Short-Term Traffic Flow Prediction for Urban Road Sections Based on Time Series Analysis and LSTM_BILSTM Method

Changxi Ma, Guowen Dai, and Jibiao Zhou

***Abstract*— The real-time performance and accuracy of traffic flow prediction directly affect the efficiency of traffic flow guidance systems, and traffic flow prediction is a hotspot in the field of intelligent transportation. To further improve the accuracy of short-term traffic flow prediction, a short-term traffic flow prediction model based on traffic flow time series analysis, and an improved long short-term memory network (LSTM) is proposed. First, perform time series analysis on traffic flow data and perform smoothing and standardization processing to obtain a stable time series as model input data, which can improve the accuracy of model training and eliminate the impact of a wide range of feature values. Then, an improved LSTM model based on LSTM and bidirectional LSTM networks are established. Combining the advantages of sequential data and the long-term dependence of forwarding LSTM and reverse LSTM, the bidirectional long-term memory network (BILSTM) is integrated into the prediction model. The first layer of the LSTM network learns and predicts the input time series and further learns and trains through the bidirectional LSTM network to effectively overcome the large prediction errors. Finally, the performance of the proposed method is evaluated by comparing the predicted results with actual traffic data. The model that is proposed in this paper is compared with the long short-term memory network (LSTM) model and the bidirectional long-term memory network (BILSTM) model. The results demonstrate that the proposed method outperforms both compared methods in terms of accuracy and stability.**

***Index Terms*— Traffic engineering, short-term traffic flow prediction, LSTM_BILSTM method, time series analysis, urban road section.**

I. Introduction

AS THE basis and critical technology of traffic control and traffic guidance, traffic flow prediction is the core issue of intelligent transportation system research. Real-time, accurate, and scientific traffic flow prediction technology can effectively improve traffic efficiency and safety, maximize the rational use of the urban road network carrying capacity, and realize traffic flow induction, thereby alleviating traffic congestion and reducing the environmental pollution.

In the research of short-term traffic flow based on traditional statistical theory, the time series prediction method is based on the analysis of historical traffic data. It finds the change law and trend of traffic and establishes a prediction model according to the flow characteristics. The urban transportation system is a nonlinear, complex, and open giant system. For studying traffic flow control, modeling, or studying the self-organizing laws of traffic flow, it is necessary to study the traffic flow fractal problem [1]–[4]. A fractal is originally a concept in geometric topology. It is a shape in which the shape of each component is similar to the entire shape. Fractals mainly reveal the inherent regularity of the irregular shapes in nature - scale invariance [1], [5]. Mathematical fractals have infinitely nested hierarchical structures, while fractals in natural systems and social systems have only a limited number of nestings and only follow a fractal law after entering a hierarchical structure. Moreover, even in the same scale-free domain, different local conditions lead to different local characteristics; thus, the multifractal theory emerges. The main research topic of the multifractal theory is the scale distribution characteristics of the fractal system. The multifractal theory analyzes the characteristics of the original system and the inherent evolution of its dynamics [5].

Previous studies have focused on the self-similarity, long-range correlation, fractal characteristics, and complexity of traffic flow time series [6], [7]. For example, Li *et al.* [8] compare various traffic flow time series clustering methods. The Classical method is applicable and straightforward, and the results are reasonable. Wu *et al.* [9] conducted a detrending fluctuation analysis of mixed traffic flow time series. They transformed the complexity of the spatiotemporal evolution of traffic flow into the study of the relevant characteristics of the time series. The current research on fractals in traffic flow mainly focuses on calculating the feature quantities of traffic flow time series, such as the fractal dimension and the Hurst index. For example, He *et al.* [3] calculates the Hurst exponent and uses it to estimate the period of change of the traffic flow time series. Pei *et al.* [4] analyze the fractal characteristics of the traffic flow by calculating the fractal dimension. However,

Manuscript received December 26, 2019; revised July 8, 2020, September 14, 2020, and November 29, 2020; accepted January 15, 2021. This work was supported in part by the Natural Science Foundation of China under Grant 52062027, Grant 52002282, and Grant 71861023; in part by the Natural Science Foundation of Zhejiang Province under Grant LQ19E080003; in part by the Philosophy and Social Science Foundation of Zhejiang Province under Grant 21NDJC163YB; in part by the Philosophy and Social Science Program of Ningbo under Grant G20-ZX37; and in part by the Foundation of A Hundred Youth Talents Training Program of Lanzhou Jiaotong University. The Associate Editor for this article was X. Jia. *(Corresponding authors: Changxi Ma; Jibiao Zhou.)*

Changxi Ma and Guowen Dai are with School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China (e-mail: machangxi@mail.lzjtu.cn; 0218002@stu.lzjtu.edu.cn).

Jibiao Zhou is with the Department of Transportation Engineering, Tongji University, Shanghai 201804, China (e-mail: zhoujibiao@tongji.edu.cn).

Digital Object Identifier 10.1109/TITS.2021.3055258

these feature quantities can only reflect some characteristics of the system as a whole, and these feature quantities are calculated mainly for stationary time series, whereas the actual traffic flow timing often contains large trend items; hence, it is not stable. For this reason, Feng *et al.* [2] use the multiscale decomposition method to analyze the self-similarity of traffic flow time series on various scales. By calculating the generalized Hurst index, it confirmed that the time series of traffic flow speed have different correlation indexes on different scales [7], [10], [11].

The artificial intelligence methods, which are represented by fuzzy artificial neural networks, exhibit excellent performance in describing nonlinear systems. The neural network model is a network model that mimics the human brain to mathematicize neurons, to express various features of the parameters in a modeling manner, and to connect them in various ways. Neural network prediction is the adjustment of the weight parameters of the network via backward propagation by inputting a large amount of historical data to the input layer and comparing the error analysis results with the corresponding output and input until the optimal solution is reached. This prediction method uses a "black box" learning mode, does not require any empirical formula, and relies only on available historical data.

Moretti F *et al.* [12] established an urban traffic flow prediction model by integrating hybrid modeling with statistics and neural networks. Liu Y *et al.* [13] used the Heursure algorithm after wavelet decomposition to suppress the noise and to generate more stable traffic flow data and used the GA-optimization-based Elman network model to predict the stable traffic flow. Yi *et al.* [14] used the k-nearest-neighbor method to find the best smoothing factor by ranking the mean square error and predicted it using a generalized regression neural network, which yielded satisfactory results. Hu *et al.* [15] used the reconstructed vector of the chaotic phase space as the input of the RBF and used the adjacent phase space reconstruction vector as the output of the RBF to train and generate the initial weight and threshold. Therefore, a traffic flow prediction algorithm based on a chaos-radial basis (Chaos-RBF, C-RBF) neural network is proposed. Zhang *et al.* analyzed the historical traffic flow data of the adjacent road sections that are to be predicted and the critical sections that are important to the relationship. Then, the result of the analysis is used as the input of the network, which not only eliminates a large amount of secondary information but also reduces the scale of the model and reduces the large number of intermediate parameters, while preserving the primary information of the original data; hence, the prediction accuracy is improved [16]. LV *et al.* [17] apply a deep self-encoding algorithm to short-term traffic prediction. Polson N G [18] *et al.* proposed a linear model of 11 regularizations and fittings to the tanh layer sequence. The short-term traffic flow is predicted by identifying the spatiotemporal relationship between the first-level predictor and the nonlinear relationships in the other layers of the model.

The cyclic RNN is an artificial neural network with a hidden-layer-node oriented connection and a closed-loop. In contrast to other feedforward neural networks, RNNs can use internal memory cells to handle timing inputs of any length. For the standard RNN architecture, the effect of a specified input on the hidden layer unit and the network output will disappear or explode with the periodic connection of the RNN network, which is the gradient disappearance problem [19]. To overcome the problem, LSTM (long short-term memory) is an RNN network that is specially designed to give memory units the ability to determine when to remember or when to forget information, thereby solving the time series. When the problem is solved, the optimal time lag can be automatically determined, thereby demonstrating the long-term memory of historical data or samples [19]. Ma *et al.* [20] proposed a method for training LSTM cyclic neural networks using remote microwave sensor data to predict traffic speed and obtained satisfactory results. Tian *et al.* [21] discussed the performance of the LSTM cyclic neural network for predicting short-term traffic flow and compared it with several other commonly used models. Fu *et al.* [22] used the LSTM and GRU neural network models to predict short-term traffic flow and confirmed that the cyclic neural network model outperforms ARIMA. The LSTM controls the state of the transmission through the gated state. Important information is remembered for a long time, whereas unimportant information is quickly forgotten.

In contrast to the standard RNN, there is only one method of memory overlay. Yu Bing *et al.* [23] proposed a new deep learning framework, the spatiotemporal graph convolution network (STGCN), to solve the time series prediction problem in the field of transportation. Represent the problem graphically and build a model with a complete convolution structure, which makes the model training faster and with fewer parameters.

This paper combines time series analysis and an improved LSTM network to propose a brand-new model for short-term traffic flow prediction. Use time-series analysis to transform the complexity of the spatiotemporal evolution of traffic flow into a study of the relevant characteristics of time series. In the data processing stage, the stationarity of the time series is detected and normalized. For non-stationary time series, stationarization should be performed. Standardization refers to scaling the data in proportion to make it fall into a relatively small specific interval, removing the unit limitation of the data, and converting it into a dimensionless pure value. Data standardization can not only improve the speed of finding the optimal solution, but also improve the accuracy of model training and eliminate the impact of a wide range of eigenvalues. The improved LSTM network combines the traditional long-term and short-term memory network with the bidirectional long-term and short-term memory network. Traditional long-term and short-term memory networks have good processing performance for medium- and short-level time series data. Still, they are not very stable and reliable for processing longer time series. The two-way long-term and short-term memory network is more effective for processing longer time-series data, and can use future information. This has a positive effect on the processing of large-scale traffic flow data sequences.

II. Time Series Analysis

A set of random variables that are arranged in chronological order, namely, X_1, X_2, ... X_t, ..., is called a time series, which is abbreviated as $\{\mathrm{X_t}\}$. To obtain a time series, the sequence is typically preprocessed. Namely, its stationarity and pure randomness are evaluated. The test results depend on the sequence type, and various analysis methods are used [30]–[36].

In a time series $\{\mathrm{X_t}\}$, the sequence value $\mathrm{X_t}$ is not a random variable; instead, its probability distribution function is random. The integral $\int_{-\infty}^{+\infty} \mathrm{x} dF_t(x)$ exists and is used to define the mean function:

$$\mu_t = EX_t = \int_{-\infty}^{+\infty} \mathrm{x} dF_t(x) \tag{1}$$

A corresponding mean function sequence $\{\mu_t\}$ can be obtained, which reflects the average levels of the time series $\{X_t,\ t \in T\}$ at various times. Similarly, if the integral $\int_{-\infty}^{+\infty} \mathrm{x}^2 dF_t(x)$ exists, the variance function can be defined as follows:

$$\sigma_t = DX_t = E(X_t - \mu_\mathrm{t})^2 \int_{-\infty}^{+\infty} (\mathrm{x}\text{-}\mu_\mathrm{t})^2 dF_t(x) \tag{2}$$

The corresponding sequence of variance functions (σt) reflects the average degree of fluctuation of the time series around the mean for random fluctuations. For any t and s∈T, define

$$\gamma(t, s) = E(X_t - \mu_\mathrm{t})(X_s - \mu_s) \tag{3}$$

as an autocovariance function of the time series. Define

$$\rho(t, s) = \frac{\gamma(t, s)}{\sqrt{DX_t \times DX_s}} \tag{4}$$

as the autocorrelation coefficient of the time series. They reflect the degree to which the same event is relevant in two periods. Define

$$\rho(x_t, x_{t-k}) = \frac{E[(x_t - \hat{E}x_t)(x_{t-k} - \hat{E}x_{t-k})]}{E[(x_{t-k} - \hat{E}x_{t-k})]^2} \tag{5}$$

as the time series of partial autocorrelation coefficients, where x_{t-1}, ..., x_{t-k+1} have been specified, $\hat{E}$xt $=$ $E[x_t\,|x_{t-1}, \ldots, x_{t-k+1}]$, and $\hat{E}$xt-k $=$ $E[x_{t-\mathrm{k}}\,|x_{t-1}, \ldots, x_{t-k+1}]$. It reflects the extent to which x_{t-k} affects x_t given x_{t-1},...,x_{t-k+1}.

It enables the definition of the stationarity of the time series. If for any positive integers m and n, the joint distribution satisfies:

$$F_{t_1,t_2,\ldots,t_m}(x_1, x_2, \ldots x_m) = F_{t_{1+n},t_{2+n},\ldots,t_{m+n}}(x_1, x_2, \ldots x_m) \tag{6}$$

then $\{X_t\}$ is called a strictly stationary time series. For *t, s*, and $k \in T$,

$$\gamma(t, s) = \gamma(k, k + s - t) \tag{7}$$

Then, $\{X_t\}$ is called a broad stationary time series. In practical applications, it is highly challenging to obtain a joint distribution of random variables, and the characteristic statistics of the sequence are relatively easy to obtain. Therefore, wide-stationary time series with more relaxed conditions are more commonly used. The stationary time series that is considered below is a broad stationary time series.

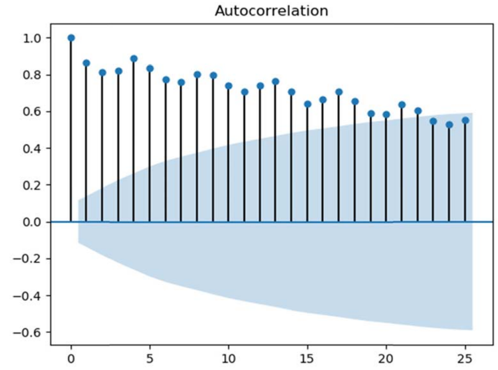

Fig. 1. Autocorrelation plot of the original time series.

The time series is an important research object of econometrics. Classical measurement theory requires that the interference term of the model satisfy the limit laws (the law of large numbers and the law of central poles). However, the time series of time series data destroys the set of random samples, and the established measurement model is unreliable. However, if the time series satisfies the stationarity requirement, the assumption of random sampling can be replaced such that the random perturbation of the model satisfies the limit rule. The parameter estimation and the statistical inference that performed at this time are correct. Namely, the smooth time series data are approximately random sample data. A time series is stationary if the mean and the standard deviation of the time series do not change with time, and all order statistics of the sequence remain unchanged. The nonstationarity of a time series is reflected in a monotonous, periodic, or stepped distribution trend from the external environment or stimulation. In the data processing stage, the stationarity of the time series must be detected, and the nonstationary time series should be smoothed.

Considering the traffic flow data of Ningbo Meteorological Road on April 5, 2018, as an example, the trend of the autocorrelation of the traffic flow data is shown in Figure 1. The autocorrelation coefficient of the stationary sequence decays rapidly. The following autocorrelation plot does not reflect this feature; hence, we have reason to believe that the sequence is not stable.

According to the previous analysis, the sequence is not stable; however, stationarity is a prerequisite for time series analysis. Therefore, we must process the unstable sequence to convert it into a stable sequence. We use the differential time method to smooth the time series.

The difference operation is an operation in which the data of the k-th order in the sequence is subtracted, and it is defined as follows:

$$\nabla_k Z_t = Z_t - Z_{t-k} \tag{8}$$

where Z is the time series, t is the number of terms of the sequence, and k is the order of the difference; for example, the first-order difference is expressed as follows:

$$\nabla Z_t = Z_t - Z_{t-1} \tag{9}$$

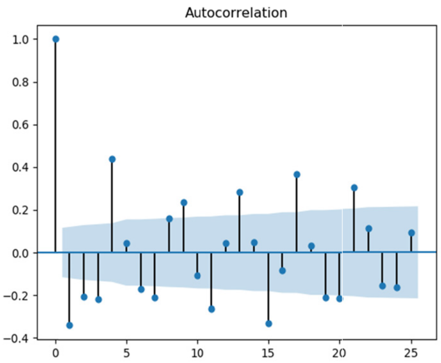

Fig. 2. Time series autocorrelation plot after smoothing.

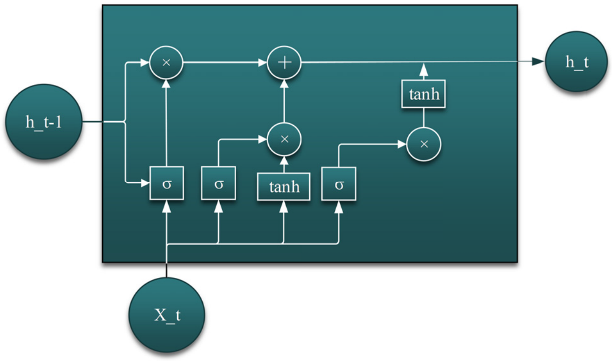

Fig. 3. LSTM neural network neuron structure.

Differencing is an essential process for smoothing the time series. Figure 2 presents the stationary time series after differential processing.

III. Model Selection

A. Long Short-Term Memory Network (LSTM)

Theoretically, an RNN can handle such long-term dependency problems. One can carefully select parameters for solving the initial form of such problems. However, in practice, an RNN cannot successfully learn this information. Therefore, LSTM was developed for solving long-term dependency problems, and LSTM avoids long-term dependency problems through deliberate design. Long-term information memory is the default behavior of LSTM in practice, not the ability to obtain information at a low cost [18], [37]–[47].

All RNNs have a chained form with a repeating neural network module. In a standard RNN, this duplicate module has a straightforward structure, such as a tanh layer.

LSTM is also such a structure; however, the duplicate modules have a different structure. In contrast to the single neural network layer of a standard RNN, there are four layers here, which interact extraordinarily, as illustrated in Figure 3.

A detailed description of the LSTM follows. The cell status (*C t*), a forgetting gate, an input gate, and an output gate included in each memory unit. These gate structures enable information to pass selectively to remove or add information to the cell state.

The memory at time t is used to store important information. Similar to a notebook, it preserves the information that we have learned previously.

The content of the previous layer of the cell state is controlled according to the previous sequence of *h t-1*, and *X t* of the sequence is used as input to the sigmoid activation function to obtain the upper layer of the cell state content, which must be removed or retained. The input is in the form of a vector, and we hope that the value of the forgetting gate output is 0 or 1, namely, that each value in the vector is completely forgotten or ultimately remembered; thus, we use the sigmoid function as the activation function since the value of this function is close to 0 or 1 in many cases (the step function cannot be used as the activation function here because its gradient at all positions is 0). Other gates use the sigmoid function for the same reason. Therefore, although the activation function can be transformed into other neural networks, it is not recommended to transform the activation function of the LSTM. The forgotten threshold layer *f t* determines which information should be discarded from the cell state at the previous moment. In the following formulas, W_f, b_f, W_i, b_i, W_o, and *bo* are the weights and offsets of each threshold layer, respectively, and σ represents the sigmoid activation function.

$$f\_t = \sigma(W_f \cdot [h\_t - 1, x\_t] + b_f) \tag{10}$$

The input of the current sequence position is processed, the information that must be updated is determined, and the cell status is updated. This process is divided into two parts: one part is to use the input gate that contains the sigmoid layer to decide which new information should be added to the cell state. Next, the selected new information must be converted into a form that can be added to the cell state. Hence, the other part is to use the tanh function to generate a new candidate vector. (LSTM's approach is to convert information into a form that can be added to the state of the cell and to use the results that were obtained in the first part to determine which new information will be added to the cell state.) The input threshold layer *i_t* determines the state of the unit that must be updated.

$$i\_t = \sigma(W_i \cdot [h\_t - 1, x\_t] + b_i) \tag{11}$$
$$C\_t' = \tanh(W_C \cdot [h\_t - 1, x\_t] + b_C) \tag{12}$$

With the forgetting gate and the input gate, we can update the cell state *C t−1* to *C t,* where *f t*×*C −1* denotes the information that has been selected for deletion and *i_t*×*C_t'* denotes the new information.

$$C\_t = f\_t * C\_t - 1 + i\_t * C\_t' \tag{13}$$

C_t is obtained from the LSTM unit state, and *C_t* and *C_t*-1 are comprehensively updated to *C_t* by using the input threshold and the forgetting threshold.

Finally, it is necessary to determine what to output based on the content that is saved by the cell state, namely, the content of the cell state that has been saved selectively. Similar to the two-part update of the input gate, the output gate also needs to use the sigmoid activation function to determine which portion of the content must be output. Then, the tanh activation

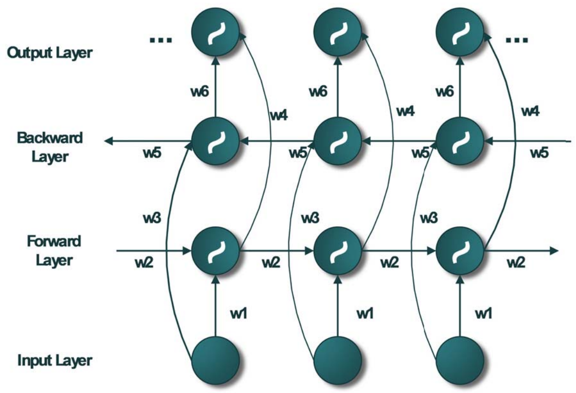

Fig. 4. BILSTM network structure.

function is used to process the content of the cell state (because each value of $C\ t$ is determined via the above calculation) that is in the range of tanh $-1\sim1$; content that is outside this range must be adjusted. The two parts are multiplied to yield the part that we want to output. The output threshold layer O_t will be filtered out based on the state of the unit.

$$O\_t = \sigma(W_o \cdot [h\_t - 1, x\_t] + b_o) \tag{14}$$

$$h\_t = O\_t * (C\_t) \tag{15}$$

B. Bidirectional Long Short-Term Memory Network (BILSTM)

Sometimes, the prediction may need to be determined by several previous inputs and several subsequent inputs, which is more accurate. Therefore, a two-way cyclic neural network is proposed, and the network structure illustrated in Figure 4. The forward layer and the backward layer connected to the output layer, which contains six shared weights, namely, w1-w6. The six weights are reused at each time step and correspond to input to the forward and backward hidden layers (w1 and w3), information flow from the hidden layers to themselves (w2 and w5), and information flow from the forward and backward hidden layers to the output layer (w4 and w6). There is no information flow between the forward and backward hidden layers, which ensures that the expanded graph is acyclic.

In the forward layer, the forward calculation conducted from time 1 to time t, and the output of the forward hidden layer at each time is obtained and saved. In the backward layer, the calculation is reversed along time t to time 1, and the output of the backward hidden layer at each time is obtained and saved. Finally, at each moment, the final output is obtained by combining the results for the corresponding time of the forward layer and the backward layer. The mathematical expressions are as follows:

$$h\_t = f(w_1 x\_t + w_2 h\_t - 1) \tag{16}$$

$$h\_t' = f(w_3 x\_t + w_5 h\_t - 1') \tag{17}$$

$$O\_t = g(w_4 h\_t + w_6 h\_t') \tag{18}$$

C. Proposed Model

The algorithm proposed in this study, which is based on the optimized LSTM model, further studies and trains the time series after the prediction of the LSTM network of the upper layer through the Bi-LSTM network. The bidirectional LSTM model combines forward LSTM and reverse LSTM to obtain the above information from the forward LSTM. The reverse LSTM acquires additional information and has the advantage of capturing data timing and long-range dependencies. Therefore, the accuracy of the prediction result can be further improved. The flow of the short-term traffic flow prediction algorithm that is based on the optimized LSTM model includes the following steps:

Step 1: Data preparation. Obtain a traffic flow data set as required by the prediction request, divide the data set and perform time series analysis.

Step 2: Use the optimized LSTM model to predict the short-term traffic flow for the segment.

Set the length of the sequence to s, so that the first s-1 number used to predict the s-th value

The first layer is the LSTM layer, the input is a time series of the form (samples, input_length, input_dim), and the output is of the form (samples, input_length, output_dim). The input and output parameters are set when the algorithm is initially running. The samples are designed to be able to train multiple samples simultaneously and to find the average gradient, which is used to update the weight. If the number of samples is equal to 1, then this update method is called stochastic gradient descent (SGD). Input_length is the number of consecutively entered data items; LSTM considers each input data item to be associated with the previous data item. Input_dim is the input dimension, which is specified when the layer is used as the model's first layer. The first layer of LSTM will return the results one by one to the next neuron in this layer. Since we want to input the output information to the next layer of LSTM as training data, we also pass the information to the next layer as input data.

The second layer is the Bi-LSTM layer, and the input is of the form (samples, input_length, input_dim). At this time, input_dim is equal to output_dim of the first LSTM layer, and the output is of the form (samples, output_dim). The standard LSTM processes sequences in chronological order, while considering historical information and ignoring future information. Therefore, the forward and reverse traffic flow information is extracted using the two-way LSTM network. The forward recursion layer starts from the first moment of the prediction period, and the backward recursion layer starts from the last moment.

The third layer is the LSTM layer, and the input is the output of the Bi-LSTM layer. The output of this layer is passed to the full-link dense layer of the next layer. Then, the true value in the dense layer is used for loss calculation and optimization.

The fourth layer is a fully connected(dense) layer. The training error is minimized via the stochastic gradient descent method (SGD). Since the time series after the smoothing process contains positive and negative values, select tanh as the activation function.

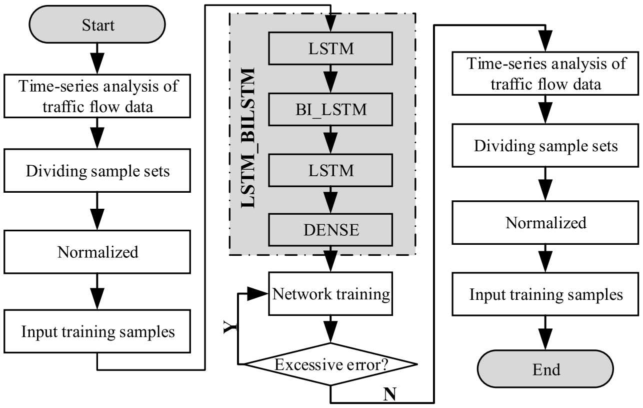

Fig. 5. Traffic flow prediction framework.

Step 3: Conduct error analysis between the predicted traffic flow and the real traffic flow, and calculate the root mean squared error (RMSE) of the predicted value and the true value. Then, adjust the parameters of the network, which include the number of neurons in each layer of the neural network, the learning rate, the training batch size, and the training data length, to minimize the loss function value.

Step 4: After conducting multiple rounds of training, the minimum value of RMSE obtained, namely, after the error function has optimized, the traffic flow data in the predicted period must be output.

Based on the above analysis, the proposed traffic flow prediction framework illustrated in Figure 5. Input the reconstructed time series for model training. According to the experimental results, different time-sequences of the same distribution characteristic have different effects on the activation function; hence, the function must be continuously adjusted according to the prediction performance during the training phase. Since the time series after the smoothing process contains positive and negative values, tanh is selected as the activation function, and to be consistent with its output value range, the time series is scaled down to [−1, 1] to realize the optimal prediction performance.

IV. Experiment and Result Analysis

A. Data Source

The data selected for this test is the traffic flow data of Ningbo Meteorological Road on the weekends from March to April 2018, for a total of 40 days. The road shown in Figure 6. The data sampling interval is 5 min, and 288 samples collected per day.

In the experimental process, the traffic data of the first 35 days (10080 sampling intervals) of the collected data used as the training data of the model, and the traffic data of the next 5 days (1440 sampling intervals) used as the test sample. A total of 10080 data included verification data Training samples and 1440 test samples.

B. Model Training

TensorFlow is Google's second-generation software library for digital computing, which is based on the data flow graph

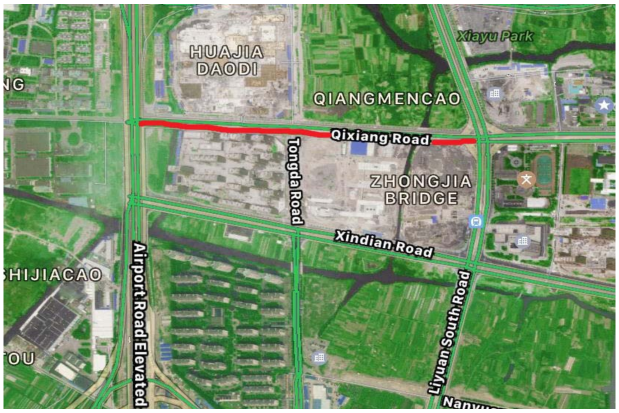

Fig. 6. Segments selected for experimental data.

TABLE I
Optimized LSTM Models at Various Depths

depth	Model structure
1	Input→LSTM→Dropout→BILSTM→Dense→Output
2	Input→LSTM→Dropout→BILSTM→Dropout→Dense→Output
3	Input→LSTM→BILSTM→Dropout→LSTM→Dropout→Dense→Output
4	Input→LSTM→Dropout→BILSTM→Dropout→LSTM→Dropout→Dense→Output
5	Input→LSTM→Dropout→BILSTM→Dropout→BILSTM→ LSTM→Dropout→Dense→Output

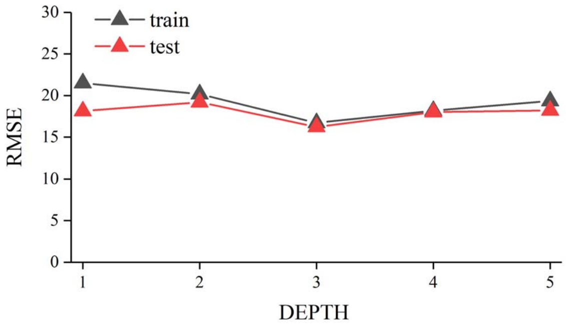

Fig. 7. RMSE for Different Depth Models.

processing framework, for realizing data operations and interaction. Under the guidance of the above traffic flow prediction framework, this paper selects the Keras high-level neural network API in TensorFlow for completing the model construction and training in the Python development environment. The error levels of models of various depths and structures (Table I) were compared. As shown in Figure 7, when the depth of the model reaches 3 layers, the training and test performances are optimal; as the depth of the model increases further, the error increases slightly.

Through the comparative experiment of each layer model, the 3-layer LSTM_BILSTM model shown in Figure 8 is finally adopted; the input and output dimensions and other parameters of each layer are specified in the figure. The dense layer is a fully connected layer, the optimizer is the Adam optimizer,

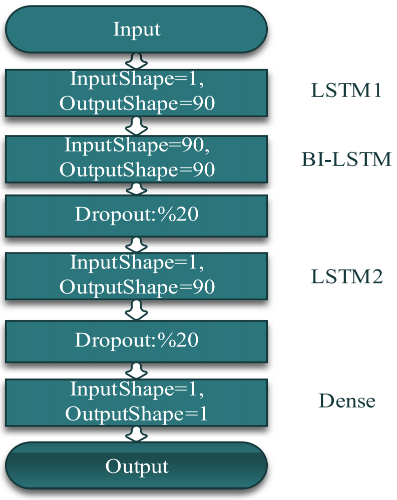

Fig. 8. LSTM_BILSTM model.

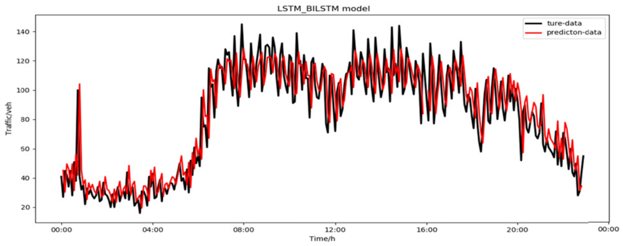

Fig. 9. LSTM_BILSTM model prediction results.

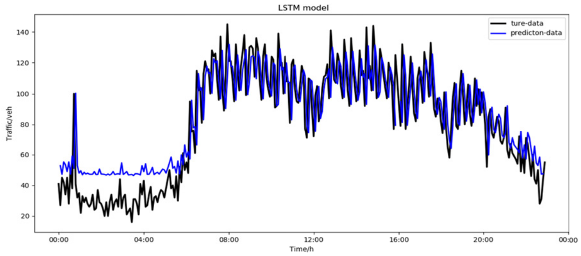

Fig. 10. LSTM model prediction results.

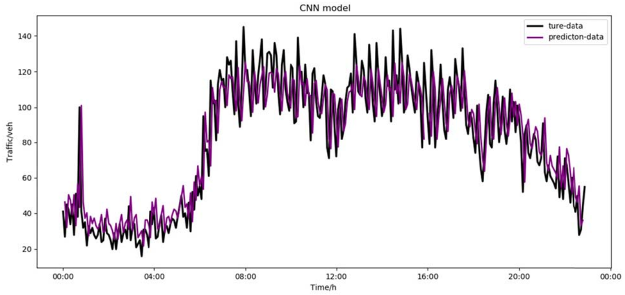

Fig. 11. CNN model prediction results.

the activation function is tanh, and the loss function is MSE. The experimental results demonstrated that with the increase of the number of epochs, the training time increases, and the training error of the model decreases; however, if the number of epochs increases past a threshold value, overfitting may occur. In this paper, the number of epochs is set to 50. The neural network tends to fall into the local minimum during the training process. A simple method is to introduce a dropout layer to enable all neurons to be further trained by randomly suppressing the outputs of some neurons. The size of the dropout parameter should be negatively correlated with the amount of information in the dataset and positively correlated with the network size; after testing, when the probability of abandonment is set to 0.2, the training requirement can be satisfied.

C. Selection of the Evaluation Function

In this paper, MAE, RMSE, and decision coefficient R2 are selected as the accuracy evaluation indicators for comparing these prediction algorithms. The decision coefficient R2, is also known as goodness of fit, can be used to evaluate the degree of fit of a model [43], [48]–[55]. The formula is as follows:

$$R_2 = 1 - \frac{\sum_{i=0}^{n_{samples}-1} (y * - y)^2}{\sum_{i=0}^{n_{samples}-1} (y - \bar{y})} \tag{19}$$

MAE is the average absolute error, and the formula is as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y * - y| \tag{20}$$

RMSE is the root mean square error. The root means the square error is also called the standard error. It is the square root of the ratio of the square of the deviation between the observed value and the actual value, divided by the number of observations. The root means square error used to measure the deviation between the observed value and the actual value. The standard error is highly sensitive to extraordinarily large or small errors in a set of measurements. As follows,

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y * - y)^2} \tag{21}$$

In the above three formulas, n is the sample size, y∗ is the predicted value, and y is the actual value.

D. Comparison and Analysis of the Results

To analyze the prediction effect of the proposed methods, this paper compares the four methods of LSTM, BILSTM, CNN, KNN, ConvLSTM, and LSTM_BILSTM. Among them, LSTM and BILSTM selected in this paper have 11 hidden units, and the activation functions are all tanh.

The prediction performance of the LSTM network presented in Figure 10. Although the gradient problem and the long-term dependency of RNN have overcome in LSTM, its performance is not sufficient. It may handle sequences that are of short to medium order; however, it fails in processing longer sequences. The two-way LSTM performs better in dealing with longer sequence problems. The two-way LSTM utilizes future information and can outperform directional LSTM in

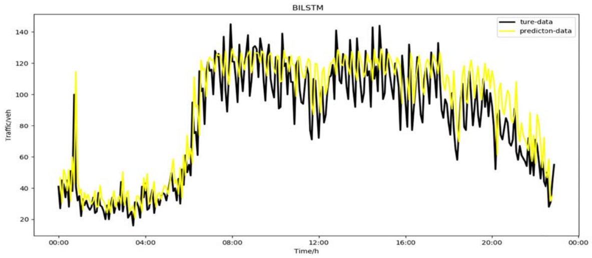

Fig. 12. BILSTM model prediction results.

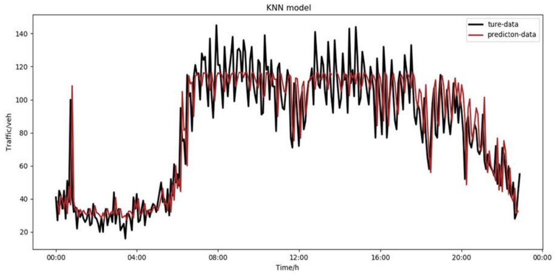

Fig. 13. KNN model prediction results.

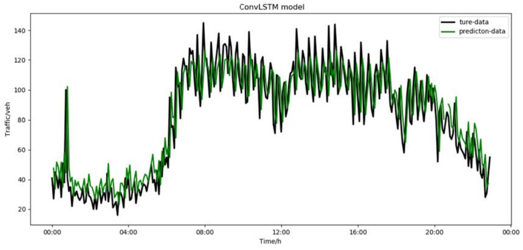

Fig. 14. ConvLSTM model prediction results.

TABLE II
COMPARISON OF THE PREDICTION RESULTS OF THREE MODELS

Model	MAE	RMSE	R2
LSTM	14.31	20.00	0.85
BILSTM	13.29	18.18	0.83
CNN	12.96	17.48	0.85
LSTM_BILSTM	12.63	16.72	0.86
KNN	26.58	34.25	0.91
ConvLSTM	13.83	19.22	0.82

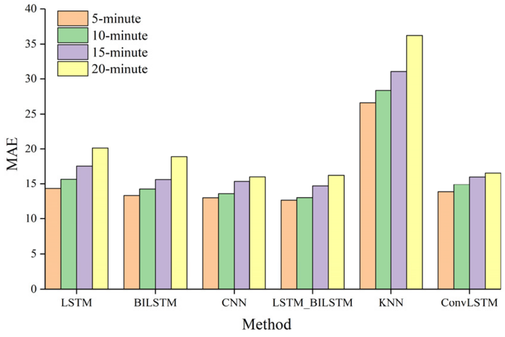

Fig. 15. MAE values of various models at different prediction steps.

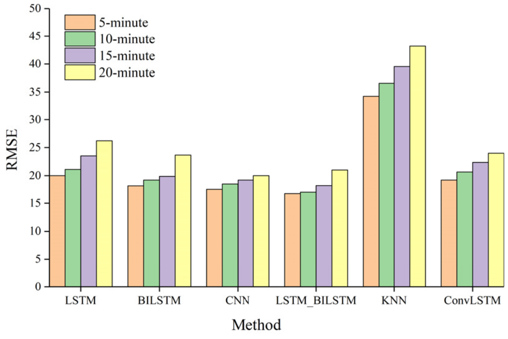

Fig. 16. RMSE values of various models at different prediction steps.

various text classification and sequence prediction problems. Compared to LSTM, BILSTM adds an inverse calculation and uses the data in the forward direction to calculate the final output. The performance of BILSTM in short-term traffic flow prediction is presented in Figure 12. The CNN model is now widely used in the transportation field. The ability of the convolutional layer and the pooling layer to extract in-depth features can realize the extraction of complex image features. But its performance in time series such as traffic flow is not very superior. The performance of CNN in short-term traffic flow prediction is shown in Figure 11.

According to Table II, BILSTM has an MAE value of 13.29, which is lower than LSTM's value of 14.31. Its RMSE value is 18.18, which is much lower than LSTM's value of 20.00. LSTM and BILSTM differed little in terms of the decision coefficient R2; the values were 0.85 and 0.83, respectively. From this, we conclude that in this study, BILSTM outperforms LSTM in short-term traffic flow prediction. Besides, according to the table, the LSTM_BILSTM model, after time series analysis substantially outperformed BILSTM and LSTM in short-term traffic flow prediction. The MAE value of LSTM_BILSTM is 12.63, which is substantially lower than the values of 14.31 for LSTM and 13.29 for BILSTM. The RMSE value of LSTM_BILSTM is only 16.72, which is lower than those of LSTM and BILSTM. The CNN's MAE and RMSE values are 12.96 and 17.48, respectively, and its performance in traffic flow prediction is better than the ordinary LSTM model, which may be related to the CNN model's good feature extraction ability. The ConvLSTM model based on CNN and LSTM uses the convolution characteristics of CNN and the characteristics of LSTM time series prediction, which performed well in this study. KNN performed relatively poorly in this study, and its MAE and RMSE values were 26.58 and 34.25, respectively. The performance of the four models on the decision coefficient R2 is basically the same, and all of them have good fitting goodness. Through the comparative analysis of LSTM, BILSTM, CNN, KNN, ConvLSTM and LSTM_BILSTM models, we found that the prediction performance of the LSTM_BILSTM model is better than the other three models, which proves that the model that is proposed in this paper is effective and feasible.

In order to further test the applicability of the LSTM_BILSTM model, this paper uses it to conduct a multi-step prediction experiment to predict the traffic flow

TABLE III
COMPARISON OF THE TWO CASES

model	MAE	RMSE	R2	MAPE (%)
Case1	12.63	16.72	0.86	6.25
Case2	13.51	18.83	0.94	6.16

of lane sections in the future at 5 min, 10 min, 15 min and 20 min. As shown in Fig. 15 and Fig. 16, as the prediction step size increases, the prediction accuracy of each model decreases, but the prediction error of the LSTM_BILSTM model is the smallest at steps of 5 min and 10 min. And as the prediction step size increases, the LSTM_BILSTM model is also the best in terms of stability. In summary, the LSTM_BILSTM hybrid prediction model has good applicability, and has good accuracy and stability in multi-step prediction.

E. Comparative Experiment

In addition to the Ningbo dataset, this paper also uses open data to conduct comparative experiments on the models proposed in this paper. In case 2, we used PeMS data provided by UCB for testing. PeMS data is mainly data from California's fixed detectors (flow rate, speed, time occupancy). This paper selects the traffic flow data of the 101 highways to the south and the area code is 7. This paper uses the typical 5min prediction value of short-term traffic flow as an interval, and uses the traffic data from May 18, 2019 to November 18, 2019 (except weekends) as the basic data set for short-term traffic flow prediction. The traffic flow data of the first 90 days of the collected data is used as the training data of the model, and the last 10 days is used as the test sample. The final test results are shown in Fig. 9. In the comparative experiment, we added the MAPE evaluation index. MAPE is the average of the absolute value of relative percentage error. It can be used to evaluate the prediction results of a model. The calculation formula is as follows:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y* - y}{y} \right| \tag{22}$$

Table III is a comparison between this case and the Ningbo case.

It can be seen from Table III that the prediction efficiency of the LSTM_BILSTM model is slightly reduced when the amount of data is larger, but the goodness of fit of the model is improved. This shows that the model has better performance in different scenarios.

V. CONCLUSION AND PROSPECTS

This paper proposes a short-term traffic flow prediction model for urban sections that is based on time series analysis and optimization of an LSTM. In the experiments, the following steps are conducted:

1) A time-series analysis of the traffic flow data is performed, and the unstable time series is smoothed to obtain a smoothed time series.

2) The model is constructed, and the error levels of models with various depths and structures are compared. Then, a 3-layer LSTM_BILSTM model with two LSTM layers and one BILSTM layer is selected.

3) The model that is proposed in this paper is trained and the prediction is completed. The performance of the proposed method is evaluated via a comparison of its results with the prediction results of the other two models.

This paper has several shortcomings: It only considers the urban road traffic flow under ideal conditions, and it does not consider other influencing factors. It only considers the traffic flow on a single road segment and workdays, namely, from Monday to Friday, the traffic flow on weekends is not predicted. A future research direction is to predict the traffic flow on weekend days and compare it with the traffic flow on workdays.

REFERENCES

[1] G. G. He, S. F. Ma, and W. D. Feng, "Preliminary study of fractals of traffic flow," *China J. Highway Transp.*, vol. 15, no. 4, pp. 82–85, 2002.

[2] W. D. Feng, C. Jian, and G. G. He, "Fractal research in traffic flow," *High Technol. Lett.*, vol. 13, no. 6, pp. 59–65, 2003.

[3] G. He and W. Feng, "Study on long-term dependence of urban traffic flow based on rescaled range analysis," *J. Syst. Eng.*, vol. 49, no. 2, pp. 166–169, 2004.

[4] Y. L. Pei and H. P. Li, "Research on fractal dimensions of traffic flow time series on expressway," *J. Highway Transp. Res. Develop.*, vol. 23, no. 2, pp. 115–119, 2006.

[5] G. X. Du and X. X. Ning, "Multifractal analysis on Shanghai stock market systems," *Eng. Theory Pract.*, vol. 27, no. 10, pp. 40–47, 2007.

[6] L. W. Lan, J.-B. Sheu, and Y.-S. Huang, "Investigation of temporal freeway traffic patterns in reconstructed state spaces," *Transp. Res. C, Emerg. Technol.*, vol. 16, no. 1, pp. 116–136, Feb. 2008.

[7] P. Shang, Y. Lu, and S. Kamae, "Detecting long-range correlations of traffic time series with multifractal detrended fluctuation analysis," *Chaos, Solitons Fractals*, vol. 36, no. 1, pp. 82–90, Apr. 2008.

[8] Q. Li, "On clustering methods of traffic flow time series and applications," School Electron. Inf. Eng., Beijing Jiaotong Univ., Beijing, China, Tech. Rep. U491.112.1000409120316, 2011.

[9] J. J. Wu, S. Y. Xu, and H. J. Sun, "Detrended fluctuation analysis of time series in mixed traffic flow," *Acta Phys. Sinica*, vol. 60, no. 1, pp. 800–806, 2011.

[10] X. Li and P. Shang, "Multifractal classification of road traffic flows," *Chaos, Solitons Fractals*, vol. 31, no. 5, pp. 1089–1094, Mar. 2007.

[11] W.-D. Chen and T. Zhang, "System analysis of multifractal structure in Expressway's traffic flux," in *Proc. Int. Conf. Transp. Eng.*, Jul. 2007, pp. 4092–4097.

[12] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, Nov. 2015.

[13] Y. Liu and K. Zhang, "Short-term traffic flow forecasting based on wavelet denoising and GA-Elman neural network," *Technol. Economy Areas Commun.*, vol. 17, no. 6, p. 19, 2015.

[14] L. Z. Yi, C. Zhang, and Z. Pei, "A modified general regression neural network with its application in traffic prediction," *J. Shandong Univ. (Eng. Sci.)*, vol. 43, no. 1, pp. 9–14, 2013.

[15] J. X. Hu, Y. Chen, and L. D. Zhang, "Traffic flow prediction algorithm study via chaos RBF network," *J. Univ. Jinan*, vol. 26, no. 2, pp. 152–155, 2012.

[16] X. L. Zhang and G. G. He, "The forecasting approach for short-term traffic flow based on principal component analysis and combined NN," *Syst. Eng. Theory Pract.*, vol. 27, no. 8, pp. 467–475, 2007.

[17] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[18] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[19] L. Li, S. He, and J. Zhang, "Online short-term traffic volume forecasting under the influences of time and space," *J. Transp. Syst. Eng. Inf.*, vol. 16, no. 5, pp. 165–171, 2016.

[20] J. Kolen and S. Kremer, *Gradient Flow in Recurrent Nets: The Difficulty of Learning Long Termdependencies*. Hoboken, NJ, USA: Wiley-IEEE Press, 2009.

[21] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[22] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Dec. 2015, pp. 153–158.

[23] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[24] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, "Traffic speed prediction under non-recurrent congestion: Based on LSTM method and BeiDou navigation satellite system data," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 70–81, Feb. 2019.

[25] E. Habler and A. Shabtai, "Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages," *Comput. Secur.*, vol. 78, pp. 155–173, Sep. 2018.

[26] N. C. Petersen, F. Rodrigues, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Syst. Appl.*, vol. 120, pp. 426–435, Apr. 2019.

[27] H. T. Nguyen and M. L. Nguyen, "Multilingual opinion mining on YouTube—A convolutional N-gram BiLSTM word embedding," *Inf. Process. Manage.*, vol. 54, no. 3, pp. 451–462, May 2018.

[28] K. Xu, Z. Yang, P. Kang, Q. Wang, and W. Liu, "Document-level attention-based BiLSTM-CRF incorporating disease dictionary for disease named entity recognition," *Comput. Biol. Med.*, vol. 108, pp. 122–132, May 2019.

[29] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017.

[30] G. N. Derry, E. Mullen, and K. A. Marcelino, "Effects of dynamical time scale mismatch on time series analysis using event intervals," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 80, no. 1, pp. 1007–5704, 2020.

[31] E. Farg, M. N. Ramadan, and S. M. Arafat, "Classification of some strategic crops in Egypt using multi remotely sensing sensors and time series analysis," *Egyptian J. Remote Sens. Space Sci.*, vol. 22, no. 3, pp. 263–270, 2019, doi: 10.1016/j.ejrs.2019.07.002.

[32] A. Talaei-Khoei and J. M. Wilson, "Using time-series analysis to predict disease counts with structural trend changes," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 674–686, May 2019.

[33] B. Tejedor, M. Casals, M. Macarulla, and A. Giretti, "U-value time series analyses: Evaluating the feasibility of *in-situ* short-lasting IRT tests for heavy multi-leaf walls," *Building Environ.*, vol. 159, Jul. 2019, Art. no. 106123.

[34] L. Kalo, H. J. Pant, M. C. Cassanello, and R. K. Upadhyay, "Time series analysis of a binary gas-solid conical fluidized bed using radioactive particle tracking (RPT) technique data," *Chem. Eng. J.*, vol. 377, Dec. 2019, Art. no. 119807.

[35] A. Ji and P. Shang, "Analysis of financial time series through forbidden patterns," *Phys. A, Stat. Mech. Appl.*, vol. 534, Nov. 2019, Art. no. 122038.

[36] L. Li, J. Zhang, Y. Wang, and B. Ran, "Missing value imputation for traffic-related time series data based on a multi-view learning method," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2933–2943, Aug. 2019.

[37] J. Zhao *et al.*, "Truck traffic speed prediction under non-recurrent congestion: Based on optimized deep learning algorithms and GPS data," *IEEE Access*, vol. 7, pp. 9116–9127, 2019.

[38] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges," *Veh. Commun.*, vol. 20, Dec. 2019, Art. no. 100184.

[39] G. Dai, C. Ma, and X. Xu, "Short-term traffic flow prediction method for urban road sections based on space–time analysis and GRU," *IEEE Access*, vol. 7, pp. 143025–143035, 2019.

[40] Y. Matsumoto and K. Nishio, "Reinforcement learning of driver receiving traffic signal information for passing through signalized intersection at arterial road," *Transp. Res. Procedia*, vol. 37, no. 12, pp. 449–456, 2019.

[41] A. Campbell, A. Both, and Q. Sun, "Detecting and mapping traffic signs from Google street view images using deep learning and GIS," *Comput., Environ. Urban Syst.*, vol. 77, Sep. 2019, Art. no. 101350.

[42] S. Dabiri and K. Heaslip, "Developing a Twitter-based traffic event detection model using deep learning architectures," *Expert Syst. Appl.*, vol. 118, pp. 425–439, Mar. 2019.

[43] C. Ma and R. He, "Green wave traffic control system optimization based on adaptive genetic-artificial fish swarm algorithm," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 2073–2083, Jul. 2019.

[44] C. El Hatri and J. Boumhidi, "Fuzzy deep learning based urban traffic incident detection," *Cognit. Syst. Res.*, vol. 50, pp. 206–213, Aug. 2018.

[45] S. Lee, D. Ngoduy, and M. Keyvan-Ekbatani, "Integrated deep learning and stochastic car-following model for traffic dynamics on multi-lane freeways," *Transp. Res. C, Emerg. Technol.*, vol. 106, pp. 360–377, Sep. 2019.

[46] J. Kurniawan, S. G. S. Syahra, C. K. Dewa, and A. Afiahayati, "Traffic congestion detection: Learning from CCTV monitoring images using convolutional neural network," *Procedia Comput. Sci.*, vol. 144, pp. 291–297, 2018.

[47] J. Zhao, Y. Gao, J. Tang, L. Zhu, and J. Ma, "Highway travel time prediction using sparse tensor completion tactics and K-nearest neighbor pattern matching method," *J. Adv. Transp.*, vol. 2018, pp. 1–16, Mar. 2018.

[48] F. Chen, M. Song, X. Ma, and X. Zhu, "Assess the impacts of different autonomous trucks' lateral control modes on asphalt pavement performance," *Transp. Res. C, Emerg. Technol.*, vol. 103, pp. 17–29, Jun. 2019.

[49] X. Xu, Ž. Šaric, F. Zhu, and D. Babic, "Accident severity levels and traffic signs interactions in state roads: A seemingly unrelated regression model in unbalanced panel data approach," *Accident Anal. Prevention*, vol. 120, pp. 122–129, Nov. 2018.

[50] W. Wu, R. Liu, W. Jin, and C. Ma, "Stochastic bus schedule coordination considering demand assignment and rerouting of passengers," *Transp. Res. B, Methodol.*, vol. 121, pp. 275–303, Mar. 2019.

[51] C. X. Ma *et al.*, "Bus-priority intersection signal control system based on wireless sensor network and improved particle swarm optimization algorithm," *Sensor Lett.*, vol. 10, no. 8, pp. 1823–1829, Dec. 2012.

[52] H. Pu, Y. Li, C. Ma, and H. Mu, "Analysis of the projective synchronization of the urban public transportation super network," *Adv. Mech. Eng.*, vol. 9, no. 6, pp. 1–8, 2017.

[53] C. T. Jiang, H. X. Ge, and R. J. Cheng, "Mean-field flow difference model with consideration of on-ramp and off-ramp," *Phys. A, Stat. Mech. Appl.*, vol. 513, no. 1, pp. 465–467, 2019.

[54] R. Cheng, H. Ge, and J. Wang, "An extended continuum model accounting for the driver's timid and aggressive attributions," *Phys. Lett. A*, vol. 381, no. 15, pp. 1302–1312, Apr. 2017.

[55] Y. Sun, H. Ge, and R. Cheng, "An extended car-following model considering driver's memory and average speed of preceding vehicles with control strategy," *Phys. A, Stat. Mech. Appl.*, vol. 521, pp. 752–761, May 2019.

Changxi Ma received the B.S. degree in traffic engineering from the Huazhong University of Science and Technology in 2002 and the Ph.D. degree in transportation planning and management from Lanzhou Jiaotong University in 2013. He is currently a Professor with Lanzhou Jiaotong University. He is the author of three books, and more than 100 articles. His research interests include ITS, traffic safety, and hazardous materials transportation.

Guowen Dai received the B.E. degree from Lanzhou Jiaotong University in 2018. He is currently pursuing the master's degree in transportation planning and management with the School of Traffic and Transportation, Lanzhou Jiaotong University, China. His research interests include short-term traffic flow forecast and data mining.

Jibiao Zhou received the Ph.D. degree in transportation planning and management from Chang'an University in 2014. He worked as a Visiting Scholar at Bauhaus-Universität Weimar, Germany, from 2015 to 2016. He is currently a Postdoctoral Fellow with Tongji University, studying microscopic traffic flow modeling. His research interests include traffic safety, pedestrian evacuation, and data mining.