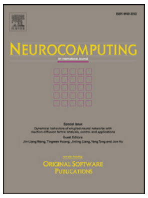
Neurocomputing 332 (2019) 320–327

Contents lists available at ScienceDirect

Neurocomputing

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



Traffic flow prediction using LSTM with feature enhancement

Bailin Yang[a], Shulin Sun[a], Jianyuan Li[b], Xianxuan Lin[a], Yan Tian[a,*]

[a] *School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310014, PR China*
[b] *ENJOYOR Company Limited, Hangzhou 310030, PR China*

ARTICLE INFO

Article history:
Received 27 June 2018
Revised 20 October 2018
Accepted 3 December 2018
Available online 29 December 2018

Communicated by K Chan

Keywords:
Short-term traffic flow prediction
Noise processing
LSTM feature enhancement
Attention mechanism

ABSTRACT

Long short-term memory (LSTM) is widely used to process and predict events with time series, but it is difficult to solve exceedingly long-term dependencies, possibly because the LSTM errors increase as the sequence length increases. Recently, researchers have noted that adding features on multiple time scales can help improve the long-term dependency of the RNN, which is inspired by the attention mechanism, considering the need for historical data in traffic flow prediction. We propose an improved approach that connects the high-impact value of remarkably long sequence time steps to the current time step, and these high-impact traffic flow values are captured using the attention mechanism. At the same time, we smooth out some data beyond the normal range to obtain better prediction results. The experimental results show that the proposed prediction model has certain competitiveness in short-term traffic flow predictions.

1. Introduction

With rapid socioeconomic development, the number of vehicles has rapidly increased, and traffic congestion problems have become increasingly serious. The most effective method to solve this problem is to use intelligent transportation systems (ITS), which use data collection, information processing, automation and other technologies to rationally allocate transportation resources and reduce traffic congestion. One of the core elements of an intelligent transportation system is the short-term traffic flow forecast. Precise traffic flow prediction models play an important role in reducing traffic congestion, improving the air quality, and supporting government decisions.

There are currently many short-term traffic flow prediction methods. In recent years, neural network-related algorithms have been widely used in various fields and have proven to be superior to traditional mathematical algorithms. Among the neural network-related algorithms, LSTM is widely used in time series prediction [1–3]. LSTM is a special type of recurrent neural network (RNN); compared with RNN, LSTM introduces the concept of gates. This concept includes a “forget gate”, which determines the information to discard; an “input gate”, which determines the new information to be saved; and an “output gate”, which determines what information to output to the next level. The three “gates” of the LSTM facilitate the regulation of long-term memory. However, the performance of the model deteriorates rapidly as the length of the sequence increases [4], which may be because of the accumulation of errors: the longer the sequence in LSTM is, the greater the error.

In this paper, to perceive useful information from a long distance, we attempt to capture high-impact traffic flow values in extremely long sequences using the attention mechanism. Then, we add the captured information to the current time step to give the LSTM the ability to rely on exceedingly long dependencies. The main contributions of the paper are as follows:

- We propose a short-term traffic flow prediction method, LSTM+, that can sense both long short-term memory and remarkably long distances. This method can effectively improve the problem of the LSTM extremely long-term memory shortage.
- In data preprocessing, we propose a simple and general method to smooth the noise based on the trends.

The remainder of this paper is outlined as follows. Section II introduces the studies on short-term traffic flow prediction. Section III provides a method for prediction based on LSTM feature enhancement. Section IV describes the local dataset and public dataset. Section V discusses the experimental results. Section VI concludes the paper.

2. Related work

Traffic volume prediction is one of the most critical links of ITS and began in the 1970s. Many scholars have used different

[*] Corresponding author.
E-mail address: tianyan@zjgsu.edu.cn (Y. Tian).

https://doi.org/10.1016/j.neucom.2018.12.016
0925-2312/

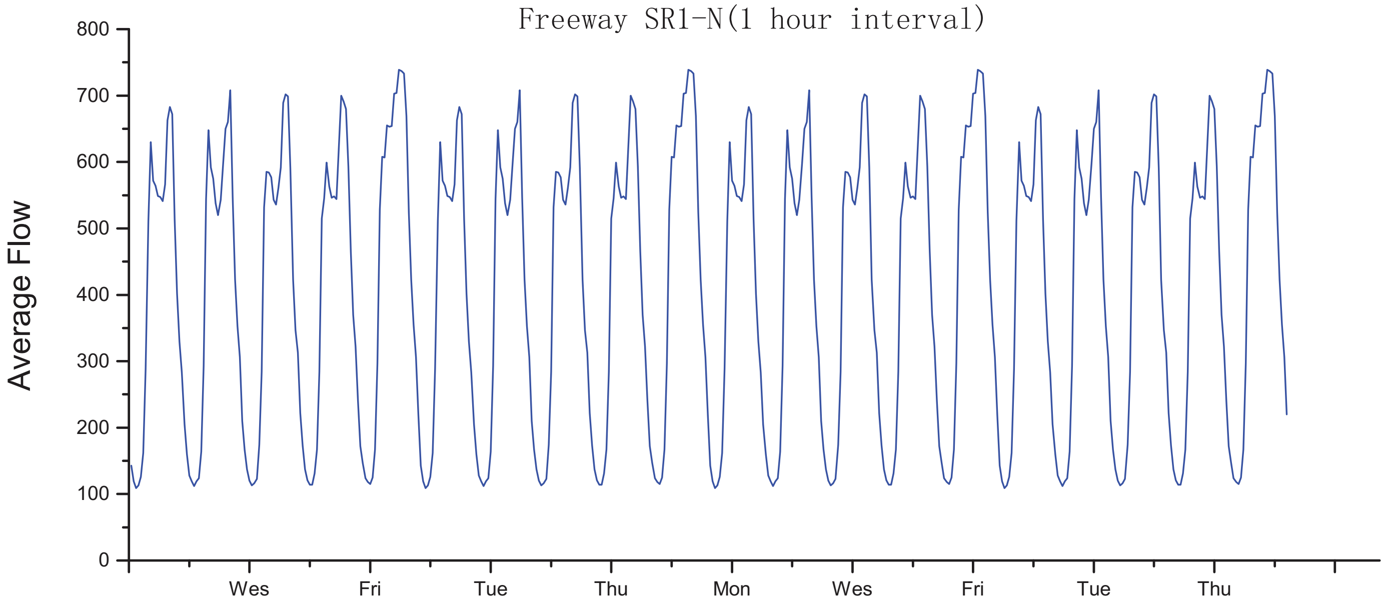


Fig. 1. The impact of the traffic volume of the previous day on the forecast value.

methods to predict traffic flows, which can be roughly divided into two types: parametric and nonparametric approaches.

2.1. Parametric approaches

Parametric approaches are based on parameters with empirical data (speed and flow), such as the kinematic-wave model [5] and the cellular automata model [6]. With the development of technology, people have proposed time-series models based on parametric methods with higher prediction accuracy; the most classic time-series model is ARIMA [7] and the subsequent improved ARIMA [8–11]. Parametric methods are based on theoretical assumptions and empirical data to determine the model parameters. In practice, because of human and environmental effects, traffic flow data are often unstable and nonlinear; thus, it is insufficient to rely on parametric models alone.

2.2. Nonparametric approaches

Examples of nonparametric approaches include support vector regression (SVR) [12], artificial neural network (ANN) [13], k-nearest neighbor (KNN) [14], and improved KNN using temporal and spatial information [15]. In recent years, Hinton [16] has had great success in deep learning, and many researchers have begun to use deep learning methods, such as deep belief networks (DBN) [17] and stacked autoencoders (SAE) [18], which perform better than the BP neural network [19], the random walk (RW) forecast method, the support vector machine (SVM) method [20], and radial basis function (RBF) NN model [21], to predict traffic flows. Jingyuan Wang et al., who used the deep bidirectional LSTM (DBL) model, captured the deep features of traffic flows [22]. Boyi Liu et al. added singular point probabilities in traffic flows [23]. However, these methods do not seriously consider the effect of empirical data. Recently, many researchers have combined the spatial feature with the temporal feature for prediction [24–28]. The disadvantage of this method is that it relies on the traffic flow data of adjacent areas.

Although LSTM has proven to be superior to most nonparametric methods [29], it is not very good at handling exceedingly long-term data because of its chain structure and error accumulation. We propose an LSTM-based feature enhancement algorithm (LSTM+) that uses attentional mechanisms to perceive extremely long-term high-impact traffic flow values and enables LSTM to improve the extremely long-term memory.

3. LSTM network for prediction

3.1. Notation and problem statement

The traffic flow prediction problem can be described as follows. Suppose that the input traffic flow sequence is denoted as $\mathbf{X} = (y_1, y_2, \ldots, y_{T-1}, \mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n)$, with $y_t \in \mathbb{R}$ representing the traffic flow quantity during the tth time interval, $\mathbf{x}^n$=$(x^n_{T-m}$,...,x^n_T,...,$x^n_{T+m}) \in \mathbb{R}^{2m+1}$, m is the length of the step, x^n_T is the traffic flow volume during the Tth time interval in the nth day before the forecast time (no weekends), and the traffic flow prediction problem is to predict the traffic flow y_T by using the data of $\mathbf{X}$.

3.2. Data preprocessing

For the actual statistics of the traffic flow, because of the various uncertain factors, such as traffic congestion, weather changes, and signal transmission, the original data include certain noisy data. When the noisy data reach a certain volume, the noisy data will affect the accuracy of the traffic flow data forecast. Therefore, the noisy data need to be identified in a certain way. Then, the noisy data must be smoothed.

Since there is no significant difference in the daily traffic flow data during the working day, as shown in Fig. 1, we believe that the traffic flow approximately follows a periodic function, and the period is one day. Considering the randomness, periodicity and instability of the traffic flow data, we use a generic noisy point smoothing method that is based on the trend to fit the traffic flow data.

For a dataset, x_t is the data to be processed, the corresponding historical average is Avg_t, and the historical average is the arithmetic mean of the same moments of the previous few working days (to facilitate the calculation, in this paper, we took 5 days of traffic flow values). First, we subtract the corresponding historical average from the traffic flow value at the current time interval. Second, we set a threshold A to control the proportion of noise. At this time, we believe that a point with a large deviation from the historical data is the noisy point. The value of A can be adjusted by combining the data quality, and the noisy point position x_t is selected if the following conditions are met:

$$|Avg_t - x_t| > A \tag{1}$$

Then, we calculate and modify the deviation of the first noisy data point, x_t. Since it is approximately consistent with the arithmetic progression, we use arithmetic progression to solve the first noisy point offset and use the historical data of the point to calculate the point-smoothed data; the updated value of x_t is:

$$x'_t = 2(x_{t-1} - Avg_{t-1}) - (x_{t-2} - Avg_{t-2}) \tag{2}$$

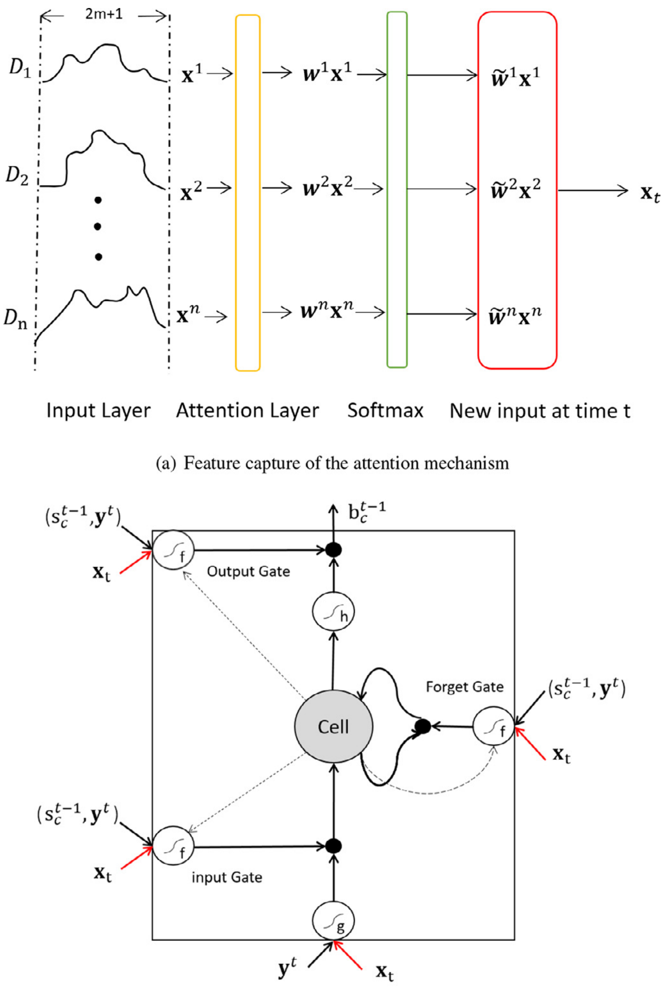
Similarly, we use the last noisy data to calculate the smoothed data x'_{t+d-1}, where d is the number of noisy segments.

$$x'_{t+d-1} = 2(x_{t+d} - Avg_{t+d}) - (x_{t+d+1} - Avg_{t+d+1}) \tag{3}$$

According to formulas (2) and (3), the first smoothed point is updated, and the second point is processed in the same way as the first noisy point. Thus, the countdown second noisy point is consistent with the last-point processing method, i.e., we alternatively smooth the noisy points on the left and right sequences. If the number of noisy data points is even, according to this method, the noisy data are smoothed out; if the number of noisy data points is odd, to make the data closer to the true value, the center noisy point is determined by averaging the first and last smoothing data as follows:

$$x'_{\frac{d-1}{2}+t} = (x'_{\frac{d-1}{2}-1+t} + x'_{\frac{d-1}{2}+1+t})/2 + Avg_{\frac{d-1}{2}+t} \tag{4}$$

Using this noisy point smoothing method, the noisy point can be quickly and adjustably obtained, and the smoothed data are closer to the real data.



(a) Feature capture of the attention mechanism

(b) Feature enhancement based on LSTM

Fig. 2. LSTM+ network structure diagram. (a) The mechanism of the attention layer uses multiple linear regression to sense the effect of $(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n)$ on the predicted value y_t and calculates the weight value of each feature that affects the predicted value. We use the Softmax layer to ensure that the weights are 1 and, finally, obtain the newly added input $\mathbf{x}_t$ encoded by the LSTM unit. (b) The LSTM unit input contains the parameter $\mathbf{x}_t$ from the past, the LSTM cell state of the previous time interval, and the current input sequence $\mathbf{y^t}$=$(y_1^{t-1}, y_2^{t-1},...,y_I^{t-1})$.

3.3. LSTM with feature enhancement

The attention mechanism, such as image captioning [30], machine translation [31], or speech recognition [32], has been widely used in recent years. We noticed that it was used for time-series prediction [33], which combines the hidden states of the previous time steps with the current time step. For traffic flow, the traffic for a certain period of the working day is likely to be similar to the traffic of the same time period last week or the same time period of the previous day, as shown in Fig. 1, which shows the traffic flow value for Friday. Inspired by this observation, we propose LSTM+, which attempts to use the high-impact features added in the sequence to compensate for the lack of learning ability for extremely long sequences. Fig. 2 shows the LSTM+ attention mechanism model. We can approximately divide the LSTM+ network into four layers: the attention mechanism layer, the mixed input layer, the hidden layer, and the output layer. In the attention mechanism layer, the input sequence is $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n)$, where we assume that the values near the tth time interval of each cycle have a high impact on the prediction value. We conducted experiments to prove that as Fig. 3 shows, the values with larger weights are distributed near time t; thus, we chose only the data near the predicted time of each cycle to reduce redundancy input data. Finally, we used the data near time t of different periods to perform multiple linear regression; $\mathbf{w}^n$ is the corresponding weight, and b is the bias. Then, the multiple linear regression function is

$$y_t = \sum_{n=1}^{N} \mathbf{w}^n \mathbf{x}^n + b \tag{5}$$

By minimizing the $L(h_\theta(\mathbf{x}), y_t)$ function, we can obtain the weight and bias parameters in the model. $h_\theta(\mathbf{x})$ is the predicted value. Additionally,

$$\tilde{\mathbf{w}}^n = \frac{exp(\mathbf{w}^n)}{\sum_{n=1}^{N} exp(\mathbf{w}^n)} \tag{6}$$

$\tilde{\mathbf{w}}^n$ is the weight of the nth day before the forecast time, and a Softmax classifier function is used to ensure that the sum of all weights is 1. We add these past traffic flow values with their weights to the LSTM network for training, which enables our LSTM network to include a super-long memory function. After joining the LSTM network, the “input gate”, “forget gate”, and “update gate” in the LSTM network add an understanding of the past traffic flows as shown below:

Input Gate:

$$a_l^t = \sum_{i=1}^{I} w_{il} y_i^{t-1} + \sum_{c=1}^{C} w_{cl} s_c^{t-1} + \sum_{n=1}^{N} \mathbf{w}_{nl} \tilde{\mathbf{w}}^n \mathbf{x}^n \tag{7}$$

$$b_l^t = f(a_l^t) \tag{8}$$

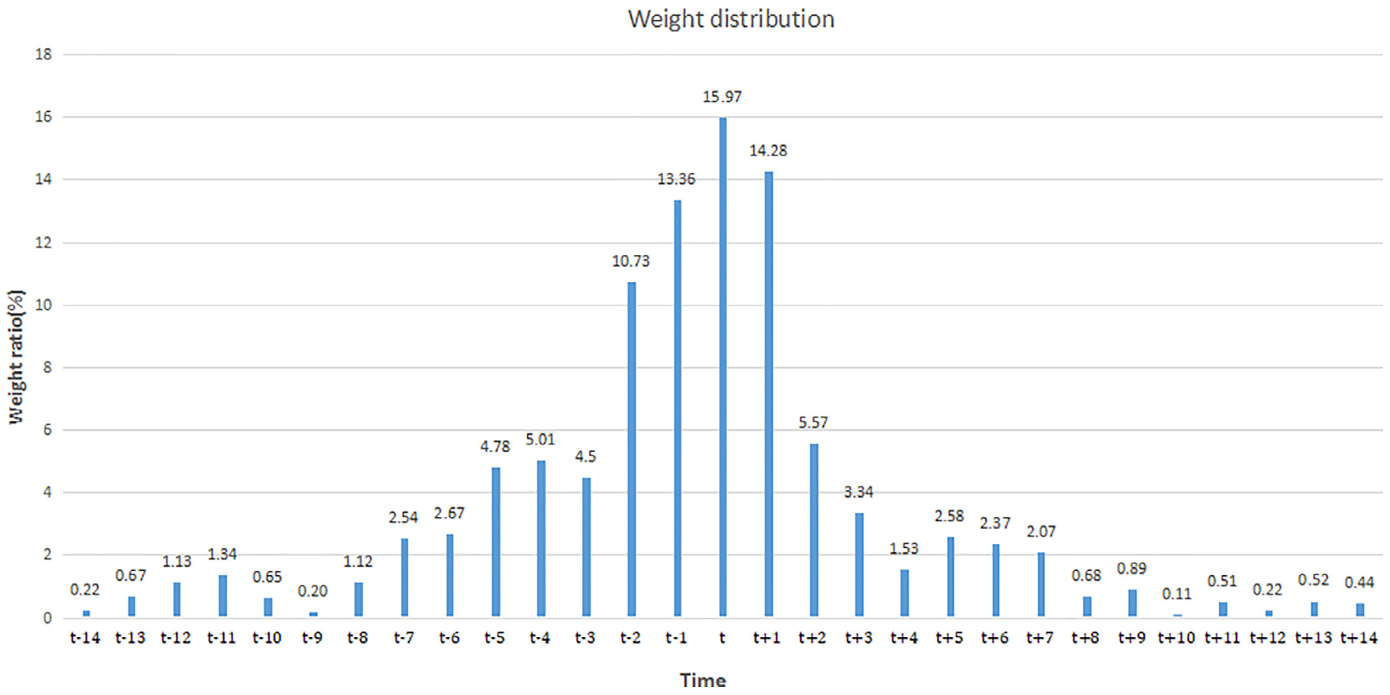


Fig. 3. The impact of the traffic volume of the previous day on the forecast value.

Forget Gate:

$$a_\varphi^t = \sum_{i=1}^{I} w_{i\varphi} y_i^{t-1} + \sum_{c=1}^{C} w_{c\varphi} s_c^{t-1} + \sum_{n=1}^{N} \mathbf{w}_{n\varphi} \tilde{\mathbf{w}}^n \mathbf{x}^n \tag{9}$$

$$b_\varphi^t = f(a_\varphi^t) \tag{10}$$

Cell State:

$$a_c^t = \sum_{i=1}^{I} w_{ic} y_i^t \tag{11}$$

$$s_c^t = b_\varphi^t s_c^{t-1} + b_l^t g(a_c^t) \tag{12}$$

Output Gate:

$$a_o^t = \sum_{i=1}^{I} w_{io} y_i^{t-1} + \sum_{c=1}^{C} w_{co} s_c^{t-1} + \sum_{n=1}^{N} \mathbf{w}_{no} \tilde{\mathbf{w}}^n \mathbf{x}^n \tag{13}$$

$$b_o^t = f(a_o^t) \tag{14}$$

where I represents the length of the current input sequence; N is the number of cycles; l is the state of the input; φ is the state of the forget gate; o is the output state, and w is its corresponding weights; s_c^t is the state of the LSTM cell at time t; c is the number of cells before t; and f and g are the activation functions, which are sigmoids in this paper. The remaining parameters in the formula are identical to the previous description, and the final output is

$$b_c^t = b_o^t h(s_c^t) \tag{15}$$

Finally, LSTM uses the backward pass method to determine the gradient of each weight in LSTM+; then, the weight values of each part are obtained and used to calculate the predicted value. With this clever construction, more parameters are provided for the prediction, and the addition of these parameters improves the LSTM exceedingly long-term prediction.

4. Dataset

4.1. Public Datasets

The public dataset used is the Caltrans Performance Measurement System (PEMS) [34], which was collected in real time from over 39,000 individual detectors; these sensors span the freeway system across all major metropolitan areas in California. It also provides over ten years of data, which include a variety of information that affects vehicle flows. In this paper, we use the detectors in the San Diego/Imperial area in 2017, and the time interval is 5 minutes. Our dataset does not include the weekend, and we attempt to avoid as many holidays as possible. The traffic flow time slot is from March to May (three months in total); the data for two months are selected as the training set, and data for one month are used as the testing set.

4.2. Our datasets

In addition, we select some local datasets. Since the local dataset is not as complete as the California dataset, and there is a large area without data, we avoid the detectors with poor data quality. The time period is only 31 days (October 24 to November 23, 2015), the ratio of the training set to the test set is 3:1, and the time interval is 5 min. We selected sensor data from 72 intercity highways among Hangzhou's Shangcheng District, Xiacheng District and Binjiang District.

4.3. Experimental datasets

In this paper, we mainly demonstrate the superior effect of LSTM+ on short-term traffic flow prediction; thus, we only consider the data for the 5-minute interval. At the same time, to verify the validity and robustness of the model, we collected data from 50 sensors from PEMS and 10 sensors from our data for our study.

5. Experimental results

5.1. Index of performance

The mean absolute error (MAE), mean relative error (MRE), and RMS error (RMSE) are used to evaluate the benefits and drawbacks of the model. The definitions are as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - \hat{f}_i|, \tag{16}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|f_i - \hat{f}_i|}{f_i}, \tag{17}$$

$$RMSE = [\frac{1}{n}\sum_{i=1}^{n}|f_i - \hat{f}_i|^2]^{\frac{1}{2}}, \tag{18}$$

f_i is the traffic data found, $\hat{f}_i$ is the traffic to be predicted, and n is the number of test samples.

5.2. Parameter setting

In terms of hardware, the CPU we use is an Intel i7-4790 3.6 GHz and a NVIDIA GTX Titan X graphics GPU. The LSTM+ networks are implemented based on open-source framework TensorFlow [35]. The learning rate is 0.01, the activation function for the LSTM layer is ‘tanh’ and ‘ReLU’ for the dense layer, the number of hidden units is [32,32], the batch size is 32, and the output layer dimension is 1.

5.3. Data processing contrast experiment

First, we adjust the threshold A to identify the location of the noisy point. In this paper, for different datasets, the proportion of the most appropriate noise is inconsistent. If the proportion of the adjusted noise data is too high, not only will the data be distorted, but the prediction accuracy will also be reduced. If the amount of data adjustment is too low, it will not improve the prediction accuracy. After many experiments, the data adjustment ratio is better at approximately 4.7%, and in all the related experiments, the ratio of our data adjustment is the same. After detecting the locations of the noisy data, the noisy smoothing begins. We use a flow-based data cross-noise smoothing method based on trend fitting. The specific noise smoothing scheme has been specifically described before. It should be noted that the noisy data smoothing used here does not apply to the smoothing of long-duration noise segments and that long-term noise smoothing will result in data smoothing distortion.

To verify the validity of data smoothing, we used LSTM algorithms to compare the experiments with other smoothing methods:

(1) *Standard LSTM algorithm (LSTM)*: in this algorithm, a no data preprocessing approach is used.

(2) *Preprocessed LSTM algorithm (pLSTM)*: its training data are preprocessed using our method.

(3) *The hybrid simple moving and LSTM algorithm (hsmLSTM)*: it uses a simple moving method to smooth the noise in the traffic flow, i.e., the *i*th preprocessed data are updated by the data of the previous four traffic flows.

$$x'_t = \frac{1}{4}\sum_{i=1}^{5}(x_{t-1} + x_{t-2} + x_{t-3} + x_{t-4}) \tag{19}$$

with $i > 4$, where $x'_1 = x_1,\ x'_2 = x_2,\ x'_3 = x_3,\ x'_4 = x_4$.

(4) *The hybrid weight moving and LSTM algorithm (hwmLSTM)*: it preprocesses noise in the traffic flow data using a weighted movement method, and the *i*th preprocessed data are updated by the data of the previous four traffic flows as:

$$x'_t = \frac{1}{10}\sum_{i=1}^{5}(4x_{t-1} + 3x_{t-2} + 2x_{t-3} + x_{t-4}) \tag{20}$$

with $i > 4$, where $x'_1 = x_1,\ x'_2 = x_2,\ x'_3 = x_3,\ x'_4 = x_4$.

Table 1
The quantitative results for pems and our datasets based on different methods.

Model	MAE	MRE(%)	RMSE
LSTM	20.59	12.89	30.86
pLSTM	19.98	12.52	30.21
hsmLSTM	20.19	12.74	30.52
hwmLSTM	20.07	12.61	30.36

Table 2
The evaluation values for the pLSTM and pLSTM+ model.

Sensor ID	pLSTM			pLSTM+		
	MAE	MRE(%)	RMSE	MAE	MRE(%)	RMSE
2272	31.93	10.19	44.54	29.54	9.50	40.95
7762	21.28	30.32	9.79	19.06	28.54	8.97
7782	24.04	10.41	33.72	22.16	9.87	31.79
8078	20.38	9.19	27.71	19.33	8.46	26.03
8544	20.12	11.01	27.61	18.62	9.98	25.54
9828	24.88	10.07	35.67	23.65	9.57	34.71
12320*	14.50	48.75	22.93	12.93	43.89	21.02
26803*	19.99	11.50	28.84	18.70	11.03	27.53

(Note: “pLSTM+“ is the LSTM+ algorithm in which the training data were preprocessed by our method.)

Table 3
Average performance index on the LSTM, pLSTM, AND pLSTM+ models.

Model	MAE	MRE(%)	RMSE
LSTM	20.59	12.89	30.86
pLSTM	19.98	12.52	30.21
pLSTM+	19.06	11.87	28.78

Table 4 shows the comparison results, as shown in the table, the average accuracy of the pLSTM is 12.52%. We found that pLSTM obtains better results than other algorithms in terms of traffic-flow prediction accuracy. The reason is that pLSTM takes into account the trend of the traffic flow data.

5.4. Super-long sequence feature enhancement contrast experiment

We use the attention mechanism to solve the high-impact characteristic factors according to the introduced algorithm with a time span of 1 week. In other words, we look for the high-impact feature values among 1152 pieces of traffic flow data (the time interval is 5 min, one hour contains 12 pieces of data, and one day contains 288 pieces of data; the data predicted for this day are not considered, and a total of four days are considered). According to the principle described in the third part of the paper, we only need to identify those factors near the predicted time of each day, at the same time, to ensure data validity and less data redundancy; in the experiment, the value of m is 5.

To evaluate the performance of our proposed model, we conducted a comparison experiment with the pLSTM model. Due to space constraints, we selected the evaluation data of 8 sensors for display in Table 2; the sensor ID with the * mark shows our datasets. In addition, Fig. 4 shows the prediction results in the “7782” detector, and Table 3 shows the average evaluation parameters for all test datasets. As show in Fig. 4(a), it may oberve

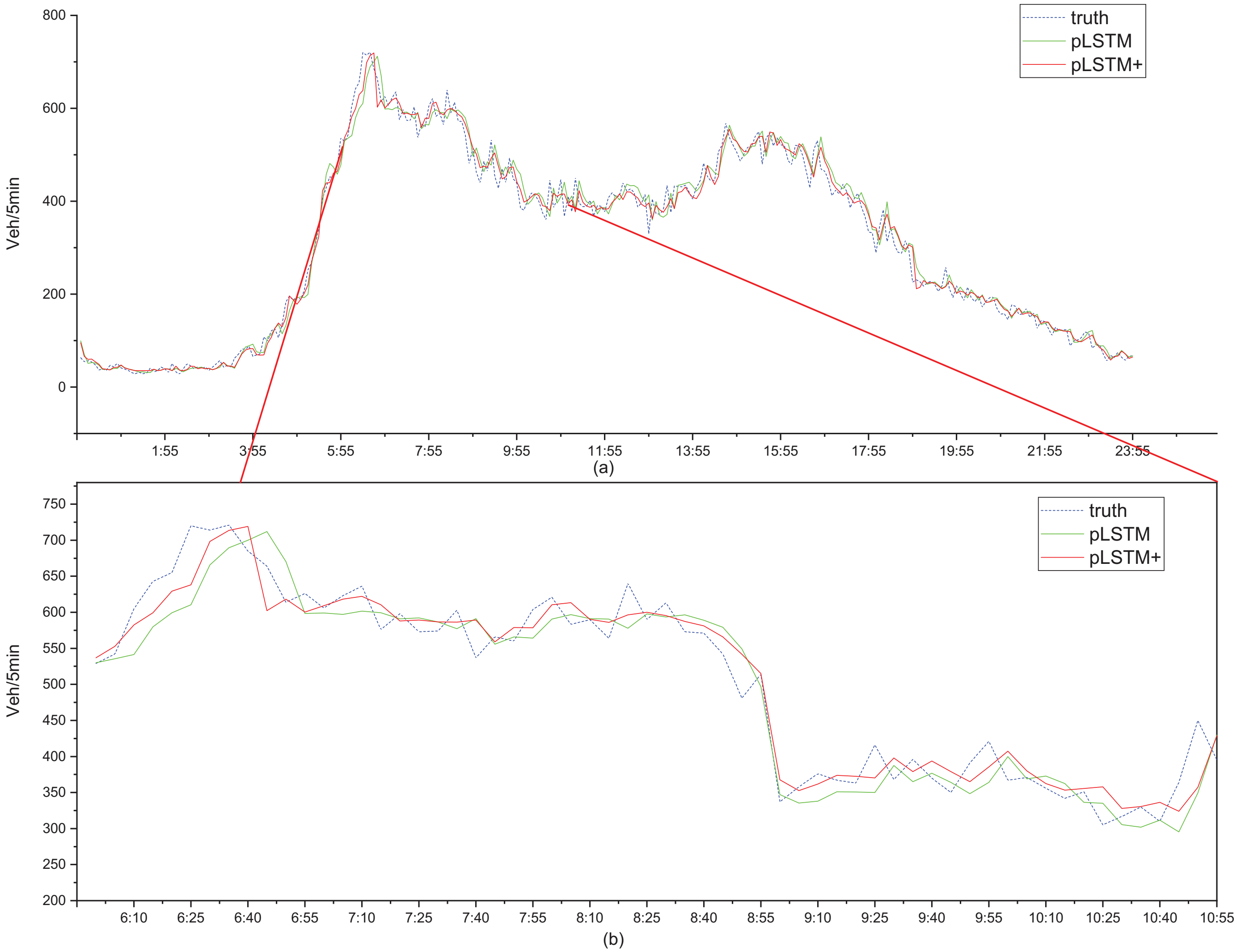


Fig. 4. Prediction performance on pLSTM, pLSTM+ in a day.(a)Experimental Results in the "7782" detector.(b) Zooming detail in the detector.

that the red curve is closer to the true blue value, to facilitate observation, we enlarged part of this diagram, from the enlarged diagram, we can see that most of the results predicted by the pLSTM+ algorithm are better than those of pLSTM. Meanwhile, from Tables 2 and 3 we observe that our algorithm pLSTM+ has smaller error. Compared with the standard LSTM algorithm, our MAE improved by 7.43% on average, MRE improved by 7.91% on average, and RMSE improved by 6.74% on average. Simultaneously, our algorithm improved by 4.60% on MAE compared to pLSTM, MRE improved by 5.19% on average, and RMSE improved by 4.73% on average. It is worth noting that the traffic flow values in different locations are quite different, the traffic flow fluctuations are uneven, and when the traffic volume is small and the fluctuation is large, this will result in similar MAE, but there is a large difference between MRE and RMSE, as shown in the data of the 7782 sensors in Table 2. These improvements are due to the importance of our algorithm considering traffic flow trends and historical data, and these test results show that our model can learn more distant dependencies.

Table 4
Comparison with other algorithms.

Model	15 min		
	MAE	MRE(%)	RMSE
BPNN [19]	60.8	10.9	94.1
SVM [20]	38.7	8	62.3
RBFNN [21]	38.3	7.4	55.9
SAE [18]	34.1	6.75	50
LSTM [36]	34.4	6.68	49.87
LSTM+	**31.8**	**6.54**	**48.37**

(Note: to ensure the fairness of the experiment, the data used by all models do not carry out noise processing.)

5.5. Comparison experiment with other algorithms

Our LSTM+ was also compared with BPNN, the random walk (RW) forecast method, the support vector machine (SVM) method, and the radial basis function (RBF) NN model. To compare other methods fairly, we used the same parameter settings and training test data as [18]. Since our algorithm aims to resolve extremely long-term memory problems, we only selected 15 minutes for comparison experiments as shown in Table 4. Table 4 shows that deep learning is better than a shallow neural network algorithm. There is no major difference between the SAE and LSTM. Within 15 min, BPNN's MRE is 10.9%, SVM is 8%, RBFNN is 7.4%, SAE is 6.75%, LSTM is 6.68%, and our LSTM+MRE is 6.54%. Compared with other algorithms, MRE has a certain advantages. In the experiment, all of our LSTM+ data performances were excellent. In particular, for MAE, our LSTM+ prediction accuracy was 6.7% higher than that of SAE and 7.6% higher than that of LSTM with no treatment; thus, the LSTM+ is not the optimal choice to use over a 15 min interval. In the test, LSTM+ can be superior in dealing with 5- or 1-minute-long historical data, which is also our original intent to improve the LSTM algorithm.

In the experiment, we can classify BPNN and RBFNN as shallow neural networks. This class lacks the ability to learn more complex networks other than deep learning networks. SVM and SAE (stacked auto encoders) can be classified into one category; SVM lacks timing considerations, whereas SAE considers the time characteristics and spatial characteristics. It is not sufficient to emphasize only the first few moments for traffic prediction. LSTM and LSTM+ can be divided into one category. Although they only consider the influence of time characteristics, their performance is similar to SAE and requires less cost. It should be noted that if a topological relationship is formed between multiple sensors, and the model considers the influence of the current time interval traffic flow and the past time interval traffic flow, such a model will be more accurate.

6. Conclusion

An improved LSTM+ solution for LSTM has been proposed to resolve the lack of extremely long-term memory. This algorithm inherits the principle of LSTM in long- and short-term memory and supplements the shortcomings of LSTM at exceedingly long distances. This remarkably long distance problem exists in the prediction of traffic flows at a time interval of 1 or 5 min. This problem is also present at a time interval of 15 min, but it is not as obvious as the performance of the 1- and 5-min intervals. The method performs better than its prototype LSTM in both PEMS and our datasets and shows improvement over other model predictions. In future work, we plan to improve the model by combining time and space features while considering the impact of historical data.

Acknowledgment

This work was supported in part by the Natural Science Foundation of China (Nos. 61602407, 61472363, and 61572436), Natural Science Foundation of Zhejiang Province (No. LY18F010007), and Opening Foundation of Engineering Research Center of Intelligent Transport of Zhejiang Province (No. 2016ERCITZJ-KF02).

References

[1] G. Lai, W.-C. Chang, Y. Yang, H. Liu, Modeling long-and short-term temporal patterns with deep neural networks, in: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2018, pp. 95–104.

[2] Y. Song, C. Ning, L. Fa, X. Liu, X. Shu, Z. Li, J. Tang, Multi-part boosting lstms for skeleton based human activity analysis, in: Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), IEEE, 2017, pp. 605–608.

[3] X. Shu, J. Tang, G.J. Qi, et al., Concurrence-Aware Long Short-Term Sub-Memories for Person-Person Action Recognition, CVPR workshop (2017) 143–152.

[4] K. Cho, B. Van Merriënboer, D. Bahdanau, et al., On the properties of neural machine translation: Encoder-decoder approaches, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (2014) 103–111.

[5] G.F. Newell, A simplified theory of kinematic waves in highway traffic, part iii: multi-destination flows, Transp. Res. Part B: Methodol. 27 (4) (1993) 305–313.

[6] K. Nagel, M. Schreckenberg, A cellular automaton model for freeway traffic, J. Phys. I 2 (12) (1992) 2221–2229.

[7] M. Van Der Voort, M. Dougherty, S. Watson, Combining Kohonen maps with arima time series models to forecast traffic flow, Transp. Res. Part C: Emerg. Technol. 4 (5) (1996) 307–318.

[8] S. Lee, D. Fambro, Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting, Transp. Res. Rec. J. Transp. Res. Board 1678 (1999) 179–188.

[9] B. Williams, Multivariate vehicular traffic flow prediction: evaluation of arimax modeling, Transp. Res. Rec. J. Transp. Res. Board 1776 (2001) 194–200.

[10] Y. Kamarianakis, P. Prastacos, Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches, Transp. Res. Rec. J. Transp. Res. Board 1857 (2003) 74–84.

[11] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results, J. Transp. Eng. 129 (6) (2003) 664–672.

[12] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, L.D. Han, Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions, Expert Syst. Appl. 36 (3) (2009) 6164–6173.

[13] K. Kumar, M. Parida, V. Katiyar, Short term traffic flow prediction for a non urban highway using artificial neural network, Proc. Soc. Behav. Sci. 104 (2013) 755–764.

[14] G.A. Davis, N.L. Nihan, Nonparametric regression and short-term freeway traffic forecasting, J. Transp. Eng. 117 (2) (1991) 178–188.

[15] S. Wu, Z. Yang, X. Zhu, B. Yu, Improved k-nn for short-term traffic forecasting using temporal and spatial information, J. Transp. Eng. 140 (7) (2014) 04014026.

[16] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in neural information processing systems, 2012, pp. 1097–1105.

[17] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: Deep belief networks with multitask learning., IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2191–2201.

[18] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, et al., Traffic flow prediction with big data: a deep learning approach., IEEE Trans. Intell. Transp. Syst. 16 (2) (2015) 865–873.

[19] Y. Cai, Ipso-bpnn for short-term traffic flow prediction, Jisuanji Gongcheng yu Yingyong (Comput. Eng. Appl.) 48 (27) (2012) 239–243.

[20] Y. Zhao-sheng, W. Yuan, G. Qing, Short-term traffic flow prediction method based on svm, J. Jilin Univ. (Eng. Technol. Ed.) 6 (2006) 009.

[21] H. Leung, T. Lo, S. Wang, Prediction of noisy chaotic time series using an optimal radial basis function neural network, IEEE Trans. Neural Netw. 12 (5) (2001) 1163–1172.

[22] J. Wang, F. Hu, L. Li, Deep bi-directional long short-term memory model for short-term traffic flow prediction, in: Proceedings of the International Conference on Neural Information Processing, Springer, 2017, pp. 306–316.

[23] B. Liu, J. Cheng, K. Cai, P. Shi, X. Tang, Singular point probability improve lstm network performance for long-term traffic flow prediction, in: National Conference of Theoretical Computer Science, Springer, 2017, pp. 328–340.

[24] X. Cheng, R. Zhang, J. Zhou, et al., Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting, International Joint Conference on Neural Networks (2018) 1–8.

[25] Y. Liu, H. Zheng, X. Feng, Z. Chen, Short-term traffic flow prediction with conv-lstm, in: Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), IEEE, 2017, pp. 1–6.

[26] Y. Wu, H. Tan, Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework, arXiv:1612.01022 (2016).

[27] J. Wang, Q. Gu, J. Wu, G. Liu, Z. Xiong, Traffic speed prediction and congestion source exploration: a deep learning method, in: Proceedings of the 16th International Conference on Data Mining (ICDM), 2016 IEEE, IEEE, 2016, pp. 499–508.

[28] A. Azzouni, G. Pujolle, A long short-term memory recurrent neural network framework for network traffic matrix prediction, arXiv:1705.05690 (2017).

[29] Y. Tian, L. Pan, Predicting short-term traffic flow by long short-term memory recurrent neural network, in: Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), IEEE, 2015, pp. 153–158.

[30] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: Proceedings of the International conference on machine learning, 2015, pp. 2048–2057.

[31] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, ICLR (2015) 56–65.

[32] J.K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based models for speech recognition, in: Proceedings of the Advances in neural information processing systems, 2015, pp. 577–585.

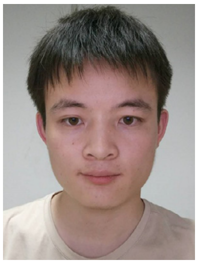
[33] Y. Qin, D. Song, H. Chen, et al., A dual-stage attention-based recurrent neural network for time series prediction, IJCAI (2017) 2627–2633.

[34] P.M.S.P. Caltrans, [Online]. Available:http://pems.dot.ca.gov (2014).

[35] M. Abadi, P. Barham, J. Chen, et al., Tensorflow: a system for large-scale machine learning, OSDI (2016) 265–283.

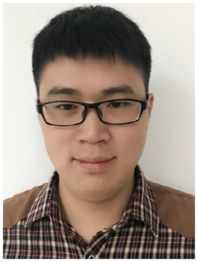
[36] X. Ma, Z. Tao, Y. Wang, H. Yu, Y. Wang, Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transp. Res. Part C: Emerg. Technol. 54 (2015) 187–197.

Bailin Yang received the Doctor's degree in department of computer science from Zhejiang University in 2007. He is an professor in the department of computer and electronic engineering of Zhejiang Gongshang University. His research interests are in virtual reality, mobile graphics, data mining and mobile game.

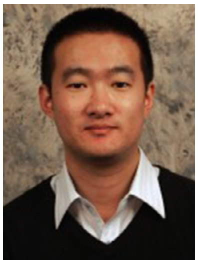


Shulin Sun received his Bachelor degree in Geographic information science from Anhui University Of Science and Technology in 2017, Huainan, China and Master degree in Technology of Computer Application from Zhejiang Gongshang University, Hangzhou, China. His research interests include machine learning and data mining.

Jianyuan Li is received the B.Sc. in Applied Electronic Technology and the M.Sc. in Computer Science & Technology from Shan Xi Normal University, Xi'an, China, in 2001 and 2007, respectively. And he received the Ph.D. degree in Computer Science & Technology from Tongji University, Shanghai, China, in 2012. Then he had a postdoctoral research fellow position in Enjoyor Co. Ltd, Hangzhou, China, from 2014 to 2017. He is currently the Chief Technology Officer (CTO) in the research institute of Enjoyor Co. Ltd, Hangzhou, China. His research interests include machine learning and data mining.



Xianxuan Lin is received Bachelor's degree in Electronic Information Engineering from Yang-En University in 2014, Quan-zhou, China. Master's degrees in Technology of Computer Application from Zhejiang Gongshang University, Hangzhou, China. His research interests are in color processing of image and Sequential data prediction.



Yan Tian is received BSc in Communication Engineering from Hang-zhou Dianzi University, Hangzhou, China, Ph.D. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 2005 and 2011, respectively. Then he had a postdoctoral research fellow position (2012–2015) in the Department of Information and Electronic Engineering, Zhejiang University, Hangzhou, China. He is currently a Lecture of Computer Science and Technology in the School of Computer Science and Information Engineering, Zhejiang Gongshang University, China. His current interests are machine learning and pattern recognition, and he also works on image and video Analysis.