

Article

An AutoEncoder and LSTM-Based Traffic Flow Prediction Method

Wangyang Wei ^{1,2,*}, Honghai Wu ² and Huadong Ma ¹

¹ Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China

² Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China

* Correspondence: weiwangyang@163.com

Received: 1 May 2019; Accepted: 26 June 2019; Published: 4 July 2019



Abstract: Smart cities can effectively improve the quality of urban life. Intelligent Transportation System (ITS) is an important part of smart cities. The accurate and real-time prediction of traffic flow plays an important role in ITSs. To improve the prediction accuracy, we propose a novel traffic flow prediction method, called AutoEncoder Long Short-Term Memory (AE-LSTM) prediction method. In our method, the AutoEncoder is used to obtain the internal relationship of traffic flow by extracting the characteristics of upstream and downstream traffic flow data. Moreover, the Long Short-Term Memory (LSTM) network utilizes the acquired characteristic data and the historical data to predict complex linear traffic flow data. The experimental results show that the AE-LSTM method had higher prediction accuracy. Specifically, the Mean Relative Error (MRE) of the AE-LSTM was reduced by 0.01 compared with the previous prediction methods. In addition, AE-LSTM method also had good stability. For different stations and different dates, the prediction error and fluctuation of the AE-LSTM method was small. Furthermore, the average MRE of AE-LSTM prediction results was 0.06 for six different days.

Keywords: AutoEncoder; long short-term memory; traffic flow prediction

1. Introduction

In recent years, the increasing number of vehicles aggravates traffic congestion. The traffic congestion often brings a series of problems such as environmental pollution, which greatly reduces the quality of urban life. To improve the urban living environment, there are emerging fields such as smart cities and Internet of Things (IoTs) [1,2], which mainly utilize various information technologies to optimize urban resources and services. As an important part of smart cities, Intelligent Transportation Systems (ITSs) can effectively alleviate urban traffic congestion [3,4]. Hence, ITSs have received significant interest and have become one of the important development directions of modern transportation. The prediction of traffic flow plays a key role in ITSs. The normal operation of many large systems relies on accurate prediction of traffic information, such as the Split Cycle Offset Optimization Technique (SCOOT) system and Sydney Coordinated Adaptive Traffic (SCAT) system [5].

The early traffic flow prediction methods mainly include AutoRegressive Integrated Moving Averaging (ARIMA) [6], Kalman filter [7,8], Support Vector Machine (SVM) [9], Markov chain model [10], etc. These prediction methods are simple and can be implemented easily. However, these models cannot mine the deep relationship between data. For this reason, neural network models are used to predict the traffic flow since these models can mine big data and discover internal structure and potential characteristics. Recently, Deep Belief Network (DBN) [11,12], Long Short-Term Memory (LSTM) neural network [13–15] and prediction methods based on deep architecture [16,17] have been widely used in traffic flow prediction. Huang et al. [17] used the Gated Recurrent Unit (GRU) to

predict the traffic flow, but this method only considers the temporal patterns of the traffic flow and ignores its spatial patterns. Because of the network structure of urban roads, the change of traffic flow in adjacent position is one of the important factors affecting traffic flow. Considering the temporal and spatial patterns of traffic flow, Lv et al. [16] exploited the deep learning architecture to predict traffic flow. However, compared with LSTM network, the deep learning model requires a large amount of data and computation. In addition, the LSTM network can effectively preserve the long-term effect of the data for time series data. Therefore, the LSTM network is a very promising prediction model for time series data.

In this paper, we consider the temporal and spatial patterns, and propose a prediction model, called AutoEncoder Long Short-Term Memory (AE-LSTM) prediction method. The AutoEncoder is used to obtain the traffic flow characteristics of adjacent positions. The adjacent positions represent the upstream and downstream location of the current location. We also use LSTM to predict the traffic flow at the current location. Compared with other prediction methods based on neural networks, AE-LSTM prediction model not only mines big data deeply, but also considers the influence of upstream and downstream traffic flow. Experimental results show that the proposed method has better performance on real datasets. The main contributions of this paper are as follows:

- We propose an AE-LSTM model to predict traffic flow. This method combines AutoEncoder with LSTM, where AutoEncoder is used for feature extraction and LSTM model is used for data prediction.
- We propose a traffic flow prediction algorithm based on AE-LSTM. The AutoEncoder and LSTM network are trained, respectively. Then, we fine-tune the whole network.
- We evaluated the performance of AE-LSTM by experiments. We conducted AE-LSTM on real datasets, and experimental results show that the performance of AE-LSTM was better than the other prediction methods.

The rest of this paper is organized as follows. In Section 2, we give a brief review of the related works of other researchers. In Section 3, we briefly introduce the AutoEncoder and LSTM, and then describe the AE-LSTM prediction model presented in this paper. In Section 4, we describe the implementation of the model in detail, including the training of the prediction network and data processing. In Section 5, we provide the experiment results. In Section 6, we conclude the whole paper.

2. Related Work

The prediction of traffic flow is necessary for traffic plan and plays an important role in the mobile opportunistic networks [18–20]. For a long time, researchers have been working on traffic flow prediction and proposed many research methods. Early prediction models were very simple, such as random walks, historical averages [21], etc. Although these models are easy to implement, the prediction accuracy is not enough. In addition, people pay more and more attention to privacy protection [22], and many traditional prediction methods have difficulty guaranteeing privacy. Subsequently, many more complex and effective prediction models emerged.

Researchers usually use some time series models to make predictions, such as ARIMA, Markov chains, Gaussian processes, etc. Williams et al. [23] presented a multivariate ARIMA model containing upstream traffic data. Stathopoulos and Karlaftis [24] proposed a state space model of multiple time series, which uses upstream flow detector to obtain data. Kamarianakis and Prastacos [25] proposed a spatiotemporal autoregressive integrated moving average model to predict urban traffic flow. Min and Wynter [26] developed a time-based extension prediction method that takes into account the interaction between time and space. Kumar et al. [27] proposed a prediction scheme using the Seasonal ARIMA (SARIMA) model. This scheme predicts traffic flow through limited data. The authors used the difference method to stabilize the input data and the maximum likelihood method to estimate the model parameters. Pan et al. [28] introduced the spatial-temporal correlation to the short-term traffic flow prediction by using the random region transmission framework. Considering

the interpretability of the prediction model, Xu et al. [29] proposed an interpretable spatial-temporal multiple adaptive bayesian regression model. Bayesian derivation is used to train model parameters. Because the traffic flow of adjacent sections affects each other, Sun et al. [30] constructed a bayesian network for traffic flow of each section to predict the traffic flow. However, when the traffic network is too complex, the accuracy of this method is not enough. Due to the dynamics and randomness of highway traffic, a short-term highway traffic prediction method based on the hidden Markov model [31] is proposed. This method uses the observed velocity statistics to define the traffic state in two-dimensional space. Xie et al. [32] used Gauss regression process to model and forecast traffic flow. Because of the interconnection between roads, traffic flow data are a kind of multivariable time series. Dynamic systems have good performance in simulating multivariable time series data. Therefore, Zhao and Sun [33] modeled the traffic flow according to the Gaussian process dynamic model. Then, a dynamic prediction model based on fourth-order Gaussian process is proposed, and the weighted neural network is introduced into the model to train the model parameters.

The Urban Traffic Control (UTC) systems and highway management systems put into use collect a large amount of traffic condition data every day. The big data in the field of transportation have attracted the interest of many scholars [34,35]. The previous prediction models cannot give full play to the advantages of big data, and have some shortcomings such as the lack of high accuracy and the inability to fully mine the historical information. In contrast, the neural network is very good at processing big data. Therefore, many prediction models based on neural networks are applied to traffic flow prediction. Tan et al. [36] proposed an aggregate prediction method based on neural network, which combines four prediction methods: neural network, ARIMA, exponential smoothing and moving average. The periodic similar time series are constructed from the original time series, and then the traffic flow is predicted by multiple prediction models. Tan et al. [11] proposed two DBN traffic prediction architectures based on Restricted Boltzmann Machines (RBMs). Lv et al. [16] exploited AutoEncoder model to forecast traffic flow. AutoEncoder is widely used in denoising processing, and can also be used as feature extractor. Chen et al. [37] used the deep learning model to predict the traffic flow in the case of special events, and proved that the RNN has good performance in traffic flow prediction. Considering the influence of weather conditions on traffic flow, Fu [14] proposed a forecast model combining RNN and GRU. GRU is a kind of RNN, but its cell structure is simpler than LSTM cell structure. Because GRU simplifies the structure of memory unit, some important information may be ignored in the prediction process, and the prediction accuracy is reduced. Similar to GRU, LSTM is also an improved RNN, which can retain the influence of data for a longer time, and improve the gradient disappearance for RNN. Therefore, LSTM is often used as a time series prediction model. Because future traffic conditions may be related to events that occurred long ago, Ma et al. [13] proposed a new LSTM neural network prediction model that can learn time series and automatically determine the prediction delay.

In addition, Yang et al. [38] proposed a new traffic state prediction method, which uses electroseidicalography data and driving behavior to prediction traffic state. However, the data acquisition process is more complex in this prediction method. Wang et al. [39] used Error feedback Recurrent Convolutional Neural Network structure (eRCNN) to predict the continuous traffic. This deep network model introduces new error feedback neurons to better deal with emergencies. Wu et al. [40] proposed a traffic flow prediction model based on DNN. This model introduces an attention based model to determine the importance of past traffic flow and uses convolutional neural network to mine the spatial characteristics of traffic flow. Zhan et al. [41] combined a variety of prediction methods and proposed an automated framework to solve the problem of traffic flow prediction.

Different from many neural network prediction models, AE-LSTM prediction model uses AutoEncoder to extract the characteristic of upstream and downstream data, which can not only take into account the spatial characteristics of vehicle flow, but also not add too many calculations in

the subsequent LSTM prediction. Meanwhile, LSTM network is used to explore the time characteristics of vehicle flow.

3. Methodology

3.1. AutoEncoder Model

AutoEncoder is usually used to reduce dimensions or extract features [42]. Given an original input sequence data $X = \{x_1, x_2, \dots, x_k\}$, where $x_i \in R^d$. The characteristic sequence of the original data is obtained through the formula f . T represents the characteristic sequence of the original sequence X , which is defined as $T = \{t_1, t_2, \dots, t_k\}$, where $t_i \in R^l$. The output of the encoder is used as the input of the decoder. The decoder reconstructs the original data according to the characteristic sequence T . The reconstructed data $Y = \{y_1, y_2, \dots, y_k\}$, where $y_i \in R^d$. The purpose of decoding is to verify whether the extracted features are valid. After the training of the AutoEncoder is completed, we only use the encoder to extract the characteristics of the original data to obtain more internal structure of the data. Figure 1 shows the basic structure of AutoEncoder. The encoding and decoding process follows the equations:

$$t_i = f(w_t \cdot x_i + b_t), \quad (1)$$

$$y_i = g(w_y \cdot t_i + b_y), \quad (2)$$

where $f(\cdot)$ and $g(\cdot)$ are the sigmoid functions, and w_t, w_y and b_t, b_y are weights and biases, respectively. We train the AutoEncoder by minimizing reconstruction error

$$L(X, Y) = \frac{1}{2} \sum_{i=1}^n \|x_i - y_i\|^2. \quad (3)$$

When the difference between the reconstructed data Y and the original data X is small enough, in other words, the output T of the coding process is valid, T is seen as the characteristics extracted from the original data.

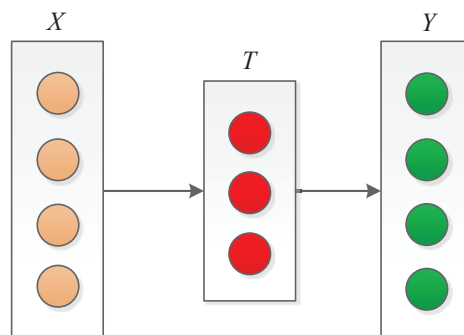


Figure 1. The structure of AutoEncoder. The encoder obtains the characteristic sequence T based on the original sequence X . The decoder gets the reconstructed sequence Y according to the characteristic sequence T .

In the urban traffic network, there is an interaction between the traffic flow at the current location and the traffic flow on surrounding roads. Therefore, we should consider not only the historical traffic flow data of the current location, but also the changes of upstream and downstream traffic flow. In this model, we use AutoEncoder to extract characteristics of upstream and downstream traffic flow data. Then, the extracted features are put into the prediction network to improve the accuracy of traffic flow prediction at the current location. Therefore, the input of AutoEncoder is the upstream and downstream traffic flow $X_u = \{x_{u1}, x_{u2}, \dots, x_{um}\}$ and $X_d = \{x_{d1}, x_{d2}, \dots, x_{dm}\}$,

where $x_{ui}, x_{di} \in R^d$. The characteristic sequence is defined as $Z_t = \{z_1, z_2, \dots, z_m\}$, where $z_i \in R^l$. In this paper, the processes of encoding and decoding follow the equations below:

$$z_i = f(w_z \cdot (x_{ui} + x_{di}) + b_z), \quad (4)$$

$$y_i = g(w_y \cdot z_i + b_y). \quad (5)$$

3.2. AE-LSTM Model

LSTM [43] has been widely used in many fields and achieved great success, such as in music generation, image caption, speech recognition and machine translation. LSTM improves the hidden-layer cell on the basis of RNN. The improvement of cell can make up for the gradient disappearance problem of RNN. LSTM adds some memory units, including forget gate, input gate and output gate. The memory units can further control the data and decide which should be retained and which should be deleted.

Since the upstream and downstream traffic flow will affect the traffic flow at the current location, we need to take these factors into account when predicting the current traffic flow. If all upstream and downstream traffic flow data are put into the prediction model, the data dimension will increase and the calculation will be too complicated. To solve this problem, we use the AutoEncoder to extract the characteristics of the upstream and downstream vehicle flow data. In other words, the dimensionality of the traffic flow data is reduced. The acquired characteristics are taken as a part of the input data of the prediction network. In this way, not only the impact of upstream and downstream traffic flow is considered, but also the data dimension is not increased too much. The input data of LSTM consists of two parts: the characteristics of upstream and downstream traffic flow data z_t and the historical traffic flow data of the current position x_t .

The current position of the traffic flow data is expressed as $X = \{x_1, x_2, \dots, x_m\}$, where $x_i \in R^d$. The characteristics of upstream and downstream traffic flow data are represented by Z_t . The input information of the forgotten gate includes three parts: the current flow x_t , the upstream and downstream characteristic z_t , and the previous unit state h_{t-1} . The forgetting gate determines the information that should be discarded. The input information of the input gate is similar to that of the forgot gate. The input gate is used to select the information that should be input. \bar{C}_t represents the input of the cell, which is added to the cell state. C_t indicates that the status of the cell was updated. This step means removing part of the cell state at the previous time and adding part of the cell state at this time. The input information of the output gate is similar to that of the forgot gate. The output gate is used to select the information that should be output. Finally, the cell state is processed by tanh and multiplied by the output of output gate, among them $\sigma(x) = \frac{1}{1+e^{-x}}$ and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. The detailed structure of the LSTM is shown in Figure 2. The relevant formulas of AE-LSTM model are shown below:

$$f_t = \sigma(w_{f1} \cdot x_t + w_{f2} \cdot z_t + w_{f3} \cdot h_{t-1} + b_f), \quad (6)$$

$$i_t = \sigma(w_{i1} \cdot x_t + w_{i2} \cdot z_t + w_{i3} \cdot h_{t-1} + b_i), \quad (7)$$

$$\bar{C}_t = \tanh(w_{c1} \cdot x_t + w_{c2} \cdot z_t + w_{c3} \cdot h_{t-1} + b_c), \quad (8)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t, \quad (9)$$

$$o_t = \sigma(w_{o1} \cdot x_t + w_{o2} \cdot z_t + w_{o3} \cdot h_{t-1} + b_o). \quad (10)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (11)$$

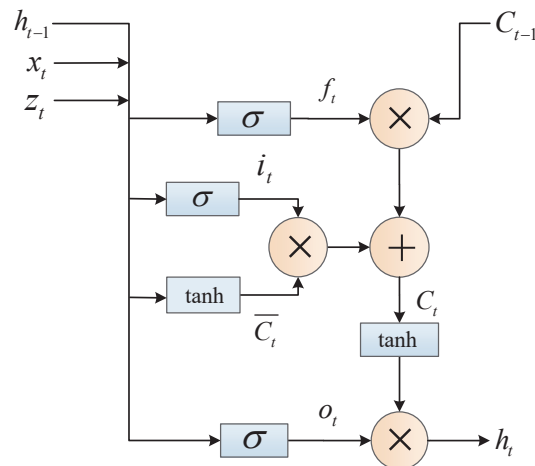


Figure 2. The structure of LSTM network. The input of the neuron is composed of x_t and z_t , x_t represents the traffic flow data of the current node, and z_t represents the characteristic of the traffic flow data of the upstream and downstream.

To consider the influence of upstream and downstream traffic flow but not increase the amount of calculation, we propose AE-LSTM prediction model, which combines the AutoEncoder and LSTM. Firstly, we use the AutoEncoder to extract the characteristics of original upstream and downstream traffic flow data. Then, we use LSTM model to predict the traffic flow. The specific prediction steps are shown below:

- The encoder of the AutoEncoder is used as the feature extractor to obtain the characteristics of upstream and downstream traffic flow data. The extracted features are put into the prediction network. Considering the influence of upstream and downstream on the traffic flow at the current location, the accuracy of traffic flow prediction can be improved.
- The characteristics of upstream and downstream traffic flow and the traffic flow data of the current position are combined as the input of LSTM. LSTM model predicts the traffic flow data at the next moment.

The AutoEncoder is a kind of unsupervised learning, which can be used as feature extractor of data. We determine the initial value of the weight matrix before the training. The weight matrix plays a very important role in the network. we hope to retain the characteristics of the original data after training the weight matrix. If the extracted feature can reconstruct the original data well, it indicates that the features of the original data can be effectively retained through the weight matrix. After the training of the AutoEncoder, we divide the AutoEncoder into two parts, which are the encoder and the decoder. The encoder, as a data feature extractor, is part of the AE-LSTM prediction model. The decoder is used to verify the validity of the extracted characteristics, and is discarded after the training. We attached a LSTM model after the encoder to form the AE-LSTM prediction model. The structure of the AE-LSTM model is shown in Figure 3.

In this paper, the Rectified Linear Unit (ReLU) function is selected as the activation function of the output layer in the whole network. The tanh function is selected as the activation function of other layers in the network. In this paper, the loss function is set to

$$L(x, \bar{x}) = \frac{\sum_{n=1}^N (\bar{x}_n - x_n)^2}{2N}, \quad (12)$$

where x_n represents the observed value, \bar{x}_n represents the predicted value, and N represents the number of predicted values.

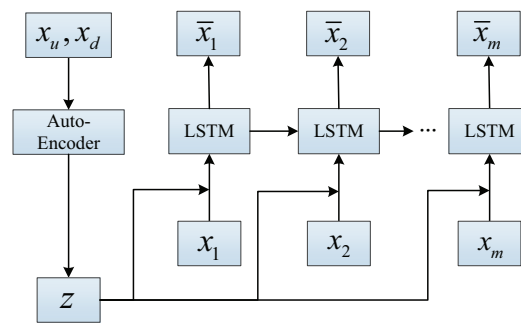


Figure 3. The structure of AE-LSTM for traffic flow prediction. AutoEncoder model is used to extract features and LSTM model is used to predict the traffic flow.

4. Model Implementation

The training of prediction network is one of the most important works, which is directly related to the final performance of prediction. We used the corresponding traffic flow data to train AutoEncoder and LSTM, respectively. Then, we fine-tuned the whole network and optimized the network parameters. The upstream and downstream traffic flow data x_u and x_d were put into AutoEncoder. We used back propagation to train parameters in AutoEncoder. The effective characteristics of traffic flow Z was obtained after encoding and decoding. In this way, the upstream and downstream traffic flow data could be considered in the prediction process, and the increasing of computational complexity could be avoided at the same time. The condition for the end of AutoEncoder training was that $L(X, Y)$ was less than a threshold. $L(X, Y)$ less than the threshold indicated that the decoder could reconstruct the original data through characteristic Z_t , and the error between the reconstructed traffic flow data and the original data was small enough. In other words, the extracted characteristics Z_t was valid, and could reflect the internal structure of the original data. The characteristics Z_t and the original data of current traffic flow X_t were put into the network for training LSTM model. We still used the back propagation method to train the LSTM network. The AutoEncoder combined with LSTM to form AE-LSTM, and the whole network was fine-tuned. Algorithm 1 shows the pseudocode of AE-LSTM network training. The input data X_u, X_d and X_t represent upstream traffic flow data, downstream traffic flow data and current position traffic flow data, respectively.

Next, to further discuss the implementation of the model, we briefly describe the data processing, which is expressed as a matrix. The upstream and downstream traffic flow datasets are represented as

$$X_u = [X_{u1} \quad X_{u2} \quad \cdots \quad X_{um}]^T \quad (13)$$

and

$$X_d = [X_{d1} \quad X_{d2} \quad \cdots \quad X_{dm}]^T, \quad (14)$$

where $X_{ui}, X_{di} \in R^d$. After extracting the characteristics by AutoEncoder, we obtain the characteristics set of upstream and downstream vehicle flow

$$Z_t = [Z_1 \quad Z_2 \quad \cdots \quad Z_m]^T, \quad (15)$$

where $Z_i \in R^l, Z_i = \{z_{i1}, z_{i2}, \dots, z_{il}\}$. The historical prediction data X_t are represented as

$$X_t = [X_1 \quad X_2 \quad \cdots \quad X_m]^T, \quad (16)$$

where $X_i \in R^d$, $X_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$. X_t and Z_t are combined to obtain a entire dataset as

$$S_t = \begin{bmatrix} S_1 & S_2 & \cdots & S_m \end{bmatrix}^T, \quad (17)$$

where $S_i \in R^{d+l}$, $S_i = \{x_{i1}, x_{i2}, \dots, x_{id}, z_{i1}, z_{i2}, \dots, z_{il}\}$. To accelerate the training and prediction process, we use the normalized method to process the original data. Before training the network, the original data are normalized. After obtaining the predicted results, the normal traffic flow predicted value can be obtained by reversing the normalization process.

Algorithm 1 AE-LSTM prediction algorithm.

Input: the training set X_u, X_d, X_t .

Output: prediction result \bar{X} .

- 1: Use X_u and X_d to build the AutoEncoder training set X_{AE} .
 - 2: Initialize the weight matrices of AutoEncoder randomly.
 - 3: Put X_{AE} into AutoEncoder.
 - 4: **if** $L(X, Y) < d$ **then**
 - 5: Calculate the error $L(X, Y)$ by Equation (3).
 - 6: Use the back propagation training the AutoEncoder.
 - 7: **else**
 - 8: End the training.
 - 9: **end if**
 - 10: Generate the characteristics of upstream and downstream vehicle flow Z_t .
 - 11: **for** $t = 0$ to *epoch* **do**
 - 12: Put Z_t and X_t into the LSTM, and use Equations (6)–(11) for forward propagation.
 - 13: Generate $\bar{x}_{t+1} = g(w_z \cdot h_t + b_z)$.
 - 14: Calculate error.
 - 15: Use the back propagation to update parameters.
 - 16: Use forward propagation to update network status h_t , through Equations (6)–(11).
 - 17: **end for**
 - 18: Add LSTM after the encoder of AutoEncoder to form AE-LSTM.
 - 19: Fine-tuning the whole network, training initialization parameters.
 - 20: Input test data in AE-LSTM to generate the predicted value \bar{X} .
 - 21: Return \bar{X} .
-

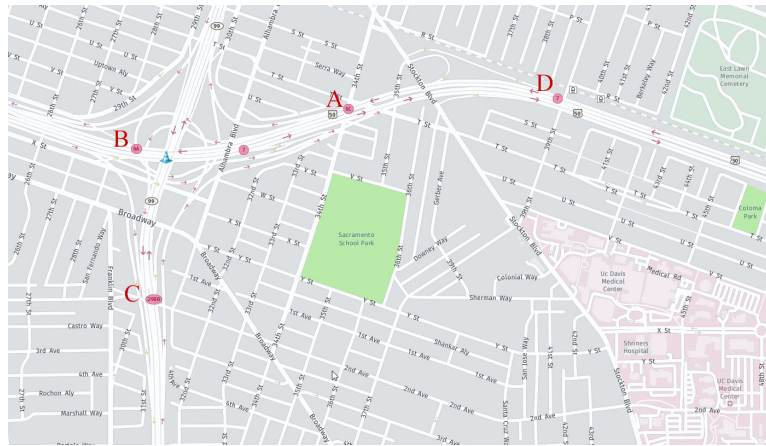
5. Experimental and Analysis

5.1. Data Collection from Caltrans Performance Measurement System

In this experiment, traffic flow data were obtained from Caltrans Performance Measurement System (PeMS) [44]. PeMS is widely used by many researchers in experiments. To facilitate the comparison with other prediction algorithm, the dataset from PeMS was selected as the experimental data. PeMS is a professional traffic data acquisition system, in which 15,000 detectors have been deployed in California. The acquisition details of dataset are summarized in Table 1. The detectors record relevant traffic data every 30 s and store them in the database. PeMS aggregates the data to 5 min. The traffic flow is the number of vehicles passing through a detector over a period of time. For traffic flow, the 5-min data sample can be obtained by adding up the 30-s traffic data samples. Of course, the original data contain some missing or invalid data for various practical reasons. These invalid data have been reasonably estimated using estimation methods. To verify the performance of the AE-LSTM traffic flow prediction model, we applied it to the actual datasets. As shown in Figure 4, we obtained the statistical data recorded at several stations near Sacramento County. In Figure 4, the red dots represent the positions of data detection, which we marked as A, B, C, D, etc., and the red arrows represent the direction of traffic flow. In this study, the experiment was validated at multiple locations in Sacramento County,. Figure 4 only shows the actual location of a predicted location.

Table 1. The acquisition details of dataset.

Directional Distance	Controllers	Stations	Detectors	Traffic Census Stations	Features
41,236.0 mi	6943	18,350	45,170	16,527	Flow, occupancy and speed

**Figure 4.** The data are collected around Sacramento County. The data collection stations are set up on the mainline of the four lanes.

5.2. Experimental Setup

In this method, the AutoEncoder model extract the characteristics of the upstream and downstream traffic flow data, and the LSTM model predicts the traffic flow data. For the structure of the AE-LSTM network, we need to determine the number of input layers, the number of hidden layers, and the number of neurons in hidden layer. The input layer was set to 288×3 . The number of hidden layers in the AutoEncoder model was 3, and the hidden neurons in each layer were 128, 128 and 32, respectively. The learning rate was set to 0.001. The number of hidden layers in LSTM model was 2, the number of neurons in each hidden layer was 128, the learning rate was set as 0.005, and the training number of epoch was 10. We obtained the traffic data from April to September 2018. The datasets was divided into two subsets, namely training set and test set. We used the first 176 days data of the six months in training process, which obtained the model. For LSTM model, the previous output was part of the input of the current iteration. Then, we used the generated model to predict the next seven days of data, and compared it to the last seven days of data during the six months. In addition, to obtain the impact of step ahead, we used the first five months data as training data, and predicted different steps ahead. Considering the time patterns of the traffic flow, we used the traffic flow data of the previous period to predict the data of the next moment. Considering the spatial pattern of traffic flow, we took the upstream and downstream traffic flow data as the input of the prediction model.

In this study, the prediction interval of traffic flow data was 5 min. To verify the performance of AE-LSTM prediction model, we chose the proposed algorithm [31], SVM model and CNN for comparison. All prediction models used the same datasets. In CNN model, we set input layer as 28×28 , and convolutional layer as $20 \times 20 \times 20$. Moreover, the pooling layer of CNN was set to $20 \times 12 \times 12$, and learning rate was 0.1. We used ReLU as activation function in CNN. In SVM model, we used RBF as Kernel function. Moreover, after training, the regularization parameter was 7.5 and the Kernel function parameter was 0.54.

5.3. Model Evaluation

In the experiment, the performance of the traffic flow prediction models were measured by three indicators: Root Mean Square Error (RMSE), Mean Relative Error (MRE) and Mean Absolute Error (MAE). MRE is one of the most common indicators for comparing prediction accuracy. However,

when the traffic flow value is large, the MRE will be small. Therefore, MAE and RMSE were used to supplement the error analysis. Their expressions are as follows:

$$RMSE(\bar{x}, x) = \sqrt{\frac{1}{N} \cdot \sum_{n=1}^N (\bar{x}_n - x_n)^2}, \quad (18)$$

$$MRE(\bar{x}, x) = \frac{1}{N} \cdot \sum_{n=1}^N \frac{|\bar{x}_n - x_n|}{x_n}, \quad (19)$$

$$MAE(\bar{x}, x) = \frac{1}{N} \cdot \sum_{n=1}^N |\bar{x}_n - x_n|, \quad (20)$$

where \bar{x}_n represents the predicted value, x_n represents the observed value, and N represents the number of data.

5.4. Experimental Results

As shown in Table 2, we compared the errors of four prediction algorithms: CNN, SVM, the prediction algorithm in [17] and the AE-LSTM prediction algorithm. We compared the predict performance at different steps ahead. In general, the error of AE-LSTM prediction algorithm was smaller than the other compared algorithms. When the step ahead was 4, the MRE was 0.065. When the prediction of small step ahead was carried out, the performance of AE-LSTM algorithm had obvious advantage. When the prediction step ahead increased, the accuracy of prediction decreased slightly. In addition, the smaller was the prediction step ahead, the higher was the prediction accuracy and the better was the prediction performance. The reason for this phenomenon is that the multi-step prediction needs to use the previous prediction results. Unless the prediction is completely accurate, there must be an error between the predicted value and the observed value. Therefore, the larger is the step ahead of the prediction, the more errors of the previous prediction will be accumulated, leading to greater errors in the end. This is a common problem in the process of timing prediction. Therefore, as the prediction step size increased, the prediction accuracy of both AE-LSTM and the algorithm proposed in [17] increased, but the error of AE-LSTM increased less. It can be seen in Table 2 that the MRE of the two prediction algorithms were not significantly different. The reason is that the traffic flow was large, thus the difference between the predicted value and the observed value was relatively small. The difference between the predicted value and the observed value could be seen more intuitively from RMSE and MAE. Compared with the prediction algorithm in [17], RMSE and MAE of AE-LSTM had significant advantages.

As shown in Figure 5, we compared the observed value with the predicted results of four other prediction algorithms: CNN, SVM model, the algorithm proposed in [17] and the AE-LSTM prediction algorithm. The horizontal axis represents time, and every interval is 5 min. We predicted the traffic for 24 h, thus there are 288 data points. The vertical axis represents the number of traffic flows. In fact, the daily traffic pattern is similar. As shown in Figure 5, AE-LSTM had better prediction performance most of the time and matched well with the observed values. Compared with other prediction methods, AE-LSTM considered the influence of upstream and downstream traffic flow in the prediction process, it had better prediction performance at both night with less traffic flow and day with more traffic flow. The predictive performance of the algorithm proposed in [17] was better than that of CNN and SVM. Neither CNN nor SVM could well match observed values, but the overall prediction performance of SVM was worse.

Table 2. Performance comparison.

Algorithm	Step Ahead	Error Value		
		RMSE	MAE	MRE
AE-LSTM	4	26.32	16.15	0.065
	6	28.23	20.16	0.072
	8	76.87	43.15	0.131
The Proposed Algorithm [31]	4	35.45	25.26	0.088
	6	48.16	28.48	0.125
	8	99.52	52.86	0.161
CNN	4	46.22	34.59	0.011
	6	59.24	36.21	0.195
	8	105.16	59.86	0.198
SVM	4	49.54	39.02	0.023
	6	59.24	36.21	0.211
	8	107.36	62.86	0.209

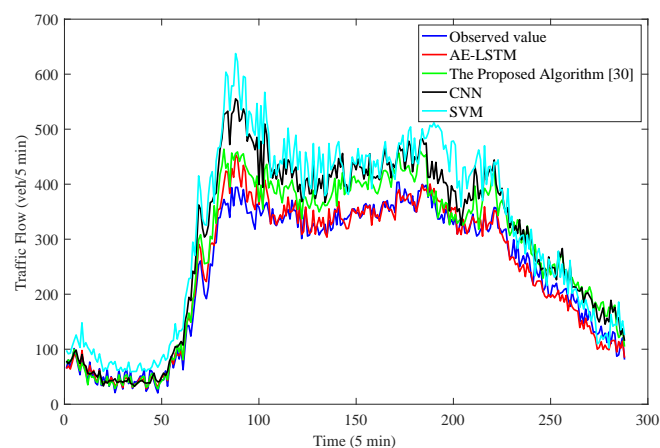


Figure 5. The blue line is the observation value on Sacramento County, the red line is the traffic flow prediction data obtained by AE-LSTM algorithm, and the green line represents the traffic flow prediction data obtained by the algorithm proposed in [17]. The black and light blue lines represent CNN and SVM, respectively.

Figures 6 and 7 show the MAR cumulative distribution functions for each prediction method under different conditions. In other words, it presents the probability statistics of MAR for each prediction method. Figure 6 shows the MAE cumulative distribution of traffic flow prediction. The solid line represents the prediction error of AE-LSTM algorithm, the dotted line represents the prediction error of the algorithm proposed in [17], and the shorter dotted line represents the prediction error of CNN. For the AE-LSTM prediction model, the probability of MAE less than 25 was 90% at three different stations. The errors of AE-LSTM prediction model on different datasets were basically the same, which indicates that the prediction model has good robustness. For the algorithm proposed in [17] and CNN, the cumulative distribution of MAE at different stations varied greatly, which indicates that the performance of these methods are not stable for different datasets. For the algorithm proposed in [17], the probability of MAE less than 25 was 60%. For CNN, the probability

of MAE being less than 25 was 30% at Stations 1 and 2, and 30% at Station 3. Overall, the prediction accuracy of CNN was not as high as that of the other two methods at the three stations. The prediction accuracy of AE-LSTM was higher than that of the algorithm proposed in [17] and CNN. The prediction error of the AE-LSTM and the prediction error of the algorithm proposed in [17] were different at Stations 1 and 2. At Station 3, AE-LSTM and the algorithm proposed in [17] had an almost equal probability of MAE less than 10. However, the probability of MAE less than 30 for the AE-LSTM prediction model was 95%, and the probability of MAE less than 25 for the algorithm proposed in [17] was less than 80%. Therefore, the prediction accuracy of AE-LSTM was higher than that of the algorithm proposed in [17].

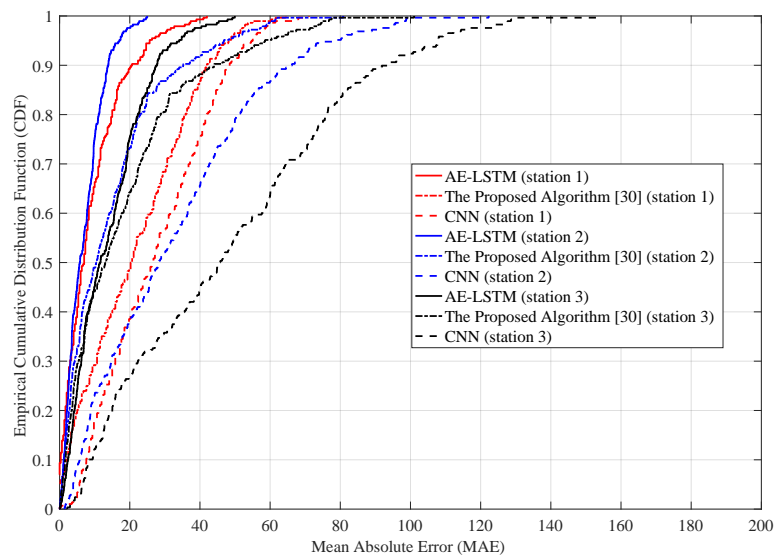


Figure 6. The cumulative distribution of MAE at different stations.

Figure 7 shows the cumulative distribution of MAE at different step ahead. We predicted the traffic flow at Steps 4, 6, and 8, respectively. As shown in Figure 7, the error distribution trend of the two prediction algorithms was similar for different prediction steps ahead. As the step ahead increased, the errors of the three prediction methods increased. The reason for this phenomenon is that the circular neural network needs to use the data generated in the previous cycle in the prediction process. The larger is the step ahead, the lower is the accuracy of the data in the previous moment. As a result, the errors of the previous prediction are accumulated and MAE becomes bigger. When the step ahead is small, the prediction accuracy is high. That is because the data used in the prediction are mostly observed values. In addition, the performance of AE-LSTM model was obviously better when the prediction step ahead was 4 and 6. However, when the prediction step ahead was 8, the prediction accuracy of the three methods was not high because the prediction step ahead was too big.

To comprehensively evaluate the performance of the prediction algorithm in practical applications, we analyzed forecast data for six consecutive days. Figure 8 shows the prediction error for six days. To analyze the performance of the prediction algorithm, the MRE was calculated every 4 h. We choose two prediction methods for comparison. Among them, the CNN prediction model directly uses the historical data of traffic flow for prediction, excluding the upstream and downstream traffic flow data. As shown in Figure 8, the trend of daily prediction error was similar: MRE was smaller during 0–4 and 20–24, while MRE was larger during the other times. Among them, The MRE of AE-LSTM algorithm fluctuated around 0.065, the MRE of the algorithm proposed in [17] fluctuated around 0.08, and the MRE of CNN was higher and fluctuated around 0.09, and the maximum value even reached 0.1.

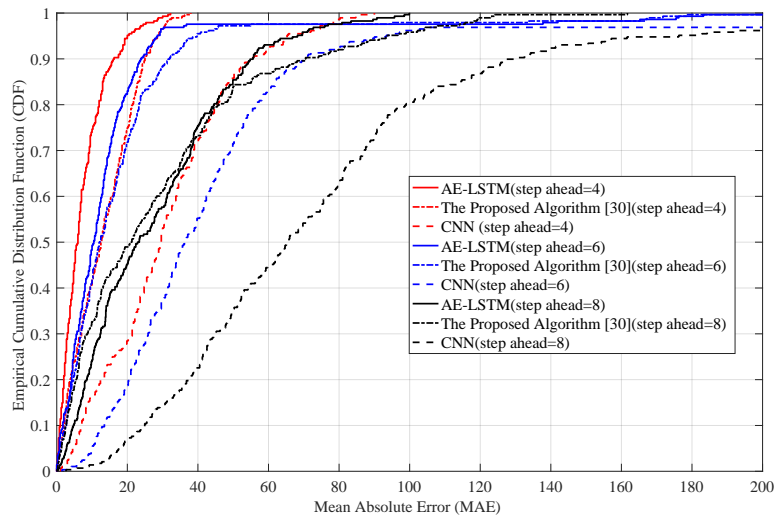


Figure 7. The cumulative distribution of MAE with different step-ahead.

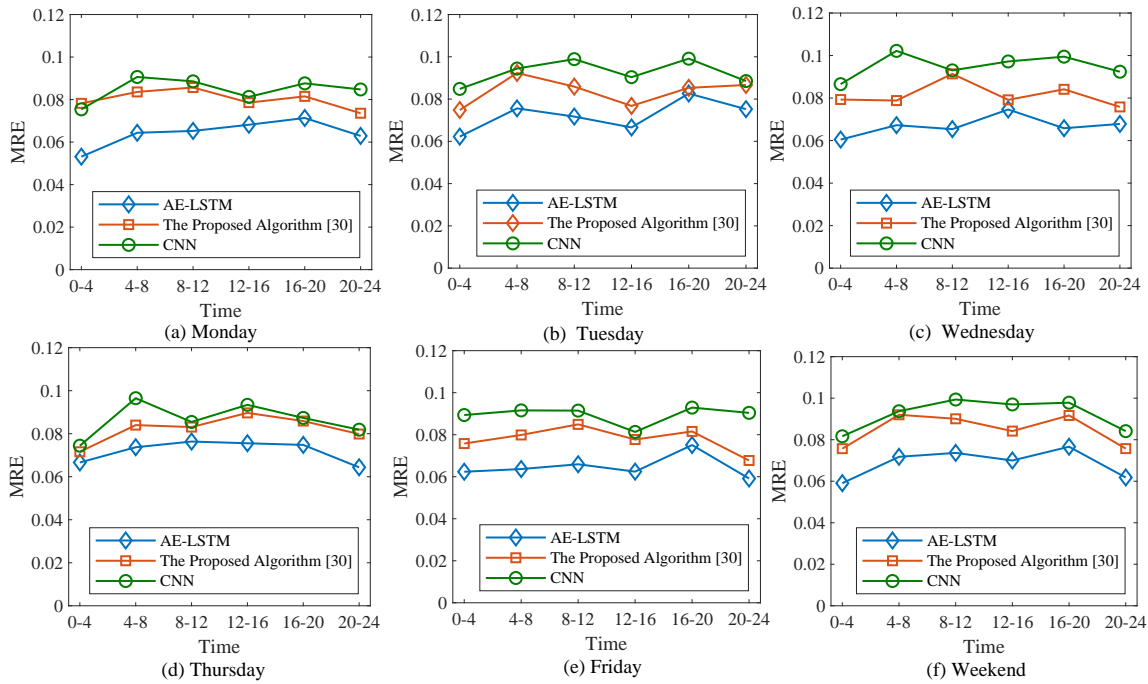


Figure 8. MRE statistics of the three prediction algorithms. We calculated the MAE of traffic flow prediction for six days.

Since the morning rush hour is from 6:00 to 9:00, the traffic flow may change greatly during this period, and traffic congestion and other problems often occur. To have more positive effects on traffic control, we analyzed the traffic flow data and forecast performance during this period. Figure 9 shows the observation values and forecast values of the traffic flow for six consecutive days between 6:00 and 9:00. As shown in Figure 9, the trend of daily traffic is similar. The traffic flow is 100 at 6:00, and the traffic flow fluctuation is 500 at 9:00. During this period, traffic flow fluctuates slightly, but the overall trend is increasing. We can see that the traffic flow on the weekend is less than that on the weekday. At 9:00 on the weekend, the traffic flow is about 400. In general, the predicted values of the AE-LSTM algorithm could better match the observed values and it had better prediction performance. AE-LSTM

fully considers the influence of upstream and downstream traffic flow on the current position, thus the result is more stable and accurate.

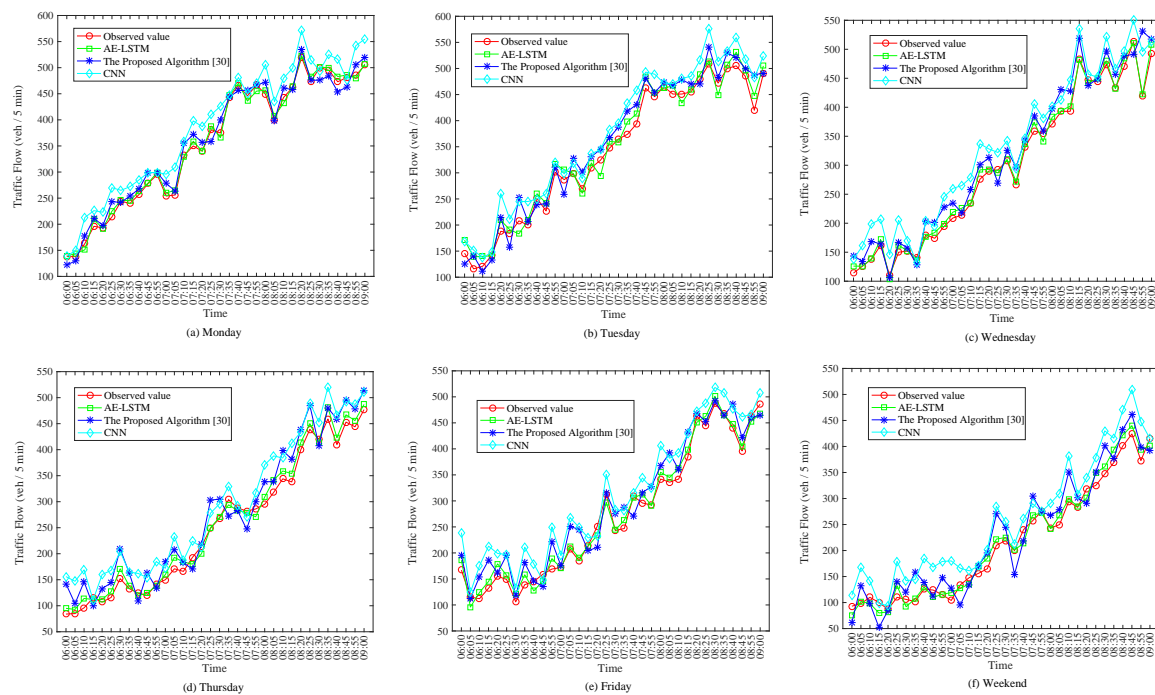


Figure 9. Observed and predicted values of traffic flow from 6:00 to 9:00.

6. Conclusions

In this paper, we propose an AE-LSTM prediction model to predict the traffic flow, which combines AutoEncoder and LSTM. The AE-LSTM prediction model not only considers the temporal characteristics but also uses the upstream and downstream data to capture the spatial characteristics of traffic flow. In addition, an AE-LSTM prediction algorithm is proposed. First, AutoEncoder and LSTM networks are trained, respectively, and then AE-LSTM is trained and fine-tuned. The algorithm is easy to implement and has good applicability. Finally, the performance of the AE-LSTM prediction model was verified by the real dataset from PeMS. Experimental results show that that AE-LSTM model had outstanding performance in traffic flow prediction. This study only considered time patterns and simple spatial patterns. In the future work, we will consider more complex spatial correlations and integrate them in neural networks.

Data Availability: In our experiment, we verified the performance of our prediction model on the dataset of PeMS [44].

Author Contributions: Conceptualization, W.W. and H.M.; Methodology, W.W. and H.M.; Model Implementation, W.W. and H.W.; Experimental and Analysis, W.W. and H.W.; Investigation, W.W., H.W. and H.M.; Writing—W.W.; Writing—Review & Editing, H.W. and H.M.; Supervision, H.M.; Funding Acquisition, H.M.

Funding: This work was supported in part by the Funds for International Cooperation and Exchange of the National Natural Science Foundation of China (No. 61720106007), the NSFC-Guangdong Joint Fund (No. U1501254), the National Key Research and Development Plan (No. 2016YFC0801005), the National Natural Science Foundation of China (No. 61602049), and the 111 Project (B18008).

Conflicts of Interest: The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Ai, Z.; Zhou, Y.; Song, F. A Smart Collaborative Routing Protocol for Reliable Data Diffusion in IoT Scenarios. *Sensors* **2018**, *18*, 1926. [[CrossRef](#)] [[PubMed](#)]
2. Zhang, M.; Quan, W.; Cheng, N.; Wu, Q.; Zhu, J.; Zheng, R.; Li, K. Distributed Conditional Gradient Online Learning for IoT Optimization. *IEEE Internet Things J.* **2019**. [[CrossRef](#)]
3. Wang, F.Y. Parallel Control and Management for Intelligent Transportation Systems: Concepts, Architectures, and Applications. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 630–638. [[CrossRef](#)]
4. Kong, Q.J.; Xu, Y.; Lin, S.; Wen, D.; Zhu, F.; Liu, Y. UTN-Model-Based Traffic Flow Prediction for Parallel-Transportation Management Systems. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1541–1547. [[CrossRef](#)]
5. Sims, A.; Dobinson, K.W. The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. *IEEE Trans. Veh. Technol.* **1980**, *29*, 130–137. [[CrossRef](#)]
6. Williams, B.M.; Durvasula, P.K.; Brown, D.E. Urban freeway travel prediction: Application of seasonal ARIMA and exponential smoothing models. *Transp. Res. Record J. Transp. Res. Board* **1998**, *1644*, 132–141. [[CrossRef](#)]
7. Okutani, I.; Stephanedes, Y.J. Dynamic prediction of traffic volume through Kalman filtering theory. *Transp. Res. Part B Methodol.* **1984**, *18*, 1–11. [[CrossRef](#)]
8. Xie, Y.; Zhang, Y.; Ye, Z. Short-Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition. *Comput.-Aided Civ. Infrastruct. Eng.* **2007**, *22*, 326–334. [[CrossRef](#)]
9. Zhang, Y.; Xie, Y. Forecasting of Short-Term Freeway Volume with v-Support Vector Machines. *Transp. Res. Record J. Transp. Res. Board* **2007**, *2024*, 92–99. [[CrossRef](#)]
10. Yu, G.; Hu, J.; Zhang, C.; Song, G. Short-term traffic flow forecasting based on Markov chain model. *IEEE Intell. Veh. Symp.* **2003**, 208–212. [[CrossRef](#)]
11. Tan, H.; Xuan, X.; Wu, Y. A Comparison of Traffic Flow Prediction Methods Based on DBN. In Proceedings of the Cota International Conference of Transportation Professionals, Shanghai, China, 6–9 July 2016; pp. 273–283.
12. Bengio, Y.; Lamblin, P.; Dan, P.; Larochelle, H. Greedy layer-wise training of deep networks. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 4–7 December 2006; pp. 153–160.
13. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 187–197. [[CrossRef](#)]
14. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the Chinese Association of Automation, Hefei, China, 19–21 May 2017; pp. 324–328.
15. Wang, J.; Hu, F.; Li, L. Deep Bi-directional Long Short-Term Memory Model for Short-Term Traffic Flow Prediction. In Proceedings of the Neural Information Processing 24th International Conference, Guangzhou, China, 14–18 November 2017; pp. 306–316.
16. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, W.Y. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [[CrossRef](#)]
17. Huang, W.; Song, G.; Hong, H.; Xie, K. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [[CrossRef](#)]
18. Quan, W.; Cheng, N.; Qin, M.; Zhang, H.; Chan, H.A.; Shen, X. Adaptive Transmission Control for Software Defined Vehicular Networks. *IEEE Wirel. Commun. Lett.* **2018**, *8*, 653–656. [[CrossRef](#)]
19. Zhang, M.; Yang, M.; Wu, Q.; Zheng, R.; Zhu, J. Smart perception and autonomous optimization: A novel bio-inspired hybrid routing protocol for MANETs. *Future Gen. Comput. Syst.* **2018**, *81*, 505–513. [[CrossRef](#)]
20. Liu, Y.; Yang, Z.; Ning, T.; Wu, H. Efficient Quality-of-Service (QoS) Support in Mobile Opportunistic Networks. *IEEE Trans. Veh. Technol.* **2014**, *63*, 4574–4584. [[CrossRef](#)]
21. 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.* **2003**, *129*, 664–672. [[CrossRef](#)]
22. Song, F.; Zhou, Y.; Wang, Y.; Zhao, T.; You, I.; Zhang, H. Smart Collaborative Distribution for Privacy Enhancement in Moving Target Defense. *Inf. Sci.* **2019**, *479*, 593–606. [[CrossRef](#)]
23. Williams, B. Multivariate Vehicular Traffic Flow Prediction: Evaluation of ARIMAX Modeling. *Transp. Res. Record J. Transp. Res. Board* **2013**, *1776*, 194–200. [[CrossRef](#)]

24. Stathopoulos, A.; Karlaftis, M.G. A multivariate state space approach for urban traffic flow modeling and prediction. *Transp. Res. Part C Emerg. Technol.* **2013**, *11*, 121–135. [[CrossRef](#)]
25. Kamarianakis, Y.; Prastacos, P. Space-time modeling of traffic flo. *Comput. Geosci.* **2005**, *31*, 119–133. [[CrossRef](#)]
26. Min, W.; Wynter, L. Real-time road traffic prediction with spatio-temporal correlations. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 606–616. [[CrossRef](#)]
27. Kumar, S.V.; Vanajakshi, L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *Eur. Transp. Res. Rev.* **2015**, *7*, 21. [[CrossRef](#)]
28. Pan, T.L.; Sumalee, A.; Zhong, R.X.; Payoong, N.I. Short-Term Traffic State Prediction Based on Temporal-Spatial Correlation. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1242–1254. [[CrossRef](#)]
29. Xu, Y.; Kong, Q.J.; Klette, R.; Liu, Y. Accurate and Interpretable Bayesian MARS for Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2457–2469. [[CrossRef](#)]
30. Sun, S.; Zhang, C.; Yu, G. A bayesian network approach to traffic flow forecasting. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 124–132. [[CrossRef](#)]
31. Qi, Y.; Ishak, S. A Hidden Markov Model for short term prediction of traffic conditions on freeways. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 95–111. [[CrossRef](#)]
32. Xie, Y.; Zhao, K.; Sun, Y. Gaussian processes for short-term traffic volume forecasting. *Transp. Res. Record J. Transp. Res. Board* **2010**, *2165*, 69–78. [[CrossRef](#)]
33. Zhao, J.; Sun, S. High-Order Gaussian Process Dynamical Models for Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2014–2019. [[CrossRef](#)]
34. Zheng, X.; Chen, W.; Wang, P.; Shen, D.; Chen, S.; Wang, X.; Zhang, Q.; Yang, L. Big Data for Social Transportation. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 620–630. [[CrossRef](#)]
35. Cheng, N.; Lyu, F.; Chen, J.; Xu, W.; Zhou, H.; Zhang, S.; Shen, X. Big Data Driven Vehicular Networks. *IEEE Netw.* **2018**, *32*, 160–167. [[CrossRef](#)]
36. Tan, M.C.; Wong, S.C.; Xu, J.M.; Guan, Z.R.; Zhang, P. An aggregation approach to short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 60–69.
37. Chen, Q.; Song, X.; Yamada, H.; Shibasaki, R. Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident inference. In Proceedings of the Association for the Advance of Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 338–344.
38. Yang, L.; Ma, R.; Zhang, H.M.; Guan, W.; Jiang, S. Driving behavior recognition using EEG data from a simulated car-following experiment. *Accident Anal. Prev.* **2018**, *116*, 30–40. [[CrossRef](#)] [[PubMed](#)]
39. Wang, J.; Gu, Q.; Wu, J.; Liu, G.; Xiong, Z. Traffic Speed Prediction and Congestion Source Exploration: A Deep Learning Method. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 499–508.
40. Wu, Y.; Tan, H.; Qin, L.; Ran, B.; Jiang, Z. A hybrid deep learning based traffic flow prediction method and its understanding. *Transp. Res. Part C Emerg. Technol.* **2018**, *90*, 166–180. [[CrossRef](#)]
41. Zhan, H.; Gomes, G.; Li, X.S.; Madduri, K.; Sim, A.; Wu, K. Consensus Ensemble System for Traffic Flow Prediction. *IEEE Trans. Intell. Transp. Syst.* **2018**, *99*, 1–12. [[CrossRef](#)]
42. Bengio, Y. Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [[CrossRef](#)]
43. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
44. Caltrans, Performance Measurement System (PeMS). 2018. Available online: <http://pems.dot.ca.gov> (accessed on 29 March 2019).

