Received: 6 October 2021 | Revised: 8 October 2023 | Accepted: 27 October 2023

IET Intelligent Transport Systems

DOI: 10.1049/itr2.12448

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Lag-related noise shrinkage stacked LSTM network for short-term traffic flow forecasting

Kai Li[1,2] | **Weihua Bai**[2] | **Shaowei Huang**[2] | **Guanru Tan**[3] | **Teng Zhou**[4] | **Keqin Li**[5]

[1]School of Computer Science and Technology, Hainan University, Haikou, China

[2]School of Computer Science, Zhaoqing University, Zhaoqing, China

[3]School of Robotics, Hunan University, Changsha, China

[4]School of Cyberspace Security, Hainan University, Haikou, China

[5]Department of Computer Science, State University of New York, New Paltz, New York, USA

Correspondence
Weihua Bai and Shaowei Huang, School of Computer Science, Zhaoqing University, Zhaoqing, China.
Email: baiweihua@zqu.edu.cn and shw.huang@qq.com

Teng Zhou, School of Cyberspace Security, Hainan University, Haikou, China.
Email: teng.zhou@hainanu.edu.cn

Funding information
Teaching Reform Project of University Public Computer Course of Guangdong Province, Grant/Award Number: 2021-GGJGJ-012; Zhaoqing University technology project 2023, Grant/Award Number: QN202346; the 2023 Guangdong Scientific Research Platform and Projects for the Higher-educational Institution and Education Science Planning Scheme, Grant/Award Number: 2023ZDZX3041; Guangdong Provincial Science and Technology Plan Project, Grant/Award Number: STKJ202209003; Technology Plan Project of Zhaoqing, Grant/Award Number: 2021SN12; Key Scientific Research Project of Universities in Guangdong Province, Grant/Award Numbers: 2020ZDZX3020, 2020ZDZX3028, 2022ZDZX1007; Laboratory Management Committee of Guangdong Institute of Higher Education, Grant/Award Number: GDJ20220360; Natural Science Foundation of Guangdong Province, Grant/Award Numbers: 2021A1515012302, 2022A1515011590, 2022A1515011978

Abstract

For the transport networks only equipped with sparse or isolated detectors, short-term traffic flow forecasting faces the following problems: (1) there are only temporal information and no spatial information; (2) the noises in the traffic flow significantly affect the forecasting performance. In this paper, a lag-related noise shrinkage stacked long short-term memory (LSTM) network is proposed for the traffic flow forecasting task only related to temporal information. To extract effective temporal features, the optimal time lags are selected in the traffic flow and converted into lag-related multi-dimensional data. Then, a discrete wavelet threshold denoising shrinkage algorithm is designed to filter the noises to construct a more reliable training set. A multi-level stacked LSTM network is employed to learn the features of the training set to map the past traffic flow to the future flow. Four benchmark datasets are to evaluate the forecasting performance by extensive experiments. The comparison with the state-of-the-art models demonstrates an average improvement of 7.28% in MAPE and 6.02% in RMSE. In addition, the proposed method has been applied in the Guilin Travel Network Bus Intelligent Dispatching System. It improves the utilization of the vehicles and reduces operating costs.

1 | INTRODUCTION

Traffic flow including traffic flow rate, density, and average speed, is one of the key factors to evaluate the state of road traffic [1]. Accurate and timely traffic flow information is critical for the successful deployment of intelligent transportation systems, which provide reliable traffic information for logistics departments, commercial organizations, tourism service

organizations, and government management agencies [2, 3]. Short-term traffic flow forecasting is on the micro-level. Subtle changes in traffic flow often have a great impact on future traffic flow. Due to the inherent randomness of traffic flow and the external noise [4], such as accidents, climate, manual traffic control, or detectors malfunction, accurately identifying effective changes in traffic flow and filtering noise are difficult if not impossible. Four typical highway datasets of Amsterdam are employed in the study. The highways with different types of noise and conditions are representative, including those with high traffic flow change rates, those prone to traffic accidents, those with complex road conditions, and those with relatively mild traffic conditions. The details are described in Section 3.1.

In the literature, a variety of short-term traffic flow forecasting methods have been proposed. The first type is based on conventional time series models, such as moving average model, regressive model [5–7], autoregressive integrated moving average model [8], Bayesian network [9], and hidden Markova model [10]. Furthermore, some variants by the combinations of these models have been applied for short-term traffic flow forecasting, such as the Kalman autoregressive integrated moving average model (KARIMA) [11], the autoregressive integrated moving average with an explanatory variable model (ARIMAX) [12], and the seasonal autoregressive integrated moving average model (SARIMA) [13]. These methods perform well in stationary traffic flow data. However, due to unexpected incidents, the actual traffic flow data manifest complicated behaviors that have nonlinear parts and are not always stationary.

The emerging machine learning methods have been employed for short-term traffic flow forecasting, such as deep belief network [14], fuzzy logic [15], Kalman filter [16, 17], ensemble learning [18], support vector regression [19], k-nearest neighbor [20], and extreme learning machines [21]. Machine learning methods require a large amount of sample data and sufficient training effort to establish the mapping function [22, 23]. These methods show more advanced predictive capabilities than classical statistical models [24]. In particular, deep neural network methods such as stacked autoencoder [25, 26], recurrent neural network (RNN) [27], and generative adversarial nets [28] have greatly improved the accuracy of forecasting. Recent studies use temporal and spatial information of traffic flow or multidimensional information to further improve forecasting performance. For example, Liu et al. [29] combine convolutional neural network and long short-term memory (LSTM) to develop a deep learning-based forecasting model by extracting the temporal and spatial dependencies of traffic flow. Lu et al. [30] employ a novel multi-diffusion convolution block and a stacked LSTM block to learn the spatial-temporal dependencies of intricate traffic data. Ma et al. [31] apply multiple features for multi-lane short-term traffic forecasting with a convolutional LSTM neural network architecture. However, it is difficult to acquire spatial information in transportation networks that only have sparse or isolated traffic detectors. In these cases, it is challenging to make the forecasting task only have temporal traffic information and without spatial traffic information. These studies aim to improve the accuracy of short-term traffic flow forecasting only based on temporal traffic flow sequence.

Various RNNs [32, 33] have been applied in this task. However, traditional RNN has the key disadvantage of gradient vanishing or exploding. The LSTM network was proposed to overcome this issue [20, 34]. Nevertheless, the existing LSTM network still has a shortcoming in terms of short-term traffic forecasting. The mean absolute error or mean square error loss is commonly used in conventional LSTMs, such as [35], which is based on the assumption that the deviation between the estimated value and the true value obeys a Gaussian distribution. Therefore, how to make the residuals of the data obey the Gaussian distribution is the key to advancing the prediction accuracy of traditional LSTM. However, emergencies and accidents can bring impulse interference and outliers to traffic flow, and the Gaussian assumption of prediction residuals does not always hold [36]. Conventional LSTM may deteriorate severely under non-Gaussian conditions [37]. In this regard, a more robust criterion is needed to prevent the interference of irregular samples. Furthermore, relying solely on LSTM networks to extract the features of the traffic flow data is probably biased [38]. Therefore, two critical questions need to be answered to improve the forecasting accuracy. The first one is what reasonable time lags is for a specific road segmentation by fully understanding the time correlation of the traffic flow. The second one is how to accurately identify and filter the noises from the traffic flow sequence. This research aims to propose a framework for the combination of statistical methods and deep learning methods to raise traffic flow forecasting accuracy by solving these two critical questions.

The contributions of this research are summarized as follows.

- The study proposes to forecast the short-term traffic flow by a multi-level stacked LSTM enhanced by data preprocessing.
- The study designs a lag-related method to effectively identify time lags for the traffic flow, and explore the optimal level of the discrete wavelet threshold denoising shrinkage algorithm to filter the noises in traffic flow sequences.
- Extensive experiments show the proposed method outperforms the state-of-the-art models by 7.28% in MAPE and 6.02% in RMSE. Furthermore, the proposed method applies to a real-world bus intelligent dispatching system in Guilin, China, and improves the utilization of the vehicles.

2 | METHODOLOGY

To solve the two critical questions above, a lag-related noise shrinkage stacked long short-term memory (LNS-SLSTM) network is proposed for the traffic flow forecasting task. First, a data preprocessing block is employed to extract effective information and filter noises simultaneously. The series correlation from the traffic flow sequence is uncovered by autocorrelation and partial autocorrelation techniques [39]. Then, the noises in the traffic flow are identified and filtered by a discrete wavelet threshold denoising shrinkage algorithm. The different levels

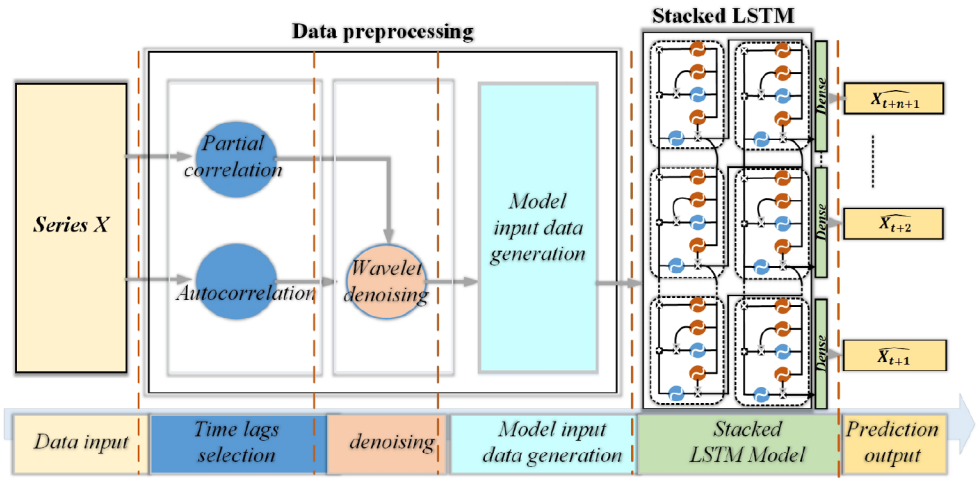

FIGURE 1 The workflow of the LNS-SLSTM.

of wavelet denoising are explored to determine the optimal one. After that, a multi-level stacked LSTM network is designed to learn the denoised data.

Four benchmark datasets are employed to evaluate the performance of the proposed methods through extensive experiments. The forecasting performance is compared to the frequently used models in the published papers in recent years. The results demonstrate the effectiveness and superiority of the proposed model for short-term traffic flow forecasting.

2.1 | Overview

The traffic flow recorded by n detectors on the highway at time t is denoted as $X_t \in \mathbb{R}^n$. The traffic flow data in the past T time intervals are $S_t = [X_{t-T+1}, \dots, X_t]$. The task of short-term traffic flow forecasting is to build a mapping function to predict the traffic flow at the next time interval through a data sequence S_t, as Equation 1.

$$S_t = [X_{t-T+1}, \dots, X_t] \xrightarrow{f} \hat{X}_{t+1}. \tag{1}$$

In traffic flow series S_t, the general name of all historical time traffic flows used in traffic flow prediction are called time lags. In Equation 1, X_t is the value corresponding to X_{t+1}'s 1 time lags. Similarly, X_{t-1} is the value corresponding to X_{t+1}'s 2 time lags, and so on.

The workflow of the proposed LNS-SLSTM is shown in Figure 1. The proposed method includes two parts which are the data preprocessing block and the stacked LSTM. The method input the traffic flow series data into the data preprocessing block from the left. There are three steps of data preprocessing. First, a current sequence is selected by the calculation of autocorrelation, and time lags are selected by the calculation of partial autocorrelation, to choose the appropriate series data. Second, the selected series data is denoised by a discrete wavelet denoising algorithm. Third, the denoised data is rearranged to generate the input data of the LSTM model. After that, the well-preprocessed data are employed to train a stack LSTM. A dense layer follows the LSTM for the prediction of the traffic flow. The detailed operation of LNS-SLSTM method the is described in Algorithm 1. The traffic flow sequence S_t is the input data of the method, and the aim is to map the function of the prediction. First, the current traffic flow sequence S_t is input into the data preprocessing block. In the data preprocessing block, three important things need to be considered. One thing is to select the current sequence X and the most effective time lags. The second thing is to employ a discrete wavelet threshold denoising method to decompose and recompose the sequence into low-frequency detail sequence H and high-frequency details sequence A. The last thing is rearranging the related time lags of H, A, and X to generate the single-step input data for the stacked LSTM network. Then, the well-preprocessed data train the multi-level stacked LSTM network, which learns the mapping from the past traffic flow to the future traffic flow.

At the bottom of Figure 1, it shows every stage of the workflow of the LNS-SLSTM. In Section 2, the contents are described in the order of the LNS-SLSTM workflow. To begin with, it is how the current sequence and time lags are selected. Then, it is how the denoising method works and the model input data generation. Last, is the multi-stacked LSTM structure.

ALGORITHM 1 The LNS-SLSTM Method.

Input: traffic flow sequence S_t

Output: mapping forecasting function

1. current sequence $X \leftarrow$ autocorrelation of S_t;
2. time lags $\leftarrow$ partial autocorrelation of S_t;
3. **repeat**
4. low-frequency detail H and high-frequency detail $A \leftarrow$ wavelet denoising for X;
5. input data for the stacked LSTM $\leftarrow$ rearrange the related time lags of H, A, and X;
6. next prediction;
7. **until** the end of the training.

2.2 | Data preprocessing

2.2.1 | Current sequence and time lags

The time lag is important for the traffic flow forecasting task. The study compares the current traffic flow with the historical flow to analyze the period and trend of the traffic flow. For a traffic flow sequence $[X_{t-n+1}, \dots, X_t]$, the autocorrelation is employed to analyze the period and trend. The autocorrelation is calculated by Equation (2).

$$\gamma_h = \sum_{t=1}^{n-h} \frac{(X_{t+h} - \hat{\mu})(X_t - \hat{\mu})}{\sum_{t=1}^{n}(X_t - \hat{\mu})^2}, \tag{2}$$

where h represents the number of lags, $\hat{\mu}$ is the overall mean used to calculate the correlation. According to the calculation, the length of the current sequence X is determined for the forecasting task. The details will be discussed in Section 3.3.

Partial autocorrelation refers to the relationship between the current observation and the observation of the previous step, which removes the intervened observation. For example, for a traffic flow sequence $[X_{t-n+1}, \dots, X_t]$, the partial autocorrelation at lag k is to remove the correlation from $(X_{t-1}, X_{t-2}, \dots, X_{t-k-1})$ after the relevant influence caused by the lag term. The partial autocorrelation of the traffic flow is calculated by Equation (3).

$$\rho_{X_t, X_{t-k} | X_{t-1}, \dots, X_{t-k+1}} = \frac{E[(X_t - \hat{E}X_t)(X_{t-k} - \hat{E}X_{t-k})]}{E[(X_{t-k} - \hat{E}X_{t-k})^2]},$$
$$k = 1, 2, \dots, n, \tag{3}$$

where $\hat{E}X_t = E[X_t | X_{t-1}, \dots, X_{t-k+1}]$, and $\hat{E}X_{t-k} = E[X_{t-k} | X_{t-1}, \dots, X_{t-k+1}]$.

The partial autocorrelation can independently indicate the degree of correlation between two traffic flow sequences. Each lag k is calculated for the collection $\{\rho_i\}$, where $-1 \le \rho_i \le 1$, and $i \in k$. The collection shows the relationship between the two traffic flow for time lag k. The closer $|\rho_i|$ is to 1 the higher correlation between two traffic flow sequences. To determine the partial autocorrelation coefficient of the most related traffic flow sequences, the largest ones are found in $\{\rho_i\}$. The index the largest $\{\rho_i\}$ is i. In this way, the most effective historical time lags $\{X_i\}$ are got, and the number of the elements for $\{X_i\}$, which is m. The details will be discussed in Section 3.3.

2.2.2 | Sequence denoising

The wavelet transform can capture the details of the time-series signal compared with other denoising methods, and it has better performance than the ones based on Fourier transform [40]. In this work, the wavelet transform is taken for the time-frequency signal analysis of the traffic flow, since it can solve the non-stationary signal processing problem. The discrete wavelet threshold denoising method is simple to implement, and can suppress noise to a large extent. There are two key parameters to determine denoising performance. One is the noise threshold λ, and the other is the threshold function [41]. There are two basic types of threshold functions, which are divided into hard threshold ones and soft threshold ones. The advantage of the hard threshold is that the denoised signal can better retain the singular characteristic information, whereas the disadvantage is that when the signal is reconstructed, it is easy to produce signal distortion such as oscillation and Pseudo-Gibbs effect at the discontinuous point [42]. The advantage of soft threshold is that the signal after denoising is smoother, but the high-frequency estimated coefficient and the original coefficient have the disadvantage of constant error. Moreover, the singular characteristic information of the signal after denoising is blurred. The Garrote threshold [43] is adopted and the parameters are further optimized to balance the advantages of the hard threshold and the soft threshold. The conventional wavelet noise threshold remains unchanged on the decomposition scale, which leads to a deviation between the estimated threshold and the actual threshold for each scale. The wavelet coefficients of noise decrease as the scale increases. Therefore, when the signal is denoising, the threshold of the layer-by-layer scale should also decrease with the increase of the decomposition scale. In this regard, the threshold is calculated as follows. The threshold λ_j under j^{th}-level scale is calculated by Equation (4).

$$\lambda_j = \sigma_j \sqrt{2 \log_{100} N_j}, \tag{4}$$

where N_j is the length of the high-frequency wavelet coefficients and σ_j is the estimation of the noise standard variance at j^{th}-level scale.

In this study, to estimate the noise in the wavelet domain for a noisy traffic flow sequence, such a method is used to perform Daubechies wavelet transform on the signal, and arrange the high-frequency wavelet coefficient modulus obtained by processing each scale according to size. Then, the estimated noise is obtained during this scale transformation, for example, $\sigma = MAD/0.6745$, where MAD is the median absolute deviation. In this way, the denoise is more flexible and effective, since the influence of the difference of coefficients obtained after the wavelet processing of each scale on the noise is fully considered.

The Garrote threshold function is as Equation 5.

$$\hat{w}_{i,j} = \begin{cases} 0 & |w_{i,j}| \le \lambda_j \\ w_{i,j} - \frac{\lambda_j^2}{w_{i,j}} & |w_{i,j}| > \lambda_j. \end{cases} \tag{5}$$

where $\hat{w}_{i,j}$ is the estimated detail coefficient, and $w_{i,j}$ is the detailed coefficient obtained by wavelet transform. The Garrote threshold is $|w_{i,j}| > \lambda_j$. For each high-frequency detail, the coefficient is subtracted to shrink. In this way, the characteristics

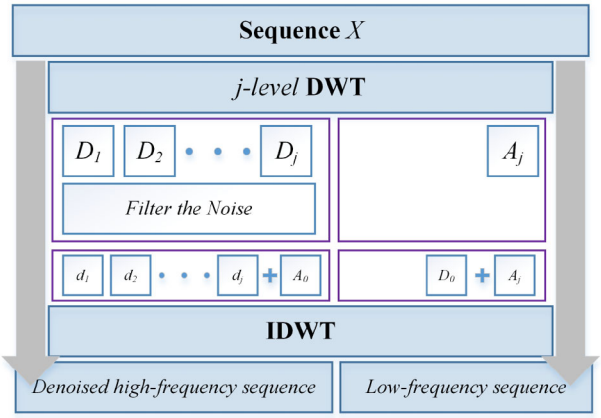

FIGURE 2 The schematic of wavelet denoising processing.

of each high-frequency detailed coefficient are better considered, and the signal characteristic information is better to keep the signal smoother.

In the wavelet domain, the effective signal energy is concentrated and represented by a few high-amplitude coefficients, while noise is represented by a great quantity of small-amplitude coefficients. The wavelet denoising shrinkage method uses this sparse characteristic of wavelet coefficients to filter noise from the signal. The discrete wavelet threshold denoising is to habitually perform multi-scale wavelet transformation on sequence signals and divide the obtained detailed coefficients of each scale into low-frequency and high-frequency detail coefficients. The usual processing method is to keep the low-frequency detail coefficients unchanged, and select an appropriate threshold to shrink the high-frequency detail coefficients. Then, the denoising signal is obtained through wavelet reconstruction. Different from the conventional ones, another signal reconstruction method is taken. The reconstruction process is to interpolate the low-frequency coefficients and the denoised high-frequency coefficients, respectively. Then, those are reconstructed through the inverse wavelet transform to obtain consecutive sequences, which are the low-frequency sequence and the high-frequency sequence. The process of wavelet denoising is depicted in Figure 2. The wavelet denoising method summarizes in Algorithm 2.

ALGORITHM 2 Wavelet Denoising Method.

Input: current traffic flow sequence X

Output: denoised high-frequency sequence and low-frequency sequence

1. Select a wavelet
2. Select the decomposition levels $j(1 \le j \le K)$, where $K = log_2 N$, N is the length of the sequence.
3. Make the j levels discrete wavelet transform (DWT) of the sequence X, decomposing it into j parts of high-frequency detail coefficients $D_1, D_2, \dots, D_j$, and the j^{th}-level low-frequency detail coefficients A_j.
4. Estimate the σ_j and calculate the λ_j for each part of high-frequency detail coefficients. And apply the noise thresholding to them.
5. A_0 is a zero values sequence with the length of N, and all the denoised high-frequency coefficients $d_1, d_2, \dots, d_j$, with A_0 to generate the denoised high-frequency sequence by the inverse discrete wavelet transform (IDWT)
6. D_0 is a zero values sequence with the length of N, and the j^{th}-level low-frequency detail coefficients with D_0 to restore the low-frequency sequence by the IDWT.

2.2.3 | Input data generation

As discussed in the previous section, for the prediction of each time interval $\hat{X}_t$, the partial autocorrelation is used to select m most related lags to form a $1 \times m$ array $\hat{X}_t = [X_1, \dots, X_m]$.

The length of the historical window n is determined from the period, trend, or other modes according to the analysis of autocorrelation. n is the number of matrix columns to form a $n \times m$ matrix. Each row is the historical time lags of the prediction, and each column is the lag sequence of the corresponding step of the original sequence. The matrix is as Equation 6.

$$SC_o = \begin{bmatrix} \hat{X}_t \\ \vdots \\ \hat{X}_{t-n+1} \end{bmatrix} = \begin{bmatrix} X_{11} & \cdots & X_{1m} \\ \vdots & \vdots & \vdots \\ X_{n1} & \cdots & X_{nm} \end{bmatrix}, \tag{6}$$

Then, the data preprocessing block utilizes discrete wavelet threshold denoising to decompose and recompose each column of SC_o into high-frequency sequence SC_{oh} and low-frequency sequence SC_{ol}.

$$SC_{oh} = \begin{bmatrix} \hat{X}_{t_h} \\ \vdots \\ \hat{X}_{t-n+1_h} \end{bmatrix} = \begin{bmatrix} X_{11_h} & \cdots & X_{1m_h} \\ \vdots & \vdots & \vdots \\ X_{n1_h} & \cdots & X_{nm_h} \end{bmatrix},$$

$$SC_{ol} = \begin{bmatrix} \hat{X}_{t_l} \\ \vdots \\ \hat{X}_{t-n+1_l} \end{bmatrix} = \begin{bmatrix} X_{11_l} & \cdots & X_{1m_l} \\ \vdots & \vdots & \vdots \\ X_{n1_l} & \cdots & X_{nm_l} \end{bmatrix}. \tag{7}$$

For the denoising algorithm in this study, when the original data is denoised, it inevitably reduces some effective information while reducing noise. To compensate for the loss of effective information, the original data and the denoised data are integrated and then input into the neural network, so that the network can calculate and determine the contribution of the denoised data and the original data to the prediction results. This is to achieve a balance between noise removal and maximum retention of valid information. Therefore, after the processing of discrete wavelet threshold denoising, A new matrix together with SC_O as a result of the data preprocessing is shown in Equation 8. Each row of the matrix SP_o is one time training step size for the LSTM network. The data for each time step contain the traffic flow corresponding to the time lags, and each time lag has three dimensions, including the original value, the high-frequency value, and the low-frequency value. For selected m

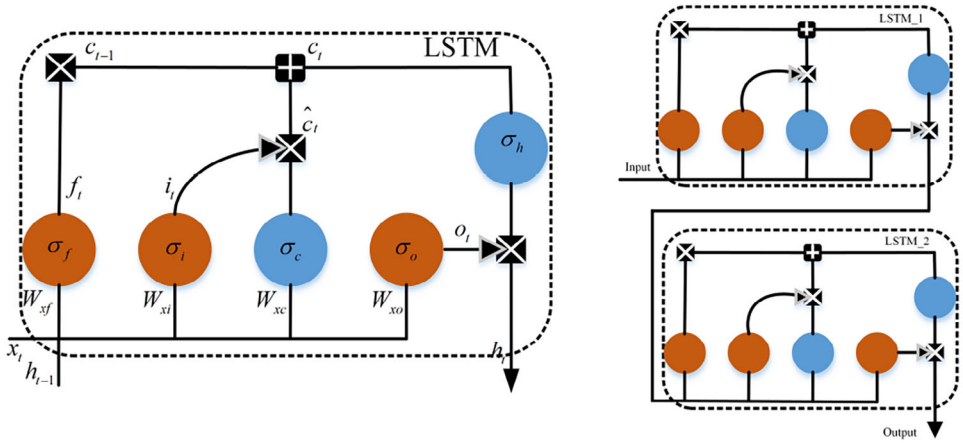

FIGURE 3 The stacked LSTM structure.

time lags, the input data dimension is $m \times 3$.

$$SP_o = \begin{bmatrix} SC_o & SC_{ob} & SC_{ol} \end{bmatrix}. \tag{8}$$

2.3 | Multi-level stacked LSTM network

The LSTM network can overcome the main disadvantage of vanishing or exploding gradients in RNNs. As it is shown in Figure 3, there are three gates, input gate, forget gate, and output gate in the LSTM network. These gates are used to control the transmission of LSTM information, and determine the cell state c_t and output signal h_t. The input gate is shown in Equations 9 and 10.

$$i_t = \sigma_i(x_t W_{xi} + h_{t-1} W_{hi} + b_i), \tag{9}$$

$$\hat{c}_t = \sigma_c(x_t W_{xc} + h_{t-1} W_{hc} + b_c), \tag{10}$$

where W_{xi}, W_{xc}, W_{hi}, W_{hc} are weight matrices. b_i, b_c are the biases. σ_i represents the sigmoid function and σ_c represents the tanh function. The input gate determines the information retained in the input information x_t and h_{t-1}. The input gate includes a sigmoid and a tanh neural network layer output which is i_t and $\hat{c}_t$. The forget gate is f_t, which determines the information to be forgotten in the input information x_t and h_{t-1}, as shown in Equation (11).

$$f_t = \sigma_f(x_t W_{xf} + h_{t-1} W_{hf} + b_f), \tag{11}$$

where W_{xf} and W_{hf} are weight matrices. b_f is the bias. σ_f represents the sigmoid function. The last moment cell state c_{t-1} is multiplied by f_t to select the forgotten and retained information. i_t is multiplied by $\hat{c}_t$. Then, they are added together to get the new cell state c_t shown in Equation 12.

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t. \tag{12}$$

The output gate integrates the cell state c_t with the output signal h_{t-1} and the input signal x_t. They pass through a tanh layer as the output signal h_t at the current time. As it shown in Equation (13).

$$h_t = \sigma_o(x_t W_{xo} + h_{t-1} W_{ho} + b_o) \odot \sigma_h c_t, \tag{13}$$

where W_{xo} and W_{ho} are the weight matrices. b_o is the bias. σ_o represent the sigmoid function and σ_h represent the tanh function. The LSTM has been proven successful in a wide range of challenging prediction tasks, such as [14, 30, 34]. Increasing the depth of the LSTM network by stacking improves the efficiency of training, and obtains higher accuracy. The added layer can learn the representations from previous layers and create new representations at a high level of abstraction. However, if the network depth is too deep, it will also cause gradient explosion/vanishing issues. In Section 3.3, the study will discuss the influence of the number of stacked layers on the forecasting performance. The study can properly increase the depth of the LSTM neural network by stacking the LSTM unit together. Essentially, the stacked LSTM network is taking one LSTM unit output and feeds it into another LSTM unit. For instance, the level 2 stacked LSTM network is shown in Figure 3.

3 | CASE STUDY

3.1 | Data description

As shown in Figure 4, four typical datasets from Amsterdam, Netherlands are employed to evaluate the forecasting performance of the proposed method, which are widely used in [25, 44, 45]. The data are collected from four detection points by MONICA detectors located on four highways. The data are collected from 20 May 2010, to 24 June 2010. The statistics are performed every minute, and the flow of vehicles passing by in that minute is counted and converted into the hourly traffic flow of that minute. These four highways are representative.

- A1 is the border highway, the first high-occupancy 3+ separation lane in Europe. The lane occupancy rate changes greatly instantaneously and the prediction are difficult.

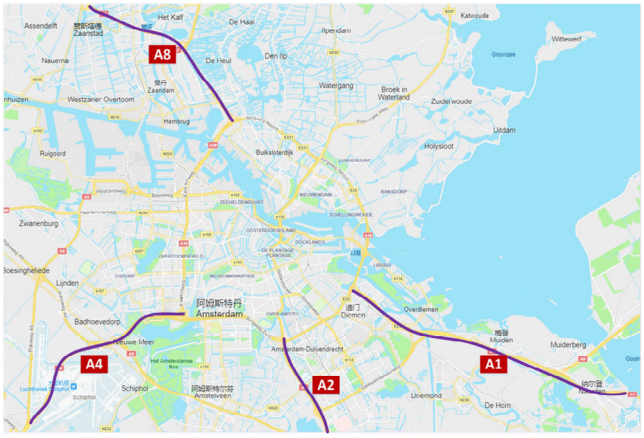

FIGURE 4 The map of the four highways in Amsterdam.

- A2 is a busy highway in the Netherlands and is prone to congestion. It can be used to verify the prediction performance of the method with traffic congestion.
- A4 is a domestic standard national highway, and its traffic flow is relatively moderate.
- A8 is a connecting highway with a total length of about 10 kilometers. It is prone to traffic accidents and can be used to verify the predictive performance of traffic emergencies.

3.2 | Evaluation criteria

The mean absolute percentage error (MAPE) and the root mean square error (RMSE) are widely used in traffic flow forecasting tasks, such as [25, 44, 45]. The study employed these two criteria to evaluate the performance of the proposed method and the control ones. The MAPE is defined as Equation (14), and the RMSE is defined as Equation (15).

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\widehat{y_i} - y_i}{y_i}\right), \tag{14}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\widehat{y_i} - y_i)^2}. \tag{15}$$

3.3 | Experimental setups and analysis

Traffic flow data is aggregated from the dataset with ten minutes time intervals for the experiment. The selected traffic flow data series has 5040 time intervals, of which 4032 time intervals are used for training and 1008 time intervals are used for testing. The experiment is based on Keras (an advanced neural network API written in Python). A Bayesian optimization tool kit, Hyperas is used to adjust the hyperparameters. The optimized hyperparameters include unit, epoch, and batch size. Unit is the characteristic dimension of the hidden layer in the LSTM model. Batch size refers to the number of training examples in one iteration. A certain number of examples are used to update the parameters of the model through the batch gradient descent algorithm in each iteration. Epoch means that the whole training set is transmitted forward and backward once through the deep neural network. The range of the unit is set as [16, 32, 64, 128, 256]. the batch size range is set as [10, 20, 40, 80, 160], and the epoch range is set as [50, 100, 200]. the tuned hyperparameters for the experiments are listed in Table 1.

TABLE 1 The hyperparameters of the LSTM.

Hyperparameter	Value
Units	64
Batch size	40
Epochs	100

The training step of the LSTM is set to 1, and each LSTM layer is set to one hidden layer whose dimension is 64. The epoch number is set to 100. The batch size is set to 40. The *decayed_learning_rate* updating in network training is given by Equation (16).

$$decayed\_learning\_rate = learning\_rate \cdot decay\_rate^{\frac{global\_step}{decay\_steps}}, \tag{16}$$

where $learning\_rate = 0.1$, $decay\_steps = 80$ and $decay\_rate = 0.96$.

Loss function set to mean absolute error as Equation (17).

$$MAE = \frac{\sum_i^n |y_i - \widehat{y_i}|}{n}. \tag{17}$$

The experiment repeats 20 times for each prediction, and averages the results of the 20 experiments as the final results to eliminate the randomness of the single experimental results.

First, 1-week traffic flow data is taken from A1, A2, A4, and A8 as the input data of Algorithm 1 to calculate the autocorrelation. Figure 5 shows that the autocorrelation of four highways is similar to the cosine function, and has a period which is around one day (144 time intervals). Thus, the study selects $n = 144$ as the length of the current sequence X.

Simultaneously, these 1-week data are used to calculate the partial autocorrelation, which is shown in Figure 6. The largest ones are found in π_i from the four highways are the time lag 1 and 2. Therefore, it concludes that the most related traffic flow of X_t is X_{t-1} and X_{t-2}, which are two adjacent historical traffic flows. And then, the Daubechies1 wavelet function is chosen for the DWT and IDWT at level 1 in the model for example. As shown in Figure 7, the original traffic flow is decomposed and recomposed to a low-frequency and a high-frequency sequence. The low-frequency sequence keeps the base model of the original signal. The high-frequency sequence includes most of the noise. For the high-frequency sequence, the study applies denoising depicted in Figure 8. The noises are flited while the character of the sequence is maintained.

As a result, as known from Algorithm 2, Equation (6), Equation (7), and Equation (8), for the

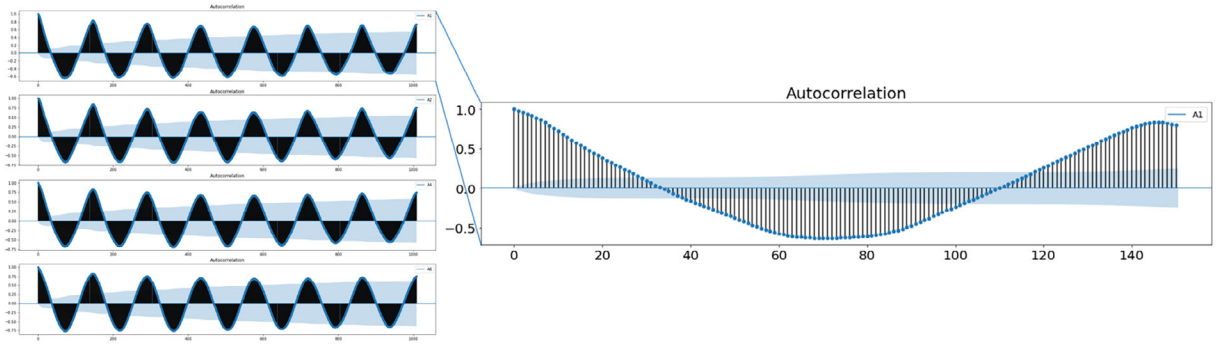

FIGURE 5 The autocorrelation of four highways.

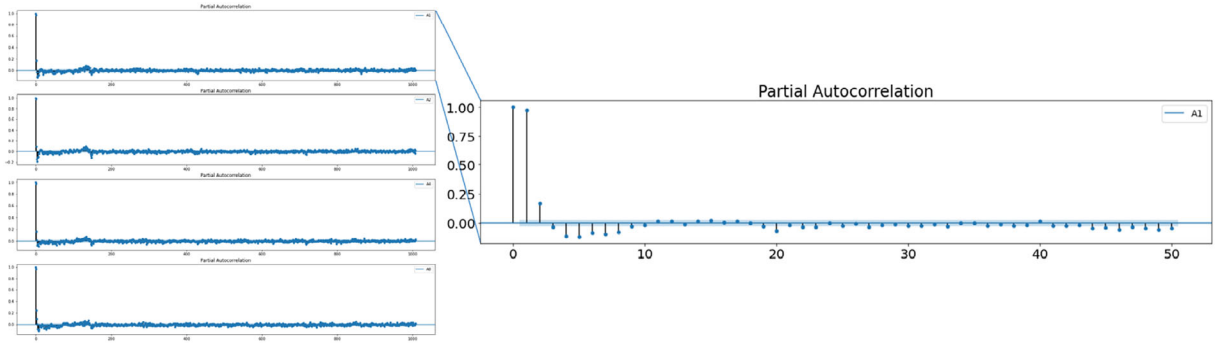

FIGURE 6 The partial autocorrelation of four highways.

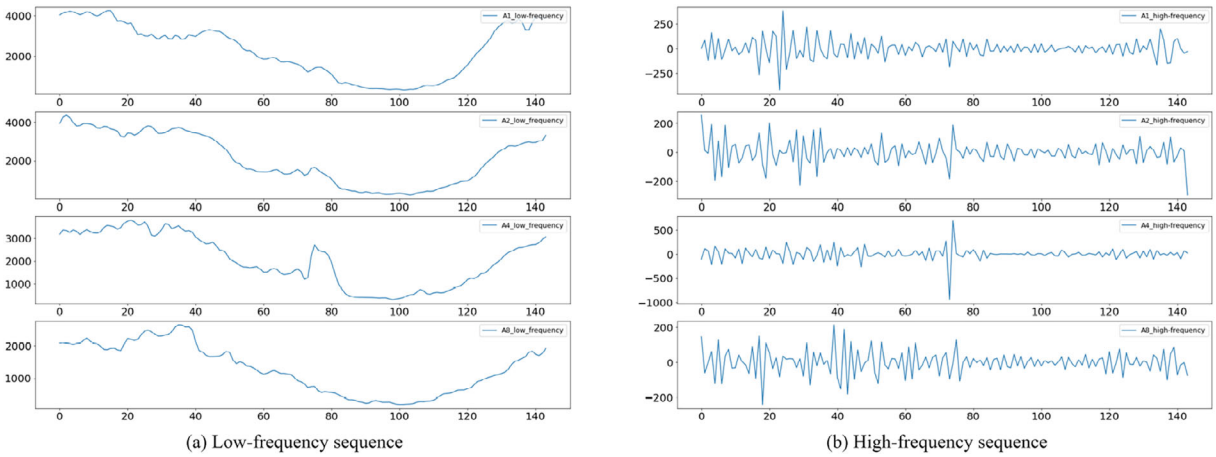

FIGURE 7 The low-frequency and high-frequency sequence of the traffic flow.

experiment data, the one training steps input of the LSTM model is $[X_{t-1}, X_{t-2}, X_{(t-1)_h}, X_{(t-1)_l}, X_{(t-2)_h}, X_{(t-2)_l}]$.

The differences in the wavelet scale impact the result. To select the optimal wavelet scale, the study has done wavelet threshold denoising of 1 to 3 levels, as depicted in Table 2. As shown in the table, the level 2 wavelet achieves a well-balanced performance on the four highways. Thus, it is more suitable to select the level 2 wavelet threshold denoising.

The study increases the depth of the LSTM network by stacking the LSTM unit together to improve the efficiency of training for higher accuracy. To illustrate the influence of the number of stacked layers, the study presents the performances for different numbers of stacked layers, for example, $n = 1, 2, 3, 4$, in Table 3. Figure 9 shows the percentage of the reduction for the MAPE and the RMSE of one stacked layer compared with other numbers of the layer. The stacked LSTM of 2 layers is the blue line. For highways, A1, A2, A4, and A8, the results show an average advantage in the stacked LSTM of 2 layers.

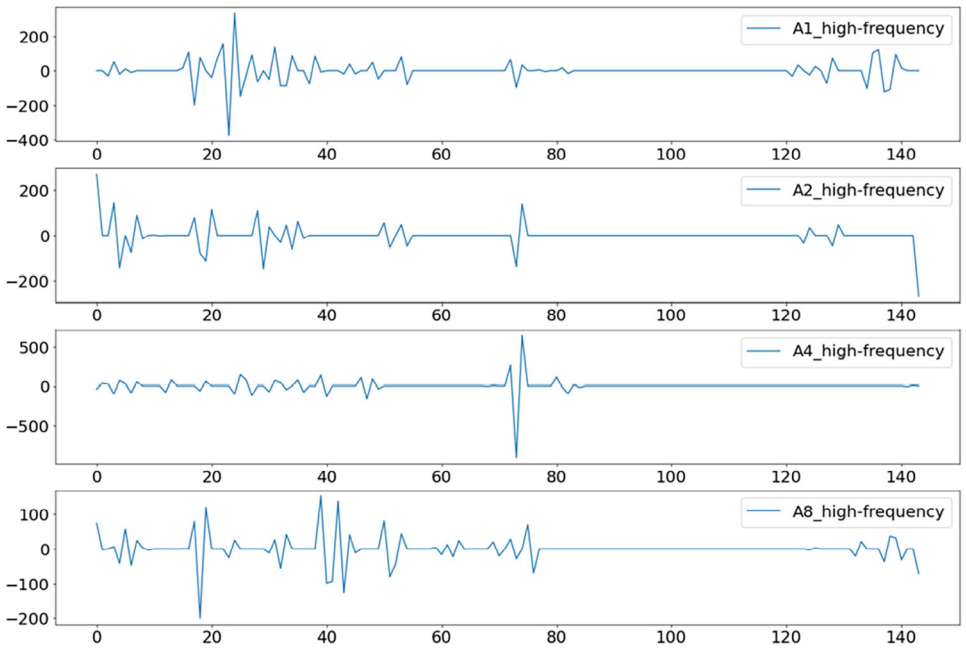

FIGURE 8 The high-frequency part after wavelet threshold denoising.

TABLE 2 Wavelet threshold denoising comparison.

		Level 1	Level 2	Level 3
A1	MAPE	10.44	10.65	10.70
	RMSE	271.74	268.68	273.09
A2	MAPE	9.41	9.71	9.56
	RMSE	200.51	197.45	201.16
A4	MAPE	10.31	10.44	10.27
	RMSE	206.74	207.18	207.68
A8	MAPE	11.52	11.26	10.95
	RMSE	157.71	152.97	163.47

TABLE 3 The comparison of different numbers of stacked layer LSTM.

		n=1	n=2	n=3	n=4
A1	MAPE	11.18	10.65	10.87	10.70
	RMSE	278.68	268.68	268.82	277.54
A2	MAPE	10.04	9.71	9.57	9.78
	RMSE	198.05	197.45	195.61	196.92
A4	MAPE	10.75	10.44	10.53	10.51
	RMSE	207.50	207.18	207.35	209.67
A8	MAPE	11.37	11.26	11.42	11.29
	RMSE	158.97	152.97	153.54	156.77

FIGURE 9 The percentage of the reduction for the MAPE and the RMSE.

3.4 | Comparison with other methods

The study employs two baseline models, the historical average (HA) and the random walk model (RW), for the experiment. The study also compares the proposed method with seven frequently used methods in recent years' publications for two criteria, MAPE and RMSE. These methods include the exponential smoothing model (ES), grey model (GM), least-squares boosting (LSBOOST), support vector regression method (SVR), stacked autoencoder model (SAE), Kalman

filtering model (KF), and LSTM model. Besides, six latest state-of-the-art models are included, SVRGSAS [19], SrOrkNNr [46], GA-KELM [21], PSOGSA-ELM [47], ABC-ELM [48], NiLSTM [20]. The brief introductions of these models are as follows.

HA: The average traffic flow in the past period is used as the forecast of current traffic flow.

RW: The traffic flow deviates from its previous value by a random step in each period. For more details on RW, refer to Reference [49].

ES: A special weighted moving average method. The smoothing factor α to 0.4 with quadratic exponential smoothing.

GM: A predictive method that builds mathematical models and makes predictions by using a small amount of incomplete information [50]. In this experiment, the GM(1,1) model was employed to predict the traffic flow.

LSBoost: One of the most popular boosting algorithms that ensembles linear regression. Zhou et al. [25] have applied this method to load forecasting in the energy day-ahead market And they declared that the least squares boosting algorithm gives more robust results than the SARIMA method for load forecasting.

SAE: The stacked autoencoder is trained in a layer-wise greedy fashion [51]. The spatial and temporal correlations are inherently considered in this model. The deep architecture of the SAE is set to [120, 60, 30] by cross-validation.

KF: The transition matrix is set to an $n \times n$ identity matrix. The variance of measurement noise R is set to 0.1. The initial state is set to $[\frac{1}{n}, \ldots, \frac{1}{n}]$. The initial error covariance and process noise are set to $0.01 \times I_{n \times n}$. The study sets the length of state variable n to 8, the same as Cai et al. [19].

SVR: The regression horizon is set to 8, which is the same as Cai et al. [19]. The kernel type is the radial basis function (RBF). The parameter C is set to the maximum difference between the traffic flow. The width parameter for the RBF kernel is set to 3×10^{-6}.

LSTM: The LSTM model with the following hyperparameters. The training step is 1, and the hidden layer dimension is 64. The epoch number is 100. The batch size is 40.

SVRGSA: A hybrid traffic flow forecasting model combining gravitational search algorithm (GSA) and the SVR model. The GSA is employed to search optimal SVR parameters. RBF is the kernel function. The ranges of these three parameters in the SVR model are set as $C \in (1, 10, 000)$, $\varepsilon \in (0, 10)$, and $\gamma \in (0, 1)$. The value of ε is closely related to the noise of the sample. GSA parameters are as follows. The population size is 40, maximum number of iterations is 100, $G0$ is 100, and δ is 20, the same as Cai et al. [19].

SrOrkNNr: The sample-rebalanced and outlier-rejected k-nearest neighbor regression model is a random subspace ensemble framework that generates multiple random subspaces for short-term traffic flow forecasting. It designs two kinds of probabilities, ω_1 and ω_2, corresponding to the distance of the hyperplane of kth local nearest neighbor (HKNN) and the kth fuzzy nearest neighbor, respectively, representing the local and global information of the data. The optimal parameters k, ω_1 and ω_2 are set to 15, 0.4, and 0.6.

TABLE 4 The comparison of MAPE with other methods.

Model	A1	A2	A4	A8
HA	16.87	15.53	16.72	16.24
RW	12.65	11.43	12.06	12.37
ES	11.94	10.75	11.97	12.00
GM	12.55	10.88	13.28	12.92
LSBOOST	13.78	14.43	12.90	14.00
SAE	13.57	11.59	12.70	12.71
KF	12.46	10.72	12.62	12.63
SVR	14.34	12.22	12.23	12.48
LSTM	12.34	11.35	11.91	12.45
SVRGSA (2019) [19]	11.15	**9.42**	10.65	11.81
SrOrkNNr (2020) [46]	11.27	10.00	11.60	11.63
GA-KELM (2023) [21]	11.67	9.83	11.31	12.59
PSOGSA-ELM (2022) [47]	11.53	10.16	11.67	12.05
ABC-ELM (2022) [48]	11.40	9.95	11.26	11.90
NiLSTM (2020) [20]	12.00	10.54	11.74	11.92
LNS-SLSTM	**10.65**	9.71	**10.44**	**11.26**

GA-KELM: A genetic-search-algorithm-improved kernel extreme learning machine unleashes the potential for improved prediction accuracy and generalization performance by substituting the inner product with a kernel function.

PSOGSA-ELM: A two-stage hybrid extreme learning model. First, the particle swarm optimization algorithm is employed for determining the initial population distribution of the gravitational search algorithm to improve the efficiency of the global optimal value search. Second, the results of the previous stage, rather than the network structure parameters randomly generated by the extreme learning machine, are used to train the hybrid forecasting model in a data-driven way.

ABC-ELM: A model uses the characteristics of the artificial bee colony algorithm to optimize the model so that the model can better and faster find the optimal solution in space. Besides, it also uses the characteristics of the limit learning machine to quickly deal with this nonlinear problem of short-term traffic flow forecasting.

NiLSTM: The noise-immune long short-term memory network embeds a noise-immune loss function deduced by maximum correntropy into the LSTM network for short-term traffic flow forecasting. The training step is 1. The hidden layer dimension is 256. The epoch number is 50. The batch size is 32.

From Tables 4 and 5, the study finds that LNS-SLSTM is superior to traditional parametric and nonparametric methods. This is because the parameterization method is difficult to deal with the nonlinear relationship within the traffic data using finite parameters and fixed model settings. For example, HA is easily affected by unexpected events. RW prediction is unstable. When the traffic state changes significantly, KF tends to overshoot,

TABLE 5 The comparison of RMSE with other methods.

Model	A1	A2	A4	A8
HA	404.84	348.96	357.85	218.72
RW	312.92	223.82	230.01	174.14
ES	315.82	226.40	237.76	174.67
GM	348.38	255.86	274.48	188.97
LSBOOST	306.33	233.88	229.78	177.52
SAE	301.44	214.22	226.12	166.71
KF	332.03	239.87	250.51	187.48
SVR	329.09	259.74	253.66	190.30
LSTM	295.48	221.66	220.84	169.01
SVRGSA (2019) [19]	284.97	**192.68**	213.69	161.07
SrOrkNNr (2020) [46]	281.30	203.54	218.37	162.38
GA-KELM (2023) [21]	284.67	193.83	220.89	163.02
PSOGSA-ELM (2022) [47]	288.03	204.09	220.52	163.92
ABC-ELM (2022) [48]	286.25	200.42	220.07	163.67
NiLSTM (2020) [20]	286.52	204.03	224.53	164.56
LNS-SLSTM	**268.68**	197.45	**207.18**	**152.97**

which greatly reduces the prediction performance. The performance of nonparametric methods, such as SVR, mainly depends on the selection of kernel functions and parameters. At present, there is no good method to solve the problem of selecting a kernel function to construct an SVR algorithm according to the actual data model for traffic flow prediction. The accuracy of LSTM network prediction is largely based on the assumption that the deviation between the estimated value and the real value follows the Gaussian distribution, which is difficult to achieve in the actual traffic flow prediction with noise. The LSN-SLSTM achieves the lowest MAPE and RMSE than other models on almost all datasets. Except for A2 datasets, LSN-SLSTM has 3.08% higher in MAPE and 2.48% higher RMSE than SVRGSA shown in Figure 11. Considering the prediction performance of all methods on four highways,the LSN-SLSTM has the best comprehensive effect. To depict the advances of the proposed model, the LNS-SLSTM method is compared with the six state-of-the-art methods, respectively, on four highways, incorporating SVRGSA, SOKNN, GA-KELM, PSOGSA-ELM, ABC-ELM, and NiLSTM. The distribution diagram for the improvement of MAPE and RMSE is shown in Figure 10. Overall, the maximum average advance is 0.99, the minimum is 0.20, and the average is 0.73 in MAPE. Meanwhile, the maximum average advances of the overall is 13.37, the minimum is 5.62, and the average is 11.61 in RMSE. Specifically, for A1, A2, A4, and A8, it achieves the average advance, respectively, is 0.82, 0.36, 1.15, 0.60 in MAPE and 17.22, 5.84, 13.12, and 10.28 in RMSE. Figure 11 shows the reduction of the MAPE prediction error of the LNS-SLSTM method. Overall, the average reduction is 7.28% compared with the above six methods on four highways; respectively, its average reduces by 9.99%, 1.76%, 9.81%, and 7.55% on A1, A2, A4, and A8. It also portrays the reduction of the RMSE prediction

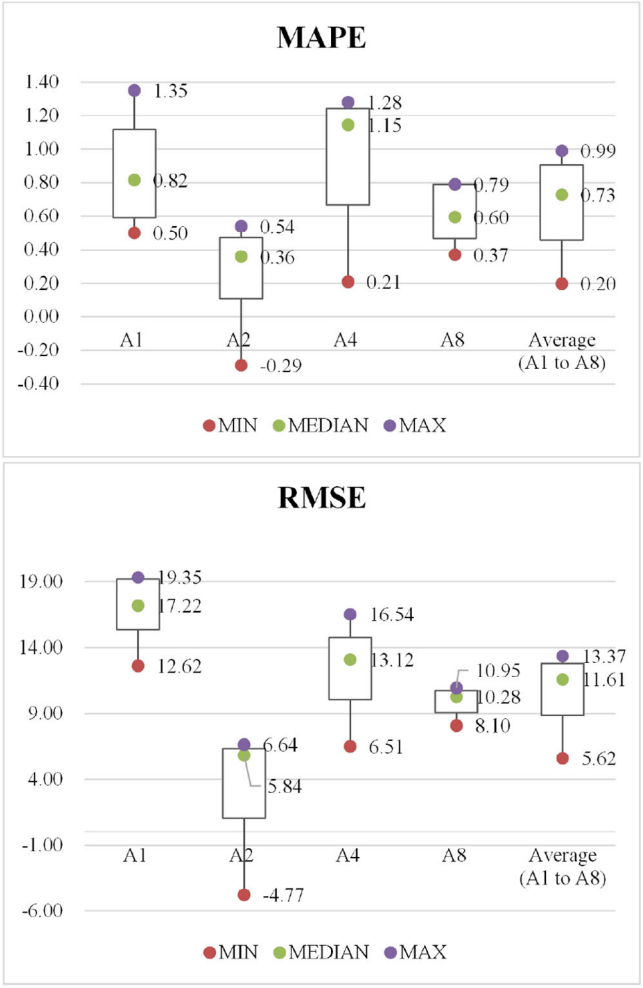

FIGURE 10 The distribution of MAPE and RMSE improvement.

error of the LNS-SLSTM method in Figure 11, The overall average reduction is 6.02%; respectively, on A1, A2, A4, and A8., it reduces 7.24%, 1.82%, 7.26%, and 7.76% on average.

In conclusion, the comparison and analysis of the experimental results illustrate that the proposed method is superior to the state-of-the-art methods mentioned in this paper. The effectiveness of the proposed method is suitable for short-term traffic flow forecasting, and the application of the proposed methods is demonstrated in the next section.

3.5 | Real-world application

Currently, public transportation companies face widespread vehicle scheduling problems, such as high investment, slow response, poor efficiency, and serious loss of passenger sources. The study embeds the LNS-SLSTM in the bus intelligent dispatching system to forecast short-term traffic flow on specific road sections for the dynamic adjustment of bus dispatch. The study improves the intelligence of bus operation and dispatch management for more efficient usage of resources, such as people, vehicles, and routes for bus companies.

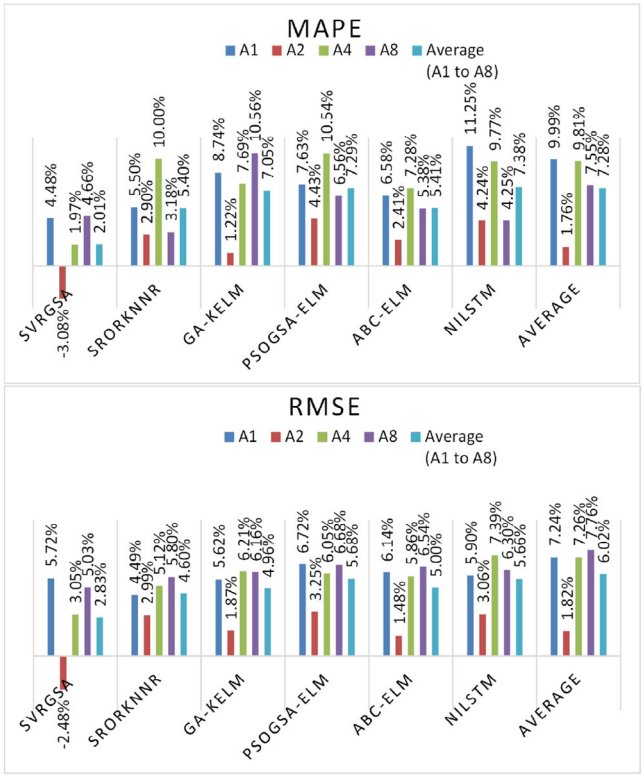

FIGURE 11 The comparison of MAPE and RMSE improvement.

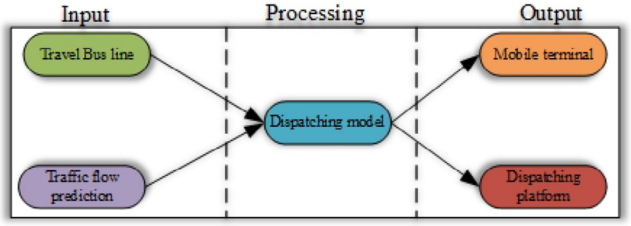

FIGURE 12 Congestion dispatching service.

The proposed LNS-SLSTM method has been applied to the Guilin Travel Network Bus Intelligent Dispatching System. This method is a part of the line congestion dispatching function module. The implementation scheme of the line congestion dispatching function is shown in Figure 12. The congestion dispatching model receives the current line of the travel bus and the traffic flow prediction to generate the dispatching result of the system for the mobile terminal and platform.

For the detail, the dispatching model judges the degree of road congestion which refers to the ratio of traffic flow to road traffic capacity, through the saturation of roads. The road traffic capacity refers to the maximum traffic flow rate of a section of the road passing through a cross-section in a unit of time under certain road and traffic conditions. According to the Code for Design of Urban Road Engineering (CJJ37-2012)(2016 version) [52], the traffic capacity of the target road can be obtained by looking up the table and calculating accordingly.

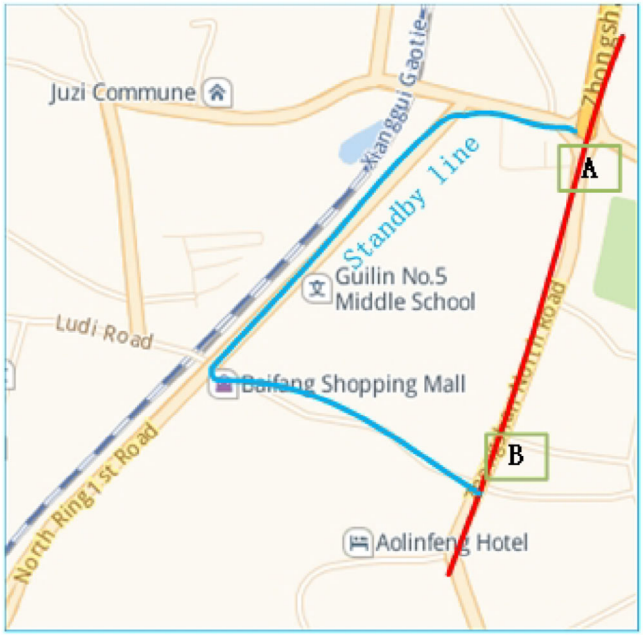

FIGURE 13 Dispatching method of this road section.

TABLE 6 The comparison of the congestion dispatching method.

Dispatching method	Congestion avoid	Operation
Original method	No	Add a bus
New method	Yes	Detour

For example, there are largely residential areas, hospitals, schools, and commercial markets in the section of North Zhongshan Road, and traffic congestion is easily caused by holidays, emergencies, accidents etc. This section is shown in Figure 13. Traffic detectors are set at positions A and B of Zhongshan North Road to collect traffic flow. The traffic congestion prediction and line dispatching are conducted according to the collected data. The travel bus line of Guilin Travel Company has no stops on this section, but ten minutes before passing through this road, the system predicts whether congestion or not of the section, if congestion the vehicles will bypass the standby line in advance, to avoid being unable to pass smoothly after entering the congested section, which will delay the travel of passengers or increase the vehicle scheduling.

Effective congestion dispatching is based on accurate traffic flow prediction. The application of the LNS-LSTM method improves the effect of traffic flow prediction and can avoid traffic congestion in specific sections in advance. The following Table 6 shows the difference between the original congestion scheduling and the new predictive congestion scheduling of Guilin Travel Company. In the previous method, there was no ability to predict congestion in advance. Once vehicles get stuck in congestion, corresponding vehicle shifts need to be increased. In contrast, the new predictive scheduling can effectively forecast the congestion to bypass the standby

line in advance, avoid congestion, and not need to increase vehicle shifts. Therefore, when the road section is congested, the previous method requires two vehicles on the line, while the new method only needs to detour without adding more vehicles. Only one vehicle is needed on the line, so the vehicle dispatching efficiency of the new method is 50% higher than that of the old method. At the same time, the operating costs of manpower and materials are reduced accordingly.

4 | CONCLUSION

In this paper, the study proposes a short-term traffic flow forecasting method that considers only temporal correlations of the traffic flow, which is common in transport networks only equipped with sparse or isolated detectors. The study designs a lag-related method to effectively identify the time lags by autocorrelation and partial autocorrelation. Then, the optimal level of the discrete wavelet threshold denoising shrinkage algorithm is explored. The denoised traffic flow data is employed to train a multi-level stacked long short-term memory network. The proposed method is compared with nine commonly used methods and six state-of-the-art models on four benchmark datasets from the highways in Amsterdam, Netherlands. The experimental results demonstrate an average improvement of 7.28% in MAPE and 6.02% in RMSE compared with state-of-the-art methods. Furthermore, the proposed method is applied in an intelligent bus dispatching system. The successful application also demonstrates the effectiveness and superiority of the proposed method.

AUTHOR CONTRIBUTIONS

Kai Li: Data curation; investigation; methodology; software; validation; writing—original draft; writing—review and editing. **Weihua Bai**: Conceptualization; resources; supervision; validation; writing—review and editing. **Shaowei Huang**: Investigation; resources. **Guanru Tan**: Formal analysis; writing—review and editing. **Teng Zhou**: Conceptualization; supervision; visualization; writing—review and editing. **Keqin Li**: Project administration.

ACKNOWLEDGEMENTS

The work was supported the fund of Natural Science Foundation of Guangdong Province (Nos. 2022A1515011590, 2021A1515012302, 2022A1515011978), Key Scientific Research Project of Universities in Guangdong Province (Nos. 2020ZDZX3020, 2020ZDZX3028, 2022ZDZX1007), Guangdong Provincial Science and Technology Plan Project (No. STKJ202209003), the 2023 Guangdong Scientific Research Platform and Projects for the Higher-educational Institution and Education Science Planning Scheme (No. 2023ZDZX3041), the Teaching Reform Project of University Public Computer Course of Guangdong Province (No. 2021-GGJGJ-012), Laboratory Management Committee of Guangdong Institute of Higher Education (No. GDJ20220360), Technology Plan Project of Zhaoqing (No. 2021SN12), Zhaoqing University technology project 2023 (No. QN202346).

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Kai Li https://orcid.org/0000-0003-2368-134X
Teng Zhou https://orcid.org/0000-0003-1920-8891

REFERENCES

1. Huang, B., Dou, H., Luo, Y., Li, J., Wang, J., Zhou, T.: Adaptive spatiotemporal transformer graph network for traffic flow forecasting by iot loop detectors. IEEE Internet Things J. 10(2), 1642–1653 (2022)
2. Agachai-Sumalee, H.W.H.: Smarter and more connected: Future intelligent transportation system. IATSS Res. 42, 67–71 (2018)
3. Lee, W.-H., Shian-Shyong Tseng, W.Y.S.: Collaborative real-time traffic information generation and sharing framework for the intelligent transportation system. Inf. Sci. 180, 62–70 (2010)
4. Fang, W., Zhuo, W., Song, Y., Yan, J., Zhou, T., Qin, J.: Δ_{free}-LSTM: An error distribution free deep learning for short-term traffic flow forecasting. Neurocomputing (2023)
5. Li, S., Lyu, D., Huang, G., Zhang, X., Gao, F., Chen, Y., et al.: Spatially varying impacts of built environment factors on rail transit ridership at station level: A case study in guangzhou, china. J. Transp. Geogr. 82, 102631 (2020)
6. Gao, F., Li, S., Tan, Z., Wu, Z., Zhang, X., Huang, G., et al.: Understanding the modifiable areal unit problem in dockless bike sharing usage and exploring the interactive effects of built environment factors. Int. J. Geogr. Inf. Sci. 1–21 (2021)
7. Li, S., Lyu, D., Liu, X., Tan, Z., Gao, F., Huang, G., et al.: The varying patterns of rail transit ridership and their relationships with fine-scale built environment factors: Big data analytics from guangzhou. Cities 99, 102580 (2020)
8. Ahmed, M.S., Cook, A.R.: Analysis of freeway traffic time-series data by using Box-Jenkins techniques. Transport. Res. Rec. 722, 1–9 (1979)
9. Li, S., Zhuang, C., Tan, Z., Gao, F., Lai, Z., Wu, Z.: Inferring the trip purposes and uncovering spatio-temporal activity patterns from dockless shared bike dataset in shenzhen, china. J. Transp. Geogr. 91, 102974 (2021)
10. Qi, Y., Ishak, S.: A hidden markov model for short term prediction of traffic conditions on freeways. Transp. Res. Part C Emerg. Technol. 43, 95–111 (2014)
11. Van.Der.Voort, M., Dougherty, M., Watson, S.: Combining kohonen maps with arima time series models to forecast traffic flow. Transp. Res. Part C Emerg. Technol. 4(5), 307–318 (1996)
12. Williams, B.M.: Multivariate vehicular traffic flow prediction: evaluation of arimax modeling. Transp. Res. Rec. 1776(1), 194–200 (2001)
13. Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. J. Transp. Eng. 129(6), 664–672 (2003)
14. 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. 172, 1–14 (2019)
15. Zhang, Y., Ye, Z.: Short-term traffic flow forecasting using fuzzy logic system methods. J. Intell. Transp. Syst. 12(3), 102–112 (2008)
16. Kumar, S.V.: Traffic flow prediction using kalman filtering technique. Procedia Eng. 187, 582–587 (2017)
17. Zhang, S., Song, Y., Jiang, D., Zhou, T., Qin, J.: Noise-identified kalman filter for short-term traffic flow forecasting. In: 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN). pp. 462–466. IEEE, Piscataway (2019)

18. Zheng, S., Zhang, S., Song, Y., Lin, Z., Jiang, D., Zhou, T.: A noise-immune boosting framework for short-term traffic flow forecasting. Complexity 2021, 1–9 (2021)
19. Cai, L., Chen, Q., Cai, W., Xu, X., Zhou, T., Qin, J.: Svrgsa: a hybrid learning based model for short-term traffic flow forecasting. IET Intell. Transp. Syst. 13(9), 1348–1355 (2019)
20. Cai, L., Lei, M., Zhang, S., Yu, Y., Zhou, T., Qin, J.: A noise-immune lstm network for short-term traffic flow forecasting. Chaos: Interdiscip. J. Nonlinear Sci. 30(2), 023135 (2020)
21. Chai, W., Zheng, Y., Tian, L., Qin, J., Zhou, T.: Ga-kelm: Genetic-algorithm-improved kernel extreme learning machine for traffic flow forecasting. Mathematics 11(16), 3574 (2023)
22. Zhou, T., Dou, H., Tan, J., Song, Y., Wang, F., Wang, J.: Small dataset solves big problem: an outlier-insensitive binary classifier for inhibitory potency prediction. Knowl.-Based Syst. 251, 109242 (2022)
23. Quan, T., Yuan, Y., Luo, Y., Song, Y., Zhou, T., Wang, J.: IEEE From regression to classification: Fuzzy multi-kernel subspace learning for robust prediction and drug screening. IEEE Trans. Ind. Inform. (2023)
24. Ma, Y., Zhang, Z., Ihler, A., Pan, B.: Estimating warehouse rental price using machine learning techniques. Int. J. Comput. Commun. Control. 13(2), 235–250 (2018)
25. Zhou, T., Han, G., Xu, X., Lin, Z., Han, C., Huang, Y., et al.: δ-agree adaboost stacked autoencoder for short-term traffic flow forecasting. Neurocomputing 247, 31–38 (2017)
26. Zhou, T., Han, G., Xu, X., Han, C., Huang, Y., Qin, J.: A learning-based multimodel integrated framework for dynamic traffic flow forecasting. Neural Process. Lett. 49(1), 407–430 (2019)
27. Yang, B., Sun, S., Li, J., Lin, X., Tian, Y.: Traffic flow prediction using lstm with feature enhancement. Neurocomputing 332, 320–327 (2019)
28. Lin, Y., Li, L., Jing, H., Ran, B., Sun, D.: Automated traffic incident detection with a smaller dataset based on generative adversarial networks. Accid. Anal. Prev. 144, 105628 (2020)
29. Liu, Y., Zheng, H., Feng, X., Chen, Z.: Short-term traffic flow prediction with conv-lstm. In: 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP). pp. 1–6. IEEE, Piscataway (2017)
30. Lu, H., Huang, D., Song, Y., Jiang, D., Zhou, T., Qin, J.: St-trafficnet: A spatial-temporal deep learning network for traffic forecasting. Electronics 9(9), 1474 (2020)
31. Ma, Y., Zhang, Z., Ihler, A.: Multi-lane short-term traffic forecasting with convolutional lstm network. IEEE Access 8, 34629–34643 (2020)
32. Mackenzie, J., Roddick, J.F., Zito, R.: An evaluation of htm and lstm for short-term arterial traffic flow prediction. IEEE Trans. Intell. Transp. Syst. 20(5), 1847–1857 (2018)
33. Tian, Y., Zhang, K., Li, J., Lin, X., Yang, B.: Lstm-based traffic flow prediction with missing data. Neurocomputing 318, 297–305 (2018)
34. Lu, H., Ge, Z., Song, Y., Jiang, D., Zhou, T., Qin, J.: A temporal-aware lstm enhanced by loss-switch mechanism for traffic flow forecasting. Neurocomputing 427, 169–178 (2021)
35. Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., Zhou, X.: Lc-rnn: A deep learning model for traffic speed prediction. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 3470–3476. AAAI Press, Menlo Park, CA (2018)
36. Wu, K., Xu, C., Yan, J., Wang, F., Lin, Z., Zhou, T.: Error-distribution-free kernel extreme learning machine for traffic flow forecasting. Eng. Appl. Artif. Intell. 123, 106411 (2023)
37. Fang, W., Zhuo, W., Yan, J., Song, Y., Jiang, D., Zhou, T.: Attention meets long short-term memory: A deep learning network for traffic flow forecasting. Phys. A: Stat. Mech. 587, 126485 (2022)
38. Li, L., Du, B., Wang, Y., Qin, L., Tan, H.: Estimation of missing values in heterogeneous traffic data: Application of multimodal deep learning model. Knowl.-Based Syst. 194, 105592 (2020)
39. Zhu, J., Bai, W., Zhao, J., Zuo, L., Zhou, T., Li, K.: Variational mode decomposition and sample entropy optimization basedtransformer framework for cloud resource load prediction. Knowl.-Based Syst. 280, 111042 (2023)
40. Xu, W., Liu, J., Yan, J., Yang, J., Liu, H., Zhou, T.: Dynamic spatiotemporal graph wavelet network for traffic flow prediction. IEEE Internet Things J. (2023)
41. He, C., Xing, J., Li, J., Yang, Q., Wang, R.: A new wavelet thresholding function based on hyperbolic tangent function. Math. Probl. Eng. 2015, 528656 (2015)
42. Ding, Y., Selesnick, I.W.: Artifact-free wavelet denoising: non-convex sparse regularization, convex optimization. IEEE Signal Process. Lett. 22(9), 1364–1368 (2015)
43. He, F., He, X.: A continuous differentiable wavelet shrinkage function for economic data denoising. Comput. Econ. 54(2), 729–761 (2019)
44. Wang, Y., Van.Schuppen, J.H., Vrancken, J.: Prediction of traffic flow at the boundary of a motorway network. IEEE Trans. Intell. Transp. Syst. 15(1), 214–227 (2013)
45. Cai, L., Zhang, Z., Yang, J., Yu, Y., Zhou, T., Qin, J.: A noise-immune kalman filter for short-term traffic flow forecasting. Phys. A: Stat. Mech. 536, 122601 (2019)
46. Cai, L., Yu, Y., Zhang, S., Song, Y., Xiong, Z., Zhou, T.: A sample-rebalanced outlier-rejected k-nearest neighbor regression model for short-term traffic flow forecasting. IEEE Access 8, 22686–22696 (2020)
47. Cui, Z., Huang, B., Dou, H., Cheng, Y., Guan, J., Zhou, T.: A two-stage hybrid extreme learning model for short-term traffic flow forecasting. Mathematics 10, 2087 (2022)
48. Li, X., Li, L., Huang, B., Dou, H., Yang, X., Zhou, T.: Meta-extreme learning machine for short-term traffic flow forecasting. Appl. Sci. 12(24), 12670 (2022)
49. Lippi, M., Bertini, M., Frasconi, P.: Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. IEEE Trans. Intell. Transp. Syst. 14(2), 871–882 (2013)
50. Huan, G., Xinping, X., Jeffrey, F.: Urban road short-term traffic flow forecasting based on the delay and nonlinear grey model. J. Transp. Syst. Eng. Inf. Technol. 13(6), 60–66 (2013)
51. 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. 16(2), 865–873 (2015)
52. Code for Design of Urban Road Engineering CJJ 37-2012 (2016 version). China Architecture Press (2016)

How to cite this article: Li, K., Bai, W., Huang, S., Tan, G., Zhou, T., Li, K.: Lag-related noise shrinkage stacked LSTM network for short-term traffic flow forecasting. IET Intell. Transp. Syst. 18, 244–257 (2024). https://doi.org/10.1049/itr2.12448