



Self-similarity Analysis and Application of Network Traffic

Yan Xu¹, Qianmu Li^{1,2}(✉), and Shunmei Meng¹

¹ School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China
{xuyan, qianmu, mengshuanmei}@njjust.edu.cn

² Intelligent Manufacturing Department, Wuyi University, Wuyi, China

Abstract. Network traffic prediction is not only an academic problem, but also a concern of industry and network performance department. Efficient prediction of network traffic is helpful for protocol design, traffic scheduling, detection of network attacks, etc. In this paper, we propose a network traffic prediction method based on the Echo State Network. In the first place we prove that the network traffic data are self-similar by means of the calculation of Hurst exponent of each traffic time series, which indicates that we can predict network traffic utilizing nonlinear time series models. Then Echo State Network is applied for network traffic forecasting. Furthermore, to avoid the weak-conditioned problem, grid search algorithm is used to optimize the reservoir parameters and coefficients. The dataset we perform experiments on are large-scale network traffic data at different time scale. They come from three provinces and are provided by ZTE Corporation. The result shows that our approach can predict network traffic efficiently, which is also a verification of the self-similarity analysis.

Keywords: Network traffic · Self-similarity · Echo State Network

1 Introduction

Traffic prediction is the foundation of network performance analysis. It provides essential evidence for network design and planning. Designing an efficient and accurate model for network traffic prediction can reduce network congestion frequency and improve network communication quality. Either short-term or long-term prediction is beneficial to network control and resource adjustment. By analyzing and forecasting historical traffic data and adjusting the allocation of network resources accordingly, operators can be aware of the future situation of the network in advance. It has a profound impact on the development of key technologies such as network planning, resource allocation and network security.

There are a great number of prediction models for network traffic data and they can be classified into statistical analysis models and machine learning methods [1]. Autoregressive integrated moving average (ARIMA) is a typical statistical analysis model [2, 3], which is the combination of autoregressive and moving average models. Since ARIMA is a linear time series model, some improvements are made to capture

the non-linearity of network traffic. Zhou et al. [2] combined ARIMA model with GARCH, non-linear model. Shu et al. [3] proposed a seasonal ARIMA model to explore the cyclical patterns of traffic data. With the rapid growth of network and complexity of traffic data, more and more researchers have placed emphasis on machine learning models, especially neural networks [1]. A hybrid ARIMA-ANN model was proposed in [4] to forecast time series data. Three methods: ARIMA, Holt-Winters and a novel neural network ensemble (NNE) approach, were performed on multi-scale internet traffic forecasting and the results showed the advantage of NNE [5]. Multi-layer Perception (MLP) is widely used for network traffic prediction [6–8]. [9] performed SVR, the regression variant of SVM, on heterogeneous Internet traffic collected at the POP of an ISP network. Nie et al. [10] decomposed the network traffic into low-pass and high-pass component, where the low-pass component describe the long-range dependence and the high-pass component gusty and irregular fluctuations of network traffic. As for prediction, a deep belief network and a Gaussian model were utilized for respectively. Poupart et al. [11] aimed at predicting the size of flow in order to detect elephant flow (very large flows). Three machine learning techniques: neural networks, Gaussian Process Regression and Online Bayesian Moment Matching (oBMM), were combined with routing, where GPR achieved the best improvements for elephant flow detection.

However, the most of the models focus on the non-linearity of traffic data to improve the accuracy but ignore the importance of self-similarity. Based on the large-scale network traffic dataset provided by ZTE Corporation, this paper analyzes the characteristics of the dataset, and performs pre-processing on the dataset to obtain suitable traffic data of each node at different time scales. Then by plotting the trend of traffic over time and calculating the Hurst exponent value, it proved that the traffic data of the three provinces provided by ZTE has self-similarity, suddenness and periodicity. Finally we can predict the data using a nonlinear time series. Because of the nonlinear characteristic of network traffic prediction, we utilize Echo State Network (ESN) to learn the output connection weight matrix. The ridge regression learning algorithm is applied instead of traditional linear regression algorithm so that weak-condition can be avoided. Meanwhile the grid search algorithm is used to optimize the reservoir parameters and regularization coefficients.

The reminding portion of the paper is organized as follows. Section 2 clarifies the definition of self-similarity and the estimation of Hurst exponent. Section 3 introduces the structure of Echo State Network, along with parameters to be estimated and the training process. Section 4 focuses on the experiments based on network traffic data from three provinces. Section 5 is the conclusion.

2 Self-similarity

Self-similarity [12–15] means that local structure is partly consistent with the overall structure. A self-similar process is a stochastic process which is statistically constant. In this regard, the concept of fractal to the random process is introduced. Network traffic has long-range dependence (LRD) as opposed to processes with short-range dependence like Poisson process. From a physical point of view, LRD [16, 17] is a

phenomenon, i.e. the sustainability and suddenness of a self-similar process exist on all time scales, also known as multi-scale behavioral features [18, 19].

Definition 2.1: $\forall \lambda > 0$, we say that a stochastic process $\{X(t), t \geq 0\}$ is self-similar if $X(t) \stackrel{d}{=} \lambda^{-H} X(\lambda t)$, where $H \in (0.5, 1)$ refers to Hurst exponent or self-similarity parameter. $\stackrel{d}{=}$ means that the equation is correct in finite dimensions.

According to Definition 2.1, $\{X(t), t \geq 0\}$ with self-similarity has following properties:

Property 2.1: Time series $\{X(t), t \geq 0\}$ has time-scale invariance, or when $\{X(\lambda t), t \geq 0\}$ is normalized by λ^{-H} , they have the same structure.

Property 2.2: $E\{X(t)\} = 0$

Property 2.3: $E\{|X(t)^q|\} = E\{|X(1)^q|\}t^{qH}$

Table 1. Comparison of commonly used estimation methods for Hurst parameters

	Self-similarity judgement	Graphical	Online	Complexity	Others
Variance-time plot	Yes	Yes	No	$O(n)$	Lots of data required in advance
R/S plot	Yes	Yes	No	$O(n^2)$	Independent of edge distribution
Periodogram	Yes	Yes	No	$O(n \log n)$	Set suitable cutoff frequency at first
Whittle estimator	Yes	No	No	$O(n^2)$	A quantitative method with high complexity
Wavelet analysis	Yes	No	No	$O(n \log n)$	Accurate estimation

The Hurst exponent [20, 21], denoted by H , is an important parameter to characterize self-similarity. A self-similar process will degenerate towards a Poisson process if $H = 0.5$. A value of H in the interval $(0.5, 1)$ refers to positive autocorrelation, i.e. the random variety series is self-similar and the degree grows with the increase of H . A value of H in the interval $(0, 0.5)$ indicates negative autocorrelation, i.e. the series is not self-similar. There are five commonly used and robust estimation methods and we compare them in detail as shown in Table 1. The first three methods are graphical. The estimation of Hurst exponent is the slope of the line, which is plotted by fitting statistical sample points. Among the three methods, the variance-time plot method is less robust and the periodogram method requires determination of the appropriate cutoff frequency. The whittle estimator can only estimate short-range dependent data, rather than the long-range dependent data and it has high complexity. Wavelet analysis can estimate the Hurst parameter more accurately while the confidence interval of the parameter cannot be obtained and the calculation is more complicated. In summary, we select R/S plot to estimate Hurst exponent.

R/S plot is widely used to estimate Hurst exponent. At the beginning, we need to divide a time series of length N into series of length N , $N/2$, $N/4$, etc. Then for a time series $\{X_1, X_2, \dots, X_n\}$, calculate its rescaled range R/S:

1. Calculate the mean: $m = \frac{1}{n} \sum_{i=1}^n X_i$, where n is the length of the time series, which is the network traffic;
2. Generate a deviation series $\{Y_1, Y_2, \dots, Y_n\}$: $Y_t = X_t - m, t = 1, 2, \dots, n$
3. Calculate the range R : $R = \max(Y_1, Y_2, \dots, Y_n) - \min(Y_1, Y_2, \dots, Y_n)$
4. Calculate the standard deviation S : $S = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - m)^2}$
5. Get the rescaled range R/S.

3 Echo State Network

Echo State Network (ESN) [22–24] is a new type of recursive neural network. An ESN is made up of an input layer, a hidden layer (dynamic reservoir) and an output layer. It can also remember data by adjusting weights inside the network. The dynamic reservoir contains a large number of sparsely connected neurons, which keep the state of the system and has a capacity of short-term memory.

3.1 Structure of ESN

Echo State Network is a new type of three-layer recurrent neural network. As shown in Fig. 1, an ESN consists of three parts: input layer, hidden layer (reservoir) and output layer. The number of neurons is K , N and L respectively. $W^{in} \in \mathbb{R}^{N \times M}$ and $W^{out} \in \mathbb{R}^{L \times (K+N+L)}$ are the input and output weight matrices respectively as shown in Fig. 1. $W \in \mathbb{R}^{N \times N}$ represents concatenation of neurons inside the reservoir. $W^{back} \in \mathbb{R}^{N \times L}$ is a feedback matrix from the output layer at one moment to the reservoir at the next.

The hidden layer is also known as a dynamic reservoir since it is made up of many dynamic neurons which are connected. The reservoir is the core structure of ESN. Like a human brain, it consists of many neurons. These neurons are connected to constitute a large-scale and complex network so that they can transfer information inside. It can constantly learn and deal with stimuli from the outside world. Considering the condition that information cannot be transferred from one neuron to another, we have weights among neurons in the range $[-1, 1]$. The weight will be 0 if there is no connection between the two neurons, otherwise it will be a non-zero value in the interval $[-1, 1]$. A positive weight results in promotion while a negative weight causes neutralization.

Reservoir connection matrix is sparse. To guarantee the reservoir's echo effects, the spectral radius of W should be less than 1. Echo effects refer to the reservoir neuron's short-memory of the states of input traffic data.

3.2 Key Parameters of Reservoir

The reservoir, a recursive structure of randomly generated, large-scale and sparse connections, is the core structure of ESN. It is necessary to set appropriate values for reservoir's key parameters to achieve good performance. Reservoir's key parameters include: spectral radius (SR), size of reservoir (N), input scale (IS) and sparse degree (SD).

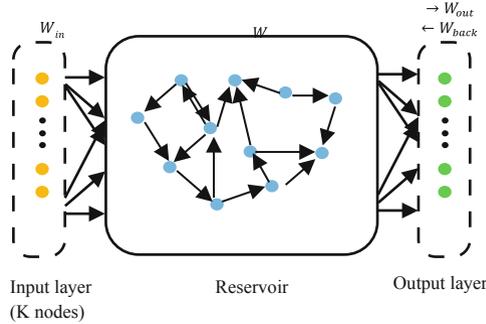


Fig. 1. Structure of Echo State Network

- Spectral radius (SR). The feature value, with the largest absolute value, of the connection weight matrix, denoted by λ_{max} . The state of reservoir neurons can stay decaying to keep the network stable if $\lambda_{max} < 1$.
- Size of reservoir (N). The number of neurons in the reservoir. The size of reservoir is closely related to the number of samples to be predicted, which has a great impact on the prediction. There are two ways to set the value of N. The first method is based on the complexity of the problem, which is gradually increasing the value of N; the second method is to select a value in the range $[T/10, T/2]$, where T refers to the size of the training set.
- Input scale (IS). A scale factor that needs to be multiplied by the input signal before it connects neurons inside reservoir, i.e. a certain scaling of the input signal. With a nonlinear time series, the IS is larger.
- Sparse degree (SD). The connection among neurons inside reservoir. Not all neurons have connections between them. SD represents the number of connected neurons out of N , i.e. $SD = \frac{n}{N}$, where n is the number of connected neurons. With a value $SD \in [5\%, 10\%]$, the reservoir can maintain its dynamic characteristics.

3.3 Training

W^{in} , W and W^{back} are randomly initialized at the beginning and they are unchanged during the training and predicting process, which means that we only need to train the weight matrix W^{out} . Linear regression algorithm is applied to calculate W^{out} . State sequences of input, output and reservoir at moment n are defined as follows:

Definition 3.1: Suppose that the ESN has K input nodes and L output nodes, then the input and output vectors are $u(n) = (u_1(n), \dots, u_k(n))^T$ and $y(n) = (y_1(n), \dots, y_L(n))^T$ respectively. $W^{in} = (w_{ij}^{in})_{N \times K}$ is the input connection weight matrix, $W = (w_{ij})_{N \times N}$ is reservoir weight matrix and $W^{out} = (w_{ij}^{out})_{L \times (K+N+L)}$ represents the output connection weight matrix.

The training process of ESN can be divided into two stages:

Step1. Data Sampling. Initiate the network state at the very beginning. Generally we set the initial state as 0, i.e. $x(0) = 0$.

- Training samples $\{u(t), t = 1, 2, \dots, P\}$ are added to reservoir by the means of input connection weight matrix W^{in} .
- According to Eqs. 1 and 2, calculate states of reservoir and corresponding output $y(n)$. Equation 1 is known as an update equation of reservoir neurons. Equation 2 is an output state function, where n is the number of samples of input network. f and f_{out} are the neuron stimulation functions of the dynamic reservoir and output layer respectively. In general, $f = \tanh$, $f_{out} = \text{Sigmoid}$. You can choose other functions according to specific situation.

$$x(n+1) = f(W^{in}u(n+1) + Wx(n) + W^{back}y(n)) \quad (1)$$

$$y(n+1) = f_{out}(W^{out}(u(n+1), x(n+1))) \quad (2)$$

Step 2. Computing Output Weights. Compute output weights W^{out} based on reservoir state matrix M and corresponding output matrix T collected at step 1, where $M \in \mathbb{R}^{m \times N}$, $T \in \mathbb{R}^{m \times 1}$. In general, the state of reservoir is not stable at the initial phase of step 1. In order to eliminate the influence of the initial transient of the network, data sampling will be simplified, which is removing some steps of sampling, and the value of m is the final number of sampling. Using linear regression method, weights are calculated according to $W^{out} = (M^{-1} \times T)^T$, where M^{-1} is the generalized inverse matrix of M .

4 Experiments and Analysis

4.1 Dataset

ZTE provided us with network traffic data of three provinces: A, B, and C. Traffic data of different time scales: second, hour, day, week, month, are stored in their own csv files. Taking the minute-scale traffic data from March 1st to March 31st for example, bytes of traffic data at several nodes at 100 fixed time points everyday are recorded and stored into 31 files respectively in a common format. Due to the space limitation, we will talk about minute-scale in detail.

After sampling, the size of traffic data in province A is 9.22 GB, along with 19.5 GB in province B and 50.7 GB in province C, which are apparently too large for the self-similarity analysis. It is necessary to find out the rules and apply corresponding pre-processing method to get suitable data. As described above, all the files have a common format with multiple fields, some of which are unnecessary. We need three fields: *noid* for Node number, *time* and *kb* for Bytes count and we can extract these data from the whole dataset. In this way we get a dataset that is much smaller.

There are abnormal values in the dataset after filtering. The abnormal values of field *kb* are -99999999 and 450000 , which can also be considered as missing values. If the ratio of abnormal values at one node exceeds 15%, then delete the node. Otherwise, replace the abnormal values with the mean. Actually the ratio is either 0 or 100%.

Besides, the value of field *kb* is relatively large, so that normalization is required for speed. The deviation standard method is applied to scale data into the range $[-1.0, 1.0]$.

4.2 Self-similarity Experiments

We shall observe the flow of traffic data by time in the first place. Considering the minute-scale traffic data, Fig. 2 demonstrates the traffic series of three provinces intuitively. It is easy to tell that all of them have periodicity. The periods all approximate to 96. Besides all of them has a big gap between the maximum value and the minimum value, which means that they are all unstable time series that have strong bursts. Since the local trends are some kind consistent to the overall trend, we can roughly tell that the minute-scale traffic data of each province have self-similarity.

In order to more rigorously judge whether the traffic sequence has self-similarity, the Hurst exponent estimation is applied. We adopt R/S plot to estimate Hurst exponent. Hurst values at different time scales of province A, B and C are shown in Tables 2, 3 and 4 respectively. We omit time scales week and month since the number of traffic data is smaller than 100, which is not persuasive. As what we said above, a time series is self-similar if its Hurst exponent is a value in range 0.5–1. It is proved that network traffic data of three provinces at different time scales is self-similar.

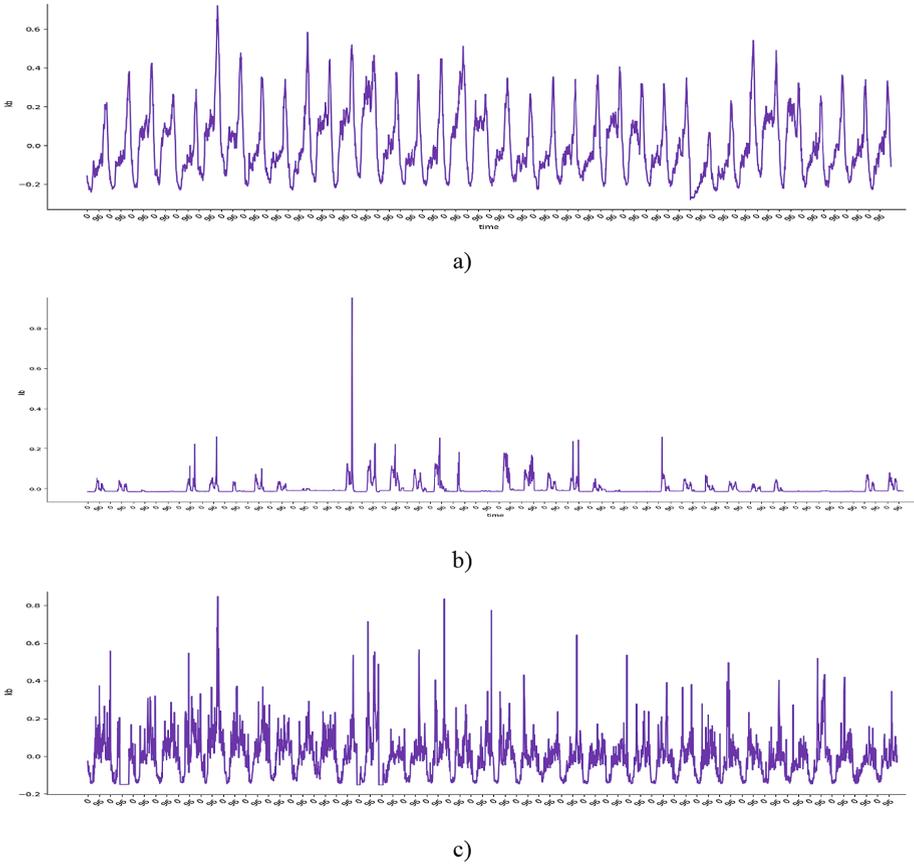


Fig. 2. Minute-scale traffic series: (a), (b), and (c) correspond to Province A, B and C respectively.

Table 2. Hurst values in Province A

Time scale	No. of traffic data	Value of Hurst
Minute	3456	0.8274
Hour	840	0.8335
Day	142	0.8430

Table 3. Hurst values in Province B

Time scale	No. of traffic data	Value of Hurst
Minute	3456	0.7189
Hour	840	0.737
Day	632	0.7511

Table 4. Hurst values in Province C

Time scale	No. of traffic data	Value of Hurst
Minute	3456	0.5755
Hour	816	0.6090
Day	784	0.7016

4.3 Network Traffic Prediction and Evaluation

Evaluation. Ahead of displaying the results of prediction, the evaluation indicators that we will use in this paper are introduced first. Considering test dataset $\{(x_1, y_1), \dots, (x_m, y_m)\}$, the prediction for vector $(x_1, \dots, x_m)^T$ is $(\hat{y}_1, \dots, \hat{y}_m)^T$. In this paper, we evaluate the performance of a prediction model with following evaluation indicators.

Mean Absolute Error (MAE). MAE is also known as L1-norm loss. It offers an intuitive comparison among real values and prediction ones. The bigger the value of MAE is, the worse this prediction model perform. It can be given by:

$$\frac{1}{m} \sum_{i=1}^m |(y_i - \hat{y}_i)|$$

Root Mean Square Error (RMSE). RMSE is the most commonly used performance metric for regression tasks. It is used to measure the deviation between observed and true values. RMSE is sensitive to abnormally large or small errors in predicted values, which makes it well reflective of prediction accuracy. RMSE is given by:

$$\sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}$$

R-square. R-square is used to describe how well the regression line fits the observations. It reflects the proportion of predictable dependent variables from the unpredictable ones. R-square can be used to describe how well the regression line fit the observations. A larger value of R^2 indicates a better performance of the prediction model. It is given by:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{\sum_{i=1}^m (y_i, \bar{y})^2} = 1 - \frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2 / m}{\sum_{i=1}^m (y_i, \bar{y})^2 / m} = 1 - \frac{MSE(\hat{y}, y)}{Var(y)}$$

Experiment Result. We predict network traffic data of three provinces with GRID-ESN: finding parameters of ESN using the grid search algorithm, and make comparisons with Elman, SVR and ESN. The key parameters to be set includes: SD, N, SR and regularization factor χ . We set $N = 1000$. SR, SD and χ are determined by grid search algorithm. Their searching ranges are $[0.01, 1]$, $[0.01, 0.1]$ and $[10^{-6}, 10^{-2}]$ respectively.

Prediction of Network Traffic in Province A. Parameter estimation of GRID-ESN: $SR = 0.948$, $SD = 0.014$ and $\chi = 0.01$. The prediction comparison is shown in Fig. 3 and Table 5. From Fig. 3, we can find that ESN can fit the traffic data better than SVR and Elman, especially at the bursts. Besides, the average evaluation values of 30 experiments are shown in Table 5, where GRID-ESN get the smallest RMSE and MAE, and the biggest value of R-square.

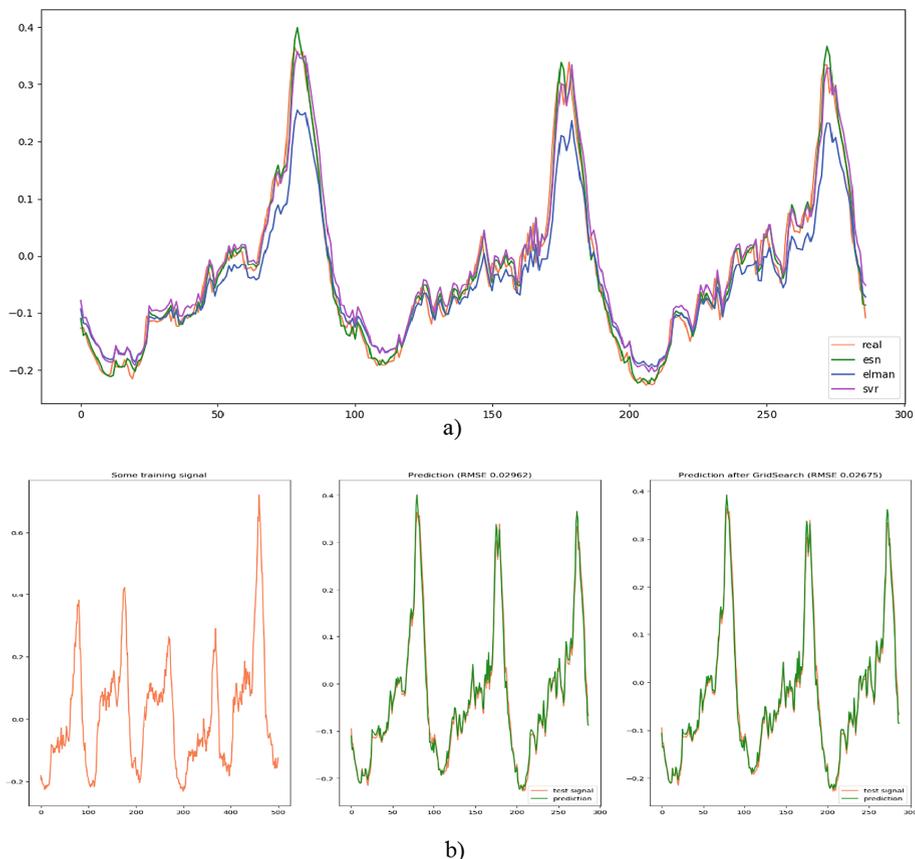


Fig. 3. Prediction comparison for traffic data in Province A: (a) is the prediction comparison among SVR, Elman and ESN; (b) is the prediction comparison between ESN and GRID-ESN

Table 5. Province A: RMSE/MAE/R-square value for ESN, Elman, SVR and GRID-ESN

	RMSE	MAE	R-square
ESN	0.03095	0.02181	0.95747
Elman	0.04237	0.03426	0.92041
SVR	0.03224	0.02643	0.95385
GRID-ESN	0.02673	0.02027	0.96827

Prediction of Network Traffic in Province B. Parameter estimation of GRID-ESN: $SR = 0.864$, $SD = 0.045$ and $\chi = 0.01$. It is not sufficient to make a decision based on the regular traffic data, so that we select traffic data on holidays, which contain many bursts. The prediction comparison is shown in Fig. 4 and Table 6. As shown in Fig. 4, ESN is still better than SVR and Elman. Compare GRID-ESN with ESN in Fig. 4(b), GRID-ESN performs better, especially at the error at burst nodes. The average evaluation values of 30 experiments are shown in Table 6, where GRID-ESN get the smallest RMSE and MAE, and the biggest value of R-square.

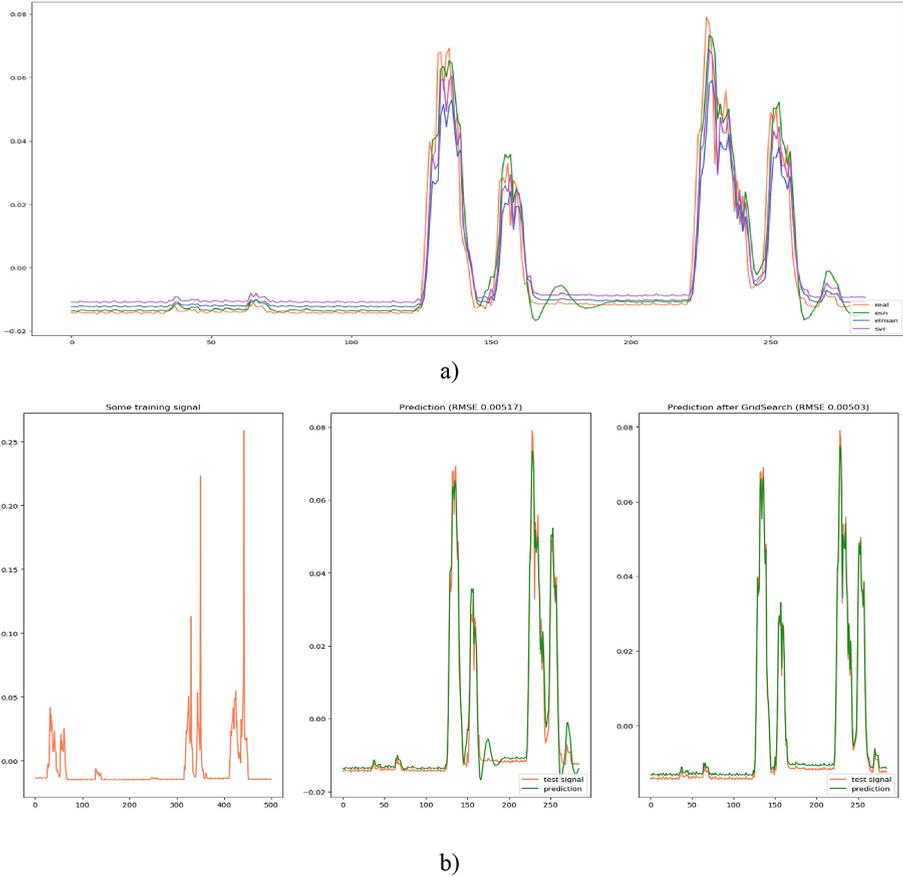


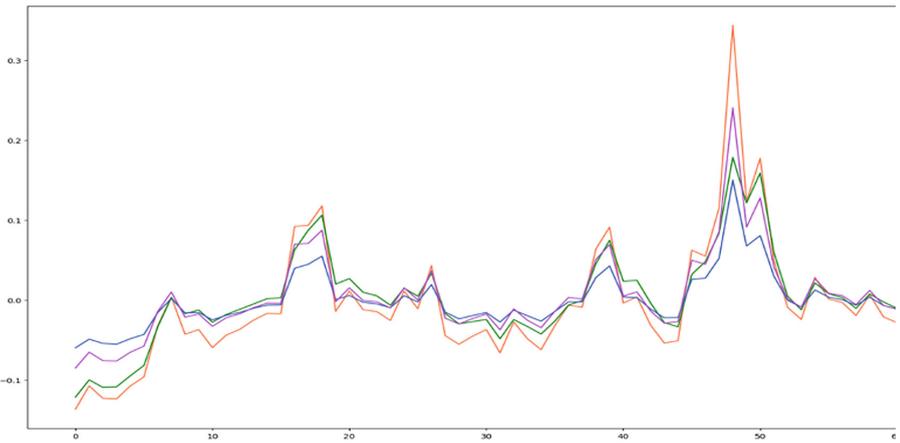
Fig. 4. Prediction comparison for traffic data in Province B: (a) is the prediction comparison among SVR, Elman and ESN; (b) is the prediction comparison between ESN and GRID-ESN

Prediction of Network Traffic in Province C. Parameter estimation of GRID-ESN: $SR = 0.948$, $SD = 0.043$ and $\chi = 0.01$. We select a piece of data that is more unstable.

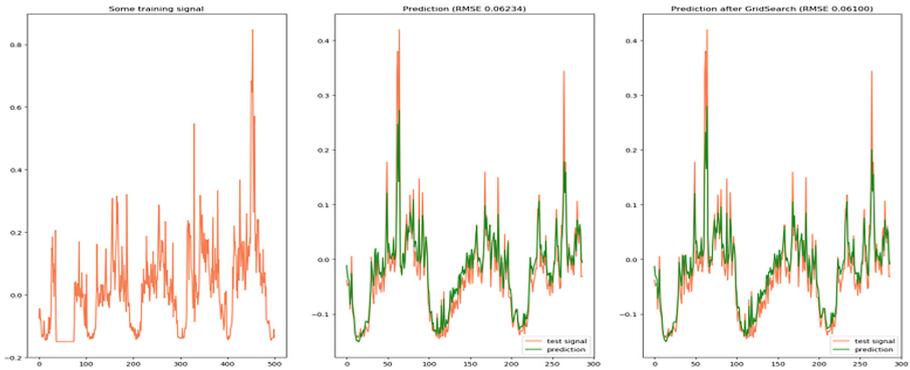
Table 6. Province B: RMSE/MAE/R-square value for ESN, Elman, SVR and GRID-ESN

	RMSE	MAE	R-square
ESN	0.00513	0.00267	0.90525
Elman	0.00676	0.00546	0.84214
SVR	0.00573	0.00425	0.88679
GRID-ESN	0.00503	0.00255	0.91251

The prediction comparison is shown in Fig. 5 and Table 7. From Fig. 5, we can find that ESN can fit the traffic data better than SVR and Elman. The evaluation of the four models including GRID-ESN is shown in Table 7, where GRID-ESN get the smallest RMSE and MAE, and the biggest value of R-square.



a)



b)

Fig. 5. Prediction comparison for traffic data in Province C: (a) is the prediction comparison among SVR, Elman and ESN; (b) is the prediction comparison between ESN and GRID-ESN

Table 7. Province C: RMSE/MAE/R-square value for ESN, Elman, SVR and GRID-ESN

	RMSE	MAE	R-square
ESN	0.06242	0.03933	0.63412
Elman	0.07233	0.05233	0.50864
SVR	0.06501	0.04610	0.60311
GRID-ESN	0.061	0.03866	0.65056

5 Conclusion

We observe and analyze the characteristics of network traffic dataset, find the law of data storage, and then perform a series of processing on data set, such as specification, integration, transformation and cleaning to obtain the traffic data of each node at different time scales. By plotting the traffic data graph, we can easily find the suddenness and periodicity of the traffic data. By calculating the Hurst exponent of the node traffic at different time scales of the dataset, it is proved that the traffic data of the three provinces provided by ZTE are self-similar, which indicated that the nonlinear characteristics of the network traffic time series can be predicted by a nonlinear time series model. In this paper we propose a traffic prediction method based on Echo State Network. The ridge regression learning algorithm is applied instead of traditional linear regression algorithm so that ill-condition can be avoided. Meanwhile the grid search algorithm is used to optimize the reservoir parameters and regularization coefficients. We compare GRID-ESN with ESN, SVR and Elman, and evaluate the prediction performance with four indicators: RMSE, MAE and R-Square, which indicates that our approach can predict traffic data with better performance.

Acknowledgement. This paper is supported by Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX18_0439).

References

1. Boutaba, R., Salahuddin, M.A., Limam, N., et al.: A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *J. Internet Serv. Appl.* **9**(1), 16 (2018)
2. Zhou, B., He, D., Sun, Z.: Traffic Modeling and Prediction using ARIMA/GARCH Model. In: Nejat Ince, A., Topuz, E. (eds.) *Modeling and Simulation Tools for Emerging Telecommunication Networks*, pp. 101–121. Springer, Boston, MA (2006). https://doi.org/10.1007/0-387-34167-6_5
3. Shu, Y., Yu, M., Yang, O., et al.: Wireless traffic modeling and prediction using seasonal ARIMA models. *IEICE Trans. Commun.* **88**(10), 3992–3999 (2005)
4. Babu, C.N., Reddy, B.E.: A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data. *Appl. Soft Comput.* **23**, 27–38 (2014)
5. Cortez, P., Rio, M., Rocha, M., et al.: Multi-scale Internet traffic forecasting using neural networks and time series methods. *Expert. Syst.* **29**(2), 143–155 (2012)

6. Eswaradass, A., Sun, X.H., Wu, M.: Network bandwidth predictor (NBP): a system for online network performance forecasting. In: Proceedings of 6th IEEE International Symposium on Cluster Computing and the Grid (CCGRID), p. 4–pp. IEEE (2006)
7. Chabaa, S., Zeroual, A., Antari, J.: Identification and prediction of internet traffic using artificial neural networks. *J. Int. Learn. Syst. Appl.* **2**(03), 147 (2010)
8. Li, Y., Liu, H., Yang, W., Hu, D., Xu, W.: Inter-data-center network traffic prediction with elephant flows. In: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, pp. 206–213. IEEE (2016)
9. Bermolen, P., Rossi, D.: Support vector regression for link load prediction. *Comput. Netw.* **53**(2), 191–201 (2009)
10. Nie, L., Jiang, D., Yu, S., et al.: Network traffic prediction based on deep belief network in wireless mesh backbone networks. In: 2017 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–5. IEEE (2017)
11. Poupart, P., Chen, Z., Jaini, P., et al.: Online flow size prediction for improved network routing. In: 2016 IEEE 24th International Conference on Network Protocols (ICNP), pp. 1–6. IEEE (2016)
12. Song, C., Havlin, S., Makse, H.A.: Self-similarity of complex networks. *Nature* **433**(7024), 392 (2005)
13. Angeles Serrano, M., Krioukov, D., Boguná, M.: Self-similarity of complex networks and hidden metric spaces. *Phys. Rev. Lett.* **100**(7), 078701 (2008)
14. Crovella, M.E., Bestavros, A.: Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Trans. Networking* **5**(6), 835–846 (1997)
15. Park, K., Willinger, W.: *Self-Similar Network Traffic and Performance Evaluation*. Wiley, New York (2000)
16. Brockwell, A.E.: Likelihood-based analysis of a class of generalized long-memory time series models. *J. Time Ser. Anal.* **28**, 386–407 (2006). <https://doi.org/10.1111/j.1467-9892.2006.00515.x>
17. Witt, A., Malamud, B.D.: Quantification of long-range persistence in geophysical time series: conventional and benchmark-based improvement techniques. *Surv. Geophys.* **34**, 541–651 (2013). <https://doi.org/10.1007/s10712-012-9217-8>
18. Crovella, M., Krishnamurthy, B.: Internet measurement: infrastructure, traffic & applications. DBLP (2006)
19. Khayari, R.E.A., Sadre, R., Haverkort, B.R.: A validation of the pseudo self-similar traffic model. In: International Conference on Dependable Systems & Networks. IEEE (2002)
20. Barunik, J., Kristoufek, L.: On Hurst exponent estimation under heavy-tailed distributions. *Phys. A Stat. Mech. Appl.* **389**, 3844–3855 (2010)
21. Alvarez-Ramirez, J., Echeverria, J.C., Rodriguez, E.: Performance of a high-dimensional R/S method for Hurst exponent estimation. *Phys. A Stat. Mech. Appl.* **387**, 6452–6462 (2008)
22. Bianchi, F.M., et al.: Prediction of telephone calls load using Echo State Network with exogenous variables. *Neural Netw.* **71**, 204–213 (2015)
23. Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**(5667), 78–80 (2004). <https://doi.org/10.1126/science.1091277>
24. Chatzis, S.P., Demiris, Y.: Echo State Gaussian process. *IEEE Trans. Neural Netw.* **22**(9), 1435–1445 (2011)