Journal of Big Data

METHODOLOGY Open Access

An adaptive composite time series forecasting model for short-term traffic flow

Qitan Shao[1], Xinglin Piao[1*], Xiangyu Yao[1], Yuqiu Kong[2], Yongli Hu[1], Baocai Yin[1] and Yong Zhang[1]

*Correspondence:
piaoxl@bjut.edu.cn

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2] School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian 116024, Liaoning, China

Abstract

Short-term traffic flow forecasting is a hot issue in the field of intelligent transportation. The research field of traffic forecasting has evolved greatly in past decades. With the rapid development of deep learning and neural networks, a series of effective methods have been proposed to address the short-term traffic flow forecasting problem, which makes it possible to examine and forecast traffic situations more accurately than ever. Different from linear based methods, deep learning based methods achieve traffic flow forecasting by exploring the complex nonlinear relationships in traffic flow. Most existing methods always use a single framework for feature extraction and forecasting only. These approaches treat all traffic flow equally and consider them contain same attribute. However, the traffic flow from different time spots or roads may contain distinct attributes information (such as congested and uncongested). A simple single framework usually ignore the different attributes embedded in different distributions of data. This would decrease the accuracy of traffic forecasting. To tackle these issues, we propose an adaptive composite framework, named Long-Short-Combination (LSC). In the proposed method, two data forecasting modules(L and S) are designed for short-term traffic flow with different attributes respectively. Furthermore, we also integrate an attribute forecasting module (C) to forecast the traffic attributes for each time point in future time series. The proposed framework has been assessed on real-world datasets. The experimental results demonstrate that the proposed model has excellent forecasting performance.

Keywords: Traffic forecasting, Deep learning, Adaptive composite framework, Traffic flow division

Introduction

Traffic flow forecasting, as a basis technology in transportation planning and traffic command management, is a core problem of intelligent transportation system [1]. Real-time and accurate traffic flow forecasting could make great contributions to the synergistic development of intelligent transportation and intelligent city. Thus, traffic flow forecasting has received wide attention from researchers.

In past decades, researchers have proposed many effective forecasting methods for traffic flow, such as Autoregressive Integrated Moving Average Model (ARIMA) [2], Kalman-filter-based method [3], Bayesian-based method [4], statistical-based method

[5] and so on. These methods adopt the shallow temporal features and statistical characteristics among traffic flow to achieve forecasting. However, traffic flow always lie on high dimentional and non-linear space. Thus the traditional linear methods would be no longer valid [6]. Therefore, some researchers have turned their focus to the models which are better suitable for nonlinear data. Radial Basis Function (RBF) [7] model, Support Vector Regression (SVR) [8] model, K-Nearest Neighbor-SVR (KNN-SVR) [9] model, and Least-Square SVR (LSSVR) [10] model have been used to forecast short-term traffic flow, and these models have good robustness for large-scale data regression problems. However, when dealing with large-scale data, these models require significant time and memory. In addition, the forecasting accuracy can be significantly affected by the kernel parameters.

With the rapid development of deep learning and its great success in big data area, a large number of deep learning based models have been carried out for short-term traffic flow forecasting [11]. Compared with traditional methods, deep learning based ones could extract more informative features from traffic flow such as Stacked AutoEncoder (SAE) based traffic data forecasting method [12], Deep Belief Network (DBN) based traffic flow forecasting method [13]. These methods could extract and transform the inherent features in the traffic flow by using deep architectures which adopt multi-layer nonlinear processing units [14]. To further explore the temporal feature among traffic flow, researchers proposed Recurrent Neural Network (RNN) based traffic flow forecasting method [15], Long Short-Term Memory Network (LSTM) based traffic flow forecasting method [16] and Bi-directional Long Short-Term Memory Network (Bi-LSTM) based traffic flow forecasting method [17]. Besides, Graph Neural Networks (GNNs) also have been introduced to formulate new traffic flow forecasting models for characterizing the spatial relationship among traffic flow [18]. A series of GNNs based methods such as Spatial Temporal Graph Convolutional Network (STGCN) based traffic flow forecasting method [19], Dual-Channel Spatio-Temporal Graph Convolutional Network (DC-STGCN) based traffic flow forecasting method [20] and Graph WaveNet based traffic flow forecasting method [21]. These GNNs based methods could further improve the accuracy of traffic flow forecasting and demonstrate their remarkable advantage of spatio-temporal feature extraction [22].

Since traffic flow has a large time span and a wide coverage area, there always contains different distribution and complicated spatio-temporal relationship among them. For example, congested and uncongested traffic flow belong to different traffic states and exhibit different distributions of traffic flow. Figure 1 presents the traffic flow variation curve of an intersection in Qingdao within one day. In this figure, it can be clearly observed that the traffic flow abruptly increases and remains high during the morning and evening peak hours. While during the non-peak hours, the traffic flow is relatively small. These reflect the distribution and fluctuation of traffic flow in the congested and uncongested states are not the same. Hence, the methods should be able to explore different feature among traffic flow for better forecasting. However, most existing methods treat all traffic flow equally and adopt a single networks data forecasting only. This kind of approaches may ignore the different interactions from different traffic situation, which would decrease the forecasting accuracy.

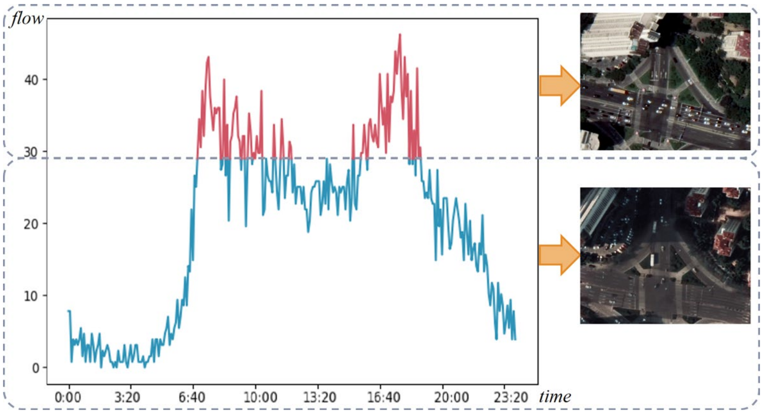

Fig. 1 Traffic flow change curve of an intersection in Qingdao during one day. The x-axis represents time, and the y-axis represents the traffic flow. Red and blue represent congested and uncongested traffic flow, respectively

Therefore, we propose a new adaptive composite framework, named Long-Short-Combination (LSC) for short-term traffic flow forecasting. The LSC framework is composed of two parts: a data forecasting module and an attribute forecasting module. The data forecasting module aims to accurately forecast traffic flows with varying attributes. We first divide the traffic flow into two parts according to their own attribute. And then each part data is extracted by two networks respectively. The attribute forecasting module is used to forecast the traffic attributes at each time point in the future time series and selects the corresponding model for traffic flow forecasting. After dividing the flow reasonably and combining two modules mentioned above, LSC can improve the accuracy of traffic flow forecasting more effectively.

The main contributions of this paper can be summarized as follows:

- An adaptive composite framework model is established for exploring different features in different distributions of traffic flow. Through an optional back propagation method, the data can adaptively select a suitable net for training to improve the forecasting effect.
- An attribute forecasting module for traffic flow is designed to forecast the traffic attributes in future time series, enabling the selection of different forecasting models for diverse traffic attributes.
- Through experiments on two real-world datasets verify the practicability and effectiveness of our proposed method.

The rest of this paper is organized as follows. Related work introduces the relevant work of short-term traffic flow forecasting. The architecture of the LSC model and the computational procedure using the divided data are described in detail in Method. Experiment and Result Analysis is devoted to data description and some performance measures. The final Conclusion makes conclusions and discusses the future work.

Related work

In this section, we introduce the related work about traffic flow forecasting. From the perspective of technical characteristics, traffic flow forecasting models can be divided into two categories: classical forecasting methods and deep learning models.

Classical forecasting methods

Early learning models usually used short-term time windows or data statistics tendency to estimate future traffic flow. For example, He et al. [23] computed the Hurst exponent to estimate the period of change in traffic flow time series. Meanwhile, Pei et al. [24] investigated the fractal properties of traffic flow by calculating its fractal dimension. On the other hand, An urban traffic flow forecasting model was established by Moretti et al. [25] through the integration of hybrid modeling techniques, including statistics and neural networks. Lin et al. [26] used Autoregressive Integrated Moving Average (ARIMA) model to achieve traffic flow forecasting by analyzing the data fluctuation characteristics. Yi et al. [27] used k-nearest-neighbor method to identify the optimal smoothing factor and combined it with a generalized regression model to improve the accuracy of traffic flow forecasting. In addition, Guo et al. [28] introduced the Kalman filter to forecast the short-term traffic flow. However, traffic flow is characterized by randomness. If the traffic flow shows irregular variations, the forecasting performance of these methods is not quite satisfactory. In order to solve this problem and discover the patterns from the nonlinear fluctuating historical data, Xiao et al. [29] constructed a dynamic forecasting model of traffic flow using radial basis function (RBF) to forecast the traffic flow of the city road and achieved better results. Liu et al. [9] proposed A hybrid forecasting model based on K-Nearest Neighbor (KNN) and Support Vector Regression (SVR). The KNN algorithm was used to reconstruct the traffic flow data and the SVR was used to forecast the short-term traffic flow.

However, traditional machine learning has unsatisfactory performance in dealing with nonlinear data, with large limitations and inaccurate forecasting results. Moreover, Shallower architectures of these models can only extract superficial traffic flow rules, and are not effective in utilizing massive traffic flow.

Deep learning models

With the development of deep learning techniques, several methods have been successfully applied in the forecasting of time-series data. Guo et al. [30] designed a Back Propagation Network (BPNN) for short-time traffic flow forecasting, which became the model basis for many future algorithms. Zhao et al. [31] designed a one-dimensional Temporal Convolutional Network (TCN) model to obtain spatiotemporal information traffic flow. Similarly Yang et al. [32] improved the SAE by incorporating an Empirical Mode Decomposition (EMD) to decompose complex time series, a collection of simple ones in the network forecasting to further improve the forecasting accuracy. Recurrent Neural Network (RNN) is an artificial neural network with hidden-layer-node connections and closed loops, RNN explores the time series input of any length by using internal memory cells. In order to better capture the temporal variation of traffic flow, Ma et al. [33] introduced RNN into traffic flow forecasting and achieve good results. To overcome the vanishing gradient and exploding gradient of RNNs during backpropagation, Zhao et al. [34] used multilayer LSTM networks for traffic flow forecasting, and the experimental results showed that LSTMs are more suitable than other deep neural networks. In order to further extract the temporal and spatial features of traffic flow, Du et al. [35] proposed a hybrid deep learning framework by combining Convolutional Neural Network (CNN) and LSTM for traffic flow forecasting. In addition, Wu et al. [36] proposed an Attention-Based LSTM (Att-LSTM) model incorporating an attention mechanism to further improve the forecasting accuracy. Ma et al. [37] further improved the LSTM network by combining the traditional LSTM network with a Bi-LSTM, and demonstrated the superiority of the model on short-term traffic flow forecasting through experiments.

Due to the complex spatiotemporal dependency of traffic data, further extraction of spatiotemporal features among traffic data is necessary to explore the correlation between traffic data in time and space. Yu et al. [19] proposed a Spatio-Temporal Graph Convolutional Networks (STGCN) to represent the traffic network as a graph and established a model with a complete convolutional structure. In addition, Pan et al. [20] designed a Dual-Channel Spatio-Temporal Graph Convolutional Networks (DC-STGCN) to explore the correlation between the daily and weekly time components, further improving the accuracy of traffic flow forecasting. However, GCN based traffic flow forecasting methods often struggle to capture the short-term and long-term time relationships in traffic flow data. To address these issues, Huo et al. [38] proposed a hierarchical traffic flow forecasting network to further improve forecasting performance.

Traffic data are complex time series. Different underlying features of traffic data has different attributes. Some researchers have considered the differences. Sun et al. [39] proposed framework built upon a multilayer perceptron (MLP), incorporating the principle of residual separation idea and wavelet decomposition techniques. This approach effectively captures various patterns and noise information within the traffic data. Similarly, Fang et al. [40] proposed framework separates the complex traffic data into stable and fluctuating trends. Then, a dual-channel spatio-temporal network is used to independently model these trends. Wang et al. [41] divided the city into grid cells by utilizing the regional attributes to create a graph structure. By using Graph Convolutional Networks (GCN) to capture spatial correlations and Temporal Convolutional Networks (TCN) to capture temporal correlations of the roads, the model improves prediction. In addition, Cai et al. [42] proposed a novel node adjustment mechanism, which increases the number of Radial Basis Function (RBF) nodes in complex scenarios and decreases it under normal conditions. This adaptability allows the model to effectively respond to time-varying traffic states, like in peak and non-peak. Wang et al. [43] designed a model called varying spatiotemporal graph-based convolution (VSTGC) to express detailed features, such as vehicle type, braking state and external variables.

Up to now, different deep learning structures have been adopted to improve the accuracy of traffic flow forecasting. Unfortunately, there has seldom research considered the different underlying features of traffic data with different attributes, especially the data attributes of congestion and non-congestion. Thus ignoring the negative impact of uniform training of congested and non-congested traffic data. In comparison, our proposed framework LSC can explore traffic data with different attributes and improve the accuracy of traffic data forecasting significantly.

Method

To address the interaction problem between different traffic data distributions, we designed the LSC framework, as shown in Fig. 2. The internal modules, combined with the designed selective backpropagation, enable the LSC framework to adaptively select the appropriate model for prediction based on different traffic attributes.

Let a denote the LSC framework transform function, then the forecast process can be represented as:

$$\hat{x}_{T+1} = [\mathbf{a}_0, \ldots, \mathbf{a}_n] \begin{bmatrix} x_{T-n} \\ \vdots \\ x_T \end{bmatrix}, \tag{1}$$

where $[x_{T-n}, x_{T-n+1}, \ldots, x_T]^T$ is a sequence of traffic flow observations with n historical time steps as the input of the model and the $\hat{x}_{T+1}$ is the forecast value at the next state.

LSC framework

In this paper, the LSC framework is designed to extract and categorize the data features of large and small traffic flows. The LSC framework consists of a data forecasting module and an attribute forecasting module (C model). The data forecasting module consists of a large traffic flow forecasting model (L model) and a small traffic flow forecasting model (S model). The structure of the LSC framework is illustrated in Fig. 2. And the training and testing processes of the entire LSC framework are detailed in Algorithm 1.

Specifically, for the task of traffic flow forecasting, we predict future traffic flow data at a 5-minute time granularity. Prior to training, we classify the data through traffic attribute division, labeling large traffic flows as 1 and small traffic flows as 0. Based on the data characteristics, the data is differenced and normalized before being input into the L, S, and C models, which are constructed based on L-B-LSTM. A selective backpropagation

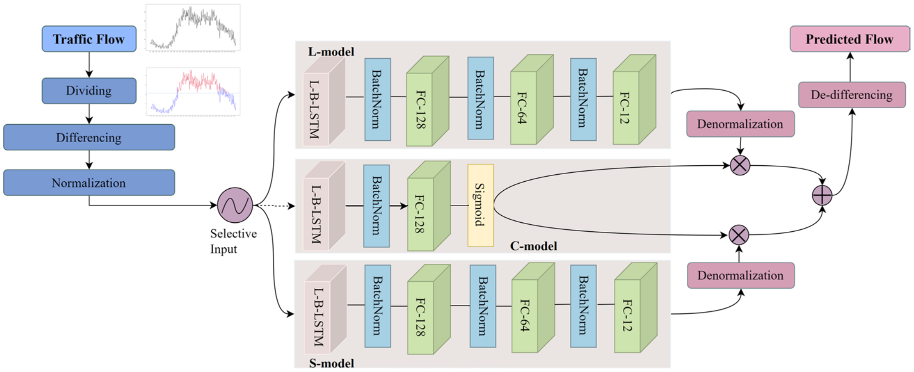

Fig. 2 The LSC framework; $\bigoplus$ denotes the element-wise addition and $\bigotimes$ denotes the element-wise product. The L-model and S-model are used to predict time-series states with different attributes, while the C-model is responsible for distinguishing traffic flow attributes and combining the outputs of the two models

method is designed to avoid interaction between different traffic attributes during training. The L model is trained to best fit the large traffic flow values in the time series, while the S model is trained to best fit the small traffic flow values in the time series. To accurately determine traffic attributes, the C model is trained to detect when future traffic time series can be categorized as large or small traffic flow. During the prediction process, we first utilize the C model to determine whether the traffic attribute at each future time point is a large traffic flow or a small traffic flow, recording large traffic flows as 1 and small traffic flows as 0. Subsequently, based on the classification results of the C model, we select the corresponding model (L model for label 1 and S model for label 0) for prediction. The prediction results are then concatenated and processed through denormalization and reverse de-differencing to obtain the final predicted values for each future time point.

Algorithm 1 The LSC framework

```
Training Stage:
Input: Training set$\{(X^{(1)}_{\text{train}}, Y^{(1)}_{\text{train}}, Y\_label^{(1)}_{\text{train}}) \ldots, (X^{(M)}_{\text{train}}, Y^{(M)}_{\text{train}}, Y\_label^{(M)}_{\text{train}})\}$

Output: the trained L-B-LSTM network parameters of model L, S and C
Initialize the L-B-LSTM of L, S and C layer parameters randomly
for epoch in training epochs do
    for $i = 1$ to $M$ do
        input $X^{(i)}_{\text{train}}$ into model C
        loss error $= BCE(Y\_label^{(i)}_{\text{train}}, \hat{Y}\_label^{(i)}_{\text{train}})$
        if the label of $X^{(i)}_{\text{train}} = 1$ then
            input $X^{(1)}_{\text{train}}$ into model L and compute $\hat{Y}^{(i)}_{\text{train}}$
            loss error $= L(Y^{(\mathrm{i})}_{\text{train}}, \hat{Y}^{(\mathrm{i})}_{\text{train}})$
        else input $X^{(i)}_{\text{train}}$ into model S
        end if
    end for
end for

Testing Stage:
Input:$\{(X^{(1)}_{\text{test}}, Y\_label^{(1)}_{\text{test}}), \ldots, (X^{(N)}_{\text{test}}, Y\_label^{(N)}_{\text{test}})\}$
Output: forecast result $Y_{\text{forecast}}$
for $j = 1$ to $N$ do
    input $X^{(i)}_{\text{test}}$ into model C to get the label
    if the label$= 1$ then
        input $X^{(i)}_{\text{test}}$ into model L to get forecast value $Y^{(L)}_{\text{forecast}}$
    else input $X^{(i)}_{\text{test}}$ into model S to get forecast value $Y^{(S)}_{\text{forecast}}$
    end if
    Merging $Y^{(L)}_{\text{forecast}}$ and $Y^{(S)}_{\text{forecast}}$ to get $Y_{\text{forecast}}$
end for
```

Traffic attribute division and time series analysis

Different distributions of traffic flow reflect different traffic states and imply different traffic attributes. For example, in the uncongested state, there are fewer vehicles and less traffic density. On the contrary, in the congested state, the traffic density is higher. To avoid the mutual influence among different distributions of traffic flow, the median value of traffic flow is taken as the standard to classify traffic attributes into two categories: large traffic flow (congestion) and small traffic flow (non-congestion). About the specific definition and division parameters of congested status, different countries and regions have their own standards. It is related to the service level of urban roads. In order to ensure the balance of different traffic attributes data after dividing, and prevent the impact of unbalanced data volume on model training, we take the median value of traffic flow as the division standard, and further compare with different division methods in Section IV. For each detection time point in the time series, the change in traffic status is instantaneous. To ensure the continuity and transition of traffic state changes, we classify the traffic flow state in each moment based on selected division standards. Specifically, the label of 1 represents congested traffic data, while the label of 0 represents uncongested traffic data. These classification labels are utilized for training the attribute forecasting module in the framework.

The time series is an important research subject in econometric. However, using time series data as a sample will violate the assumption of random sampling [44]. This is due to the fact that each observation can only be uniquely observed at any given time. If the time series is stationary, the autocorrelation coefficient will rapidly converge to zero with time intervals [45]. In this case, the sample drawn can represent the population sample, and the stationarity of the time series can replace random sampling. This can effectively reduce spurious regression and enable the model to perform correct parameter estimation or statistical inference. Therefore, during the data processing stage, the stationarity of time series must be detected and the non-stationary time series must be smoothed.

Taking the one-day traffic flow data of an intersection in Qingdao as an example, the autocorrelation graph is plotted as shown in the left chart of Fig. 3. The decreasing autocorrelation coefficient indicates that the series is unstable. Therefore, we use the difference operation to smooth the time series [46], and it is defined as follows:

$$\nabla_{\mathrm{k}} X_t = X_t - X_{t-k}, \tag{2}$$

where X is the time series, t is the number of terms in the series, and k is the order of the difference. For this experimental data we use the first-order difference, and the formula is follows:

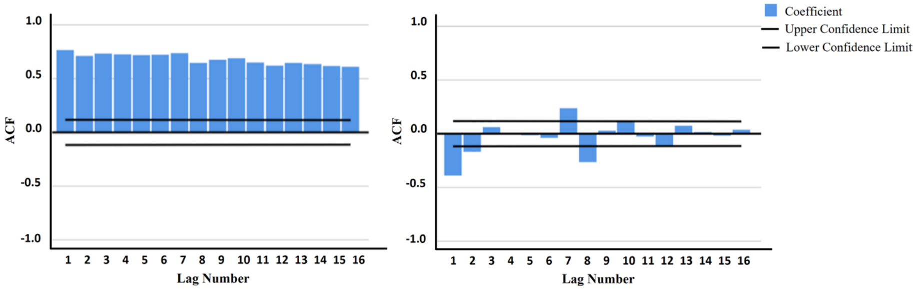

Fig. 3 Comparative autocorrelation analysis of traffic flow time series at an intersection in Qingdao. The left chart presents the autocorrelation plot of the original traffic flow time series, illustrating the raw data patterns. The right chart displays the traffic flow time series autocorrelation plot after smoothing

$$\nabla X_t = X_t - X_{t-1}. \tag{3}$$

In the right chart of Fig. 3, the autocorrelation coefficient rapidly converges to 0, indicating that a smooth time series is obtained after differential processing. The time series is then normalized by scaling the data and converting the time series to a dimensionless value. This improves the speed of the model in finding the optimal solution as well as the accuracy of the results. The definition of standardization is as follows:

$$X' = \frac{X - \mu}{\sigma}. \tag{4}$$

Attribute forecasting module and data forecasting module

The attribute forecasting module is implemented based on the L-B-LSTM forecasting model, as shown in Fig. 4. The LSTM is served as the first layer of the model to extract the hidden features of the time series. Bi-LSTM is applied to the second layer, and the input data dimension of Bi-LSTM is equal to the output data dimension of the first layer of LSTM. LSTM processes the time series in chronological order, but only the historical information is considered while the future information is ignored. As the second layer, the Bi-LSTM can fully learn the effective information extracted by LSTM, and avoid to reduce the forecasting accuracy because of the noise in the original time series data. In addition, Bi-LSTM can effectively extract the forward and backward dependencies in the traffic flow data [47–49].

The Bi-LSTM model contains two independent LSTMs as hidden layers with data propagation details shown below. The input sequences $[l_{t-n}, \ldots, l_{t-1}, l_t]$ are input into the next two independent LSTMs in forward and backward order for feature extraction. The forward output sequence $\overrightarrow{h}_t$ and backward output sequence $\overleftarrow{h}_t$ are

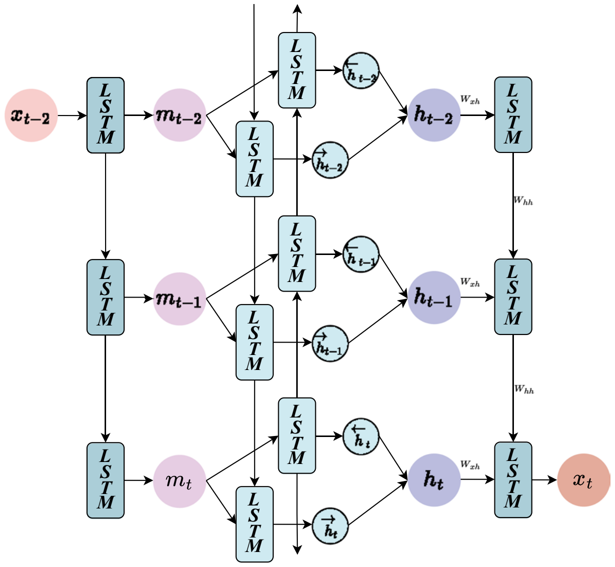

Fig. 4 The L-B-LSTM network architecture

concatenated together to form the final extracted feature vectors. The output of the Bi-LSTM can be expressed in the following equation:

$$\begin{cases} \overleftrightarrow{h}_t = \delta(\overrightarrow{h}_t, \overleftarrow{h}_t) \\ \overrightarrow{h}_t = \sigma_{\overrightarrow{h}}(W_{m\overrightarrow{h}} m_t + W_{\overrightarrow{h}\overleftarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \\ \overleftarrow{h}_t = \sigma_{\overleftarrow{h}}(W_{m\overleftarrow{h}} m_t + W_{\overleftarrow{h}\overrightarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}). \end{cases} \tag{5}$$

where W is the weight matrix, b is the bias vector, $\sigma_{\overrightarrow{h}}$ is the forward activation function, $\sigma_{\overleftarrow{h}}$ is the backward activation function, and δ is the connection function.

The LSTM has excellent inference and forecasting ability to obtain contextual information from long-range time series [50–52]. In L-B-LSTM, LSTM is introduced as the forecasting layer of the model, as shown in Fig. 4. The input of this layer is the output of the Bi-LSTM.

Within each LSTM unit, the current hidden layer state n_t is obtained based on the contextual information $[h_{t-n}, \ldots, h_{t-1}, h_t]$ and the previous unit's hidden state n_{t-1}, which is used to forecast the next traffic flow state x_{t+1}. By iteratively accumulating useful contextual information, the final forecasting result is obtained. The computational process is expressed as shown in the following equations:

$$\begin{cases} x_{t+1} = W_{yh} n_t + b_y \\ n_t = \sigma_h(W_{xh} h_t + W_{hh} n_{t-1} + b_h). \end{cases} \tag{6}$$

To constrain and guide the optimization of the model, the cost function $L\left(x_{t+1}, \hat{x}_{t+1}\right)$ is set as the root mean square error function in the training phase, as shown in the following equation:

$$L\left(x_{t+1}, \hat{x}_{t+1}\right) = \sqrt{\frac{1}{n} \times \sum_{i=1}^{n} \left(\hat{x}_{t+1} - x_{t+1}\right)^2}, \tag{7}$$

where $\hat{x}_{t+1}$ is the forecast value of the model, x_{t+1} is the measured value of traffic flow, and n is the number of training samples. Since the traffic attributes are divided into large and small traffic flows, the attribute forecasting is a binary classification task. In this paper the embedding dimension of the L-B-LSTM hidden layer in the attribute forecasting module is set to 128. Afterward, Fully Connected (FC) layers with Sigmoid activation functions are connected to achieve the classification of traffic attributes. These layers enable the network to map the input features extracted by the L-B-LSTM into the desired output classes. The Binary Cross-Entropy (BCE) loss function is selected, which is often the most appropriate loss function for binary classification tasks. The expression can be written as follows:

$$BCE(t,p) = -\left(\mathrm{t} \times \log(\mathrm{p}) + (1 - \mathrm{t}) \times \log(1 - \mathrm{p})\right), \tag{8}$$

where t represents the measured value and p represents the forecast value.

To forecast the traffic flow, the same embedding dimension of the L-B-LSTM hidden layer is set in the traffic forecasting module to 128, and the traffic features extracted by the L-B-LSTM are then fed into three fully connected (FC) layers. The number of nodes in the final FC layer is reduced to 12 to obtain the future traffic forecasting results.

Additionally, we have incorporated dropout method in our model to mitigate the risk of overfitting, thus enhancing the model's generalization ability and stability.

The selectable backpropagation

In order to ensure the correct training of the L and S models and to avoid the interaction between large and small traffic flow data during the training process, we use selectable backpropagation and depicts it visually in Fig. 5, where the blue and orange inputs respectively represent large and small traffic flows. The direction of the arrows represents the data transmission direction. Red arrows indicate that the data classification label matches the assigned model, allowing loss calculation and backpropagation, while black arrows indicate the opposite. After the data are divided and labeled, each forecasting sample contains traffic flow data for 12 time points, and both large and small traffic flows are distributed in each sample. If direct backpropagation is performed directly, the loss of small traffic flow data will inevitably affect the L model, and vice versa.

As a result, the optional back-propagation ignores forecastings on the small traffic flow data in the L model and on the large traffic flow data in the S model, forcing the model to only focus on the classification data corresponding to itself. Specifically, when training the L model, only the large traffic flow data add to the loss, and similarly for the S model,

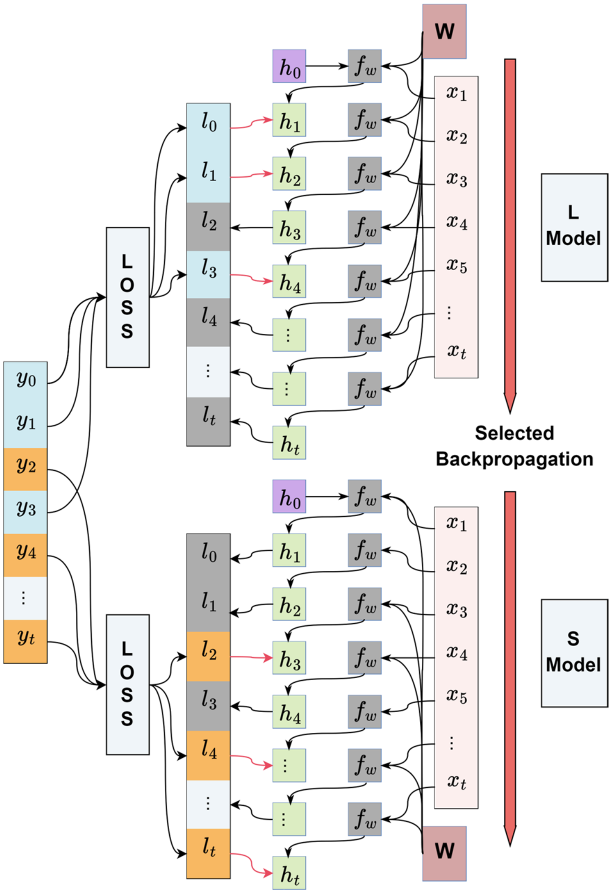

Fig. 5 The selectable backpropagation for the L and S models. Blue and orange inputs represent large and small traffic flow values, respectively

only the data classified as small traffic flow increases the loss. This ensures that the hidden parameters in the network are updated properly for both large and small traffic flow data.

Experiment and result analysis

In this section, in order to test the effectiveness of traffic flow classification as well as to enhance the persuasive power of the LSC framework, the forecasting accuracy is used to represent the solving ability in experiments. Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) are chosen as the evaluation indexes, and the test datasets are collected by traffic fellow detectors at multiple spots in two different datasets. The training procedure is implemented with PyTorch and a GPU (GeForce RTX 3090). The comparison experiments are carried out to compare the framework with other superior models, and the experimental results are analyzed. The best experimental results in the table are bolded, and the suboptimal ones are underlined. This framework addresses the problem of forecasting traffic data at an individual intersection, and does not involve the research of spatial characteristics in traffic data currently.

Data description

Qingdao dataset and PeMS08 dataset are used to conduct the experiments. Qingdao dataset is a month (30 days) of traffic flow data collected by sensors at an intersection of the road network in Qingdao, China, with a data sampling interval of 5 min and 288 sample data collected per day. In the experiment, the traffic flow data of the first 18 days (5184 sampling intervals, 60% of the total data) are collected as the training set, the traffic flow data of the 18th to 23rd days (1728 sampling intervals, 20% of the total data) are used as the validation set, and the others (1728 sampling intervals, 20% of the total data) are used as the test data.

PeMS08 dataset was collected from California by the Caltrans Performance Measurement System (PeMS) at a frequency of sampling every 30 s. PeMS08 contains traffic data from 170 sensors in San Bernardino from July to August 2016. The raw data is aggregated into 5-minute intervals, which means that there are 288 records of sampling time points in a day, with a total of 17857 valid records at each point. In the experiment, three spots are selected to conduct the experiments. And 60% of the total traffic flow data are collected as the training set, 20% is used as the validation set, and the others are used as testing dataset.

Evaluation Indexes

To evaluate the LSC framework performance, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) are selected as the evaluation metrics [53], which are commonly selected to assess traffic flow forecasting performance. The specific calculation equations are as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{h}_i - h_i\right|, \tag{9}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{h_i - \hat{h}_i}{h_i}\right|, \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{h}_i - h_i\right)^2}. \tag{11}$$

In the above three equations, n is the sample size of the data, $\hat{h}_i$ represents the forecast traffic flow value, and h_i represents the observation value. To evaluate the effectiveness of the C model in classifying traffic data of large and small flows, the $F1$ score are selected as an evaluation index, which is calculated as follows:

$$recall = \frac{TP}{TP + FN}, \tag{12}$$

$$precision = \frac{TP}{TP + FP}, \tag{13}$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{14}$$

where TP, FP, and FN respectively represent the true positive, false positive, and false negative values. precision represents the proportion of correct samples forecasted by the classifier. Recall represents the ratio of all correct cases forecasted by the classifier to all actual correct cases. The $F1$ score is the harmonic mean of precision and recall, and higher $F1$ means better model performance.

Framework parameters

The size of the input sequence and number of hidden nodes in each layer are important to the LSC framework for forecasting accuracy. We find LSTM layer widths of 128 nodes are better suited for the L, S and C models. All models are trained using PyTorch 1.9.1. Finally, the L-B-LSTM layers were trained using an SGD optimizer with learning rate 10^{-3}, while the fully connected layers were trained using an Adam optimizer with learning rate 5×10^{-4}. The best-performing model on the validation set is selected to conduct testing and evaluate our framework.

Effect of L-B-LSTM

As shown in Fig. 2, L-B-LSTM is the backbone structure of the LSC framework. To verify the effectiveness, we compare the LSC framework with L-B-LSTM as the backbone with LSC using the LSTM, Bi-LSTM, and Att-LSTM as the backbone. Specifically, in the LSC framework, the features are explored through L model and S model respectively, and the C model is used as the attribute forecasting module. To prove the effectiveness of L-B-LSTM, MAE, MAPE and RMSE are selected as the accuracy evaluation indicators for comparing L and S models in Qingdao dataset (A, B). And F1score is the evaluation indicator of C model. The experimental results are shown in Table 1. It shows that L-B-LSTM has good performance in the traffic forecasting module (L and S models), especially in the MAE evaluation indicator. In contrast, Att-LSTM performs slightly better in certain evaluation indicators. However, the overall performance of Att-LSTM falls

Table 1 Effectiveness of L-B-LSTM in LSC in Qingdao Dataset (A, B)

Dataset	Models	L			S			C
		MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE	F1score
SPOT A	LSTM	1.82	1.38	1.05	5.92	0.80	0.61	**0.98**
	Bi-LSTM	1.63	1.27	0.94	7.56	1.05	0.81	0.97
	Att-LSTM	**1.15**	1.17	0.86	4.98	0.68	0.55	**0.98**
	L-B-LSTM	1.25	**1.06**	**0.73**	**4.41**	**0.54**	**0.41**	**0.98**
SPOT B	LSTM	1.94	0.83	0.64	18.79	0.83	0.64	0.96
	Bi-LSTM	1.53	0.68	0.49	15.71	**0.66**	0.52	0.96
	Att-LSTM	1.61	0.67	0.53	**12.55**	0.75	0.62	**0.98**
	L-B-LSTM	**1.27**	**0.53**	**0.41**	13.76	0.69	**0.51**	**0.98**

The best experimental results in the table are bolded

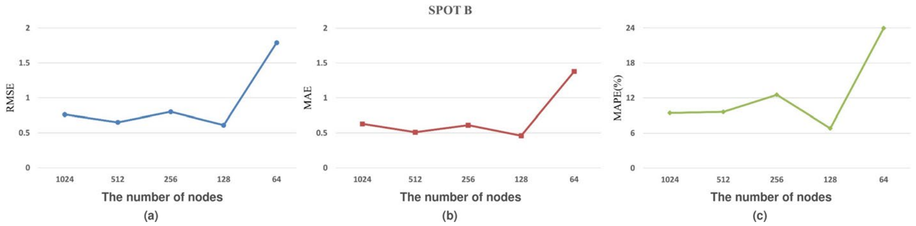

Fig. 6 Effect of the number of nodes in the LSC framework in Qingdao Dataset B. The y-axis represents different evaluation indexes, and the x-axis represents the number of nodes

slightly short of L-B-LSTM. Furthermore, the attribute forecasting module (C model) is a simple binary classification task based on median value dividing, so several models perform similarly with good classification results. Therefore, L-B-LSTM is selected as the main structure of the model in the construction of the overall LSC framework.

In the LSC framework, the embedding dimension of the L-B-LSTM hidden layer is set to 128. In order to verify the rationality of the nodes number, we choose the hidden layers with different numbers of nodes for training in Qingdao Dataset B. The results are shown in Fig. 6. It shows that the widths of 128 nodes are better suited for the framework. And all evaluation indexes are less than others.

To demonstrate the effect of choosing median as the division standard, we compare the results of using the mean, the mode, the variance and the standard deviation as division standard of the L module, S module and LSC framework in Qingdao dataset B. The experimental results are shown in Fig. 7. It shows that using the median as the division standard performs better than others, both on the all framework and the modules. Because, the median can ensure that the amount of data used for training in the L and S modules is balanced.

Effect of data division

In the model introduction section, it is hypothesized that dividing the traffic flow can effectively avoid the interaction between different attributes of traffic flow, and LSC can effectively extract the hidden features of time series based on the divided traffic data. To verify this hypothesis, we use MAPE as the main evaluation index, and

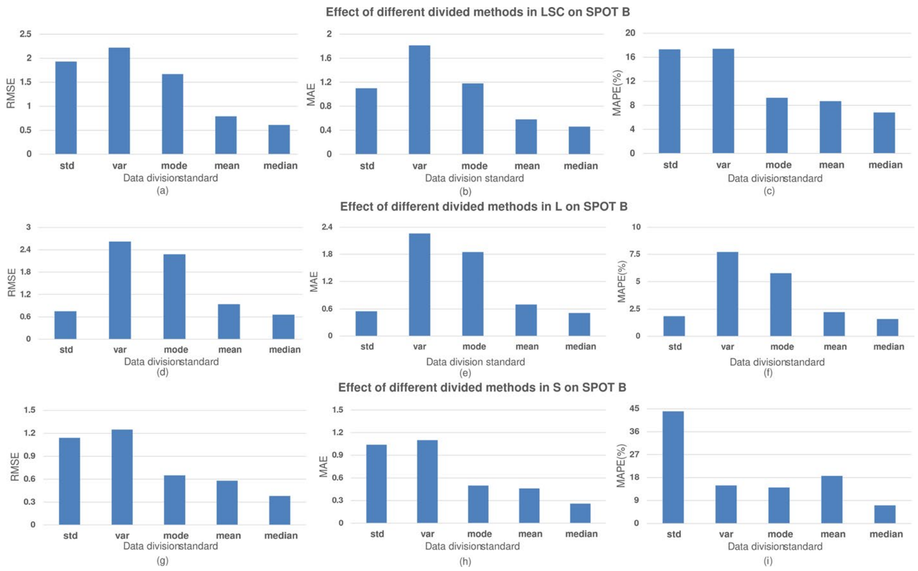

Fig. 7 Effect of Different Divided Methods in L module (**a**–**c**), S module (**d**–**f**) and LSC framework (**g**–**i**) in Qingdao Dataset B. The x-axis represents data division standard, and the y-axis represents different evaluation indexes

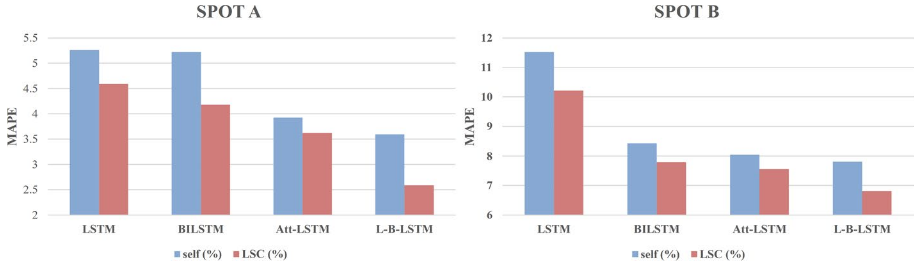

Fig. 8 Effect of data divisions, where "self" represents the model itself without data division, while "LSC" indicates the forecasting using the LSC framework. The y-axis represents MAPE evaluation indexes

performed comparative experiments on LSTM, Bi-LSTM, Att-LSTM, and L-B-LSTM models in Qingdao dataset (A, B). The experimental results are shown in Fig. 8, where "self" represents the model itself using the traffic flow time series for forecasting without data division, while "LSC" indicates that the forecasting using the LSC framework after dividing the traffic flow data into large and small flows. It can be clearly observed from the figure that the forecasting performance is significantly improved after dividing the traffic flow data. Among them, the L-B-LSTM model combined with LSC has the best forecasting performance, and the average result is 1% higher compared with only using L-B-LSTM model.

Furthermore, using L-B-LSTM as the backbone network significantly outperformed the other methods compared, with an average improvement of 2.7%, 1.27% and 0.89% higher than LSTM, BiLSTM, and Att-LSTM respectively. This also demonstrates the effectiveness of using L-B-LSTM as the backbone structure of the LSC framework. Therefore, it can be proved that the reasonable division of traffic flow can better explore the hidden features of traffic and improve the forecasting performance of the framework.

Effect of LSC

To demonstrate the performance of our proposed LSC framework and verify the differences between deep learning and traditional machine learning algorithms, This section compares the designed LSC model with various methods. Since this framework only solves the short-term traffic prediction problem at a single intersection, and does not utilize the road structural to further capture spatial correlations, it is not compared with graph based networks currently. Among the traditional machine learning algorithms include: ARIMA model (a combination of Autoregressive and Moving Average models, which is a traditional time series forecasting analysis method), RBF (Radial Basis Function neural network, a single-layer neural network structure with RBF kernel), and SVR (Support Vector Regression, which applies support vector machines to regression problems). Deep learning algorithms include: BPNN (Multilayer Back Propagation Neural Network, which is the most basic deep learning forecasting model), RNN (Recurrent Neural Network, with good time series information processing capabilities), LSTM (Long Short-Term Memory Network, a variation of traditional RNN, mitigating the phenomenon of gradient disappearance or explosion), Bi-LSTM (bi-directional LSTM, which captures both forward and backward information), L_BILSTM (has better performance in processing short-time series information), Att-LSTM (retains the intermediate output results of the LSTM encoder for the input sequence, and selects the attention mechanism to selectively learn the input), LSTM-GRU [54] (consists of a mixture of LSTM and Gated Recurrent Unit for better mining of time series features) and PSO-Bi-LSTM [55] (is based on the Bi-LSTM model and is optimized using PSO technique to expand the global search capability of the model). The experimental comparison results on the Qingdao dataset are shown in Table 2, and those on the PeMS08 dataset are presented in Table 3. The average results of Qingdao dataset and PeMS08 dataset are shown in Table 4.

As shown in the tables, it can be evidently found that the deep learning algorithms outperform the traditional machine learning algorithms in exploring traffic flow information. The SVR and RBF are better than conventional ARIMA, which has the lowest precision. Although the forecasting results of deep learning methods BPNN and LSTM are decent in classical methods, they are still less than LSTM-GRU and PSO-Bi-LSTM. And the LSC framework performs better in traffic flow forecasting than other deep learning methods. These demonstrate the effectiveness and feasibility of the proposed framework.

Furthermore, to demonstrate the transferability of the framework, we trained the model on the PeMS08 dataset (E) and adjusted the data dividing value before testing it on the Qingdao dataset (A). As shown in Table 5, the comparison with other models indicates that the LSC model exhibits excellent transferability. It can still perform well even after changing the dataset.

In traffic data, there are significant differences in traffic flow characteristics between weekdays and weekends. It is typically manifested as increased traffic volume, increased congestion, and less distinct morning and evening peak hours. To make further

Table 2 Forecast performances of different models in Qingdao Dataset (A, B, C, D)

Models	SPOT A			SPOT B			SPOT C			SPOT D		
	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE
ARIMA	18.4	11.45	7.38	27.85	7.12	4.59	12.45	7.36	5.04	13.94	5.66	3.62
RBF	11.45	7.04	5.31	21.72	4.57	3.15	9.25	5.26	3.42	7.21	3.91	2.11
SVR	8.75	7.81	4.72	17.65	4.87	2.72	7.70	4.89	3.67	6.12	3.52	2.44
BPNN	7.68	1.68	1.55	13.52	1.32	0.88	6.32	2.66	1.05	4.96	1.17	0.94
RNN	6.42	1.38	1.11	13.78	0.99	0.76	4.93	0.98	0.71	3.67	0.61	0.47
LSTM	5.26	1.45	1.07	11.52	0.84	0.65	3.57	1.07	0.76	2.44	0.53	0.42
Bi-LSTM	5.22	1.17	0.94	8.43	0.76	0.57	4.05	0.84	0.66	2.51	0.46	0.37
L_BILSTM	3.59	0.97	0.79	7.81	0.75	0.57	3.15	**0.77**	0.58	2.45	0.45	0.36
Att-LSTM	3.92	1.05	0.96	8.04	0.79	0.62	3.33	0.79	0.58	2.12	0.51	0.36
LSTM-GRU	3.45	1.01	0.78	7.57	0.71	0.48	3.38	0.88	0.64	2.71	0.59	0.33
PSO-Bi-LSTM	4.58	1.09	0.85	8.17	0.73	0.61	**2.73**	0.81	0.57	2.04	0.42	0.31
LSC	**2.59**	**0.88**	**0.59**	**6.81**	**0.61**	**0.46**	3.16	0.78	**0.46**	**1.43**	**0.38**	**0.27**

The best experimental results in the table are bolded, and the suboptimal ones are underlined

Table 3 Forecast performances of different models in PeMS08 Dataset (E, F, G)

Models	SPOT E			SPOT F			SPOT G		
	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE
ARIMA	27.00	86.62	60.18	24.72	71.09	50.59	32.96	73.19	50.82
RBF	17.20	68.62	46.53	19.12	57.49	42.30	23.34	54.77	40.42
SVR	19.23	66.83	48.37	18.63	55.99	41.38	20.76	54.55	39.93
BPNN	14.52	46.49	41.54	16.28	34.68	25.96	15.69	39.78	32.09
RNN	14.68	45.56	35.55	11.55	36.11	27.12	15.38	38.70	29.33
LSTM	11.20	42.56	31.69	9.35	32.76	24.75	13.71	35.14	26.25
Bi-LSTM	10.58	40.34	28.91	9.48	32.65	24.69	12.83	31.84	23.94
L_BILSTM	10.41	39.21	27.77	9.15	34.45	24.79	13.30	31.09	23.52
Att-LSTM	10.46	36.00	28.10	9.26	32.98	**22.32**	12.71	32.95	24.68
LSTM-GRU	11.19	35.46	26.56	9.98	32.21	23.48	13.10	31.96	23.65
PSO-Bi-LSTM	10.51	34.71	25.42	10.96	32.44	24.55	12.12	31.41	24.10
LSC	**10.14**	**31.82**	**24.01**	**8.70**	**31.55**	22.45	**11.57**	**27.11**	**20.22**

The best experimental results in the table are bolded, and the suboptimal ones are underlined

Table 4 Average Forecast Performances of Different Models in Qingdao Dataset and PeMS08 Dataset

Models	Qingdao			PeMS08		
	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE
ARIMA	18.16	7.90	5.16	28.23	76.97	53.86
RBF	12.41	5.20	3.50	19.89	60.29	43.08
SVR	10.06	5.27	3.39	19.54	59.12	43.23
BPNN	8.12	1.71	1.11	15.50	40.32	33.20
RNN	7.20	0.99	0.76	13.87	40.12	30.67
LSTM	5.70	0.97	0.73	11.42	36.82	27.56
Bi-LSTM	5.05	0.81	0.64	10.96	34.94	25.85
L_BILSTM	4.25	0.74	0.58	10.95	34.91	25.36
Att-LSTM	4.35	0.79	0.63	10.81	33.98	25.37
LSTM-GRU	4.28	0.80	0.56	11.42	33.21	24.56
PSO-Bi-LSTM	4.38	0.76	0.59	11.20	32.85	24.69
LSC	**3.50**	**0.66**	**0.45**	**10.14**	**30.16**	**22.23**

The best experimental results in the table are bolded, and the suboptimal ones are underlined

Table 5 Verification of model transferability to different datasets by training on the PeMS08 Dataset (E) and testing on the Qingdao Dataset (A)

Models	E → A		
	MAPE(%)	RMSE	MAE
LSTM	24.55	3.31	2.45
Bi-LSTM	36.40	3.65	3.08
L_BILSTM	24.01	**3.08**	2.33
Att-LSTM	21.98	3.47	2.73
LSTM-GRU	16.51	3.61	2.54
PSO-Bi-LSTM	20.69	3.99	2.45
LSC	**12.98**	**3.08**	**2.18**

The best experimental results in the table are bolded, and the suboptimal ones are underlined

Table 6 Forecast performances of different models in Qingdao Dataset (A) on weekdays and weekends

Models	Weekdays			Weekends		
	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE
ARIMA	22.15	11.54	7.74	36.72	15.87	11.83
RBF	15.60	8.19	4.89	32.34	12.34	9.76
SVR	11.52	7.44	5.48	27.80	11.29	9.88
BPNN	8.37	3.79	2.11	25.78	9.04	8.35
RNN	9.93	2.35	2.07	26.35	8.68	6.58
LSTM	5.78	1.32	0.98	21.60	7.46	6.46
Bi-LSTM	5.40	1.41	0.99	20.23	6.05	5.15
L_BILSTM	4.38	**1.23**	0.80	19.71	5.89	5.35
Att-LSTM	3.89	1.34	0.89	17.99	6.56	5.62
LSTM-GRU	4.62	1.12	0.92	17.82	6.53	5.97
PSO-Bi-LSTM	5.00	1.34	0.83	18.63	6.06	5.11
LSC	**3.53**	1.37	**0.72**	**17.48**	**5.66**	**4.87**

The best experimental results in the table are bolded, and the suboptimal ones are underlined

Table 7 Forecast Performances of Different Models in PeMS08 Dataset (E) on Weekdays and Weekends

Models	Weekdays			Weekends		
	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE
ARIMA	26.95	95.06	64.63	26.26	74.92	52.41
RBF	16.83	76.58	48.73	19.76	51.27	40.12
SVR	19.66	73.75	51.83	19.82	50.83	40.76
BPNN	11.34	50.39	48.29	12.21	47.15	35.70
RNN	13.47	51.02	47.81	12.63	39.83	31.73
LSTM	12.69	43.30	31.48	12.28	36.19	28.35
Bi-LSTM	10.96	41.01	29.69	12.09	36.40	28.63
L_BILSTM	10.00	39.94	28.23	10.88	35.52	27.36
Att-LSTM	10.61	41.19	29.24	12.54	35.14	27.70
LSTM-GRU	10.76	40.76	30.62	**10.06**	33.41	25.72
PSO-Bi-LSTM	10.56	38.01	27.04	12.08	33.46	26.44
LSC	**9.46**	**30.74**	**23.70**	10.19	**29.57**	**23.28**

The best experimental results in the table are bolded, and the suboptimal ones are underlined

refinements, forecasting experiments are conducted on traffic flow data in Qingdao dataset (A) and PeMS08 dataset (E) on weekdays and weekends. As shown in Table 6 and Table 7, the LSC framework achieves better forecasting performance than other methods both on weekdays and weekends. This further demonstrates the effectiveness of the proposed model.

Conclusion

In this study, we propose an adaptive composite framework LSC for short-term traffic flow forecasting, which aims to avoid the influence of traffic flow with different attributes on the model training and better extract the hidden information in the time series. We divide the processed time series into large traffic flow data and small traffic flow data based on traffic attributes. The LSC model adaptively learns different regression models for forecasting large and small traffic volumes, and selects the corresponding model for each time node in the future time series through an attribute forecasting module. In addition, an L-B-LSTM model with two LSTM layers and one Bi-LSTM layer is selected for the internal structure of the framework, which is combined with an optional back-propagation method to further improve the performance of the composite framework. Compared with existing methods, the LSC framework achieves superior forecasting performance on the traffic flow dataset. In the future, we will attempt to combine the present framework with the spatial structure of roads to further explore the network-wide features. And we will investigate more advanced traffic attribute classification methods, extending the current binary classification to multi-class classification. Additionally, we will further attempt to capture hidden temporal features during traffic state transitions to further enhance the prediction performance.

Author contributions

In our manuscript, Yong Zhang and Xinglin Piao contributed to the Abstract and Introduction sections, Xiangyu Yao and Yuqiu Kong contributed to the Related Work section, Yongli Hu and Baocai Yin contributed to the Method section, and Qitan Shao was responsible for the Experiment and Results Analysis section.

Funding

This work is supported in part by the National Key R&D Program of China (No. 2021ZD0111902), NSFC (No. 62072015, U21B2038) and in part by Beijing Natural Science Foundation (No.4222021), R&D Program of Beijing Municipal Education Commission (No. KZ202210005008, No. KM202410005031).

Data availibility

No datasets were generated or analysed during the current study.

Declarations

Competing interests

The authors declare no competing interests.

Received: 10 December 2023 Accepted: 16 July 2024
Published online: 03 August 2024

References

1. Ma X, Zhao J, Gong Y, Sun X. Carrier sense multiple access with collision avoidance-aware connectivity quality of downlink broadcast in vehicular relay networks. IET Microw Antennas Propag. 2019;13(8):1096–103.
2. Chandra SR, Al-Deek H. Predictions of freeway traffic speeds and volumes using vector autoregressive models. J Intell Transp Syst. 2009;13(2):53–72.
3. Xie Y, Zhang Y, Ye Z. Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. Computer-Aided Civ Infrastruct Eng. 2007;22(5):326–34.
4. Nahar J, Chen Y-PP, Ali S. Kernel-based naive bayes classifier for breast cancer prediction. J Biol Syst. 2007;15(01):17–25.
5. Tjondronegoro DW, Chen Y-PP. Knowledge-discounted event detection in sports video. IEEE Trans Syst Man Cybern Part A Syst Hum. 2010;40(5):1009–24.
6. Bi J, Zhang X, Yuan H, Zhang J, Zhou M. A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM. IEEE Trans Automation Sci Eng. 2021;19(3):1869–79.
7. Chen D. Research on traffic flow prediction in the big data environment based on the improved RBF neural network. IEEE Trans Ind Inf. 2017;13(4):2000–8.
8. Bermolen P, Rossi D. Support vector regression for link load prediction. Computer Netw. 2009;53(2):191–201.
9. Liu Z, Du W, Yan D-m, Chai G, Guo J-h. Short-term traffic flow forecasting based on combination of k-nearest neighbor and support vector regression. J Highw Transp Res Dev. 2018;12(1):89–96.
10. Wang H, Hu D. Comparison of SVM and LS-SVM for regression. In: 2005 International Conference on Neural Networks and Brain, vol. 1. IEEE; 2005. p. 279–83.
11. Pan G, Fu L, Chen Q, Yu M, Muresan M. Road safety performance function analysis with visual feature importance of deep neural nets. IEEE/CAA J Autom Sin. 2020;7(3):735–44.
12. Lv Y, Duan Y, Kang W, Li Z, Wang F-Y. Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst. 2014;16(2):865–73.

13. Li L, Qin L, Qu X, Zhang J, Wang Y, Ran B. Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. Knowl Based Syst. 2019;172:1–14.
14. Schmidhuber J. Deep learning in neural networks: an overview. Neural Netw. 2015;61:85–117.
15. Ramakrishnan N, Soni T. Network traffic prediction using recurrent neural networks. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE; 2018. p. 187–93.
16. Tian Y, Pan L. Predicting short-term traffic flow by long short-term memory recurrent neural network. In: 2015 IEEE International Conference on Smart city/SocialCom/SustainCom (SmartCity), IEEE; 2015. p. 153–8.
17. Li Z-Y, Ge H-X, Cheng R-J. Traffic flow prediction based on BILSTM model and data denoising scheme. Chin Phys B. 2022;31(4): 040502.
18. Jiang W, Luo J. Graph neural network for traffic forecasting: a survey. Expert Syst Appl. 2022;207: 117921.
19. Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv:1709.04875 [Preprint]. 2017..
20. Pan C, Zhu J, Kong Z, Shi H, Yang W. DC-STGCN: dual-channel based graph convolutional networks for network traffic forecasting. Electronics. 2021;10(9):1014.
21. Wu Z, Pan S, Long G, Jiang J, Zhang C. Graph wavenet for deep spatial-temporal graph modeling. arXiv:1906.00121 [Preprint]. 2019.
22. Wang J, Zhang Y, Wang L, Hu Y, Piao X, Yin B. Multitask hypergraph convolutional networks: a heterogeneous traffic prediction framework. IEEE Trans Intell Transp Syst. 2022;23(10):18557–67.
23. He G-g, Feng W-d. Study on long-term dependence of urban traffic flow based on rescaled range analysis. Xitong Gongcheng Xuebao. 2004;19:166–9.
24. Pei Y, Li H. Research on fractal dimensions of traffic flow time series on expressway. J Highw Transp Res Dev. 2006;23(2):115–9.
25. Moretti F, Pizzuti S, Panzieri S, Annunziato M. Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling. Neurocomputing. 2015;167:3–7.
26. Lin X, Huang Y. Short-term high-speed traffic flow prediction based on ARIMA-GARCH-M model. Wirel Pers Commun. 2021;117(4):3421–30.
27. Yi L, Zhang C, Pei Z. A modified general regression neural network with its application in traffic prediction. J Shandong Univ Eng Sci. 2013;43(1):9–14.
28. Guo J, Huang W, Williams BM. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. Transp Res Part C Emerg Technol. 2014;43:50–64.
29. Xiao J-M, Wang X-H. Study on traffic flow prediction using RBF neural network. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826), vol. 5. IEEE; 2004. p. 2672–5.
30. Xiaojian G, Quan Z. A traffic flow forecasting model based on bp neural network. In: 2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS), vol. 3. IEEE; 2009. p. 311–4.
31. Zhao W, Gao Y, Ji T, Wan X, Ye F, Bai G. Deep temporal convolutional networks for short-term traffic flow forecasting. IEEE Access. 2019;7:114496–507.
32. Yang H-F, Dillon TS, Chen Y-PP. Optimized structure of the traffic flow forecasting model with a deep learning approach. IEEE Trans Neural Netw Learn Syst. 2016;28(10):2371–81.
33. Ma X, Yu H, Wang Y, Wang Y. Large-scale transportation network congestion evolution prediction using deep learning theory. PLoS ONE. 2015;10(3):0119044.
34. Zhao Z, Chen W, Wu X, Chen PC, Liu J. LSTM network: a deep learning approach for short-term traffic forecast. IET Intell Transp Syst. 2017;11(2):68–75.
35. Du S, Li T, Gong X, Yang Y, Horng SJ. Traffic flow forecasting based on hybrid deep learning framework. In: 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), IEEE; 2017. p. 1–6.
36. Wu P, Huang Z, Pian Y, Xu L, Li J, Chen K. A combined deep learning method with attention-based LSTM model for short-term traffic speed forecasting. J Adv Transp. 2020;2020:1–15.
37. Ma C, Dai G, Zhou J. Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method. IEEE Trans Intell Transp Syst. 2021;23(6):5615–24.
38. Huo G, Zhang Y, Wang B, Gao J, Hu Y, Yin B. Hierarchical spatio-temporal graph convolutional networks and transformer network for traffic flow forecasting. IEEE Trans Intell Transp Syst. 2023. https://doi.org/10.1109/TITS.2023.3234512.
39. Sun K, Liu P, Li P, Liao Z. ModWaveMLP: MLP-based mode decomposition and wavelet denoising model to defeat complex structures in traffic forecasting. Proc AAAI Conf Artif Intell. 2024;38:9035–43.
40. Fang Y, Qin Y, Luo H, Zhao F, Xu B, Zeng L, Wang C. When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE), IEEE; 2023. p. 517–29.
41. Wang Y, Zhao A, Li J, Lv Z, Dong C, Li H. Multi-attribute graph convolution network for regional traffic flow prediction. Neural Process Lett. 2023;55(4):4183–209.
42. Cai P, Wang Y, Lu G. Tunable and transferable RBF model for short-term traffic forecasting. IEEE Trans Intell Transp Syst. 2018;20(11):4134–44.
43. Wang J, Chen Q. A traffic prediction model based on multiple factors. J Supercomput. 2021;77(3):2928–60.
44. Berk RA, Freedman DA. Statistical assumptions. In: Blomberg TG, Cohen S, editors. Punishment and social control. Piscataway: Transaction Publishers; 2003. p. 235.
45. Bence JR. Analysis of short time series: correcting for autocorrelation. Ecology. 1995;76(2):628–39.
46. Li L, Su X, Zhang Y, Lin Y, Li Z. Trend modeling for traffic time series analysis: an integrated study. IEEE Trans Intell Transp Syst. 2015;16(6):3430–9.
47. Zou H, Wu Y, Zhang H, Zhan Y. Short-term traffic flow prediction based on PCC-BILSTM. In: 2020 International Conference on Computer Engineering and Application (ICCEA), IEEE; 2020. p. 489–93.
48. Xue X, Jia Y, Wang S. Expressway traffic flow prediction model based on Bi-LSTM neural networks. In: IOP Conference Series: Earth and Environmental Science, vol. 587. IOP Publishing; 2020. p. 012007.

49. Hu X, Liu T, Hao X, Lin C. Attention-based Conv-LSTM and Bi-LSTM networks for large-scale traffic speed prediction. J Supercomput. 2022;78(10):12686–709.
50. Ma X, Tao Z, Wang Y, Yu H, Wang Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp Res Part C Emerg Technol. 2015;54:187–97.
51. Guo S, Lin Y, Li S, Chen Z, Wan H. Deep spatial-temporal 3d convolutional neural networks for traffic data forecasting. IEEE Trans Intell Transp Syst. 2019;20(10):3913–26.
52. Do LN, Vu HL, Vo BQ, Liu Z, Phung D. An effective spatial-temporal attention based neural network for traffic flow prediction. Transp Res Part C Emerg Technol. 2019;108:12–28.
53. Willmott CJ, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Clim Res. 2005;30(1):79–82.
54. Zafar N, Haq IU, Chughtai J-u-R, Shafiq O. Applying hybrid LSTM-GRU model based on heterogeneous data sources for traffic speed prediction in urban areas. Sensors. 2022;22(9):3348.
55. Redhu P, Kumar K, et al. Short-term traffic flow prediction based on optimized deep learning neural network: PSO-Bi-LSTM. Phys A Stat Mech Appl. 2023;625: 129001.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.