Singular Point Probability Improve LSTM Network Performance for Long-term Traffic Flow Prediction

Boyi Liu1, Jieren Cheng1,2*, Kuanqi Cai3, Pengchao Shi3, Xiangyan Tang1

1. College of Information Science & Technology, Hainan University, Haikou, 570228
2. State Key Laboratory of Marine Resource Utilization in South China Sea, 570228, Haikou, China
3. Mechanical and Electrical Engineering College, Hainan University, Haikou, 570228

cjr22@163.com*

Abstract: Traffic flow forecasting is the key in intelligent transportation system, but the current traffic flow forecasting method has low accuracy and poor stability in the long time period. For this reason, a combined forecasting method (SDLSTM-ARIMA) based on improved LSTM neural network (SDLSTM) and time series (ARIMA) is proposed in this paper. Firstly, the concept and calculation method of time singularity ratio of traffic data stream is proposed, and the adaptive over-fitting improvement of LSTM neural network is presented. Namely, the time singularity ratio is used as the probability value of dropout neuron in Dropout module, So as to get SD-LSTM model. Then, the SD-LSTM model can predict an unsatisfying result in a certain period of time. This paper through a large number of training experiments to determine the specific time period and the forecast method of the time period is replaced by ARIMA. The combination of SDLSTM and ARIMA is combined with non-equal interval, and the SDLSTM-ARIMA combined traffic flow forecasting method is proposed to realize the accurate prediction of 24-hour traffic flow data. Theoretical analysis and experimental results show that the SDLSTM-ARIMA has a high accuracy rate, stability and wide application prospect in the long-time traffic flow forecast with hourly period.

Keywords: traffic flow forecasting; LSTM; ARIMA; depth learning

1 INTRODUCTION

With the development of social economy and transportation, traffic problems appear more frequently. The traditional mode of transport has encountered more and more challenges, attracted worldwide attention. In recent years, many countries have invested lots of manpower and resources to carry out the development of management and control technology in the road transportation system. ITS (Intelligent Transportation System) has been developed rapidly[1, 2]. Accurate traffic flow forecasting is the prerequisite and the key step to realize ITS, it is conducive to improving the efficiency of transport operations and the quality of people's travel. Traffic flow forecasting is also helpful to alleviate the road congestion, reduce carbon emissions, and conserve the energy and so on. Especially with the rapidly development of big data technology, some methods predict the traffic flow data and plan the vehicle travel path relying on the current and historical traffic flow data. These method forecast the traffic flow reasonably and designed the best route for vehicles, realizing the traffic's balanced distribution in the road network and improving road utilization. The longer the time of traffic flow prediction, the greater the value of its utility. However, the current researches are mainly to solve the short-term traffic flow forecasting problem. Accuracy of the long-term traffic flow forecasting is low. The paper made a research on this problem, and proposed a new traffic flow prediction algorithm with higher accuracy and longer prediction time.

Traffic flow is an important measure of the state of the road network. It refers to the number of vehicles through a road section during a period of time [3]. The excellent traffic flow prediction algorithm can predict the traffic flow data for a certain period of time earlier and more accurately. Traffic flow data is affected by many factors, for example the noise and some non-linear interferences. So its rule is difficult to grasp, especially in the long-term traffic flow forecast [4], which has been a difficult point. In recent years, many traffic flow prediction algorithms have been proposed [5, 6]. They can be broadly divided into two categories according to their forecasting basis: one prediction model is based on mathematical statistics and traditional mathematical such as calculus [7]; the other is a prediction model based on modern science and technology methods [8].

The first class certainly includes many traffic flow prediction algorithms. One of the representative results is the time-series model [9] used in the traffic flow prediction field for the first time by Ahmed and Cook in 1979. It includes the auto regressive model (AR) [10], the moving average model (MA) [11], and the auto regressive moving average model (ARMA) [12]. The technology is mature and has high accuracy when the sample data is sufficient. It is usually used in relatively stable traffic perdition. The method required a lot of uninterrupted data and it is easily to be interfered by random factors. Stephanedes proposed History Average Model [13] applied to urban traffic control system in 1981. The algorithm is simple and fast, but cannot cope with emergencies. Okutani and Stephanedes proposed Kalman Filtering Model [14] for the traffic flow's prediction in 1984, its predictive factor selection is flexibility and has high precision and good robustness. However, this method requires a lot of matrix calculations and its forecast value is delayed for several time periods sometimes, which making it difficult to realize real-time online prediction. In addition, a series of traffic flow forecasting methods have been proposed in recent years, spatial-temporal characteristics-based analysis [15], random forest model [16] and similarity model [17], etc.

One of the representative traffic flow prediction algorithm of the second class is Davis and Nihan's Nonparametric Regressive Model [18] applied to traffic flow prediction in 1991. Without prior knowledge, it can perform more accurate than parametric modeling only with sufficient historical data, but its complexity is also high. Dougherty proposed neural network[19] for traffic flow prediction in 1995, which is suitable for complex and non-linear conditions, and it can be effective to predict when the data is incomplete and inaccurate with good adaptability and fault-tolerance, but it requires a lot of learning data and the training process is complex; The classification regression tree method[20] for the traffic flow forecast proposed by Xu Yanyan et al. in 2013 has a better prediction effect and interpretability, but requires a lot of training data and certain skills for parameter adjustment. In addition, plenty of traffic flow forecasting methods based on the above methods, deep belief network model[21], support vector machine[22], wavelet neural network model[23], hybrid neural network model[24] have been proposed in recent years.

The traffic flow forecasting model proposed above has certain improvement in accuracy, but its prediction time of high precision is limited to 5min ~ 15 minutes while its prediction accuracy is not high during 30min ~ 60min, and its stability is poor. Aiming at this problem, this paper proposed an unequal interval combining model based on improved LSTM [25] and ARIMA [26], which can guarantee the higher accuracy rate on the basis of increasing the prediction time and the length of the time period.

2 Improvement of LSTM Neural Network

2.1 LSTM Neural Network

The LSTM neural network [27] is a special type of RNN (recurrent neural networks) [28]. RNN is an efficient and accurate depth neural network, which has outstanding effect in long-term dependence on data learning[29] and has been applied well in the field of machine translation[30], pattern recognition[31] and so on. However, it has a problem called "gradient disappearance"[32]. And LSTM was raised to solve the problem of RNN, which is characterized by the ability to learn long-term dependency information. LSTM was proposed by Hochreiter and Schmidhuber in 1997[33]. In recent years, LSTM has derived many variants, of which the relatively popular variant with the added "peephole connection" is proposed by Gers and Schmidhuber in 2000[34]. In addition, Yao proposed a variant using the Depth gate [35], it is different from LSTM that it also decides what to forget and what new information to add. Another novel modified variant is the Gated Recurrent Unit (GRU) proposed by Cho, et al. in 2014[36], which combined the forget gate and input gate to a single update gate. The specific structure of the LSTM model is shown in Fig.1. Below:

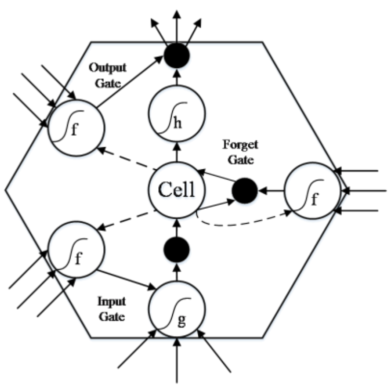


Fig. 1. Structure of LSTM neural network

In the figure above, Cell represents the memory of the neuron state and it sets the state of the state record; Input Gate and Output Gate are used to receive parameters, output parameters, modify the parameters; Forget Gate is a correction parameter that forgets the state of the upper neuron. In the model above, the three weight values in each storage unit come from input training, including the complete hidden state in the previous time step. Three weights are brought into the input node, forget gate, and output gate respectively. The activation function (S-type function) is connected to a black node, and the internal state of the unit is the most central node. The weight across the time step is set to 1, while the self-feedback is made, and the constant error conveyor (CEC) is the connection edge of the internal state. In the model, if the input sequence is set to $(x_1, x_2, \ldots, x_T)$ and the state of the hidden layer is set to $(h_1, h_2, \ldots, h_T)$, then at time t, there are:

$$i_t = sigmoid(W_{hi}h_{t-1} + W_{xi}X_t) \tag{1}$$

$$f_t = sigmoid(W_{hf}h_{t-1} + W_{hf}X_t) \tag{2}$$

$$o_t = sigmoid(W_{ho}h_{t-1} + W_{hx}X_t + W_{co}c_t) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{hc}h_{t-1} + W_{xc}X_t) \tag{4}$$

$$h_t = o_t \odot \tanh(c_t) \tag{5}$$

In the formula above, it represents input gate, *ft* represents forget gate, *ot* represents output gate, *ct* represents cell, *Wh* represents the weight of recursive link, *Wx* represents the weight from the output layer to the hidden layer, the activation functions are sigmoid and *tanh*.

As can be seen from the above figure and formula, LSTM neural network is a special RNN, it can learn from the information of a long time to solve the problem of gradient disappearance by increasing the memory unit. Therefore, this paper applied it to the hourly traffic flow forecast of the middle and long time period for the first time.

2.2 LSTM Neural Network Based on Self-adaptive Probabilities

The LSTM neural network has the function of preventing the gradient disappearance and long-term memory, but it also has the problem of over-fitting [37]. The so-called over-fitting phenomenon is that the trained model has a good performance on the training data set, but its performance on the test set is poor [38]. The causes of this phenomenon include excessive noise interference, high model complexity and so on. In this paper, the situation that the LSTM neural network is applied to the traffic flow prediction, making the noise interference an important incentive for the over-fitting phenomenon.

To solve the over-fitting problem, Hinton proposed a solution that uses Dropout in 2014[39]. Dropout refers to discard the neural network unit from the network temporarily according to a certain probability during the training process of the depth learning network. That is, Dropout randomly selects a part of the neurons, then sets its output as 0, and remains its previous values at the same time, and restores the previous retention value in the next training process, and then randomly selects, and repeats this process. In this way, the network structure changes in each training process, so as to avoid the situation that a feature is effective only with the support of the specific characteristics of other features, thus reducing the probability of over-fitting in the training process.

Although Hinton, et al. proposed Dropout to reduce the probability of over-fitting, but they do not go into the calculation method seriously of the key parameter involved in Dropout - the probability of selective discarding neurons, while they use the empirical value of 0.5. The reason is that the network structure generated randomly is the most in this case. In recent years, the empirical value is also used in the related applications based on LSTM. In order to solve this problem, this paper made a study and proposed the method of calculating the probability value of selective discarding neurons in Dropout to improve the self-adaptive over-fitting of LSTM neural network.

In the improved scheme proposed in this paper, the probability value of selective discarding neurons is replaced by the traffic data time singularity ratio. The reason is that the over-fitting phenomenon has a certain relationship with the amount of the noise. Too much noise will lead to the situation that the training result performs well on the training noise while it performs badly on the real data, which will lead to the poor performance on the test set; and it is really easy to fall into the local feature optimal solution when the noise is too small. Therefore, the proportion of singular points has an important impact on the training results. And the probability value of selectively discarding neurons in Dropout also expresses the proportion of the screening of data to a certain extent. Therefore, there is a large degree of critical link between the two. At the same time, it was found in this paper that if we use the time singularity ratio as the probability value of selectively discarding neurons in Dropout, we can guarantee that the singular points are not discarded totally and they exist to a certain degree. The reason is as follows:

$$\frac{N_d}{N_j} = \frac{N_q}{N} \tag{6}$$

$$N_j = N_d + N_u \tag{7}$$

It can be deduced form (6), (7) that:

$$\frac{N_{qd}}{N_j} \leq \frac{N_d}{N_j} \tag{8}$$

In the formula above, N_d represents the number of discarded nodes, N_j represents the number of nodes of each layer, N_q represents the number of singular points, N represents the number of all nodes in the single-layer network, N_u represents the number of nodes that is not discarded in the single-layer network, N_{qd} represents the number of nodes in the single-layer network that are both discarded and belong to noise. It can be seen that the improved method proposed in this paper can make the probability of selecting the node needed to delete randomly in Dropout more reasonable, and its effect to prevent over-fitting problem is more prominent.

We call the improved neural network Adaptive to prevent over-fitting LSTM neural network (SD-LSTM). S represents Singularity ratio; D represents Dropout. The structures of the normal neural network and SD-LSTM are shown as follows:

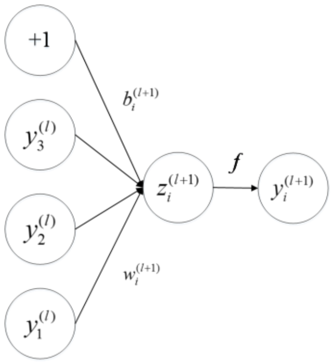


Fig. 2. Standard neural network layer

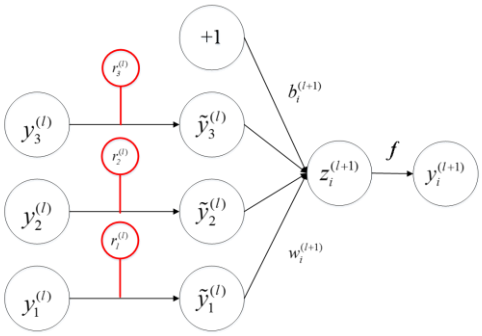


Fig. 3. SD-LSTM layer

Formula expressions are as shown below, the unimproved formulas are as follows:

$$\begin{aligned} z_i^{(l+1)} &= w_i^{(l+1)} y^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}). \end{aligned} \tag{9}$$

The formulas of Adaptive to prevent over-fitting LSTM neural network are as follows:

$$\begin{aligned} & p = \frac{N_q}{N}, \\ & r_j^{(l)} \sim Bernoulli(p), \\ & \tilde{y}_i^{(l)} = r_j^{(l)} \times y^{(l)}, \\ & z_i^{(l+1)} = w_i^{(l+1)} \tilde{y}_i^{(l)} + b_i^{(l+1)}, \\ & y_i^{(l+1)} = f(z_i^{(l+1)}). \end{aligned} \tag{10}$$

In Fig.2. Fig.3. and (9), (10) above, $z_i^{(l+1)}$ represents the value of the i-th neuron of the l+1-th layer, $w_i^{(l+1)}$ represents the weight of the i-th connection of the l+1-th layer, $b_i^{(l+1)}$ represents the bias of the i-th neuron of the l+1-th layer, $y_i^{(l+1)}$ represents the output of the i-th neuron of the l+1-th layer, f represents the activation function, p represents the expectation of probability, $r_j^{(l)}$ reflects the case whether the j-th neuron of the l-th layer is discarded or not, $\tilde{y}_i^{(l)}$ represents the output of the i-th neuron of the l-th layer after Dropout.

2.3 Traffic Data Flow Time Singularity Ratio Definition and Algorithm

To obtain the value of the time singularity ratio of the traffic flow proposed in this paper, it is necessary to obtain the number of singular points and the number of all the sample points, where the latter is known. Therefore, we only need to calculate the number of singular points. And for the detection methods of singular point, domestic and foreign scholars have been studied [40], but in this paper, we need to carry out the detection of singular point in traffic flow. In view of the high temporality of traffic flow, a method of self-adaptive singular point detection using time series is proposed. The flow chart of the algorithm is shown in Fig.4. Below.

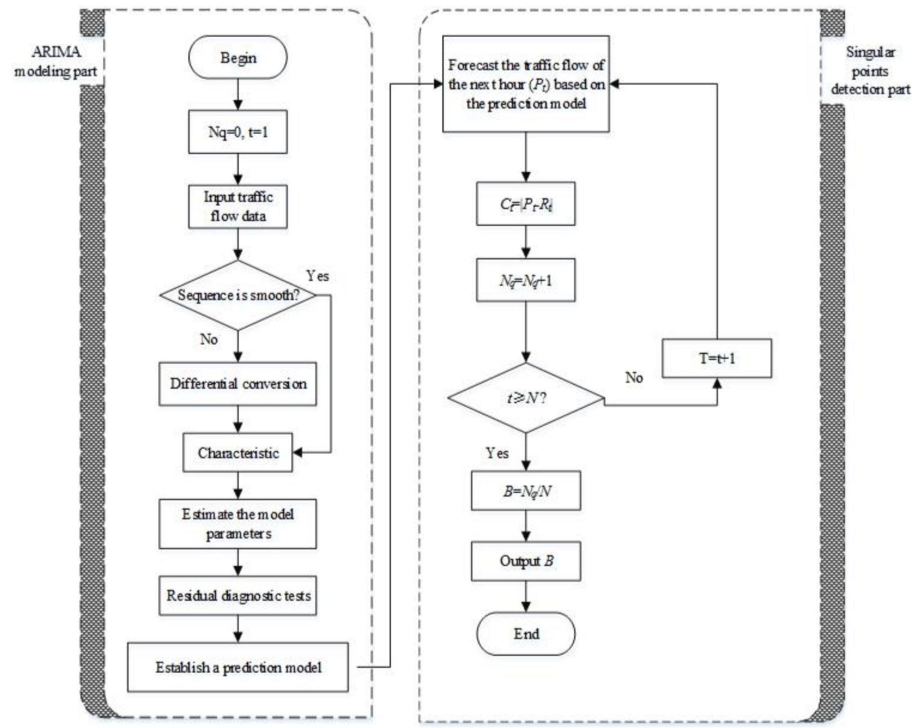


Fig.4. Flowchart of singular ratio determination method in improved LSTM neural network

In the figure above, N_q represents the number of singular points, t is the sequence of time periods in the traffic flow data, Ct represents the difference between the traffic flow predicted during the time period and the actual traffic flow, P_t represents the traffic flow predicted during the time period, R_t represents the actual traffic flow during the time period, N_t represents the number of time periods in the data set, N represents the number of all data in the data set, and B is the required singularity ratio.

As shown in Fig.4. The method of determining the time singularity ratio of the traffic data flow is mainly composed of two parts. One part is the establishment of the ARIMA model, and the steps are the same as that of the general ARIMA model, which include the smoothness detection, differential transformation, feature analysis, parameter estimation, etc. [41].

The other part is the singular point detection part of the self-adaptive traffic flow. In this part, based on the ARIMA prediction model obtained in the previous section and the previous data, the data of the next time period are predicted successively to obtain Pt, and then we calculated the difference Ct of Pt and the actual traffic flow Rt of the time period. When the difference is obtained, it is compared with the threshold to determine whether the data belongs to the singular point. For the selection of thresholds, this paper considered that the order of magnitude of traffic flow is different at different time intervals, so it is unreasonable to set constant threshold, which will lead to big error of the result of singular point detection. Thus, this paper used 10% of the average of the traffic flow over a period of time in the data set as the threshold of the singular point detection in the time period, so that the threshold is changing over the time period, the traffic flow data is self-adaptive in different time periods and the accuracy of singular point detection is much higher. After the threshold comparison is made, the number of singular points can be counted and then divided by the number N of all data points in the data set to obtain the determined singularity ratio.

3 Combination Forecasting Model Based on SD-LSTM and ARIMA

3.1 Traffic Flow Predicting Method Based on SDLSTM

After putting forward SD-LSTM, this paper applied it into the traffic flow prediction according to its features of great learning capacity on long term dependence of data and anti-overfitting. As shown in fig.5. Below, the traffic flow prediction result is obtained by using the improved LSTM model. The picture is plotted by time period as the horizontal axis and the values of MAPE representing error measurement as the vertical axis. There are 7 lines indicates the changing condition of error measurement MAPE in different time during a day by predicting the traffic flow every day of a week according to the improved LSTM model.

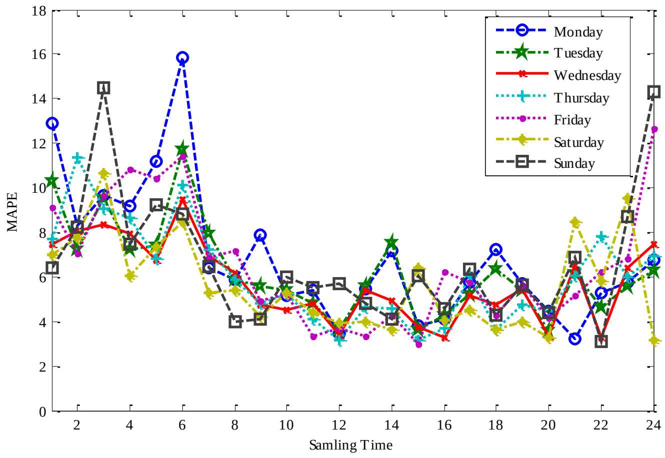


Fig.5. Traffic flow prediction error distribution based on improved LSTM

From the figure we can see that the result of using improved LSTM model to predict the traffic flow is better, and the most measurement figures of MAPE are under 20. Meanwhile, most of the MAPE figures are under 10 in the time of heavy traffic flow. At the same time, this paper analyzed the time slot of high MAPE figures and found that low traffic flow base would cause high MAPE figures. However, the traffic flow base of 6 o'clock was not low while the MAPE figure of it was high. Aiming at this, SDLSTM-ARIMA model is raised in this paper to predict the traffic flow.

3.2 SDLSTM-ARIMA Combination Predicting Method

This paper found that the main reason led to high MAPE in 6 o'clock is that the traffic flow changed severely during this period. The complicated features and high real time meant that LSTM cannot learn the whole features of this time slot. Thus, the prediction using deep learning method is not suit for this time slot. Meanwhile, ARIMA algorithm does not demand too much on data volume, and it has high real time and low algorithm complexity [42]. In result, this paper aimed at solving the non-ideal result of 6 o'clock predication by bringing in traffic flow prediction method based on ARIMA. ARIMA doesn't have the training process of data learning, so it is much suit for shorter period prediction. And the result is not ideal in the medium and long time prediction. Aiming at this problem, this paper solved it by combining LSTM and ARIMA with non-equal interval, that is the non-equal interval traffic flow prediction method based on SDLSTM neural network and ARIMA (a.k.a. SDLSTM-ARIMA).

Non-equal interval, that is, in the prediction period of LSTM, regarding 1 hour as unit time; in prediction period of ARIMA, regarding 15 min as unit time. Under this circumstance, the traffic flow prediction in different periods during one day forms the condition of the combination of non-equal intervals, as shown in Fig.6. Below.

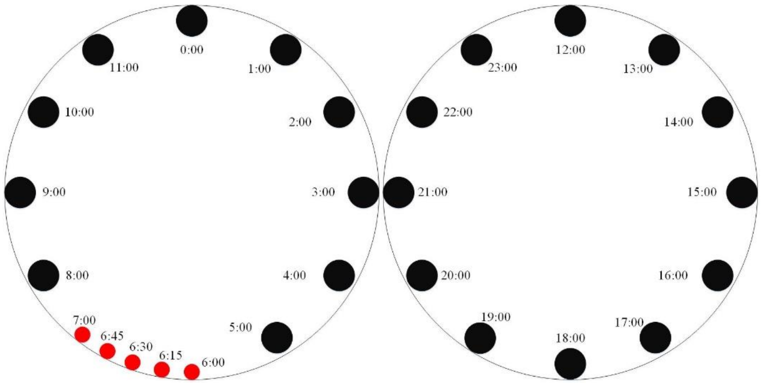


Fig. 6. SDLSTM-ARIMA non-equal interval combination diagram

In the figure above, the prediction in 6 o'clock used 15 min as a circle (red dot), the prediction of other time slots regarded 1 hour as a circle (black dot). Then, by the mode of non-equal interval combination, we combined the advantages of LSTM and ARIMA models together to improve the real time and accuracy of the traffic flow prediction.

The above theoretical analysis proves that SDLSTM-ARIMA can reach higher accuracy in the traffic flow prediction. At last, this paper proved it through experiment.

4 Experiment

4.1 Data and environment of the experiment

The experimental data in this paper comes from the traffic flow data set published publicly in the official website of British Columbia of Canada [43], and the experiment is based on the data from Vancouver Richmond region. The road condition at the monitoring point are shown in Fig.7. And the specific location is shown in Fig.8. During the process of the experiment, a total of 43824 hours of data (from January 1, 1999 to December 31, 2003) of the monitoring point are used, with data of May 21, 2003 and after as test data and data before that date as training data. The total test data has 5400 hours and the total training data has 38424 hours.

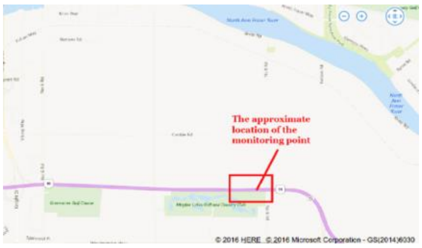


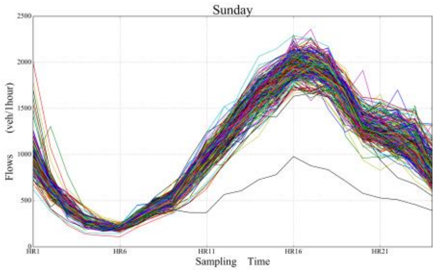
Fig. 7. Monitoring point and its surrounding roads Fig.8**.** The location of the monitoring point

Firstly, this paper took a week as a cycle and number of weeks as label, and extracted traffic flow in the data by day. As shown form Fig.9. To Fig.15. The abscissa is time and the ordinate is traffic flow. It can be seen from the figures that the number of different days in a week has a relatively fixed trend, which further proves the feasibility of using the model obtained by LSTM training.

Thus, this model can be used to model the different days of a week based on the trend. Through a large number of data training, we added the characteristics of a week to the model, so that the training model can adapt to the changing trend each week during the four years, and the law of the four years also has enough universality. The forecasting model obtained by four years' traffic flow data training is more consistent with long historical trends, and a lower error rate is guaranteed in this dimension. In the training process, we increased the training input vector to 24 hours, and regarded the output of the forecast data as the traffic flow for the next hour. By using the method of deep learning, the characteristics of traffic flow information are excavated, and

24 hours' traffic flow is entered in real time. This paper took the factors of real-time traffic flow and historical traffic flow into account, ensured real-time and improved accuracy at the same time.

The experiment of this paper is implemented by python 3.5.2 language in the 2-core 4-thread, 2.5GHz, 8G memory computer with Linux system (Ubuntu 16.04 64-bit), where the traffic flow is collected every 1 hour.



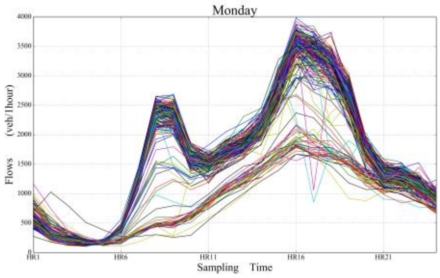
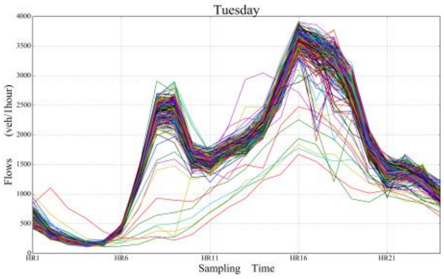


Fig. 8. Traffic flow data of Sunday

Fig.10. Traffic flow data of Monday



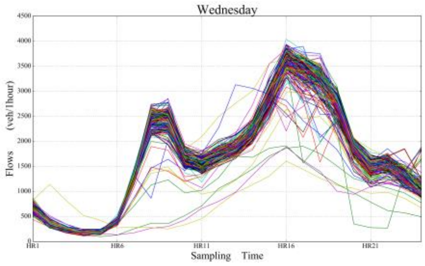
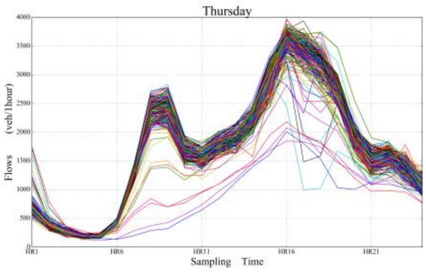


Fig.11. Traffic flow data of Tuesday

Fig.12. Traffic flow data of Wednesday



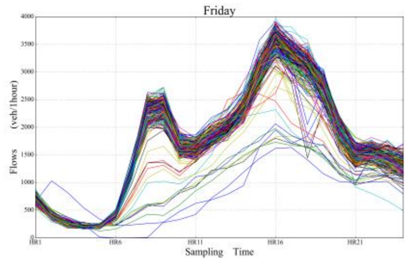


Fig.13. Traffic flow data of Thursday

Fig.14. Traffic flow data of Friday

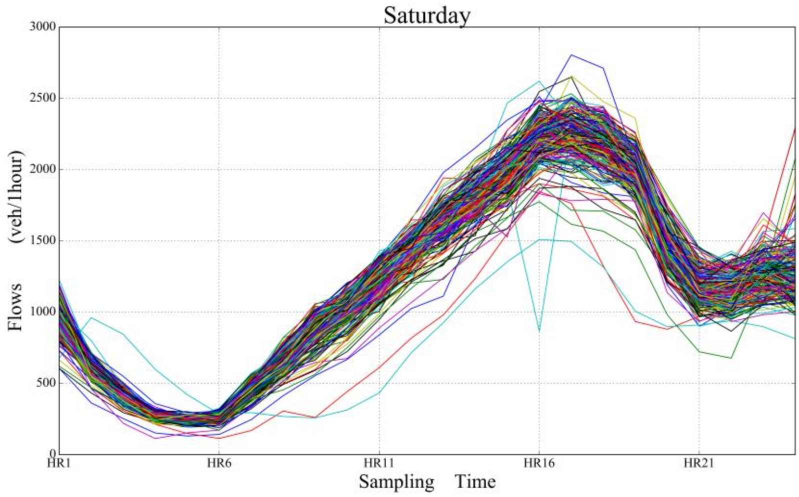


Fig. 15. Traffic flow data of Saturday

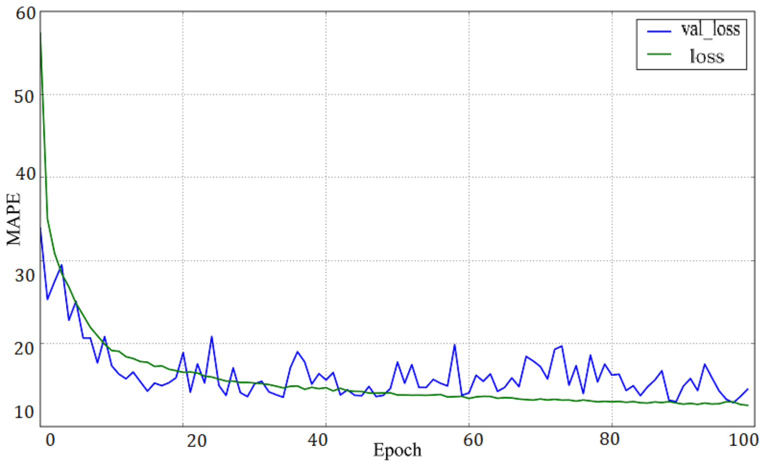


Fig.16. The change of MAPE with epoch during training

4.2 Result and Analysis of SDLSTM-ARIMA Traffic Flow Prediction

The formula below shows the method of calculating the MAPE value of the data deviation. As shown in Fig.10. The MAPE value [44] is changing during the training process of LSTM by the data training set. It can be seen from the figure that the MAPE value decreases with the increase of epoch [45], and the MAPE value tends to be stable after the epoch value reaches 40. It shows that the number of the training done to sample set in this paper is enough sufficient.

$$MAPE = (\sum \frac{\frac{X - Y}{X}}{N}) \times 100\% \div N \tag{11}$$

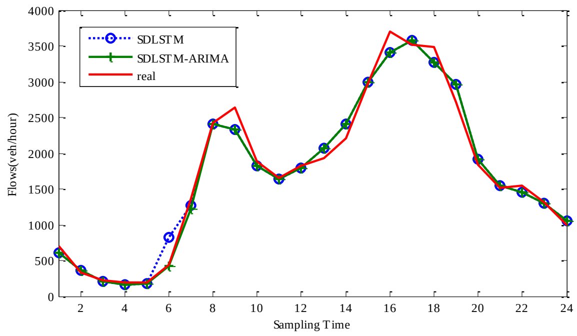


Fig.17. Comparison of predicted scenarios between SDLSTM-AR and LSTM

The improved LSTM neural network for data training is used in this paper, and the process of MAPE changes are shown in Fig.16. Above. The ordinary LATM neural network method obtained by training is compared with the SDLSTM-ARIMA method proposed in this paper, as shown in Fig.17., where the red part of the figure shows the improvement of the accuracy of the prediction after the introduction of the ARIMA model, and the result of the comparison proved the effectiveness of SDLSTM-ARIMA.

Finally, the SDLSTM-ARIMA model obtained from the training of the data set is tested in the test data set in this paper. Subsequently, this article selected some of the results and made them visualized, Fig. 18. Shows the forecast value and the actual value of one day of the results; Fig. 19. Shows the forecast value and the actual value of one week of the results; Fig. 20. Shows the forecast value and the actual value of one month of the results; as can be seen from the graph, in the LATM-AR experimental test results, the predicted traffic flow data is basically consistent with the actual data and the method has high accuracy.

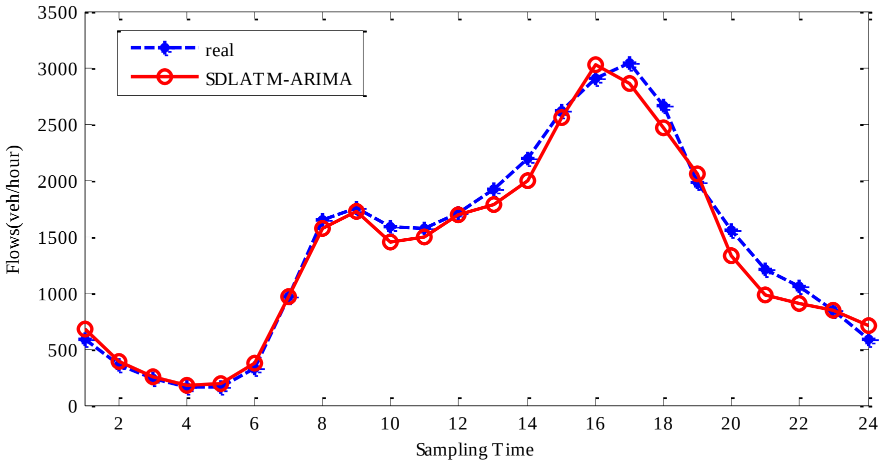


Fig.18. The predicted values obtained by SDLSTM-AR method compare with the real values in a day

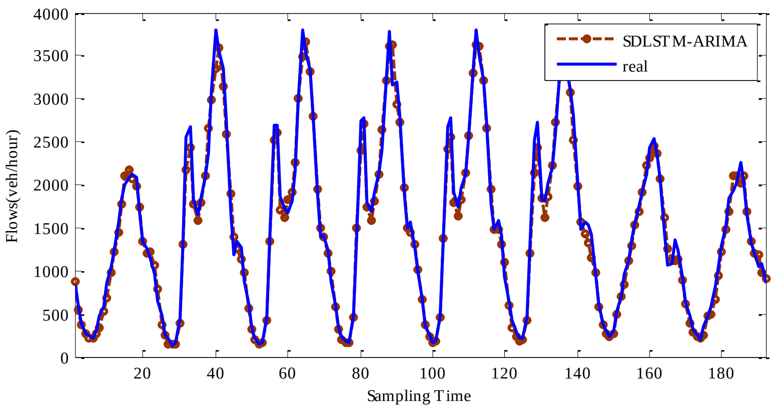


Fig.19. The predicted values obtained by SDLSTM-AR method compared with the real values in a week

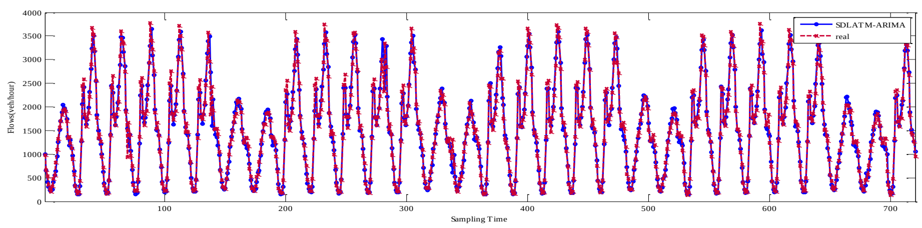


Fig.20. The predicted values obtained by SDLSTM-AR method compared with the real values in a month

4.3 Comparison of the results of different kinds of traffic flow forecast method

Using the training data set and the test data set, this paper compared the commonly used ARIMA prediction method and the latest proposed AR-RBLTFa method in reference [17] with the SDLSTM-ARIMA method proposed in this paper. Fig. 21. Shows the comparison of the data of working days, Fig. 22. Shows the comparison of the data of Non-working days. As can be seen from the figure, compared with the commonly used ARIMA prediction method, AR-RBLTFa method and SDLSTM-ARIMA have higher accuracy.

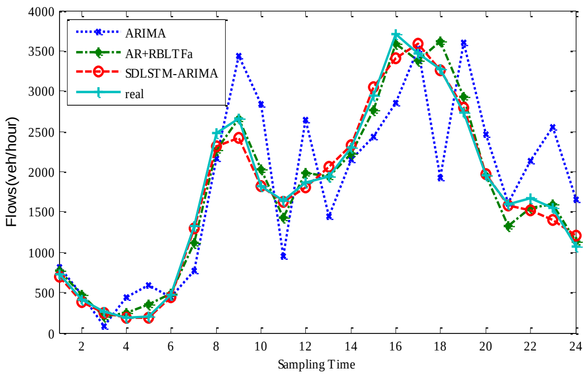


Fig.21. Comparison of experimental results of different methods of working days

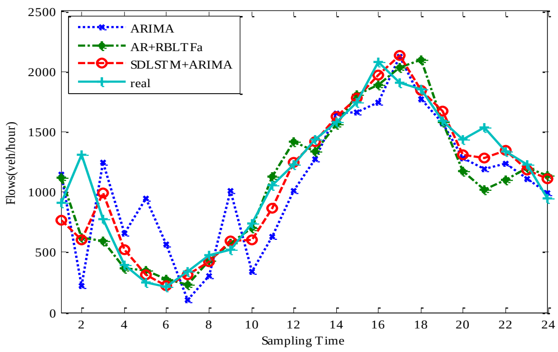


Fig.21. Comparison of experimental results of different methods of non-working days

After obtaining the traffic flow data of the three methods, in order to measure the error of the three methods much better, the MAPE value and the absolute error of the three methods are calculated and compared in this paper. Fig. 23. Below shows the comparison of MAPE values of working days of the three methods; Fig. 24. Below shows the comparison of MAPE values of non-working days of the three methods; Fig. 25. Below shows the comparison of the absolute error of working days of the three methods; Fig. 26 below shows the comparison of the absolute error of non-working days of the three methods; Fig. 27. Below shows the comparison of RMSE values of working days of the three methods; Fig. 28. Below shows the comparison of RMSE values of non-working days of the three methods; And RMSE calculation method is as shown in the Formula (12) below:

$$RMSE=\sqrt{\frac{\sum_{i=1}^{n}(x_{real}-x_{pre})^2}{n}} \tag{12}$$

In the formula above, n represents the number of time points, represents the real traffic flow at the time point, and represents the predicted traffic flow at the time point.

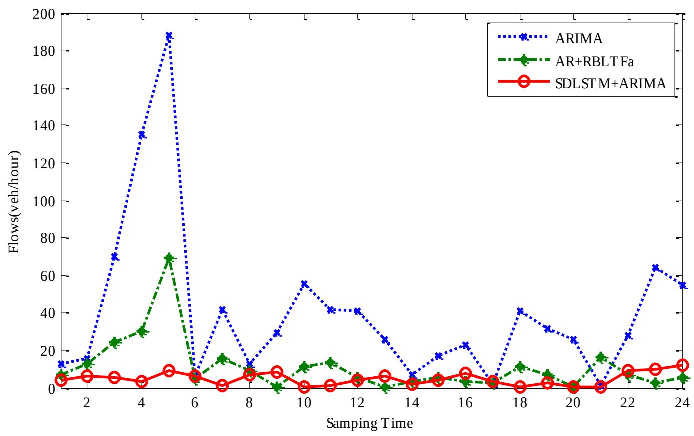


Fig.23. Comparison of MAPE values of workday of three methods

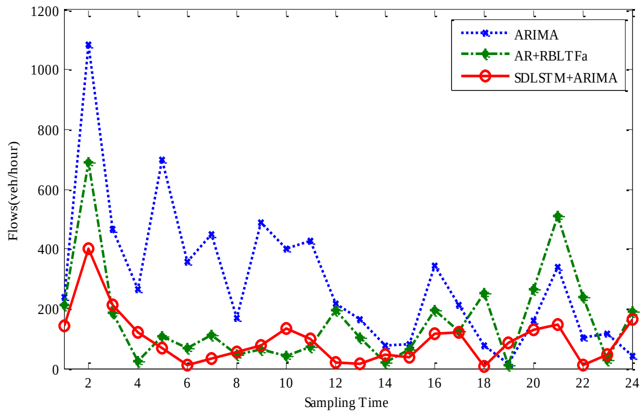


Fig.24. Comparison of MAPE values of non-workday of three methods

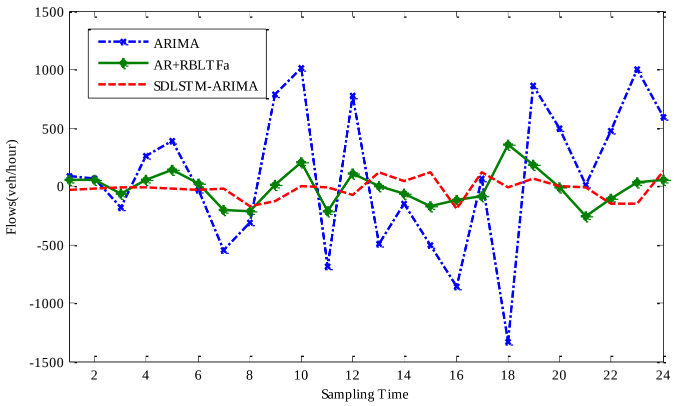


Fig.25. Comparison of absolute errors of workday of three methods

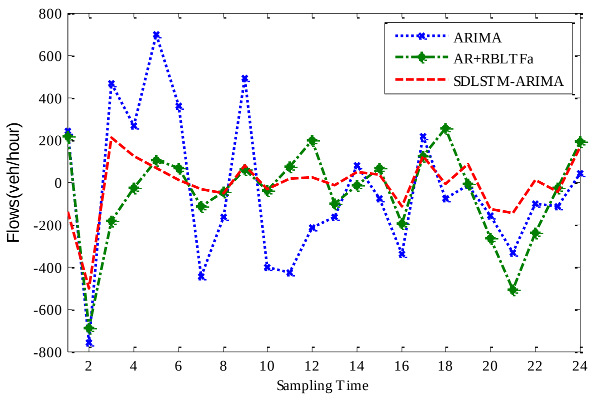


Fig.26. Comparison of absolute errors of non-workday of three methods

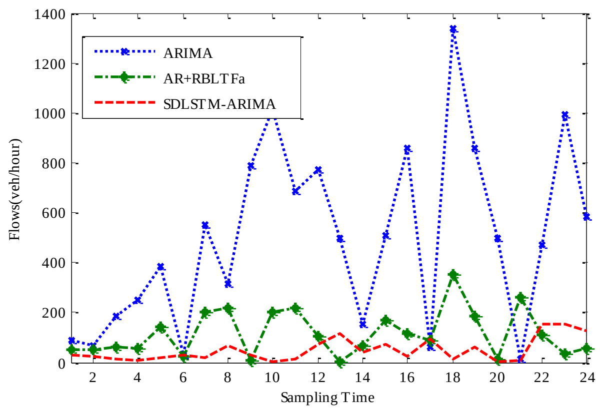


Fig.27. Comparison of RMSE of workday of three methods

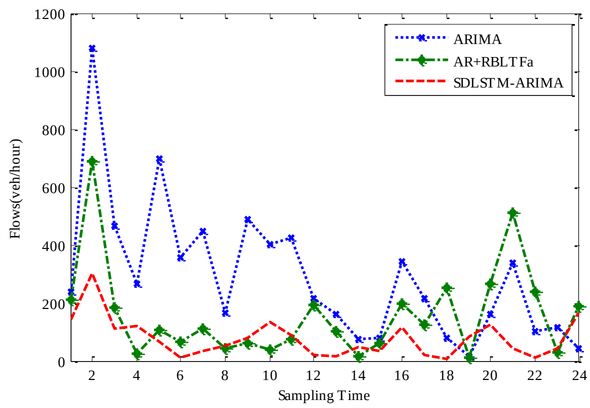


Fig.28. Comparison of RMSE of non-workday of three methods

As can be seen from Fig. 16. To Fig. 21, ARIMA has the greatest error among the three kinds of error criteria, and the error of the AR-RBLTFa method is slightly larger than that of the SDLSTM-ARIMA. However, the AR-RBLTF method is not stable enough. For example, the error of the AR-RBLTFa method is similar to that of the AR method in Fig. 16. Which is easy to produce serious potential hazard in the practical application. Then, the practical error of the three methods is quantitatively compared, as shown in Table 1. As can be seen from the table, the accuracy rate is: SDLSTM-ARIMA > AR-RBLTFa > ARIMA, while the error stability is: SDLSTM-

ARIMA > AR-RBLTFa > ARIMA. Therefore, the SDLSTM-ARIMA proposed in this paper has higher accuracy and stability.

Table 1. Quantitative comparison of three methods

Evaluating Indicator	ARIMA	AR-RBLTFa	SDLSTM-ARIMA
MPAE	40.43	14.46	10.26
RMSE	611.4	184.49	176.90
Absolute Error	498.59	156.41	138.25
Error Entropy (normalized)	1	0.24	0.09

At last, this paper summarized the advantages and disadvantages of the three methods, as shown in Table 2 below:

Table 2. Comparison of three prediction methods

Algorithm	Accuracy	Dependence on historical data	Stability of forecast data	Real-time of the prediction
ARIMA	general	slight low	low	Predict on time
AR-RBLTFa	high	high	general	Predict on time
SDLSTM-ARIMA	slight high	slight high	high	Delay in variable scale adjustment

5 Conclusion

Aiming at the problem of traffic flow prediction algorithm cannot reach ideal result in medium and long time slot, the SDLSTM method is put forward. This article defined the calculation method of time singularity ratio of the traffic flow firstly, improved LSTM neural network and put forward the probability values of selectively discarding neurons of the Dropout model by using time singularity ratio as self-adaptive data environment to deal with the problem of over-fitting in LSTM neural network and achieve adaptively of the traffic flow data set. Then, this article applied SDLSTM neural network in the traffic flow prediction. Aiming at the 6 o'clock error, the ARIMA model is introduced to predict traffic flow of 6 o'clock accurately by using the combination of non-equal intervals, which raised up the accuracy of the whole method. At last, this article verified the method by experiment and compared it with other methods. The result shows that SDLSTM-ARIMA proposed in this article has higher accuracy and stability. This method converts the traffic big data to practical value by using big data technology and machine learning. And it has broad application prospects. Especially in recent years, the rapid development of cloud computing and large data technology [46-52], making the proposed traffic flow prediction algorithm has a greater application prospects.

REFERENCES

[1] Alam M, Ferreira J, Fonseca J. Introduction to intelligent transportation systems[C]//Intelligent Transportation Systems. Springer International Publishing, 2016: 1-17.

[2] Moral-Muñoz J A, Cobo M J, Chiclana F, et al. Analyzing highly cited papers in Intelligent Transportation Systems [J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 993-1001.

[3] Bright P W, Balmforth Z, MacPherson J L. The effect of changes in traffic flow on mammal road kill counts[J]. Applied Ecology and Environmental Research, 2015, 13: 171-179.

[4] Habtemichael F G, Cetin M. Short-term traffic flow rate forecasting based on identifying similar traffic patterns[J]. Transportation Research Part C: Emerging Technologies, 2016, 66: 61-78.

[5] Lv Y, Duan Y, Kang W, et al. Traffic flow prediction with big data: a deep learning approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(2): 865-873.

[6] Hou Y, Edara P, Sun C. Traffic Flow Forecasting for Urban Work Zones[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(4): 1761-1770.

[7] Romero D, Rico N, Garcia-Arenas M I. Modellation and Forecast of Traffic Series by a Stochastic Process[M]//Time Series Analysis and Forecasting. Springer International Publishing, 2016: 279-292.

[8] Lopez-Garcia P, Onieva E, Osaba E, et al. A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(2): 557-569.

[9] Fan Na, Zhao Xiangmo, Dai Ming, et al. Short-term traffic flow prediction model[J]. Journal of Traffic and Transportation Engineering, 2012,04:114-119.

[10] Kumar S V, Vanajakshi L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data[J]. European Transport Research Review, 2015, 7(3): 1-9.

[11] Moorthy C K, Ratcliffe B G. Short term traffic forecasting using time series methods[J]. Transportation planning and technology, 1988, 12(1): 45-56.

[12] Williams B, Durvasula P, Brown D. Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models[J]. Transportation Research Record: Journal of the Transportation Research Board, 1998 (1644): 132-141.

[13] Stephanedes Y J, Michalopoulos P G, Plum R A. Improved estimation of traffic flow for Real-Time control (Discussion and closure)[J]. Transportation Research Record, 1981 (795).

[14] Okutani I, Stephanedes Y J. Dynamic prediction of traffic volume through Kalman filtering theory[J]. Transportation Research Part B: Methodological, 1984, 18(1): 1-11.

[15] Misra C, Fan Y, Davatzikos C. Baseline and longitudinal patterns of brain atrophy in MCI patients, and their use in prediction of short-term conversion to AD: results from ADNI[J]. Neuroimage, 2009, 44(4): 1415-1422.

[16] Teles J, Pina C, Edén P, et al. Transcriptional regulation of lineage commitment-a stochastic model of cell fate decisions[J]. PLoS Comput Biol, 2013, 9(8): e1003197.

[17] Hou Z, Li X. Repeatability and Similarity of Freeway Traffic Flow and Long-Term Prediction Under Big Data[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(6): 1786-1796.

[18] Davis G A, Nihan N L. Nonparametric regression and short-term freeway traffic forecasting[J]. Journal of Transportation Engineering, 1991, 117(2): 178-188.
[19] Dougherty M. A review of neural networks applied to transport[J]. Transportation Research Part C: Emerging Technologies, 1995, 3(4): 247-260.
[20] Xu Yanyan, Zhaixi, Kong Xiqing, et al. Short-term prediction method of freeway traffic flow[J]. Journal of Traffic and Transportation Engineering, 2013,02:114-119.
[21] Liu F, Liu B, Sun C, et al. Deep belief network-based approaches for link prediction in signed social networks[J]. Entropy, 2015, 17(4): 2140-2169.
[22] Bai C, Peng Z R, Lu Q C, et al. Dynamic bus travel time prediction models on road with multiple bus routes[J]. Computational intelligence and neuroscience, 2015, 2015: 63.
[23] Yanchong C, Darong H, Ling Z. A short-term traffic flow prediction method based on wavelet analysis and neural network[C]//Control and Decision Conference (CCDC), 2016 Chinese. IEEE, 2016: 7030-1034.
[24] Moretti F, Pizzuti S, Panzieri S, et al. Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling[J]. Neurocomputing, 2015, 167: 3-7.
[25] Tian Y, Pan L. Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network[C]//2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity). IEEE, 2015: 153-158.
[26] Butt M. Selection of forecast model for consumption (four sectors) and transmission (two Piplines) of natural gas in Punjab (Pakistan) based on ARIMA model[J]. International Journal of Advanced Statistics and Probability, 2015, 3(1): 115-125.
[27] Zhu X, Sobhani P, Guo H. Long short-term memory over recursive structures[C]//Proceedings of the 32nd International Conference on Machine Learning. 2015: 1604-1612.
[28] Rakkiyappan R, Chandrasekar A, Cao J. Passivity and passification of memristor-based recurrent neural networks with additive time-varying delays[J]. IEEE transactions on neural networks and learning systems, 2015, 26(9): 2043-2057.
[29] LukošEvičIus M, Jaeger H. Reservoir computing approaches to recurrent neural network training[J]. Computer Science Review, 2009, 3(3): 127-149.
[30] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.
[31] Liang M, Hu X. Recurrent convolutional neural network for object recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3367-3375.
[32] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks[J]. ICML (3), 2013, 28: 1310-1318.
[33] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[34] Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM[J]. Neural computation, 2000, 12(10): 2451-2471.
[35] Yao K, Cohn T, Vylomova K, et al. Depth-Gated Recurrent Neural Networks[J]. 2015.Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.
[36] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
[37] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.

[38] Bailey D H, Borwein J M, de Prado M L, et al. Pseudomathematics and financial charlatanism: The effects of backtest over fitting on out-of-sample performance[J]. Notices of the AMS, 2014, 61(5): 458-471.
[39] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
[40] Awad A I, Baba K. Singular point detection for efficient fingerprint classification[J]. International Journal of New Computer Architectures and their Applications (IJNCAA), 2012, 2(1): 1-7.
[41] Wang W, Chau K, Xu D, et al. Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition[J]. Water Resources Management, 2015, 29(8): 2655-2675.
[42] Pati J, Shukla K K. A comparison of ARIMA, neural network and a hybrid technique for Debian bug number prediction[C]//Computer and Communication Technology (ICCCT), 2014 International Conference on. IEEE, 2014: 47-53.
[43] Traffic data program of British Columbia. Public Traffic Data[EB/OL]. http://www.th.gov.bc.ca/trafficData/legacy/TDP-97-03.html
[44] Chen J, Wang Y, Gu C, et al. Enhancement of the mechanical properties of basalt fiber-wood-plastic composites via maleic anhydride grafted high-density polyethylene (MAPE) addition[J]. Materials, 2013, 6(6): 2483-2496.
[45] Dunleavy K, Pittaluga S, Maeda L S, et al. Dose-adjusted EPOCH-rituximab therapy in primary mediastinal B-cell lymphoma[J]. New England Journal of Medicine, 2013, 368(15): 1408-1416.
[46] Zhangjie Fu, Kui Ren, Jiangang Shu, Xingming Sun, and Fengxiao Huang, "Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement," IEEE Transactions on Parallel and Distributed Systems, 2016,27(9): 2546–2559.
[47] Zhihua Xia, Xinhui Wang, Xingming Sun, and Qian Wang, "A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data," IEEE Transactions on Parallel and Distributed Systems, 2015,27(2): 340-352.
[48] Bin Gu and Victor S. Sheng. A Robust Regularization Path Algorithm for ν-Support Vector Classification. IEEE Transactions on Neural Networks and Learning Systems, 2016,PP(99):1-8.
[49] Bin Gu, Xingming Sun, and Victor S. Sheng, Structural Minimax Probability Machine. IEEE Transactions on Neural Networks and Learning Systems, 2016, PP(99):1-11.
[50] Zhangjie Fu, Xingming Sun, Qi Liu etal. Achieving Efficient Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing. IEICE Transactions on Communications, 2015, E98-B (1):190-200.
[51] Qi Liu, Weidong Cai, Jian Shen, Zhangjie Fu, Xiaodong Liu, Nigel Linge, "A speculative approach to spatial‐temporal efficiency with multi‐objective optimization in a heterogeneous cloud environment," Security and Communication Networks, 2016,9(17).
[52] Zhiguo Qu, John Keeney, Sebastian Robitzsch, Faisal Zaman, Xiaojun Wang. "Multilevel Pattern Mining Architecture for Automatic Network Monitoring in Heterogeneous Wireless Communication Networks", China Communications, 2016, 13(7): 108-116.
[53] Bin Gu, Victor S. Sheng, Zhijie Wang, Derek Ho, Said Osman, and Shuo Li, "Incremental learning for ν-Support Vector Regression," Neural Networks, vol. 67, pp. 140-150, 2015. DOI: 10.1016/j.neunet.2015.03.013
[54] Bin Gu, Victor S. Sheng, Keng Yeow Tay, Walter Romano, and Shuo Li, "Incremental Support Vector Learning for Ordinal Regression," IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 7, pp. 1403-1416, 2015. DOI:

10.1109/TNNLS.2014.2342533
[52] Yu Xue, Jiongming Jiang, Binping Zhao and Tinghuai Ma, "A self-adaptive articial bee colony algorithm based on global best for global optimization", Soft Computing, 2017. DOI: 10.1007/s00500-017-2547-1