Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 4858546, 12 pages
https://doi.org/10.1155/2019/4858546

Research Article

A Spatial-Temporal Hybrid Model for Short-Term Traffic Prediction

Fei Lin, Yudi Xu, Yang Yang, and Hong Ma

School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Yudi Xu; plkm357@126.com

Received 15 September 2018; Revised 22 November 2018; Accepted 19 December 2018; Published 14 January 2019

Academic Editor: Roberta Di Pace

Accurate and timely short-term traffic prediction is important for Intelligent Transportation System (ITS) to solve the traffic problem. This paper presents a hybrid model called SpAE-LSTM. This model considers the temporal and spatial features of traffic flow and it consists of sparse autoencoder and long short-term memory (LSTM) network based on memory units. Sparse autoencoder extracts the spatial features within the spatial-temporal matrix via full connected layers. It cooperates with the LSTM network to capture the spatial-temporal features of traffic flow evolution and make prediction. To validate the performance of the SpAE-LSTM, we implement it on the real-world traffic data from Qingyang District of Chengdu city, China, and compare it with advanced traffic prediction models, such as models only based on LSTM or SAE. The results demonstrate that the proposed model reduces the mean absolute percent error by more than 15%. The robustness of the proposed model is also validated and the mean absolute percent error on more than 86% road segments is below 20%. This research provides strong evidence suggesting that the proposed SpAE-LSTM effectively captures the spatial-temporal features of the traffic flow and achieves promising results.

1. Introduction

Traffic problems are becoming more and more serious with the increasing number of vehicles. The growing traffic problems not only cause the environmental pollution but also hinder the development of the economy. Intelligent Transportation System (ITS) is considered as an effective tool to solve the problems. Moreover, traffic prediction is the key of ITS. The real-time and accurate prediction of short-term traffic in the road network helps to better analyze the traffic condition of the road network, and it also plays an important role in the road network traffic planning and traffic optimization control. Traffic prediction is not a new issue, and the research on traffic prediction also has a long history. In the past few decades, many researchers turned their attention to short-term traffic prediction. With the development, numerous approaches were proposed to enhance the accuracy and efficiency of the prediction. In general, all the approaches can be approximately divided into two categories: parametric approaches and nonparametric approaches [1].

Parametric approaches mean their structures are predetermined according to some certain theoretical assumptions. And the model parameters usually can be computed with the empirical data. Simulation approaches [2–4] are typical parametric approaches and they are usually used to depict traffic situation based on the traffic flow theory with three key parameters (speed, density, and flow). Many researches of traffic flow prediction are based on time series models; for example, Ahmed et al. applied the ARIMA model to predict expressway traffic flow [5] and Yang et al. made short-term traffic prediction by similar search of time series [6]. Actually, traffic condition is heavily coupled with human behaviors, and it is hard and inefficient to describe by models with fixed structures and parameters. Thus applying these approaches to make short-term traffic prediction is unreliable.

In order to overcome the shortcomings of the parametric approaches, researchers turned their attention to the nonparametric approaches. Unlike parametric approaches, nonparametric approaches' model structures and parameters depend on concrete issues, so they are called data-driven methods. In 1984, Kalman Filtering was used to predict traffic volume [7], and Lint et al. [8] and Deng et al. [9] further successfully proved its effectiveness in traffic flow prediction. Support Vector Machine (SVM) was also a useful method for short-term traffic flow prediction [10]. Classical statistical models like Bayesian network [11] are also used to predict traffic flow.

The approaches mentioned above may be effective when the traffic condition is simple and small-scale, but when the traffic condition becomes large-scale and the stochasticity is more complex, they do not work. With the development of the traffic sensor technologies, traffic data is more detailed, abundant, and large-scale. We have entered the era of big data transportation, in which most traditional traffic flow prediction systems and models are somewhat unsatisfying. So we need to further develop the traditional models or find new approaches. In recent years, intelligence algorithms have drawn plenty of academic and industrial interests, especially in the domain of transportation. Different from other intelligent algorithms, algorithms based on artificial neural networks are more widely used. ANN algorithm can use its multilayer architecture to extract inherent features in data from the lowest level to the highest level and find large amounts of hidden structures of the data. This is necessary for prediction of the complicated and stochastic traffic flow. In the domain of traffic flow prediction, numerous variant ANN were developed, such as backpropagation neural networks (BPNNs) [12], the modular neural network (MNN) [13], radial basis frequency neural network (RBFNN) [14], and recurrent neural network (RNN). Among them, RNN is special, and it has the temporal component [15, 16]. RNN uses internal memory units to process time-series inputs, so it is suitable to capture the temporal and spatial evolution of traffic flow, volume, and speed [17]. As RNN can not handle the long-term time series effectively, LSTM [18] is naturally considered as an improved approach. In 2015, Ma et al. proposed a novel LSTM NN to make short-term travel speed prediction [17]. In their study, the LSTM network consists of three layers and the hidden layer is composed of memory blocks; even the LSTM network can automatically determine the optimal time lags by proper training method. From then on, many researchers apply LSTM and its variant to make the prediction in traffic domain. Considering the spatial correlation between roads, a two-dimensional LSTM NN was proposed to make traffic flow prediction [19].

Actually, traffic flow is affected by many factors and single-component models are not enough to complete the task of traffic prediction. So hybrid models are increasingly used to improve the accuracy of traffic speed prediction. Although there are already many hybrid models for traffic speed prediction, most of them simply merge the temporal and spatial features, ignoring the temporal trend of the traffic flow and the inner relationships between the spatial features.

According to the aforementioned discussion, the contributions of this paper can be summarized as follows. Firstly, we analyze the spatial correlations of nearby road segments and propose a spatial-temporal matrix based on the statistical theories and methods. Secondly, a hybrid model called SpAE-LSTM is proposed for traffic speed prediction in this paper. By combining the ability for capturing spatial features of sparse autoencoder and the ability for capturing temporal dependence of long short-term memory (LSTM), the spatial-temporal features and their inherent structures are extracted from the raw traffic data. The spatial and temporal features of traffic flow are inherently fused by a new fusion method in the modeling. Thirdly, we use the average traffic speed from GPS probe data rather than other sensor data to make traffic speed prediction. It is flexible and cost-effective, because we do not need the additional sensors fixed on the roads. In addition, the proposed hybrid model shows a superior performance for traffic speed prediction.

The remainder of this paper is organized as follows. Section 2 describes the traffic prediction problem studied in this paper. Section 3 presents the methods used and the hybrid model for traffic prediction. Experiments based on traffic dataset from Chengdu and a comparison of performance with other representative prediction models are given in Section 4. Conclusion and future work are reported at the end of this paper.

2. Problem Description

The aim of traffic prediction is to evaluate the future traffic condition so that we can provide valuable information to the traffic managers and participants. They can take some measures in advance to make sure traffic flows smoothly via the predicted information. Therefore, it is important to improve the accuracy of traffic prediction. According to traffic flow theory, traffic condition is usually assessed by three parameters (speed, density, and flow). Among these parameters, the speed of traffic flow is an important criterion for assessing traffic condition. Thus, the goal of this paper is to make traffic speed prediction. As Figure 1 shows, traffic flows of different roads interact with each other through intersections. In order to achieve better prediction, we divide the road network into road segments according to intersections. For example, Figure 2 depicts the selected area is divided into 10 road segments $r_1, r_2, \ldots, r_{10}$. The average traffic speed v_i within a time frame can be applied to evaluate the traffic condition of the road segment r_i [20]. A time frame means a period of time. GPS probe data collected from floating cars contains a time stamp, current GPS position, instantaneous speed, etc. Therefore, this paper calculates average traffic speed with GPS probe data. In a time frame j, if the GPS positions of the vehicles $h_1, h_2, \ldots, h_n$ are on the road segment r_i, and their corresponding instant speeds are $v_{h1}, v_{h2}, \ldots, v_{hn}$, then the average speed $\overline{v}_{i,j} = (1/n)\sum_{i=0}^{n} v_{hi}$. If an area has n road segments $[r_1, r_2, \ldots r_n]$, then the traffic condition of the road network in j time frame is represented as a vector $S_j = [\overline{v}_{j,1}, \overline{v}_{j,2}, \overline{v}_{j,3}, \ldots, \overline{v}_{j,n}]$. At the same time, the historical data of each road segment corresponds to a time series, like $V_i = [\overline{v}_{1,i}, \overline{v}_{2,i}, \overline{v}_{3,i}, \ldots, \overline{v}_{p,i}]$ which corresponds to the road segment r_i, where p indicates the number of time frames in historical data. So the traffic data of an area with n road segments over p time frames can be described as a spatial-temporal matrix C:

$$C = \begin{bmatrix} V_1^T & V_2^T & \ldots & V_p^T \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \ldots \\ S_p \end{bmatrix}$$

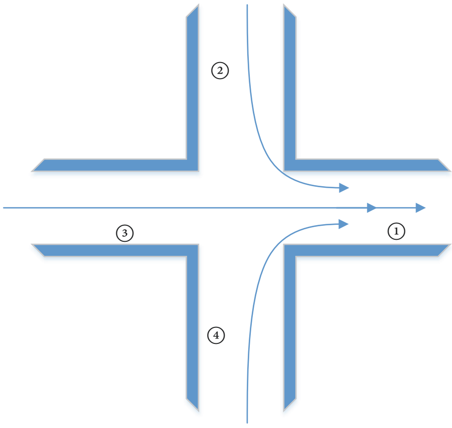


Figure 1: An example of the interaction of traffic flow in different road segments at intersections.

Figure 2: An example of dividing the road network into road segments.

$$= \begin{bmatrix} \overline{v}_{1,1} & \overline{v}_{1,2} & \cdots & \overline{v}_{1,n} \\ \overline{v}_{2,1} & \overline{v}_{2,2} & \cdots & \overline{v}_{2,n} \\ \cdot & \cdot & \cdot \quad \cdots & \\ \overline{v}_{p,1} & \overline{v}_{p,2} & \cdots & \overline{v}_{p,n} \end{bmatrix} \qquad (1)$$

3. Methods

3.1. Sparse Autoencoder. Sparse autoencoder can automatically learn features from unlabeled data and gives a better characterization than the original data. In practice, the features found by sparse autoencoder can be applied to replace the original data, which often leads to better results.

As Figure 3 shows, the sparse autoencoder is a neural network which has one hidden layer, one input layer, and one output layer, which is trained to reproduce its input. Typically, an autoencoder takes a vector input x, encodes it as a hidden layer h, and decodes it into a reconstruction y. For instance, if a training dataset is $\{x_1, x_2, x_3, \ldots x_n\}$, and an input x_i is first encoded to a latent representation $h(x_i)$ according to (2), then latent representation $h(x_i)$ is decoded back to a reconstruction $y(x_i)$ based on (3). The following equations are shown as

$$h(x) = g_h(W_{sh} * x + b_{sh}) \qquad (2)$$

$$y(x) = g_o(W_{so} * h(x) + b_{so}) \qquad (3)$$

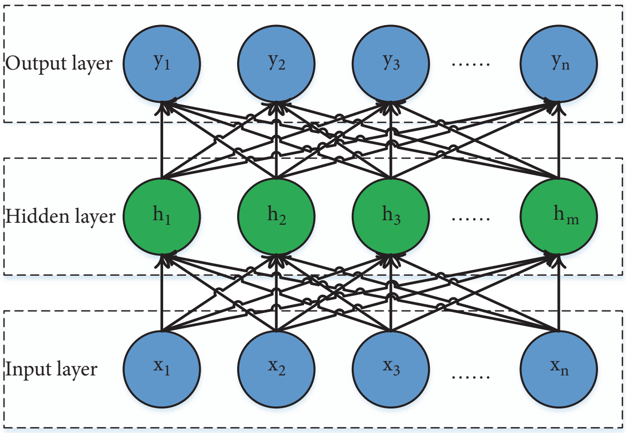


Figure 3: The structure of the autoencoder.

where $g_h(\cdot)$ and $g_o(\cdot)$, respectively, denote the activation of hidden layer and output layer; W_{sh}, W_{so} are the encoding matrix, decoding matrix respectively and b_{sh}, b_{so} are the encoding bias vector and decoding bias vector, respectively.

In this paper, we use the squared loss function $L(x, y)$ to measure the reconstruction error [21]. By minimizing the difference between the input x and the reconstruction y, we can obtain the model parameters, denoted by θ, as shown in

$$\theta = \arg\min_{\theta} L(x, y) = \arg\min_{\theta} \frac{1}{2}\sum_{i=1}^{n} \left\| x_i - y(x_i) \right\|^2 \qquad (4)$$

The autoencoder with more hidden units than inputs can yield useful representations when it is trained with stochastic gradient descent. However, if the hidden layer is the same size or larger than the input layer, an autoencoder could potentially learn the identity function and become useless (e.g., just by copying the input) [22]. The current practice indicates that the above issue could be avoided by constraining the encoding via other restrictions such as sparsity constraints. In this study, the hidden units are constrained by sparseness. For the sigmoid activation function, the neuron whose output close to 1 is considered as being active, while the neuron whose output close to 0 is considered as being inactive. The sparseness means the most neurons of hidden layer are inactive. To achieve this goal, let the average activation of each neuron be close to a ρ (equation (6)) and ρ is a sparsity parameter, usually a small value close to 0 (e.g., $\rho = 0.05$). The average activation $\widehat{\rho}_j$ of the hidden neuron j is calculated as follows:

$$\widehat{\rho}_j = \frac{1}{l}\sum_{i=1}^{l} \left[h_j(x_i) \right] \qquad (5)$$

$$\widehat{\rho} = \rho \qquad (6)$$

where x_i denotes the input, l is the scale of the train data, and $h_j(\cdot)$ represents the function to calculate the activation.

In order to achieve the sparseness restriction, a penalty term is added into the cost function. This penalty term will penalize those cases where $\widehat{\rho}_j$ and ρ are significantly different

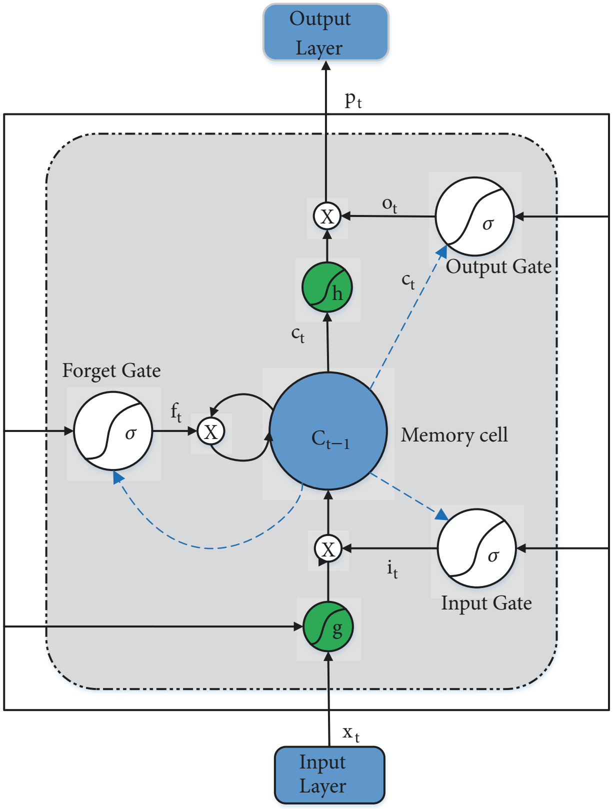


FIGURE 4: The structure of the LSTM.

so that the average activation of the hidden neuron is kept in a relatively small range. The widely used penalty term as the following is based on the *KL divergence*:

$$\sum_{j=1}^{m} KL\left(\rho \parallel \widehat{\rho}_j\right) = \sum_{j=1}^{m} \rho \log \frac{\rho}{\widehat{\rho}_j} + (1-\rho)\log \frac{1-\rho}{1-\widehat{\rho}_j} \quad (7)$$

where m is the number of hidden neurons.

In this study, the autoencoder is implemented using the backpropagation algorithm, and the overall cost function (reconstruction error with a sparsity constraint) is shown as

$$L_S(x,y) = L(x,y) + \alpha \sum_{j=1}^{m} KL\left(\rho \parallel \widehat{\rho}_j\right) \quad (8)$$

where α is the weight of the sparsity term.

3.2. LSTM. The historical traffic average speed data of one road segment can be regarded as a time series. Long short-term memory (LSTM) is an effective method to deal with the prediction of time series. LSTM has three layers, including one input layer, one recurrent hidden layer, and one output layer. The recurrent hidden layer of LSTM has the function of memory, which is the key to the fact that LSTM can learn long-term dependency of time series. As Figure 4 shows, a memory cell is the core of the LSTM structure, which memorize the temporal state (cell state) with self-connections and a pair of adaptive, multiplicative gating units. In addition, there are three gates named input gate, output gate, and forget gate in the LSTM structure. Input gate and output gate control the input and output activations. The forget gate decides which information in cell state should be discarded for preventing the cell state values growing without bound.

The input gate and forget gate receive the input x_t, the previous state c_{t-1} of the memory cell, and the previous output h_{t-1} and the operations of these gates can be denoted by the following equations:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i\right) \quad (9)$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1} + b_f\right) \quad (10)$$

The cell state update is calculated by the following equations:

$$\widetilde{c}_t = g\left(W_c x_t + U_c h_{t-1} + b_c\right) \quad (11)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t \quad (12)$$

The output gate receives the input x_t, the previous output h_{t-1}, and current cell state c_t; its operation and finally output are denoted by the following equations:

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_t + b_o\right) \quad (13)$$

$$h_t = o_t \odot h\left(c_t\right) \quad (14)$$

where i_t, f_t, and o_t are the output of three gates, $\widetilde{c}$ is the new cell state, c_t is the updated cell state, and h_t is the final output. $\odot$ denotes the scalar product between two vectors; $\sigma(\cdot)$ is the gate activation function and denotes the standard logistics sigmoid function; $g(\cdot)$ and $h(\cdot)$ usually denote the tanh function. $W_i, U_i, V_i, W_f, U_f, V_f, W_c, U_c, W_o, U_o$, and V_o are weight matrixes and b_i, b_f, b_c, b_o are bias vectors.

3.3. The Hybrid Model. According to the traffic flow theory, the traffic flows of different road segments will affect each other, especially the road segments that are connected and adjacent in the geography. In this paper, we use the method mentioned in the second section to process the traffic data. The data is divided into some time frames by time and every time frame corresponds to a vector $S_j = [\overline{v}_{j,1}, \overline{v}_{j,2}, \overline{v}_{j,3}, \ldots, \overline{v}_{j,n}]$, where $\overline{v}_{j,n}$ represents the average traffic speed of the road segment n in time frame j.

In this paper, a model called **SpAE-LSTM** is proposed to make traffic speed prediction. The spatial relationships between each road segment are used to generate the features of the traffic flow for a certain road segment. In other words, if a target road segment is decided, its average traffic speed is the target value, and the other average traffic speed of related road segments will be used to extract the features. According to the traffic flow theory, the relationships between traffic flows of road segments are not simply linear or monotonic relationships. So, the sparse autoencoder is applied to extract spatial features and the focus of LSTM is to capture the trend in the temporality and extract the temporal features. Through extracting the features and transform, the data will be better represented. The model's structure is shown in

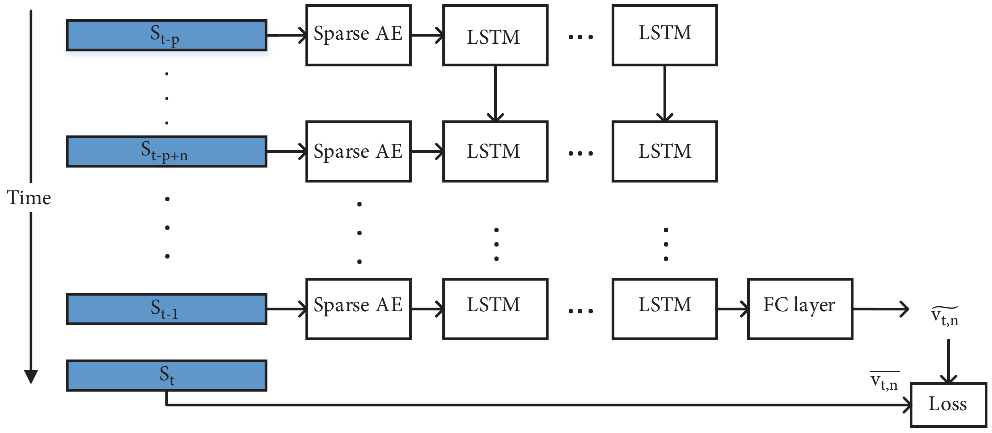


Figure 5: The structure of the model.

Figure 5. $[S_{t-p}, S_{t-p+1}, \ldots, S_{t-1}]$ is the input to train the sparse autoencoder, then the output (see (7)) of the hidden layer is the input to LSTM. Finally, a fully connected layer is applied to connect the output of LSTM and the target value $\overline{v}_{t,n}$, where $\overline{v}_{t,n}$ is an element of S_t and n is the mark of the target road segment.

The steps of the algorithm are described as follows, and Figure 6 illustrates the optimized structure of the SpAE-LSTM model.

Step 1. Set the values of sparsity α and sparsity parameter ρ. The weights of matrices and bias vectors (W_h, W_o, b_h, b_o) are initialized randomly.

Step 2. Let the input be the output; the input is the spatial-temporal matrix composed by vectors of every time frames. Train the parameters of the sparse autoencoder unit by the stochastic gradient descent algorithm, to minimize the cost function.

Step 3. Get the output from the hidden layer of the sparse autoencoder unit.

Step 4. The weights of matrixes and bias vectors $(W_i, U_i, V_i, W_f, U_f, V_f, W_c, U_c, W_o, U_o, V_o, b_i, b_f, b_c, b_o)$ are initialized randomly.

Step 5. Let the output from *Step 3* be the input. Train the parameters of the LSTM units by the full gradient backpropagation through time (Full BPTT) algorithm.

Step 6. Initialize the weight matrixes and bias vectors of the fully connected layer.

Step 7. Use the backpropagation method with the gradient-based optimization to adjust the parameters of the whole network in a top-down fashion.

4. Experiment

4.1. Traffic Data Set and Preprocessing. The traffic dataset is the GPS probe data which is collected from more than 14,000 taxis traveling in the Chengdu from August 3, 2014, to August 30, 2014. The GPS device on each taxi is sampling every 20 to 30 seconds from 6:00 a.m. to 24:00 p.m.; every record includes the taxi identification (taxi ID), current time (time stamps), and current location (latitude, longitude) [23].

The area in experiment is located in the Qingyang District of Chengdu, China (Figure 7). In this paper, the road map is obtained from the OpenStreetMap, which is a free, open source, and editable map service. Then, we use the operation mentioned in the second section to divide the road network into segments.

Because the GPS probe data obtained do not contain the instantaneous speed, the instant speed of every record needs to be calculated first. We use the average speed between current point and the previous point to approximate the instantaneous speed at the current point. Time frames with the size of 10 minutes are enough to provide a suitable granularity for timely traffic estimation [24], so the traffic data is divided for each 10 minutes time frame in this paper. Then, the traffic dataset is processed into the time-series data of each road segment by the operation mentioned in the second section.

4.2. Error Indicators. In order to evaluate the performance of the proposed model of short-term traffic prediction, two error indicators are introduced and they are Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) as shown in the following.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\tilde{x}_i - x_i\right|}{x_i} \tag{15}$$

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{\left(\tilde{x}_i - x_i\right)^2}{n}} \tag{16}$$

where $\tilde{x}_i$ is the value of prediction, x_i is the true value at the time i, and n indicates the number of predictive values.

Different measures evaluate the model performance from different aspects. The most important indicator is MAPE, and it reflects the relative errors of the models. RMSE provide the prediction error in terms of differences in the speed of traffic flow.

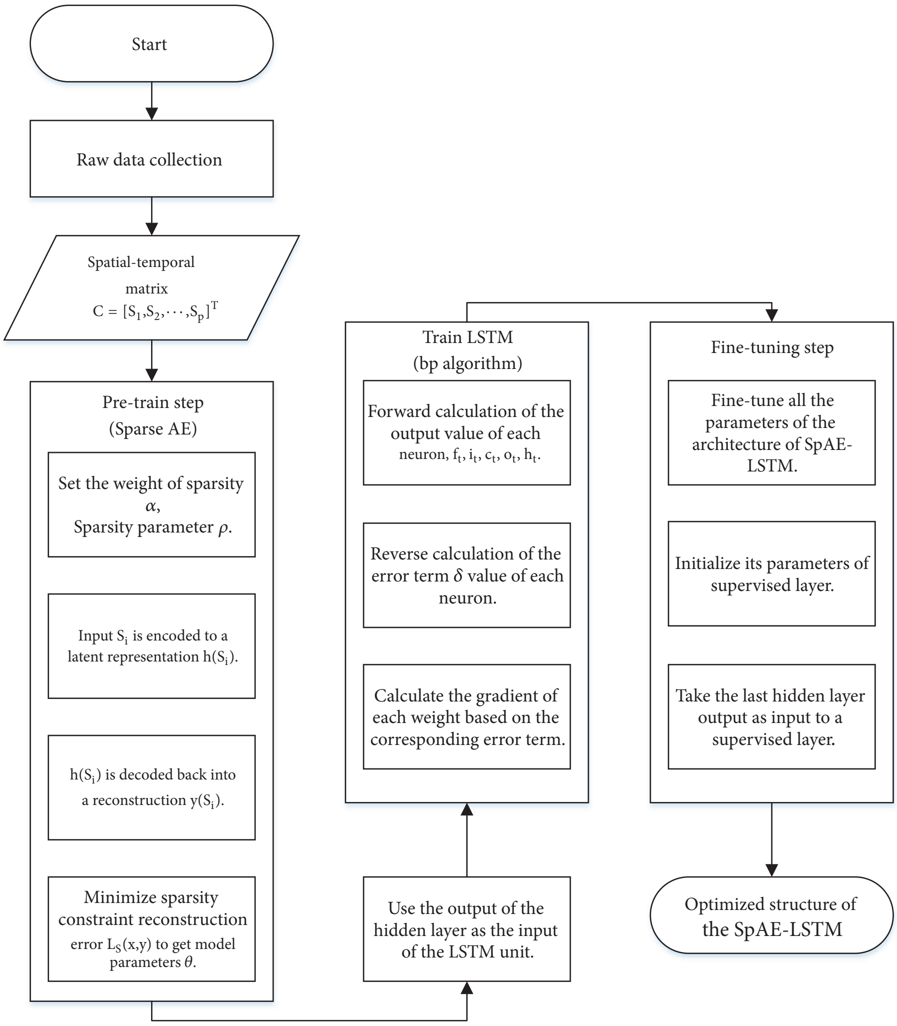

Start
Raw data collection
Spatial-temporal matrix
$C = [S_1, S_2, \cdots, S_p]^T$
Pre-train step
(Sparse AE)
Set the weight of sparsity α,
Sparsity parameter ρ.
Input S_i is encoded to a latent representation $h(S_i)$.
$h(S_i)$ is decoded back into a reconstruction $y(S_i)$.
Minimize sparsity constraint reconstruction error $L_S(x,y)$ to get model parameters θ.
Use the output of the hidden layer as the input of the LSTM unit.
Train LSTM
(bp algorithm)
Forward calculation of the output value of each neuron, f_t, i_t, c_t, o_t, h_t.
Reverse calculation of the error term δ value of each neuron.
Calculate the gradient of each weight based on the corresponding error term.
Fine-tuning step
Fine-tune all the parameters of the architecture of SpAE-LSTM.
Initialize its parameters of supervised layer.
Take the last hidden layer output as input to a supervised layer.
Optimized structure of the SpAE-LSTM

Figure 6: The optimized structure of the SpAE-LSTM model.

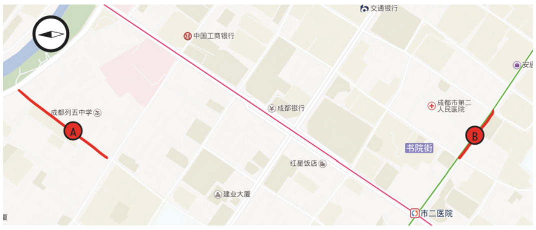


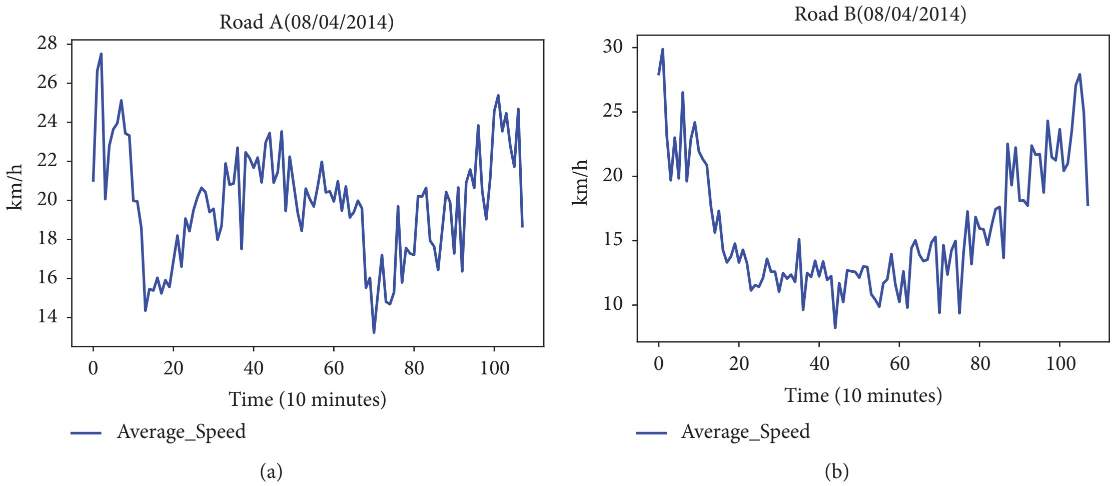
Figure 7: Road map.

4.3. Get Research Dataset. We choose the road segment A and road segment B as the research objects as shown in Figure 6. Road segment A is near the school, which has two peak time periods on weekdays, so the curve of the average speed of traffic flow within one day is like the letter "W" (Figure 8(a)); road segment B is near the hospital, which is always busy from morning to evening, so the curve of the average speed is like the letter "U" (Figure 8(b)).

According to the *Big Data Report of Smart Travel 2016* proposed by Didi Chuxing, the average speed of vehicles in Chengdu is 25.7 km/h, so we calculate the distance from road segment r_i to the target road segment (A or B) by Dijkstra's Algorithm and choose the road segments within 4 km distant from the target road. Then, we calculate *Spearman's correlation coefficient* of road segment r_i and the target road segment and discard the road whose p *value* is above the significance level 0.01. After screening, there are 96 road segments related to road segment A and 113 road segments related to road segment B.

Then, the time series vectors of target road and its related roads are formed into spatial-temporal matrixes, named MatrixA and MatrixB. Take MatrixA and MatrixB as the research datasets.

4.4. Identifying the Optimum Number of Hidden Units. With regard to the structure of the sparse autoencoder in the proposed model, the number of hidden units needs to be determined. For identifying the optimum number of hidden units, we use a cross-validation method named Day Forward-Chaining. The Day Forward-Chaining means that we select each day as the test set, and the days front as the training



(a)

(b)

Figure 8: The daily traffic flow patterns. (a) Road segment A and (b) Road segment B.

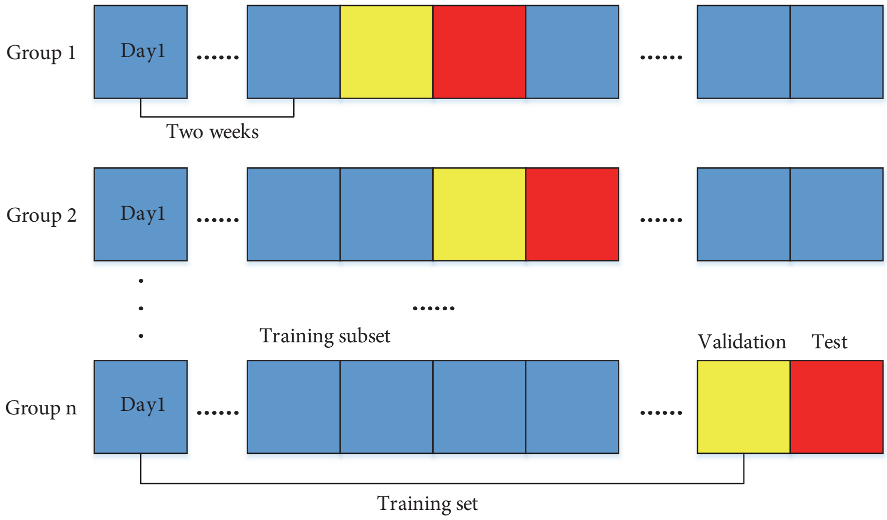


Figure 9: The test groups.

set. The training set is divided as the training subset and the validation. For example, in this research, we select each day after two weeks as the test set and the days before as the training set. The test groups as shown in Figure 9.

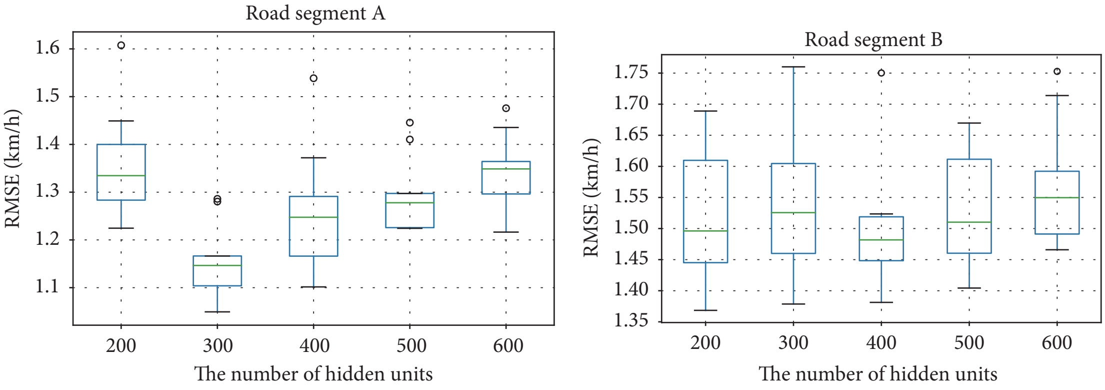
We choose the number of hidden units of sparse autoencoder from $\{200, 300, 400, 500, 600\}$. After performing the test groups, the results are shown as in Figure 10.

From the results, we can obtain the best number of hidden units for road segment A which is 300 and for road segment B is 400.

4.5. Result Analysis and Comparison. For better evaluation of model performance, we use the cross-validation method mentioned in Section 4.4 to process the data into multiple groups. The proposed model is implemented on each group and gets the result. Then, we calculate the error of results for all groups and use the average of error to evaluate performance of the model. The predictive result of road segments A and B is shown in Figure 11. The red curve of the Figure 11 is the predicted result of the method proposed and the other blue curve is the true data.

Figure 11(a) shows that the traffic patterns of road segment A on weekdays is very different from weekends. The performance of prediction on weekdays is obviously better than weekends. The reason is that the traffic flow conditions are more complex on weekends. There is no big difference between the traffic patterns of road segment B on weekdays and the weekends as Figure 11(b) shows. By carefully observing Figure 11, we speculate that the model performance on rush hour is better than the non-rush hour from the results. In order to verify this conjecture, we divide the prediction results of each day into hours and take the average error per hour. The error distribution per hour is shown in Figure 12.

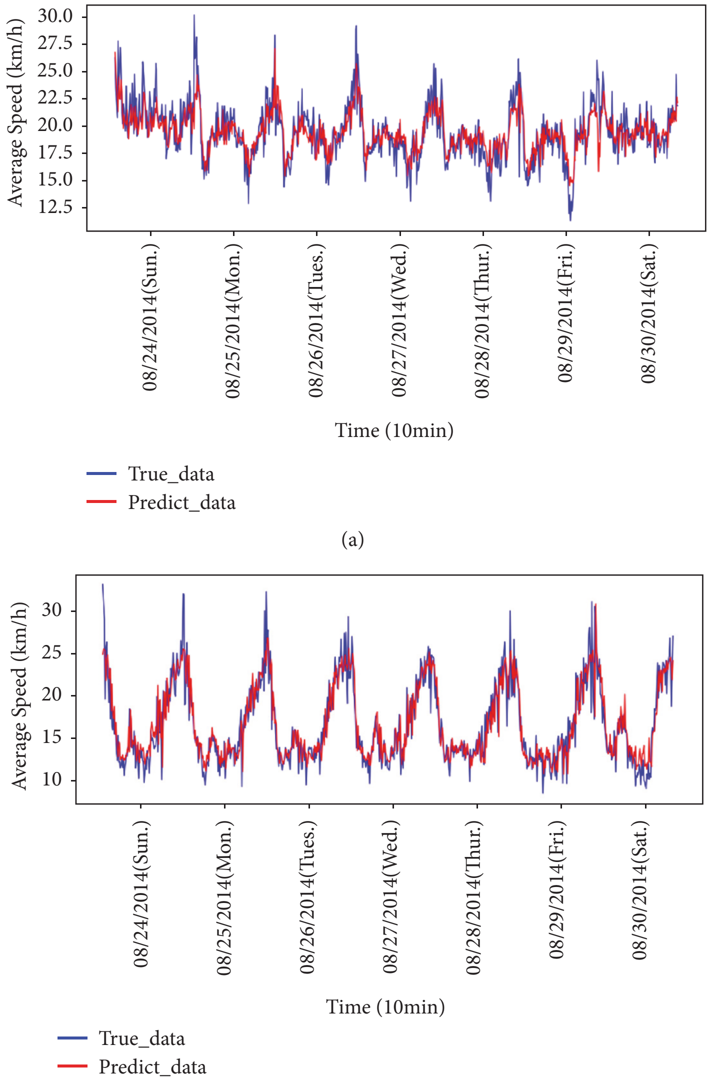
Figures 12(a) and 12(c) clearly show that the accuracy of prediction is higher during rush hours than during non-rush hours. Furthermore, the deviations of prediction vary with



(a)

(b)

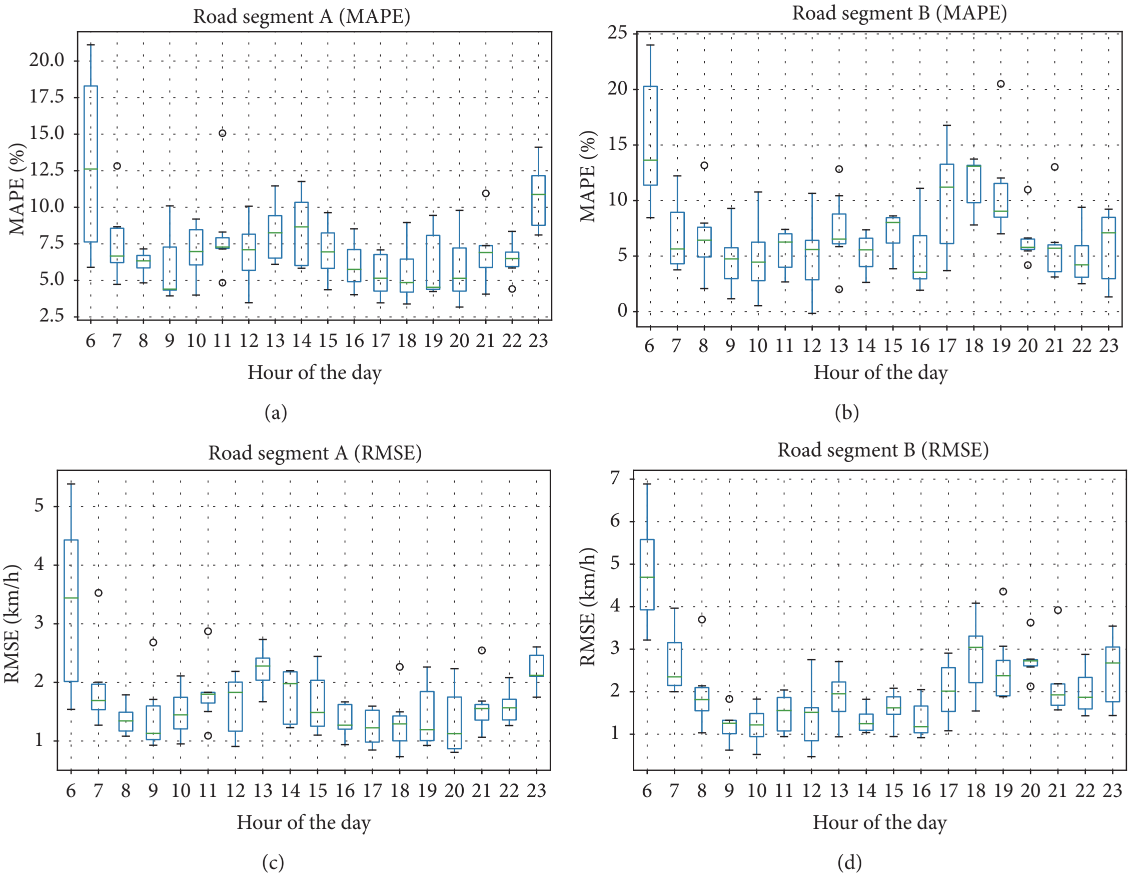
Figure 10: Impact of the number of hidden units. (a) Road segment A and (b) Road segment B.



(a)

(b)

Figure 11: Traffic flow prediction of road segments A and B. (a) Road segment A and (b) Road segment B.



(a)

(b)

(c)

(d)

Figure 12: The error distribution per hour. (a) The MAPE for road segment A, (b) the MAPE for road segment B, (c) the RMSE for road segment A, (d) and the RMSE for road segment B.

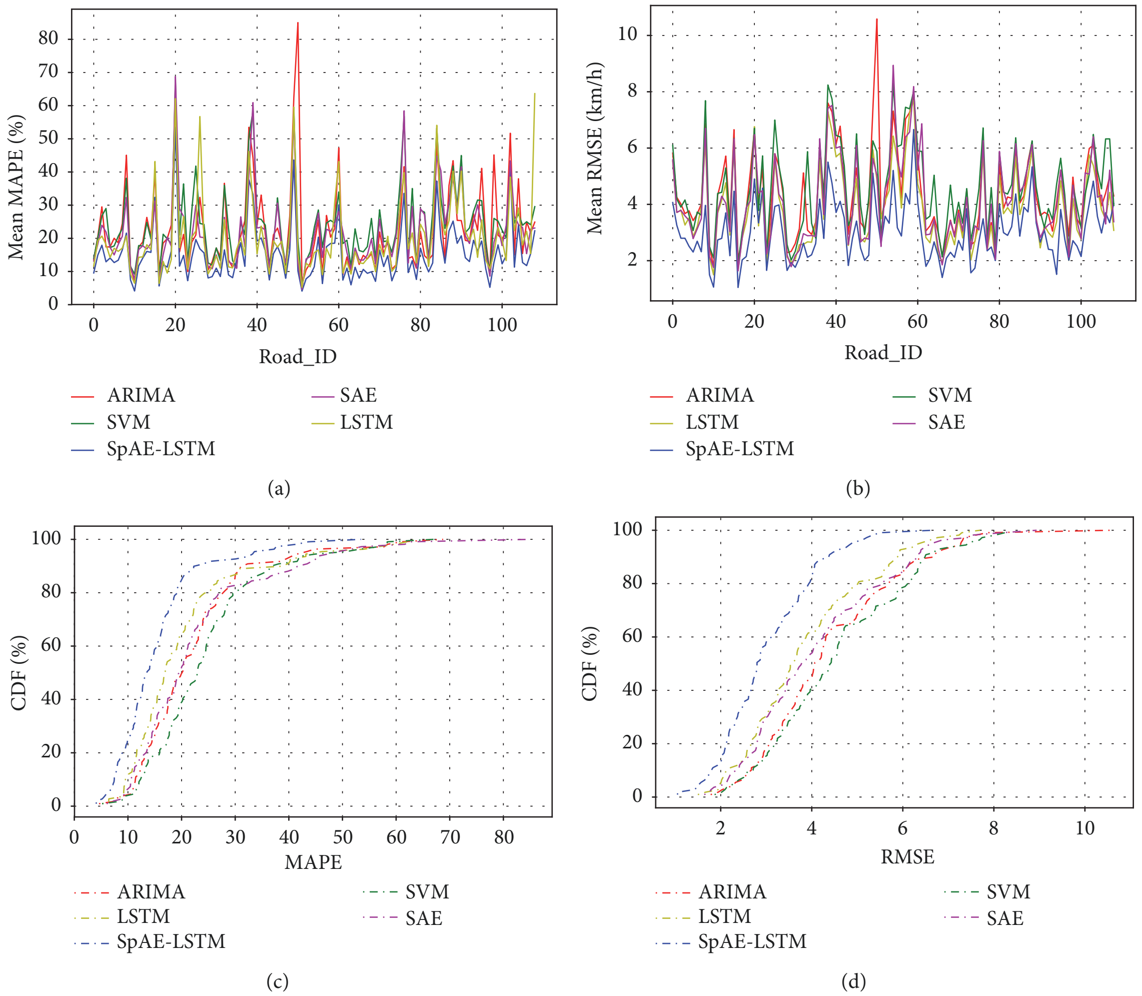
time. The main reason is that we use the average speed of taxis, which is like sampling traffic flow data; the speed of the taxis is more representative during rush hours. It is also because cars on non-rush time can easily change their speeds. That enhanced the stochasticity of traffic flow. The traffic conditions of road segment B are obviously more complex as Figures 12(b) and 12(d) show. Road segment B is busy from morning to evening, so it is hard to define which time is rush hour or non-rush hour, whereas we can also find the regularity that the accuracy of prediction during rush hour (8:00 a.m.-10:00 a.m.) is higher than during non-rush hour (6:00 a.m.-7:00 a.m.).

To validate the effectiveness of the proposed model, its predictive performance is compared with the ARIMA model, SVM model, SAE model, and LSTM model. By estimating the parameters, we select the ARIMA(2,1,1) as the baseline method for road segment A and ARIMA(1,2,5) for road segment B. The SAE model is proposed by Lv et al. [21] and by testing the best architecture which consists of three hidden layers; the number of hidden units in each hidden layer is [200, 200, 200]. We use the LSTM model about traffic speed prediction proposed by Ma Xiaolei et al. [17]. The performance of the proposed model and other models is shown in Tables 1 and 2.

Tables 1 and 2 show the different error result indicators of predictive result of different algorithms. According to the comparison, SpAE-LSTM has better performance than ARIMA, SVM, SAE, and LSTM model. Especially comparing with the ARIMA and SVM, SpAE-LSTM reduces error by almost 50%. The proposed model also reduces the error by more than 15% compared to the SAE and LSTM. The reason is that the conventional models like ARIMA and SVM are insufficient to handle complex traffic conditions. SAE and LSTM did not fully consider the impact of the spatial-temporal features of traffic flow on traffic prediction. Road segment B is located near the hospital and traffic conditions are more complicated than road segment A, so the predictive result of road segment A is better than road segment B, as the tables show.

For confirming the robustness of the proposed model, we select 108 road segments in the area mentioned in Section 4.1 for experiments. The prediction performance of different models is shown in Figure 13.

As the result shows that the proposed model is robust, Figures 13(a) and 13(b) display that the proposed model performs better on most road segments than other methods. According to Figure 13(c), the prediction MAPE of more than 86% of the road segments is below 20%. Figure 13(d)



(a)

(b)

(c)

(d)

Figure 13: Comparison of mean prediction errors for each method of short-term traffic prediction. (a) The Mean MAPE of predictive results from 108 road segments, (b) the mean RMSE of predictive results from 108 road segments, (c) The Cumulative Distribution Function of Mean MAPE, and (d) The Cumulative Distribution Function of Mean RMSE.

shows the prediction RMSE of more than 90% of the road segments is below 4.5 km/h. Considering the complexity and irregularities of urban traffic flow, this result is acceptable.

5. Conclusion

According to the spatial relationships between road segments and the temporal trend of traffic flow, this paper proposes a hybrid model called SpAE-LSTM. Using the characteristics of the sparse autoencoder and the LSTM, the spatial features and its inherent structures are extracted from the raw traffic data; the long- and short-term dependence of traffic flow is captured. Then the spatial and temporal features are combined to predict the traffic average speed of traffic flow. The robustness of the proposed model is demonstrated by applying it to the traffic data of multiple road segments collected from Chengdu. We evaluate the performance of the proposed model and compare it with other models. This research provides evidence that suggests the following key findings.

(1) For intricate traffic flow with multiple properties, we can use the hybrid model of fusing different methods to handle the prediction task well. In this paper, we demonstrate that the hybrid model based on a combination of LSTM and sparse autoencoder has good performance in traffic prediction.

Table 1: Predictive performances of different algorithms for road segment A.

Model	RMSE	MAPE(%)
ARIMA	1.910	9.539
SVM	2.122	10.340
SAE	1.483	7.108
LSTM	1.398	6.841
SpAE-LSTM	**1.133**	**5.826**

(2) The performance of the proposed model cannot be enhanced by applying more hidden units in sparse autoencoder. An optimum number of hidden units are a prerequisite for the proposed model to achieve accurately predicted results.

Table 2: Predictive performances of different algorithms for road segment B.

Model	RMSE	MAPE(%)
ARIMA	2.225	11.357
SVM	2.016	9.852
SAE	1.702	7.636
LSTM	1.816	7.998
SpAE-LSTM	**1.457**	**6.536**

(3) The promising results show that the predictive performance of SpAE-LSTM during rush hours is better than during non-rush hours. So the SpAE-LSTM is more suitably applied on the busy road segments.

In this paper, the average speed of taxis is used to represent the traffic condition and applied in the prediction. The traffic flow consisted of many types of vehicles and the condition of traffic flow includes many aspects, e.g., traffic volume, traffic occupancy, and traffic speed. This paper just focuses on the traffic speed from taxis due to the limitations of data. In the future, short-term traffic prediction needs to consider more aspects of traffic flow for improving the accuracy of prediction. In the meantime, the traffic flow data is complex and mixed with useful and useless information. If we can find a more efficient and suitable method to extract the features of traffic flow, short-term traffic prediction will achieve a better performance.

Data Availability

The GPS probe data from taxis used to support the findings of this study were supplied by DataCastle Big Data Competition Platform (http://www.pkbigdata.com/) and freely available from [23].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the grant from the National Natural Science Foundation of China (no. 61602141), Zhejiang Provincial Science and Technology Plan Project of China (no.2018C01111).

References

[1] V. L. Hans and C. V. Hinsbergen, “Short-Term Traffic and Travel Time Prediction Models,” *Transportation Research E-Circular*, 2012.

[2] K. Nagel and M. Schreckenberg, “A cellular automaton model for freeway traffic,” *Journal de Physique I*, vol. 2, no. 12, pp. 2221–2229, 1992.

[3] G. F. Newell and F. Mannering, “A simplified theory of kinematic waves in highway traffic, part I: General theory,” *Transportation Research Part B Methodological*, vol. 4, no. 1991, pp. 281–287, 1991.

[4] B. S. Kerner, *The Physics of Traffic*, vol. 197(2638), article 48, New Scientist, 2013.

[5] M. S. Ahmed and A. R. Cook, “Analysis of freeway traffic time-series data by using box-jenkins techniques,” *Transportation Research Record*, vol. 773, no. 722, pp. 1–9, 1979.

[6] Z. S. Yang, Q. C. Bing, C. Y. Lin, N. Y. Yang, and D. Mei, “Research on short-term traffic flow prediction method based on similarity search of time series,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 184632, 8 pages, 2014.

[7] I. Okutani and Y. J. Stephanedes, “Dynamic prediction of traffic volume through Kalman filtering theory,” *Transportation Research Part B: Methodological*, vol. 18, no. 1, pp. 1–11, 1984.

[8] J. W. C. van Lint, “Online learning solutions for freeway travel time prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 38–47, 2008.

[9] D. Ming-jun and Q. Shi-ru, “Fuzzy State Transition and Kalman Filter Applied in Short-Term Traffic Flow Forecasting,” *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 875243, 7 pages, 2015.

[10] Z. Mingheng, Z. Yaobao, H. Ganglong, and C. Gang, “Accurate multisteps traffic flow prediction based on SVM,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 418303, 8 pages, 2013.

[11] S. Sun, C. Zhang, and G. Yu, “A Bayesian network approach to traffic flow forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[12] D. Park and L. R. Rilett, “Forecasting freeway link travel times with a multilayer feedforward neural network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 5, pp. 357–367, 1999.

[13] D. Park and L. R. Rilett, “Forecasting multiple-period freeway link travel times using modular neural networks,” *Transportation Research Record*, no. 1617, pp. 163–170, 1998.

[14] M. Carroll and I. I. Thomas Urbanik, “Short-Term Freeway Traffic Volume Forecasting Using Radial Basis Function Neural Network,” *Transportation Research Record Journal of the Transportation Research Board*, vol. 1651, no. 1, pp. 39–47, 1998.

[15] J. W. C. Van Lint, S. P. Hooqendoorn, and H. J. Van Zuvlen, “Freeway travel time prediction with state-space neural networks modeling state-space dynamics with recurrent neural networks,” *Transportation Research Record*, vol. 1811, no. 1, pp. 347–369, 2002.

[16] J. A. Bullinara, “Recurrent Neural Network,” Neural Computation: Lecture 12. 2013. http://www.cs.bham.ac.uk/~jxb/INC/l12.pdf.

[17] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, “Long short-term memory neural network for traffic speed prediction using remote microwave sensor data,” *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, “LSTM network: a deep learning approach for short-term traffic forecast,” *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[20] J. Han, J. W. Polak, J. Barria, and R. Krishnan, “On the estimation of space-mean-speed from inductive loop detector data,” *Transportation Planning & Technology*, vol. 33, no. 1, pp. 91–104, 2010.

[21] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic Flow Prediction with Big Data: A Deep Learning Approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[22] H. F. Yang, T. S. Dillon, and Y. P. Chen, "Optimized Structure of the Traffic Flow Forecasting Model With a Deep Learning Approach," *IEEE Transactions on Neural Networks Learning Systems*, vol. 28, no. 10, pp. 2371–2381, 2016.

[23] The password: "meq5" http://pan.baidu.com/s/1o84gtPS, 2016.

[24] Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu, "Mining Road Network Correlation for Traffic Estimation via Compressive Sensing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1880–1893, 2016.