A Network Traffic Prediction Method Based on LSTM

WANG Shihao, ZHUO Qinzheng, YAN Han, LI Qianmu, and QI Yong
(School of Computer Science and Engineering, Nanjing University of Science and Technology,
Nanjing, Jiangsu 210094, China)

Abstract: As the network sizes continue to increase, network traffic grows exponentially. In this situation, how to accurately predict network traffic to serve customers better has become one of the issues that Internet service providers care most about. Current traditional network models cannot predict network traffic that behaves as a nonlinear system. In this paper, a long short-term memory (LSTM) neural network model is proposed to predict network traffic that behaves as a nonlinear system. According to characteristics of autocorrelation, an autocorrelation coefficient is added to the model to improve the accuracy of the prediction model. Several experiments were conducted using real-world data, showing the effectiveness of LSTM model and the improved accuracy with autocorrelation considered. The experimental results show that the proposed model is efficient and suitable for real-world network traffic prediction.

Keywords: recurrent neural networks; time series; network traffic prediction

DOI: 10.12142/ZTECOM.201902004

http://kns.cnki.net/kcms/detail/34.1294.TN.20190619.0902.002.html, published online June 19, 2019

Manuscript received: 2018-02-13

1 Introduction

As Transmission Control Protocol/Internet Protocol (TCP/IP) networks become more and more important in modern society, how to better understand and correctly predict the behavior of the network is a vital point in the development of information technology. For medium/large network providers, network prediction has become an important task and get more and more attention [1]. By improving the accuracy of network prediction, network providers can better optimize resources and provide better service quality [2]. Not only that, network traffic prediction can help detect malicious attacks in the network too. For example, denial of service or spam attacks can be detected by comparing real traffic and predicting traffic [3]. The earlier these problems are detected, the more reliable network services can be obtained [4], [5].

With the development of computational science, some relatively reliable methods of network traffic prediction have been proposed to replace the intuitive prediction, especially in the field of sequential sequence prediction.

Nancy and George [6] used the time series analysis method to make an accurate prediction of the National Science Foundation Network (NSFNET) traffic data, and then carried out a series of research on the method of time series data, especially network traffic data prediction [7], [8]. Modeling methods include the autoregressive moving average model (ARMA), Autoregressive Integrated Moving Average model (ARIMA), Fractional Autoregressive Integrated Moving Average model (FARIMA), etc. Large-scale network system is a complex nonlinear system, and it is influenced by many external factors. Because of that its macroscopic flow behavior is often complex and changeable, and the data contains many kinds of periodic fluctuations. In addition, it shows non-linear trend and contains uncertain random factors, which may prevent the flow model with linear characteristics from being predicted accurately. There-

This paper supported by ZTE Industry-Academia-Research Cooperation Funds under Grant No. 2016ZTE04-11, National Key Research and Development Program: Key Projects of International Scientific and Technological Innovation Cooperation Between Governments under Grant No. 2016YFE0108000, Fundamental Research Funds for the Central Universities under Grant (30918012204), and Jiangsu Province Key Research and Development Program under Grant (BE2017739).

fore, how to select and optimize the nonlinear model is the key to predict network traffic in recent years. Support Vector Machine (SVM), Least Squares Support Vector Machine (LSSVM) , artificial neural network, echo state network and so on all have certain improvement to the prediction accuracy. However, these models ignore the network traffic data from the autocorrelation, although they consider the nonlinear characteristics of data. Therefore, how to use nonlinear time sequence data to predict network traffic linked with autocorrelation characteristic is a problem to be solved eagerly.

2 Flow Prediction Model

This section describes recurrent neural networks (RNN), which are used for network traffic prediction, and also introduce a special type of RNN, long short-term memory networks (LSTM). On the basis of this, we put forward the model of LSTM combination with the characteristics of network traffic autocorrelation.

2.1 Recurrent Neural Networks

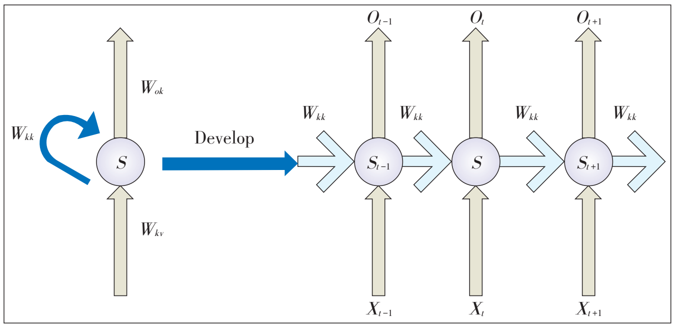
RNN is a popular learning method in the fields of machine learning and deep learning in recent years, which is different from the traditional feedforward neural network (FNN). FNN neurons transmit information through the input layer, hidden layer, and the connection of the output layer. Each input item is independent of others, and there is no connection between neurons in the same layer. However, RNN introduces the recurrent structure in the network, and establishes the connection of the neuron to itself. Through this circular structure, neurons can "remember" information from the previous moment in the neural network and influence the output of the current moment. Therefore, RNN can better deal with the data with time series, and in the prediction of timing data, it often performs better than FNN. The structure diagram of RNN is shown in **Fig. 1**.

The prediction process of RNN is similar to that of FNN, which is calculated by the forward propagation algorithm (**Algorithm 1**).

Algorithm 1. Forward propagation algorithm

```
For $t$ from 1 to $T$ do
    $u_t \leftarrow w_{hv} v_t + w_{hh} h_{t-1} + b_n$
    $h_t \leftarrow e(u_t)$
    $o_t \leftarrow W_{ot} h_t + b_o$
    $z_t \leftarrow g(o_t)$
End for
```

In Algorithm 1, T is the length of the input data, v_t is the input for time t, z_t is the output for time t. W_{hv}, W_{hh}, and W_{oh} are the link matrix of input layer to hidden layer, hidden layer to hidden layer and hidden layer to output layer. Functions e and g are the activation functions of the hidden layer and the output layer respectively.



▲Figure 1. The recurrent neural networks structure diagram.

There are some differences between RNN training and FNN. FNN is implemented by back propagation (BP) algorithm. However, RNN affects the error of the output layer because of the hidden layers of the previous moments, so the results of the backward propagation need to be propagated on the time dimension, named back propagation through time (BPTT) algorithm. The BPTT of RNN defines the partial derivative of the loss function to the t input value of the neuron j at time t firstly, and then solves the function with the derivative of the chain (**Algorithm 2**).

Algorithm 2. Back propagation through time (BPTT) algorithm

```
For $t$ from 1 to $T$ do
    $do_t \leftarrow g^{'}(o_t) * \frac{dL(z_t, y_t)}{dz_t}$
    $db_o \leftarrow db_o + do_t$
    $dW_{oh} \leftarrow dw_{oh} + do_t h_t^T$
    $dh_t \leftarrow dh_t + W_{oh}^T do_t$
    $du_t \leftarrow e^{'}(u_t) dh_t$
    $dW_{hv} \leftarrow dW_{hv} + du_t v_t^T$
    $db_h \leftarrow db_h + du_t$
    $dW_{hh} \leftarrow dW_{hh} + du_t h_{t-1}^T$
    $dh_{t-1} \leftarrow W_h^h du_t$
End for
Return $d\theta = [dW_{hv}, dW_{hh}, dW_{oh}, db_h, db_o, dh_o]$
```

The partial derivative between the loss function and the neuron is affected by the output layer of the current time t and the hidden layer of $t+1$ at the next moment. For each time step, we use the chain rule to sum up all the results in the time dimension and get the partial derivative of the loss function to the weight w of the neural network. The weight of the recursive neural network is updated by gradient descent method until the condition is met.

The final step in the RNN training process is that the gradient is multiplied repeatedly in the propagation process. If the eigenvalues $W_{hh} > 1$, this will result in a gradient explode; if the eigenvalues $W_{hh} < 1$, this will result in a gradient vanish [9]–[12],. For this problem, Hochreiter et al. [13] put forward a LSTM neural network.

2.2 Long Short Term Memory

LSTM neural network is a variant of RNN. The key is to replace the neurons with cell states. The cell state is delivered over the time chain, with only a few linear interactions, and information is easily maintained on cell units. Each cell contains one or more memory cells and three nonlinear summation units. The nonlinear summation unit is also called the "gate", which is divided into three kinds: "Input gate", "Output gate" and "Forget gate". They control the input and output of memory cells by matrix multiplication. The structure diagram of LSTM is shown in **Fig. 2** [14].

The forward propagation algorithm of LSTM is similar to RNN, and the input data is a time series of length *T*:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2}$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{3}$$

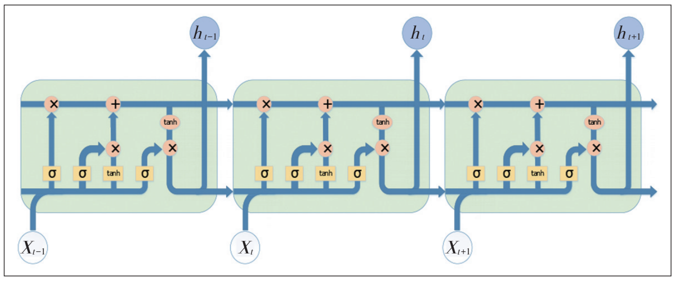
$$C_t = f_t * C_{t-1} + i_t * C'_t, \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t * \tanh(C_t). \tag{6}$$

Eq. (1) is the parameter of forget gate, where 1 means "complete reservation", while 0 means "completely abandoned". Eqs. (2) and (3) calculate the value of the input gate. Then Eq. (4) is used to discard the information that we need to discard, plus useful information entered at time *t*. Eqs. (5) and (6) determine which part of the cell state we are going to output to the next neural network and the next time.

The BPTT algorithm of LSTM is similar to that in RNN. It starts from the end of the time series (time *T*), gradually reverses the gradient of each parameter, and then updates the network parameters with the gradient of each time step. The output value corresponding to the partial derivative of memory cells is calculated first, then the partial derivative of the output gate is calculated, and the corresponding partial derivatives of the memory cell state, forget gate and input gate are calculated respectively. Eventually, the model is updated using the gradient descent method connection weights.



▲**Figure 2. LSTM structural diagram of neural network model.**

2.3 Long Short Term Modeling Applicable to Autocorrelation Sequence

Time series is an ordered data set in time. Each data corresponds to the corresponding time. The time series model assumes that the data of the past can be applied to the future, and we can predict future data by studying the information of historical data.

In the conventional sequential sequence prediction problem, let's say the time series is $(y_1, y_2, y_3 \cdots y_t)$, y_t represents the data value at time *t*. We predict the value of y_{m+1} by using $(y_{m-k+1}, y_{m-k+2} \cdots y_m)$, where *k* represents the number of steps to be used for each prediction.

Four types of prediction can be defined according to the granularity of time [15]: current-time prediction, short - term prediction, medium-term prediction, and long-term prediction. The current-time prediction requires the shortest time interval between data, which can be used to establish an online real-time forecasting system. The short term interval is usually between an hour and a few hours, often used for optimal control or abnormal detection. The medium-term forecast time interval is one day, which can be used to guide resource planning. The long term interval is between months and years and can be used as a reference for strategy and economic investment.

The autocorrelation coefficients r_k is used to indicate the autocorrelation between the time sequence itself and the data of the lag *k* period [15]:

$$r_k = \frac{\sum_{t=1}^{T-k} (y_t - y')(y_{t+k} - y')}{\sum_{t=1}^{T} (y_t - y')}, \tag{7}$$

where $y_1, y_2 \cdots, y_T$ represent time series data, *T* represents the size of the dataset, y' represents the average of the sequence. The higher the value of r_k, the higher autocorrelation of that data in the period *k*. This means that the data set is in a time dimension and cycles through *k*. Data at the same time in different cycles tend to have the same data characteristics.

The traditional RNN sequence prediction model is based on the changing trend of the learning sequence to predict the value of next moment. According to our experience, the network traffic data has a certain autocorrelation in the 24 hours and 7 days cycles. This means that the data of 24 hours or 7 days at a certain time can be very good at representing the data characteristics of the present moment. However, because the data of 24 hours or 7 days ago does not constitute a continuous sequential relationship with *T* time steps before, the traditional RNN neural network does not apply to the modeling of autocorrelation characteristics.

Using the improved LSTM + deep neural network (DNN) neural network and keeping the original LSTM network unchanged, the predicted value and several values before the autocorrelation cycle make the input of DNN neural network. By training the DNN network, the autocorrelation characteristics and timing characteristics of network data traffic are combined to achieve the accurate prediction of network data traffic. **Fig. 3** shows the Neural network structure of LSTM and DNN .

3 Network Traffic Prediction

In order to verify the accuracy of network traffic prediction model, three real network traffic data sets were selected for experiment. Data set A comes from the network traffic history data of network service provider in 11 European cities. The data was collected from 06:57 am on June 7, 2004 to 11:17 am on June 29, 2004 and collected every 30 seconds. Data set B comes from the web traffic history data from the United Kingdom Education and Research Networking Associations (UKERNA) website for academic research. Data was collected from 9:30 am November 2004 to 11:11 am on January 27, 2005, every five minutes. Data set C comes from the network traffic data collected by the backbone node of China's education network, Beijing University of Posts and Telecommunications, in 2012. Data is collected every minute

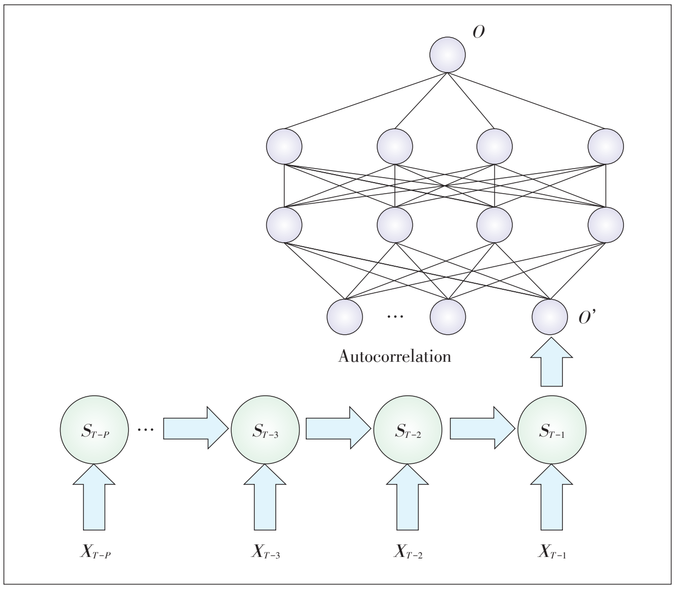
Based on the period and time of the three experimental data sets, this paper only considers the first two prediction types. According to Cortezs selection of time granularity [16], 5 minutes and 1 hour were used as current-time prediction and short-term prediction of time granularity. When we divide the training set and test set, we use the first two thirds of the data set for training, and then one third to test the accuracy of the model.

In order to evaluate the accuracy of the prediction model, the mean absolute percentage error (MAPE) is used as the evaluation indicator of the model [17]:

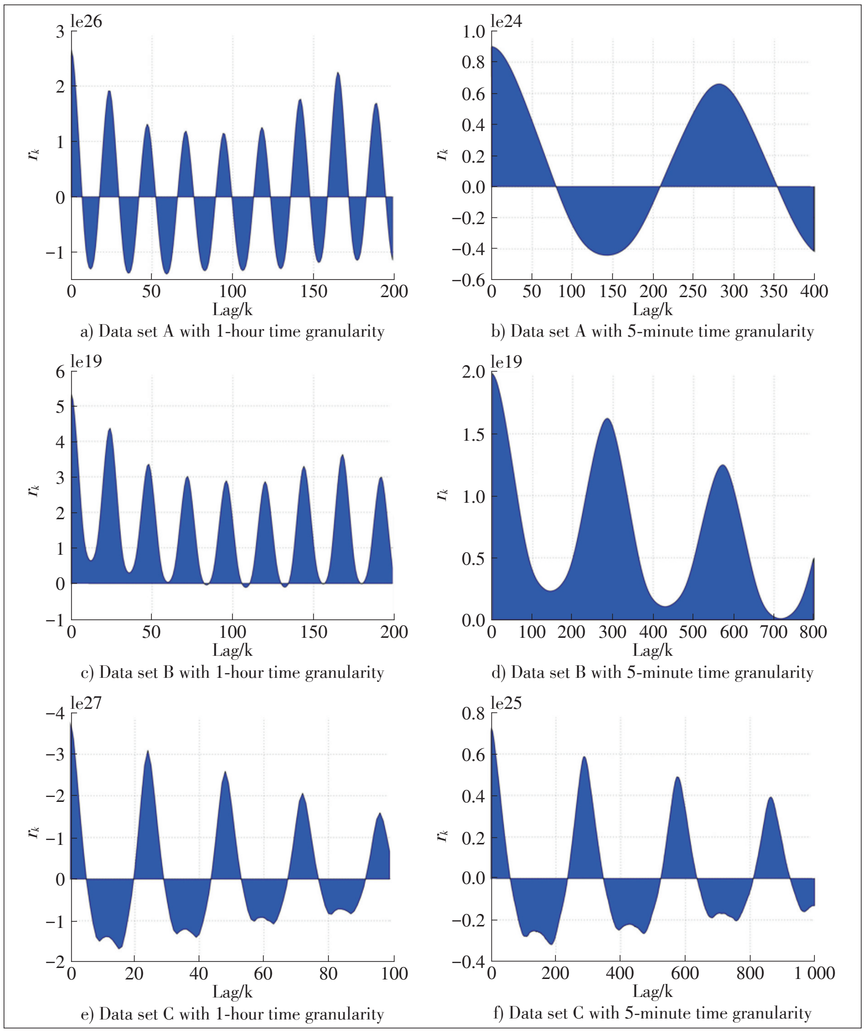
$$MAPE = \frac{1}{N}\sum_{n=1}^{N}\left|\frac{y_n - Y_n}{y_n}\right| \times 100\%. \tag{8}$$

As shown in Eq. (8), the average absolute percentage error is a commonly used index for estimating the accuracy of the model. In Eq. (8), y represents the real value of data, Y_n represents predicted value, and N is the test machine size. The smaller the error, the better the prediction.

Fig. 4 shows the autocorrelation analysis of



▲**Figure 3. Neural network structure of LSTM and DNN.**



a) Data set A with 1-hour time granularity

b) Data set A with 5-minute time granularity

c) Data set B with 1-hour time granularity

d) Data set B with 5-minute time granularity

e) Data set C with 1-hour time granularity

f) Data set C with 5-minute time granularity

▲**Figure 4. Autocorrelation analysis.**

these data sets. For the data of 5minute granularity of time, the degree of autocorrelation is high when $k=288$. And for the data of 1 hour for particle size, $k=24$ and $k=168$ were the two peaks. This indicates that the autocorrelation cycle of network traffic is 24 hours and 7 days, which is correspond with our characteristics of network traffic data.

In this paper, echo state network (ESN) is used as a comparison of LSTM model. The ESN is also a form of RNN, consisting of input layers, hidden layers, and output layers, and has a connection between the hidden layer and the hidden layer to retain information from the previous moments. Unlike RNN, the connection statuses of neurons in hidden layers are random, however, the connection weights are changeless. In the course of training, we only need to train the connection weights of the hidden layer to the output layer. After experiment, the parameters of the ESN model in **Table 1** are finally selected:

The neurons in the LSTM model were treated by Dropout in order to prevent the overfitting. Dropout was first proposed by Hintion [18], and the main idea was to let the neural network randomly make some neurons not work during the training. It is proved by experiment that this can effectively prevent the overfitting and improve the generalization ability of the model. Zaremba [19] improved the Dropout rate so that it could be applied to RNN. In the model of this paper, the probability that each neuron does not work is 10% during the training. **Table 2** shows LSTM neural network parameters.

In the LSTM-DNN model, the parameters of the LSTM neural network are consistent with Table 2. Depending on the time granularity of the data set, the values of different autocorrelation and the calculated value of LSTM feedforward algorithm were selected and entered into DNN. For example, a data set that time granularity is 5 minutes, $\left(xj-287,xj-288,xj-289\right)$ is a good indicator of the data volume characteristics. Therefore, put these data input current-time network traffic prediction model, the result we get is predicted traffic data. In a data set of time granularity 1 hour, the data $\left(xj-23,xj-24,xj-25\right)$ is fed into the current-time network traffic prediction model to predict xj.

The average absolute percentage errors of the three algorithms are shown in **Table 3**.

The relationship between loss function and training frequency in LSTM and LSTM-DNN is shown in **Fig. 5**.

Due to the limitation of data set length, the test set divided

▼Table 1. Echo status network parameters

Parameter	Value
In size	1
Reservoir size	1 000
Leaking rate	0.3
Spectral radius	0.136
Linear regression algorithm	Ridge regression

▼Table 2. LSTM neural network parameters

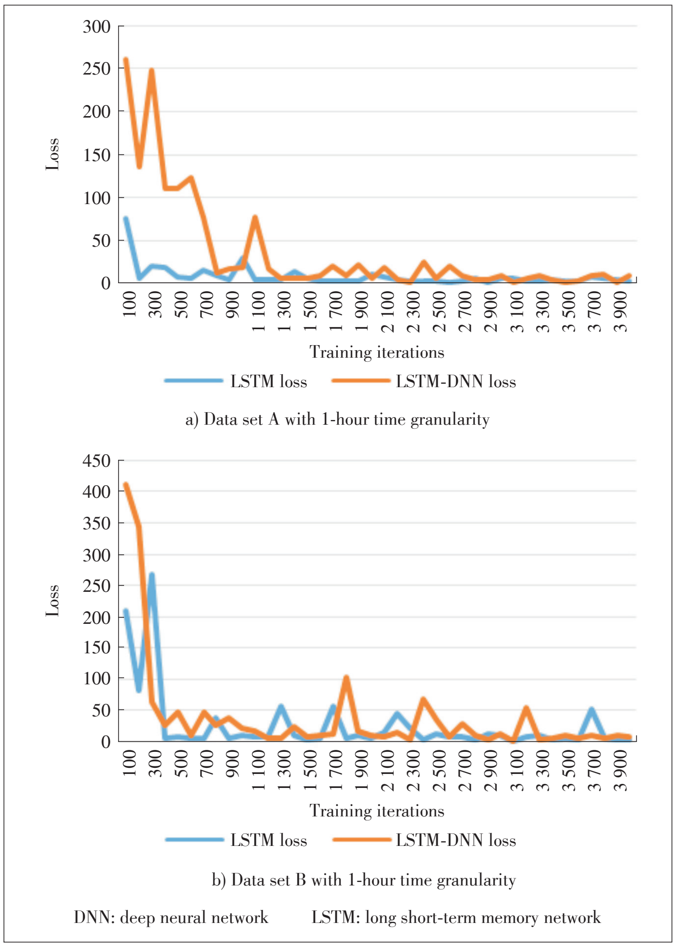
Parameter	Value
Dropout fraction	10%
Time steps	10
RNN units	200
RNN layers	1
Dense units	[10,10]
Batch size	10

RNN: recurrent neural network

▼Table 3. Average absolute percentage error analysis

Date	Time granularity	ESN	LSTM	LSTM and DNN
Date set A	5 min	2.81%	1.41%	1.39%
	1 h	7.63%	4.69%	4.51%
Data set B	5 min	3.50%	3.29%	1.32%
	1 h	5.04%	4.47%	2.80%
Data set C	5 min	4.41%	12.04%	
	1 h	0.59%	14.00%	

DNN: deep neural network ESN: echo state network LSTM: long short-term memory network



a) Data set A with 1-hour time granularity

b) Data set B with 1-hour time granularity

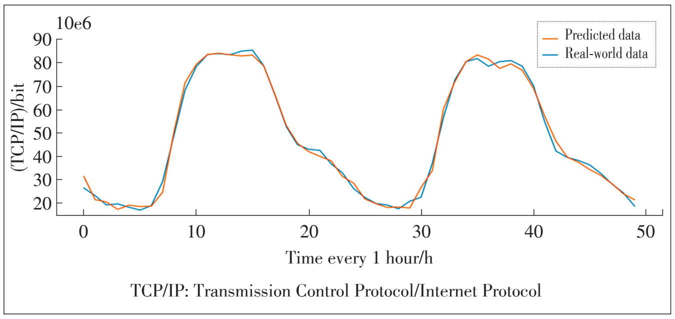
DNN: deep neural network LSTM: long short-term memory network

▲Figure 5. Comparison of performance of loss function under different data sets and training iterations.

by the data set C is not satisfied with the requirements of LSTM-DNN training and prediction. It can be seen from Table 3 that LSTM can effectively predict network traffic and have good performance at different time granularity of two data sets. According to the optimized LSTM-DNN model, the autocorrelation of data is added; when the time granularity was 5 m, the accuracy was slightly increased, but the accuracy of the prediction was significantly improved when the time granularity was 1 h. This indicates that the autocorrelation makes up for the insufficient data training that LSTM requires. The LSTM-DNN model has some advantages over traditional LSTM in the case of coarse time granularity or less data volume. We can see from Fig. 5 that as the complexity of neural network layer increases and the number of layers increases, LSTM-DNN is relatively slow and not easy to converge compared to LSTM in the training effect. **Figs. 6–9** show the prediction results of LSTM-DNN.

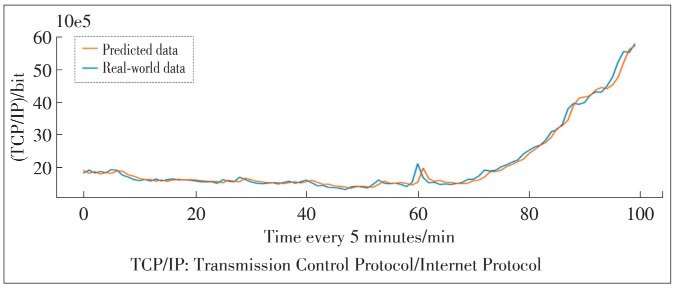
4 Conclusions

Aiming at implementing the network traffic with autocorrelation, this paper proposes a model of neural network which combines LSTM with DNN. By using several real data sets our experiments show that LSTM can be used as a timing sequence forecast model well. It can provide better performance than the other traditional model. Making use of the autocorrelation features, the neural network of LSTM and DNN has certain advantages in the accuracy of the large granularity data sets, but there is a requirement for the data size. High accuracy network traffic prediction provides some support to deal with possible network congestion, abnormal attack, etc. LSTM is just one of the many variations of RNN. Some of other RNN models are also popular and well represented in many problems, such as Gated Recurrent Unit (GRU). The next step is to explore different RNN structures and combine the characteristics of network traffic data to explore the possibility of further improving the prediction accuracy.



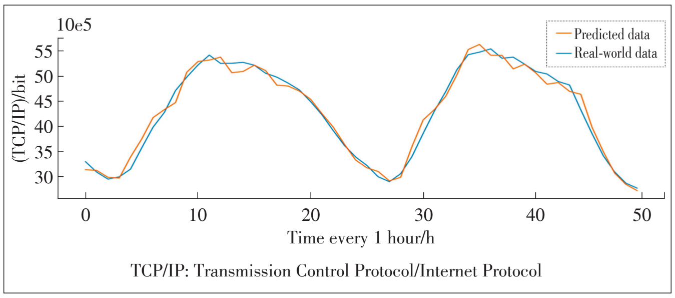
TCP/IP: Transmission Control Protocol/Internet Protocol

▲Figure 6. Prediction effect of the data set A with 1-hour time granularity.



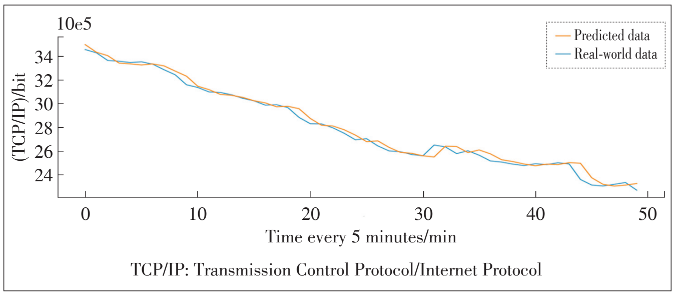
TCP/IP: Transmission Control Protocol/Internet Protocol

▲Figure 7. Prediction effect of the data set A with 5-minute time granularity.



TCP/IP: Transmission Control Protocol/Internet Protocol

▲Figure 8. Prediction effect of the data set B with 1-hour time granularity.



TCP/IP: Transmission Control Protocol/Internet Protocol

▲Figure 9. Prediction effect of the data set B with 5-minute time granularity.

References

[1] BABIARZ R, BEDO J. Internet Traffic Midterm Forecasting: a Pragmatic Approach Using Statistical Analysis Tools [M]//BOAVIDA F, PLAGEMANN T, STILLER B, et al. (eds). Networking 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems. Lecture Notes in Computer Science: vol 3976. Berlin, Heidelberg, Germany: Springer, 2006: 111–121

[2] ALARCON-AQUINO V, BARRIA J A. Multiresolution FIR Neural-Network-Based Learning Algorithm Applied to Network Traffic Prediction [J]. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 2006, 36(2): 208–220. DOI: 10.1109/tsmcc.2004.843217

[3] KRISHNAMURTHY B, SEN S, ZHANG Y, et al. Sketch-Based Change Detection: Methods, Evaluation, and Applications [C]//3rd ACM SIGCOMM Conference on Internet Measurement. Miami Beach, USA, 2003: 234– 247. DOI: 10.1145/948205.948236

[4] YANG M M, ZHU T Q, ZHOU W L, et al. Attacks and Countermeasures in Social Network Data Publishing [J]. ZTE Communications, 2016, 14(S0): 2– 9. DOI: 10.3969/j.issn.1673-5188.2016.S0.001

[5] CHRISTIAN J, MOHAMED B. A Software-Defined Approach to IoT Networking [J]. ZTE Communications, 2016, 14(1): 61− 68. DOI: 10.3969/j.issn.1673 - 5188.2016.01.009

[6] GROSCHWITZ N K, POLYZOS G C. A Time Series Model of Long-Term NSFNET Backbone Traffic [C]//International Conference on Communications. New Orleans, USA, 1994: 1400−1404. DOI: 10.1109/ICC.1994.368876

[7] BASU S, MUKHERJEE A, KLIVANSKY S. Time Series Models for Internet Traffic [C]//Conference on Computer Communications. San Francisco, USA, 1996: 611−620. DOI: 10.1109/INFCOM.1996.493355

[8] SHU Y T, WANG L, ZHANG L F, etc. Internet Network Business Forecast Based on FARIMA Model [J]. Chinese Journal of Computers, 2001, 24(1): 46−54

[9] LIU X W, FANG X M, QIN Z H, et al. A Short-Term Forecasting Algorithm for Network Traffic Based on Chaos Theory and SVM [J]. Journal of Network and Systems Management, 2011, 19(4): 427−447. DOI: 10.1007/s10922-010-9188-3

[10] WANG H F, HU D J. Comparison of SVM and LS-SVM for Regression [C]//International Conference on Neural Networks and Brain. Beijing, China, 2005: 279−283. DOI: 10.1109/ICNNB.2005.1614615

[11] BENGIO Y, SIMARD P, FRASCONI P. Learning Long-Term Dependencies with Gradient Descent is Difficult [J]. IEEE Transactions on Neural Networks, 1994, 5(2): 157−166. DOI: 10.1109/72.279181

[12] HOCHREITER S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions [J]. International Journal of Uncertainty, Fuzziness and Knowledge - Based Systems, 1998, 6(2): 107− 116. DOI: 10.1142/s0218488598000094

[13] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory [J]. Neural Computation, 1997, 9(8): 1735−1780. DOI: 10.1162/neco.1997.9.8.1735

[14] COLAH. Understanding LSTM Networks [EB/OL]. (2015-08-27). http://colah.github.io/posts/2015-08-Understanding-LSTMs

[15] DING X, CANU S, DENOEUX T. Neural Network Based Models for Forecasting [C]//Proc. ADT'95. New York, USA: Wiley and Sons, 1995: 243−252

[16] CORTEZ P, RIO M, ROCHA M, et al. Multi-Scale Internet Traffic Forecasting Using Neural Networks and Time Series Methods [J]. Expert Systems, 2010. DOI: 10.1111/j.1468-0394.2010.00568.x

[17] FARIA A E Jr. Review of Forecasting: Methods and Applications (by MAKRIDAKIS S, WHEELWRIGHT S C, HYNDMAN R J, Third Edition) [J]. International Journal of Forecasting, 2002, 18(1): 158−59. DOI: 10.1016/s0169-2070 (01)00130-3

[18] HINTON G E, SRIVASTAVA N, KRIZHEVSKY A, et al. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors [EB/OL]. (2012-07 -03). https://arxiv.org/abs/1207.0580

[19] ZAREMBA W, SUTSKEVER I, VINYALS O. Recurrent Neural Network Regularization [EB/OL]. (2014-09-08). https://arxiv.org/abs/1409.2329

Biographies

WANG Shihao is a postgraduate student of Nanjing University of Science and Technology, China. He received the bachelor's degree from Nanjing Institute of Technology, China in 2017. His research interests include network traffic prediction and network intrusion detection.

ZHUO Qinzheng was a postgraduate student of Nanjing University of Science and Technology, China. He received the bachelor's and master's degree from Nanjing University of Information Science and Technology, China in 2015 and 2018. His research interests include network traffic prediction, data mining, and deep learning.

YAN Han received the Ph.D. degree from Nanjing University of Information Science and Technology in 2000. He is an associate professor with Nanjing University of Science and Technology, China. His research interests include software modeling, web computation, information security, and agile software development, and his current focus is on computing system management. He received the second class prizes for national defense science and technology. More than 40 academic papers have been published.

LI Qianmu (liqianmu@126.com) received the B.Sc. and Ph.D. degrees from Nanjing University of Science and Technology, China in 2001 and 2005, respectively. He is a professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He is also a Ph.D. supervisor and the director of Department of Software Engineering. His research interests include information security, computing system management, and data mining. He received the China Network and Information Security Outstanding Talent Award and multiple Education Ministry Science and Technology Awards. More than 120 high-level papers have been published and more than 150 patents have been applied and authorized.

QI Yong is a Ph.D. supervisor and professor with Nanjing University of Science and Technology, China. He is also the director of Jiangsu Intelligent Transportation Information Perception and Data Analysis Engineering Laboratory. His main research interests include data mining and intelligent transportation systems. He received more than 10 scientific awards, such as China University Research Cooperation Innovation Award, the Ministry of Educations Scientific and Technological Progress Award, and the Jiangsu Science and Technology Award. More than 100 academic papers have been published, and more than 60 patents for inventions and applications have been granted.