison on four datasets. The blue solid line represents the actual traffic flow, the orange and black solid lines represent the predicted traffic flow and error of the TPA-LSTM+ model, and the green and red dashed lines represent the predicted traffic flow and error of the LSTM model. In most cases, the error between the predicted data and the actual value is small. When the traffic flow data changes rapidly, the error of the TPA-LSTM+ model is smaller than that of the LSTM model, and it can adapt to this change more quickly, demonstrating stronger robustness.

7. Conclusions and perspective

In this study, we initially extract the temporal patterns underlying changes in traffic flow. Simultaneously, we introduce a traffic flow model to explore the intrinsic variation patterns within traffic flow, capturing features associated with traffic flow operational states. Subsequently, we construct a multi-feature input dataset and assess the efficacy of the added features in improving prediction performance through experimental validation. Regarding the input of multi-feature traffic flow data, we propose a novel method for utilizing attention mechanisms in this study. This method eliminates the shortcomings of previous attention mechanisms, which only weighted features on the temporal dimension and lacked attention to the original input data features, demonstrating the flexibility to select suitable features for different datasets in experiments. The proposed model has excellent performance in different datasets. However, there are some urgent issues to be addressed. On the one hand, some datasets representing external environmental features are needed to improve the features used for prediction. On the other hand, The application of optimization algorithms can enable deep learning models to more quickly discover optimal hyperparameters, thus reducing model training time [37]. Currently, there are numerous optimization algorithms such as the Diffused Memetic Optimizer (DMO) [38], Adaptive Polyploid Memetic Algorithm (APMA) [39], Ant-based Generation Constructive Hyper-heuristic, and more [40,41], which are worth considering and utilizing.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

We sincerely thank the editors and reviewers for their invaluable comments and suggestions, which put the article in its present shape.

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. Z. Ge, Y. Li, C. Liang, Y. Song, T. Zhou, J. Qin, Acsnet: adaptive cross-scale network with feature maps refusion for vehicle density detection, in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, (2021), 1–6. https://doi.org/10.1109/ICME51207.2021.9428454
2. H. Lu, Z. Ge, Y. Song, D. Jiang, T. Zhou, J. Qin, A temporal-aware LSTM enhanced by loss-switch mechanism for traffic flow forecasting, *Neurocomputing*, **427** (2021), 169–178. https://doi.org/10.1016/j.neucom.2020.11.026
3. B. S. Chen, S. C. Peng, K. C. Wang, Traffic modeling, prediction, and congestion control for high-speed networks: A fuzzy AR approach, *IEEE Trans. Fuzzy Syst.*, **8** (2000), 491–508. https://doi.org/10.1109/91.873574

4. B. M. Williams, L. A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results, *J. Transp. Eng.*, **129** (2003), 664–672.
5. L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, L. Damas, Predicting taxi-passenger demand using streaming data, *IEEE Trans. Intell. Transp. Syst.*, **14** (2013), 1393–1402. https://doi.org/10.1109/TITS.2013.2262376
6. F. Wu, H. Wang, Z. Li, Interpreting traffic dynamics using ubiquitous urban data, in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2016. https://doi.org/10.1145/2996913.2996962
7. N. P. Trinh, A. K. N. Tran, T. H. Do, Traffic flow forecasting using multivariate time-series deep learning and distributed computing, in *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, IEEE, (2022), 665–670. https://doi.org/10.1109/RIVF55975.2022.10013796
8. B. Geng, J. Ma, S. Zhang, Ensemble deep learning-based lane-changing behavior prediction of manually driven vehicles in mixed traffic environments, *Electron. Res. Arch.*, **31** (2023), 6216–6235. https://doi.org/10.3934/era.2023315
9. F. Aljuaydi, B. Wiwatanapataphee, Y. H. Wu, Multivariate machine learning-based prediction models of freeway traffic flow under non-recurrent events, *Alexandria Eng. J.*, **65** (2023), 151–162.
10. C. Chen, Z. Liu, S. Wan, J. Luan, Q. Pei, Traffic flow prediction based on deep learning in internet of vehicles, *IEEE Trans. Intell. Transp. Syst.*, **22** (2020), 3776–3789. https://doi.org/10.1109/TITS.2020.3025856
11. J. An, L. Fu, M. Hu, W. Chen, J. Zhan, A novel fuzzy-based convolutional neural network method to traffic flow prediction with uncertain traffic accident information, *IEEE Access*, **7** (2019), 20708–20722. https://doi.org/10.1109/ACCESS.2019.2896913
12. Y. Li, S. Chai, Z. Ma, G. Wang, A hybrid deep learning framework for long-term traffic flow prediction, *IEEE Access*, **9** (2021), 11264–11271. https://doi.org/10.1109/ACCESS.2021.3050836
13. X. Wang, Y. Wang, J. Peng, Z. Zhang, X. Tang, A hybrid framework for multivariate long-sequence time series forecasting, *Appl. Intell.*, **53** (2023), 13549–13568. https://doi.org/10.1007/s10489-022-04110-1
14. Y. Zhang, Y. Yang, W. Zhou, H. Wang, X. Ouyang, Multi-city traffic flow forecasting via multi-task learning, *Appl. Intell.*, **2021** (2021), 1–19. https://doi.org/10.1007/s10489-020-02074-8
15. M. Méndez, M. G. Merayo, M. Núñez, Long-term traffic flow forecasting using a hybrid CNN-BiLSTM model, *Eng. Appl. Artif. Intell.*, **121** (2023), 106041. https://doi.org/10.1016/j.engappai.2023.106041
16. Q. Du, F. Yin, Z. Li, Base station traffic prediction using XGBoost-LSTM with feature enhancement, *IET Networks*, **9** (2020), 29–37. https://doi.org/10.1049/iet-net.2019.0103
17. S. Wang, M. Zhang, H. Miao, Z. Peng, P. S. Yu, Multivariate correlation-aware spatio-temporal graph convolutional networks for multi-scale traffic prediction, *ACM Trans. Intell. Syst. Technol.*, **13** (2022), 1–22. https://doi.org/10.1145/3469087
18. Y. Li, K. Li, C. Chen, X. Zhou, Z. Zeng, K. Li, Modeling temporal patterns with dilated convolutions for time-series forecasting, *ACM Trans. Knowl. Discovery Data*, **16** (2021), 1–22. https://doi.org/10.1145/3453724

19. Q. Zhao, G. Yang, K. Zhao, J. Yin, W. Rao, L. Chen, Multivariate time-series forecasting model: Predictability analysis and empirical study, *IEEE Trans. Big Data*, 2023. https://doi.org/10.1109/TBDATA.2023.3288693
20. L. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, D. Phung, An effective spatial-temporal attention based neural network for traffic flow prediction, *Transp. Res. Part C Emerging Technol.*, **108** (2019), 12–28. https://doi.org/10.1016/j.trc.2019.09.008
21. H. Zheng, F. Lin, X. Feng, Y. Chen, A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction, *IEEE Trans. Intell. Transp. Syst.*, **22** (2020), 6910–6920. https://doi.org/10.1109/TITS.2020.2997352
22. J. Wu, J. Fu, H. Ji, L. Liu, Graph convolutional dynamic recurrent network with attention for traffic forecasting, *Appl. Intell.*, **2023** (2023), 1–15. https://doi.org/10.1007/s10489-023-04621-5
23. D. Qin, Z. Peng, L. Wu, Deep attention fuzzy cognitive maps for interpretable multivariate time series prediction, *Knowl.-Based Syst.*, (2023), 110700. https://doi.org/10.1016/j.knosys.2023.110700
24. W. Fang, W. Zhuo, J. Yan, Y. Song, D. Jiang, T. Zhou, Attention meets long short-term memory: A deep learning network for traffic flow forecasting, *Physica A*, **587** (2022), 126485. https://doi.org/10.1016/j.physa.2021.126485
25. R. Wan, C. Tian, W. Zhang, W. Deng, F. Yang, A multivariate temporal convolutional attention network for time-series forecasting, *Electronics*, **11** (2022), 1516. https://doi.org/10.3390/electronics11101516
26. S. Shun-Yao, S. Fan-Keng, L. Hung-yi, Temporal pattern attention for multivariate time series forecasting, *Mach. Learn.*, **108** (2019). https://doi.org/10.1007/s10994-019-05815-0
27. X. Geng, X. He, L. Xu, J. Yu, Graph correlated attention recurrent neural network for multivariate time series forecasting, *Inf. Sci.*, **606** (2022), 126–142. https://doi.org/10.1016/j.ins.2022.04.045
28. D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, et al., Spectral temporal graph neural network for multivariate time-series forecasting, *Adv. Neural Inf. Process. Syst.*, **33** (2020), 17766–17778.
29. X. Liu, J. Xu, M. Li, L. Wei, H. Ru, General-logistic-based speed-density relationship model incorporating the effect of heavy vehicles, *Math. Probl. Eng.*, **2019** (2019). https://doi.org/10.1155/2019/6039846
30. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
31. R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in *International Conference on Machine Learning,* PMLR, (2013), 1310–1318.
32. A. Graves, Generating sequences with recurrent neural networks, preprint, arXiv:13080850. https://doi.org/10.48550/arXiv.1308.0850
33. L. Cai, M. Lei, S. Zhang, Y. Yu, T. Zhou, J. Qin, A noise-immune LSTM network for short-term traffic flow forecasting, *Chaos*, **30** (2020). https://doi.org/10.1063/1.5120502
34. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, preprint, arXiv:14090473. https://doi.org/10.48550/arXiv.1409.0473
35. M. T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, preprint, arXiv:150804025. https://doi.org/10.48550/arXiv.1508.04025

36. Y. Lv, Y. Duan, W. Kang, Z. Li, F. Y. Wang, Traffic flow prediction with big data: A deep learning approach, *IEEE Trans. Intell. Transp. Syst.*, **16** (2014), 865–873. https://doi.org/10.1109/TITS.2014.2345663
37. W. Chai, Y. Zheng, L. Tian, J. Qin, T. Zhou, GA-KELM: Genetic-algorithm-improved kernel extreme learning machine for traffic flow forecasting, *Mathematics*, **11** (2023), 3574. https://doi.org/10.3390/math11163574
38. M. A. Dulebenets, A Diffused Memetic Optimizer for reactive berth allocation and scheduling at marine container terminals in response to disruptions, *Swarm Evol. Comput.*, **80** (2023), 101334. https://doi.org/10.1016/j.swevo.2023.101334
39. M. A. Dulebenets, An Adaptive Polyploid Memetic Algorithm for scheduling trucks at a cross-docking terminal, *Inf. Sci.*, **565** (2021), 390–421. https://doi.org/10.1016/j.ins.2021.02.039
40. M. Chen, Y. Tan, SF-FWA, A Self-Adaptive Fast Fireworks Algorithm for effective large-scale optimization, *Swarm Evol. Comput.*, **80** (2023), 101314. https://doi org/101016/j swevo
41. J. Pasha, A. L. Nwodu, A. M. Fathollahi-Fard, G. Tian, Z. Li, H. Wang, et al., Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings, *Adv. Eng. Inf.*, **52** (2022), 101623. https://doi.org/10.1016/j.aei.2022.101623