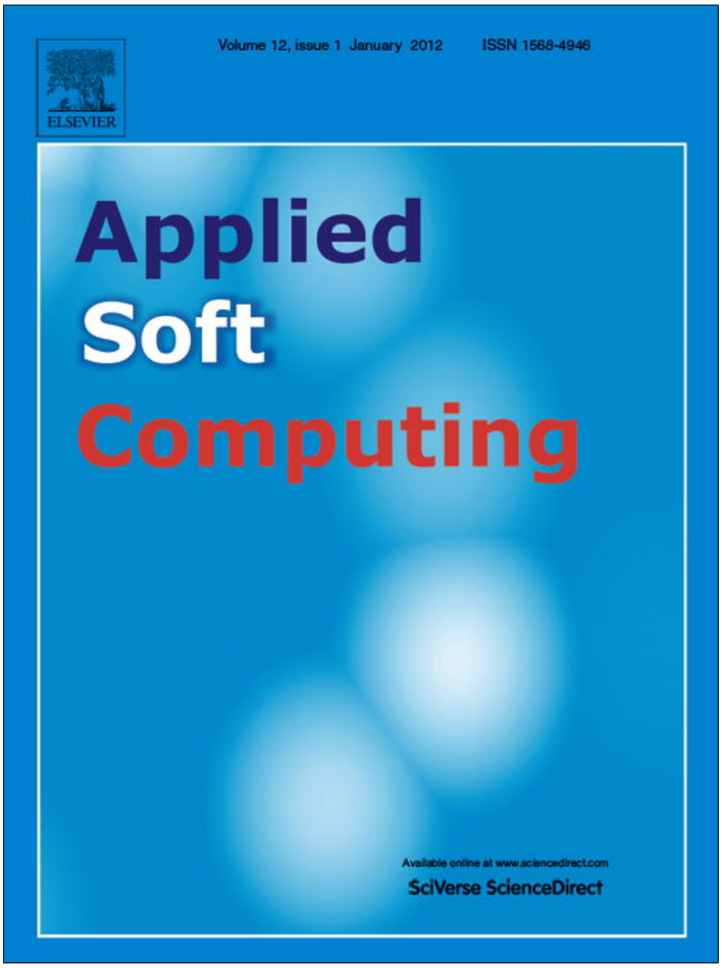
Volume 12, issue 1 January 2012
ISSN 1568-4946
ELSEVIER
Applied
Soft
Computing
Available online at www.sciencedirect.com
SciVerse ScienceDirect

Applied Soft Computing 12 (2012) 274–279

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

# Small-time scale network traffic prediction based on flexible neural tree

Yuehui Chen*, Bin Yang, Qingfang Meng

*Shandong Provincial Key Laboratory of Network Based Intelligent Computing, School of Information science and Engineering, University of Jinan, 106 Jiwei Road, 250022 Jinan, PR China*

ARTICLE INFO

*Article history:*
Received 10 December 2009
Received in revised form 18 January 2011
Accepted 14 August 2011
Available online 25 August 2011

*PACS:*
0545

*Keywords:*
Flexible neural tree model
Genetic programming
Particle Swarm Optimization
Network traffic
Small-time scale

ABSTRACT

In this paper, the flexible neural tree (FNT) model is employed to predict the small-time scale traffic measurements data. Based on the pre-defined instruction/operator sets, the FNT model can be created and evolved. This framework allows input variables selection, over-layer connections and different activation functions for the various nodes involved. The FNT structure is developed using the Genetic Programming (GP) and the parameters are optimized by the Particle Swarm Optimization algorithm (PSO). The experimental results indicate that the proposed method is efficient for forecasting small-time scale traffic measurements and can reproduce the statistical features of real traffic measurements. We also compare the performance of the FNT model with the feed-forward neural network optimized by PSO for the same problem.



## 1. Introduction

Network traffic analysis and modeling play a major role in charactering network performance, so it has been a focus of many researches. Models that accurately capture the salient characteristics of the traffic is useful for analysis and simulation, and they further our understanding of network dynamics, so it has a fundamental meaning for many network designs and engineering problems, e.g., the traffic balance scheme, router, switcher designing, the manage devices and its support software development.

Complexity is a key issue in network geometry and information traffic. Evidence of traffic complexity appears in many forms, such as the long-range correlations and self-similarity found in the statistical analysis of traffic measurements. There is also strong evidence of these phenomena at several different time scales. The complexity revealed from the traffic measurements has led to the suggestion that the network traffic cannot be analyzed in the framework of available traffic models [1–3]. Alternative reliable traffic models and tools for quality assessment and control should be developed [4,5].

Recently, communication and network technologies are developing rapidly, which brings the traffic characteristics to change greatly. The research emphasis of the network traffic analysis and modeling has change from the large-time scale to the small-time scale. The researches have shown that the traffic characteristics of the small-time scale were different from those of the large-time scale [6,7].

There is growing evidence that the nonlinear deterministic component is exist in the network traffic time series [8,9]. Various nonlinear time series prediction methods have been proposed to predict irregular nonlinear time series [11–18]. A competition was also arranged to test the success of prediction algorithms. Among those registered for the competition, two methods prove to be the most successful [10]. One uses a connectionist neural network [11–14,32] and the other utilizes the delay coordinate embedding based methodology [15–18]. Recently, many technique, for example neural networks [11–14], SVM [19], adaptive algorithms [20,21], and et al., have been used to predict nonlinear time series successfully.

The neural network is one of the most usually used nonlinear models to predict the traffic data. Ref. [22] has applied the feed-forward neural network to predict the traffic measurements data. Ref. [27] has used an adaptable neural network to predict the traffic data and to model the MPEG video sources. A wavelet-based combined model was employed to predict the network traffic in Ref. [29]. Ref. [30] has used multifractal modeling approach to the network traffic prediction. In our recent work, a local support vector machine regression model was used for the prediction of traffic measurements data [31].

* Corresponding author.
*E-mail addresses:* yhchen@ujn.edu.cn (Y. Chen), batsi@126.com (B. Yang), ise_mengqf@ujn.edu.cn (Q. Meng).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.08.045

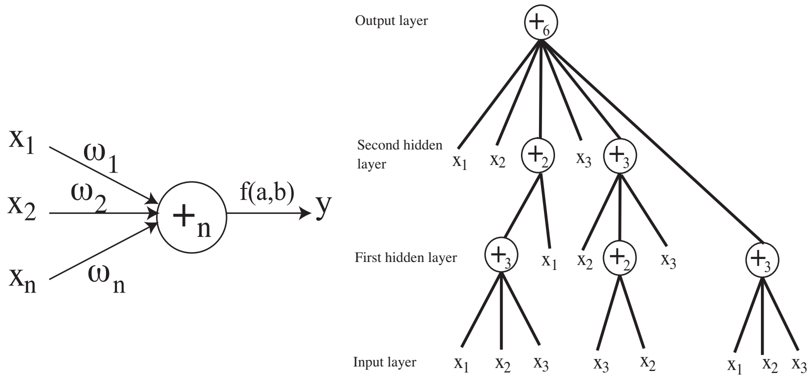


**Fig. 1.** A flexible neuron operator (left), and a typical representation of the FNT with function instruction set $F = \{+_2, +_3, +_4, +_5, +_6\}$, and terminal instruction set $T = \{x_1, x_2, x_3\}$ (right).

Although neural networks have the nonlinear approximation capability, neural networks also present many weaknesses, for example, the neural network's structure is difficult to regulate, it suffers from slow convergence characteristics and over-fitting phenomenon leading the decline of its generalization, it is prone to be trapped in local minima.

So Chen [23,24,28] has proposed the flexible neural tree (FNT). And the flexible neural tree (FNT) is a special multi-layer feed-forward neural network and allows over-layer connections, input variables selection and different activation functions for different nodes. So the network is sparse, the connections between layers are not complete.

The proposed flexible neural tree (FNT) has drastically changed the problem of optimizing and designing neural network. Comparing with the neural network, the flexible neural tree (FNT) has the following advantages: (1) for a given problem, the network's input, output and structure do not need to be designed in advance, and the FNT can automatically design and optimize the network's structure and parameters; (2) the FNT is automatically design by the evolutionary algorithm, the individuals, which has simplicity structure and similar model accuracy, will be select, and the individuals, which has complexity structure and similar model accuracy, will be eliminate; (3) the structure of FNT is generally more simple than that of neural network, and the generalization of FNT is better than that of neural network; (4) the FNT can automatically select the input variables or features.

In this paper we apply the flexible neural tree (FNT) model to predict the traffic measurements data. In our previous work, the hierarchical structure was evolved using the Probabilistic Incremental Program Evolution algorithm (PIPE) with specific instructions [23,24]. In this research work, the hierarchical structure is evolved using the genetic programming. The fine tuning of the parameters encoded in the structure is accomplished using the PSO algorithm.

The proposed method interleaves both optimizations. Starting with random structures and corresponding parameters, it first tries to improve the structure and then as soon as an improved structure is found, it fine tunes its parameters. It then goes back to improving the structure again and, fine tunes the structure and rules' parameters. This loop continues until a satisfactory solution is found or a time limit is reached.

## 2. Flexible neural tree model

In this research, a tree-structural based encoding method with specific instruction set is selected for representing a FNT model [23,25].

### 2.1. Flexible neuron instructor and FNT model

The function set $F$ and terminal instruction set $T$ used for generating a FNT model are described as follows:

$$S = F \bigcup T = \{+_2, +_3, \ldots, +_N\} \bigcup \{x_1, \ldots, x_n\}, \tag{1}$$

where $+_i (i = 2, 3, \ldots, N)$ denote non-leaf nodes' instructions and taking $i$ arguments. $x_1, x_2, \ldots, x_n$ are leaf nodes' instructions and taking no other arguments. The output of a non-leaf node is calculated as a flexible neuron model (see Fig. 1 (left)). From this point of view, the instruction $+_i$ is also called a flexible neuron operator with $i$ inputs. In the creation process of neural tree, if a nonterminal instruction, i.e., $+_i (i = 2, 3, 4, \ldots, N)$ is selected, $i$ real values are randomly generated and used for representing the connection strength between the node $+_i$ and its children. In addition, two adjustable parameters $a_i$ and $b_i$ are randomly created as flexible activation function parameters. Some examples of flexible activation functions are shown in Table 1.

For developing the FNT predictor, in this paper the following flexible activation function is used.

$$f(a_i, b_i, x) = e^{-(x - a_i/b_i)^2} \tag{2}$$

The output of a flexible neuron $+_n$ can be calculated as follows. The total excitation of $+_n$ is

$$net_n = \sum_{j=1}^{n} w_j \times x_j \tag{3}$$

where $x_j (j = 1, 2, \ldots, n)$ are the inputs to node $+_n$. The output of the node $+_n$ is then calculated by

$$out_n = f(a_n, b_n, net_n) = e^{-(net_n - a_n/b_n)^2}. \tag{4}$$

A typical flexible neuron operator and a neural tree model are illustrated in Fig. 1 (right). The overall output of flexible neural tree can be computed from left to right by depth-first method, recursively.

**Table 1**
The flexible activation functions.

| | |
|---|---|
| Gaussian function | $f(x, a, b) = exp\left(-\frac{(x-a)^2}{b^2}\right)$ |
| Unipolar sigmoid function | $f(x, a) = \frac{2\lvert a\rvert}{1+e^{-2\lvert a\rvert x}}$ |
| Bipolar sigmoid function | $f(x, a) = \frac{1-e^{-2xa}}{a(1+e^{-2xa})}$ |
| Nonlocal radial coordinates | $f(x, a, b) = (b^2 + \lVert x - a \rVert^2)^{-\alpha} (\alpha > 0)$ |
| General multiquadratics | $f(x, a, b) = (b^2 + \lVert x - a \rVert^2)^{\beta} (0 < \beta < 1)$ |
| Thin-plate s-spline function | $f(x, a, b) = (b \lVert x - a \rVert)^2 \ln(b \lVert x - a \rVert)$ |

### 2.2. The optimization of FNT model

The optimization of FNT including the tree-structure and parameter optimization. Finding an optimal or near-optimal neural tree is formulated as a product of evolution. A number of neural tree variation operators are developed as follows:

**Mutation**: Five different mutation operators were employed to generate offspring from the parents. These mutation operators are as follows:

(1) Changing one terminal node: randomly select one terminal node in the neural tree and replace it with another terminal node;
(2) Changing all the terminal nodes: select each and every terminal node in the neural tree and replace it with another terminal node;
(3) Growing: select a random leaf in hidden layer of the neural tree and replace it with a newly generated subtree.
(4) Pruning: randomly select a function node in the neural tree and replace it with a terminal node.
(5) Pruning the redundant terminals: if a node has more than 2 terminals, the redundant terminals should be deleted.

**Crossover**: Select two neural sub-trees randomly and select one nonterminal node in the hidden layer for each neural tree randomly, and then swap the selected subtree. The crossover operator is implemented with a pre-defined a probability 0.3 in this study.

**Selection**: Evolutionary programming (EP) style tournament selection was applied to select the parents for the next generation [26]. Pairwise comparison is conducted for the union of $\mu$ parents and $\mu$ offsprings. For each individual, $q$ opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is no smaller than the opponent's, it receives a selection. Select $\mu$ individuals out of parents and offsprings, that have most wins to form the next generation. This is repeated for each generation until a predefined number of generations or when the best structure is found.

#### 2.2.1. Parameter optimization by PSO

The Particle Swarm Optimization (PSO) conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA). A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\mathbf{x}_i$. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector $\mathbf{v}_i$. At each time step, a function $f_i$ representing a quality measure is calculated by using $\mathbf{x}_i$ as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector $\mathbf{p}_i$. Furthermore, the best position among all the particles obtained so far in the population is kept track of as $\mathbf{p}_g$. In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle. At each time step $t$, by using the individual best position, $\mathbf{p}_i$, and the global best position, $\mathbf{p}_g(\mathbf{t})$, a new velocity for particle $i$ is updated by

$$\mathrm{v_i(t+1)} = \mathrm{v_i(t)} + c_1\phi_1(\mathrm{p_i(t)} - \mathrm{x_i(t)}) + c_2\phi_2(\mathrm{p_g(t)} - \mathrm{x_i(t)}) \tag{5}$$

where $c_1$ and $c_2$ are positive constant and $\phi_1$ and $\phi_2$ are uniformly distributed random number in [0,1]. The term $\mathbf{v}_i$ is limited to the range of $\pm\mathbf{v}_{max}$. If the velocity violates this limit, it is set to its proper limit. Changing velocity this way enables the particle $i$ to search around its individual best position, $\mathbf{p}_i$, and global best position, $\mathbf{p}_g$. Based on the updated velocities, each particle changes its position according to the following equation:

$$\mathrm{x_i(t+1)} = \mathrm{x_i(t)} + \mathrm{v_i(t+1)}. \tag{6}$$

In the experiments, a particle consists of all the free parameters in each flexible neural tree, which will be optimized by PSO discussed in above.

### 2.3. Fitness function

To find an optimal FNT, the following objective function, the normalized mean squared error is employed.

$$NMSE = \frac{1/N\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{1/N\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{7}$$

where the $x_i$, $\hat{x}_i$ and $\bar{x}$ are the actual, FNT model output and average traffic data.

### 2.4. Computational complexity

The computational complexity of the FNT model can be described as follows: (1) the optimization of FNT trees has reduced complexity compared to the GP trees, this is because the FNT models have good computational ability than the GP models; (2) the FNT model needs parameter optimization but GP not.

### 2.5. Input/feature selection with FNT

It is often a difficult task to select variables (features) for the classification or prediction problems, especially when the feature space is large. A fully connected NN predictor usually cannot do this. In the perspective of FNT framework, the nature of model construction procedure allows the FNT to identify important input features in building a predictor that is computationally efficient and effective.

The mechanisms of input selection in the FNT constructing procedure are as follows: (1) Initially the input variables are selected to formulate the FNT model with same probabilities; (2) The variables which have more contribution to the objective function will be enhanced and have high opportunity to survive in the tree at next generation by an evolutionary procedure; (3) The evolutionary operators, i.e., crossover and mutation, provide a input selection method by which the FNT should select appropriate variables automatically.

## 3. Experimental results and analysis

In this paper we use the TCP traffic data which is downloaded from the webside 'http://ita.ee.lbl.gov/'. This traffic data contain an hour's worth of all wide-area traffic between Digital Equipment Corporation and the rest of the world. The data package used in this paper is DEC-Pkt1, and the time stamps have millisecond precision. The traffic data aggregated with time bin 0.1 s, that is the arrived package's amount within the 0.1 s time interval, are shown in Fig. 2. Generally we may consider the traffic measurements as a sum of a regular process and a stochastic part, related to the high-frequency noise. The elimination of the noisy part may simplify the analyzed time series. In order to separate the regular component of the dynamical process from the stochastic noise component, we apply the wavelet soft threshold noise reduction method to the traffic measurements data. The difference between the original time series and the filtered signal, corresponds to the noisy component. Fig. 2 presents the original traffic series, Fig. 3 presents the corresponding filtered signal and Fig. 4 presents the noisy component. Then the filtered traffic measurements data were normalized to the interval [0, 1] with following formula $x' = (x - x_{min})/(x_{max} - x_{min})$.

The length of this traffic measurements data is 36,000. The front 33,000 data points are used as the training set, and the last 3000 data points are used as the test set.

The parameters used for the experiment is listed in Table 2.

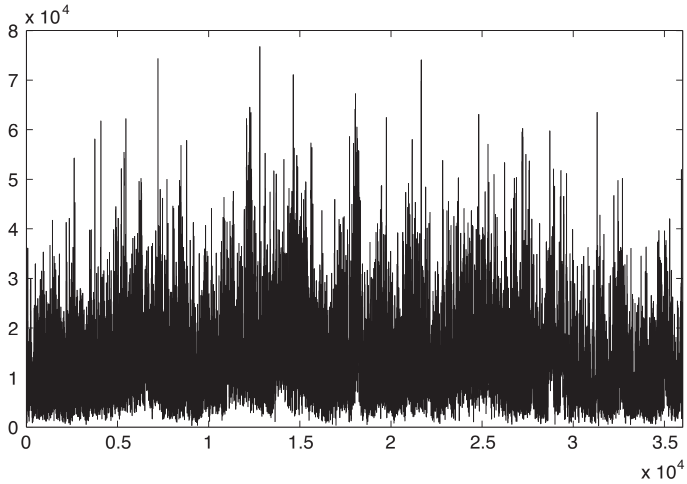

**Fig. 2.** Traffic measurements data aggregated with 0.1 s.

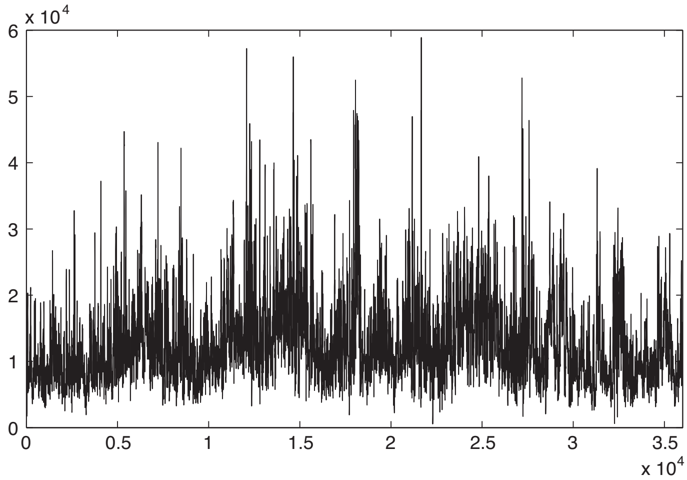

**Fig. 3.** Filtered traffic measurements data.

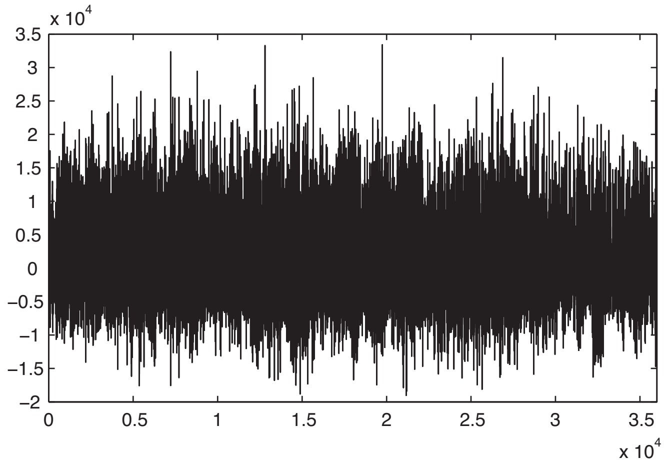

**Fig. 4.** The noisy component.

**Table 2**
Parameter settings.

| | |
|---|---|
| Population size | 50 |
| Crossover probability | 0.4 |
| Mutation probability | 0.03 |
| $c_1$ | 2.0 |
| $c_2$ | 2.0 |
| $v_{max}$ | 5.0 |

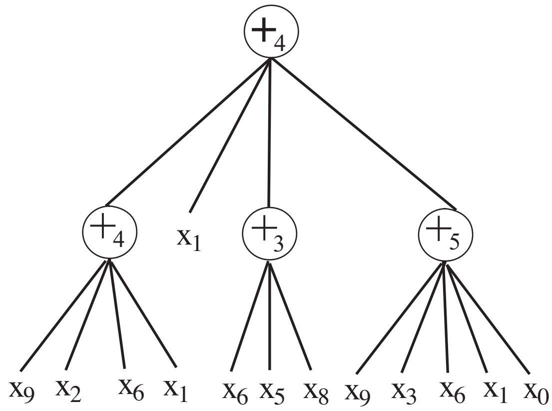


**Fig. 5.** The evolved optimal FNT model for prediction network traffic measurement data.

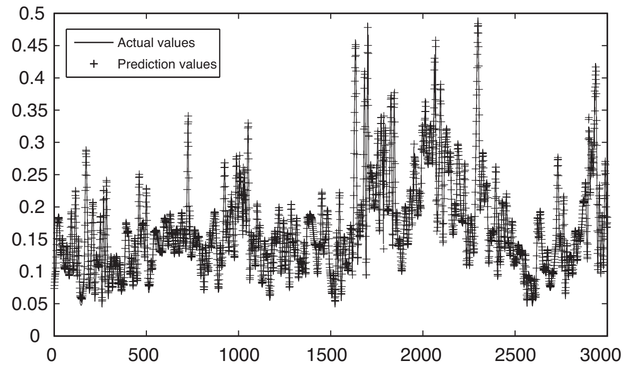


**Fig. 6.** Comparison of actual network traffic time series data and the predicted ones.

For this simulation, the 10 input variables are used for constructing a FNT model. The instruction sets used to create an optimal FNT model is $S=F\bigcup T=\{+_2, \ldots, +_6\}\bigcup\{x_0,x_1, \ldots, x_9\}$. Where $x_i(i=0, 1, \ldots, 9)$ denotes the 10 input variables. Where the 10 input variables compose the input vector$[x_0, x_1, \ldots, x_9]$ corresponding $[x(i), x(i+1), \ldots, x(i+9)]$, $i=1, 2, \ldots, N-9$, for time series $x(n)$, $n=1, 2, \ldots, N$.

The optimal FNT model for prediction network traffic measurements data are shown in Fig. 5. The *NMSEs* for training and testing data sets are 0.062945 and 0.011215, respectively. A comparison of actual network traffic time series data and the predicted ones is shown in Fig. 6 and the prediction error is shown in Fig. 7. It should be noted that the important features for constructing the FNT model were formulated in accordance with the procedure mentioned in the previous section. These important variables are shown in Table 3. From Figs. 6 and 7, it can be clearly seen that the FNT model can effectively predict the traffic measurements data,

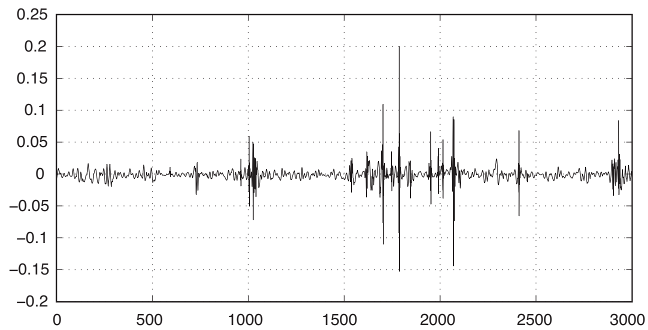

**Fig. 7.** The prediction error.

**Table 3**
The important features selected by the FNT algorithm.

| |
|---|
| $x_0,x_1,x_2,x_3,x_5,x_6,x_8,x_9$ |

**Table 4**
Comparative results of the feed-forward neural network model and the flexible neural tree model (*NMSEs*).

| *NMSEs* | Training data sets | Testing data sets |
|---|---|---|
| Neural network | 0.092304 | 0.073236 |
| Our FNT | 0.062945 | 0.011215 |

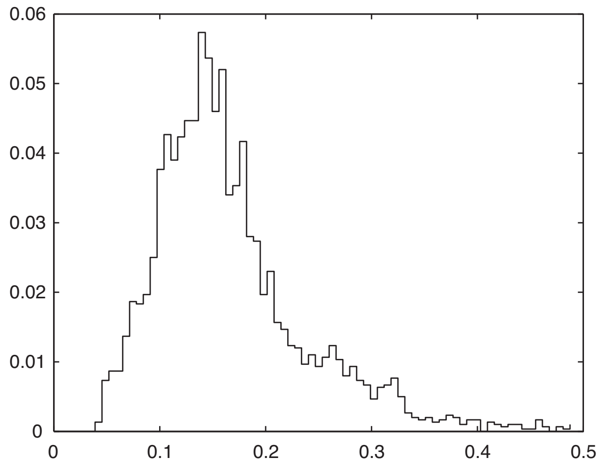


**Fig. 8.** The statistical distributions of the traffic measurements data.

the prediction accuracy is well, and the prediction error is very low.

For comparison purpose, a feed-forward neural network with the structure {10-12-1}, trained by using PSO algorithm is also used to predict the same network traffic data, the *NMSEs* for training and testing data sets are 0.092304 and 0.073236, respectively, and the comparative results are shown in Table 4. It can be seen that the prediction performance of FNT model is better than the feed-forward neural network model which is also optimized by PSO.

The statistical distributions of the traffic measurements and the prediction values are shown in Figs. 8 and 9. The statistical distribution of the prediction errors is shown in Fig. 10, where the abscissa is the value of the prediction errors and the vertical axis is the probability of the prediction errors.

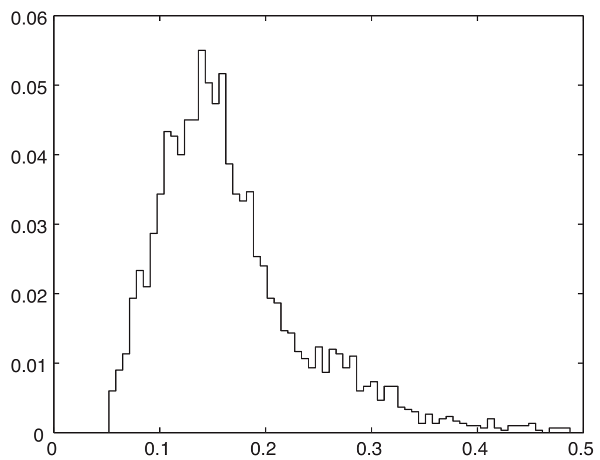


**Fig. 9.** The statistical distributions of the generated time series.

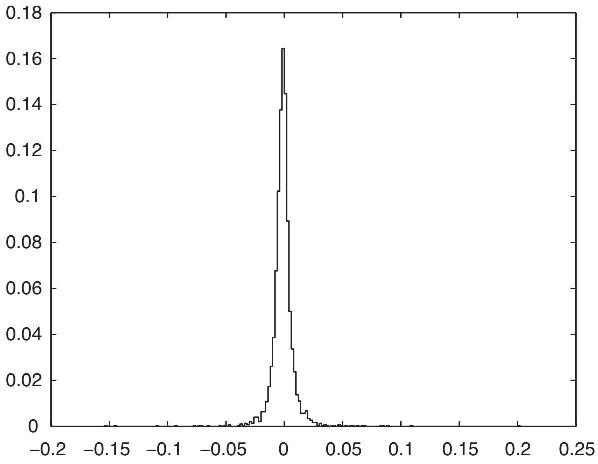


**Fig. 10.** The statistical distributions of the prediction errors.

From Figs. 8 and 9, it can be clearly seen that the time series generated by the FNT model have very similar probability distribution with the actual traffic measurements time series. So the FNT model can reproduce quite well the statistical distribution of the real traffic measurements data.

From Fig. 10, it can be clearly seen that the prediction error of the FNT model to the traffic data mainly concentrates on the vicinity of zero, and the probability of the prediction error with large absolute value is very small.

## 4. Conclusion

In summary we apply the flexible neural tree model to predict the traffic measurements data. The experimental results show that the FNT model can be successfully used for a deeper understanding of main features of the traffic data. Based on the pre-defined instruction/operator sets, a FNT model can be created and evolved. This framework allows input variables selection, over-layer connections and different activation functions for the various nodes involved. The FNT structure is developed using Genetic Programming (GP) and the parameters are optimized by Particle Swarm Optimization algorithm (PSO). The experiment results show that the FNT model can effectively predict the traffic measurements data, the prediction accuracy is well, the prediction error mainly concentrates on the vicinity of zero, and the time series generated by the FNT model have very similar probability distribution with the actual traffic measurements time series, and the prediction accuracy of the FNT model is superior to that of the feedforward neural network optimized by PSO.

## Acknowledgements

This research was partially supported by the Natural Science Foundation of China (61070130, 61173079, 60903176, 60973093, 60873089), Key Project of Natural Science Foundation of Shandong Province (ZR2011FZ001, ZR2011FZ003), and the Key Subject Research Foundation of Shandong Province.

## References

[1] G. Orosz, B. Krauskopf, R.E. Wilson, Bifurcations and multiple traffic jams in a car-following model with reaction-time delay, Physica D 211 (2005) 277–293.
[2] G. Orosz, R.E. Wilson, B. Krauskopf, Global bifurcation investigation of an optimal velocity traffic model with driver reaction time, Phys. Rev. E 70 (2004) 026207.
[3] I. Gasser, G. Sirito, B. Werner, Bifurcation analysis of a class of 'car following' traffic models, Physica D 197 (2004) 222–241.

[4] A.D. Doulamis, N.D. Doulamis, S.D. Kollias, An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources, IEEE Trans. Neural Netw. 14 (2003) 150–165.
[5] Y.B. Xie, W.X. Wang, B.H. Wang, Modeling the coevolution of topology and traffic on weighted technological networks, Phys. Rev. E 75 (2007) 026111.
[6] Z.L. Zhang, V.J. Ribeiro, S. Mooon, C. Diot, Small-time scaling behaviors of Internet backbone traffic: an empirical study, IEEE INFOCOM 3 (2003) 1826–1836.
[7] S. Uglig, Non-stationarity and high-order scaling in TCP flow arrivals: a methodological analysis, in: ACM SIGCOMM, Computer Communications Review, 34, 2004.
[8] P. Shang, X. Li, S. Kamae, Chaotic analysis of traffic time series, Chaos Solitons Fractals 25 (2005) 121–128.
[9] P. Shang, X. Li, S. Kamae, Nonlinear analysis of traffic time series at different temporal scales, Phys. Lett. A 357 (2006) 314–318.
[10] N.A. Gershenfeld, A.S. Weigend, The future of time series: learning and understanding, in: A.S. Weigend, N.A. Gershenfeld (Eds.), Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley, Redwood City, CA, 1993.
[11] D.S.K. Karunasinghe, S.-Y. Liong, Chaotic time series prediction with a global model: artificial neural network, J. Hydrol. 323 (2006) 92–105.
[12] A. Freking, W. Kinzel, I. Kanter, Learning and predicting time series by neural networks, Phys. Rev. E 65 (2002) 050903.
[13] M. Small, C.K. Tse, Minimum description length neural networks for time series prediction, Phys. Rev. E 66 (2002) 066701.
[14] Q.-L. Ma, Q.-L. Zheng, H. Peng, T.-W. Zhong, J.-W. Qin, Multi-step-prediction of chaotic time series based on co-evolutionary recurrent neural network, Chin. Phys. B 17 (2) (2008) 536–542.
[15] Y.-S. Wang, J. Sun, C.-J. Wang, H.-D. Fan, Predicting chaotic time series, Phys. Rev. Lett. 59 (1987) 845–848.
[16] J.-F. Zhang, S.-T. Hu, Markov models from data by simple nonlinear time series predictors in delay embedding spaces, Phys. Rev. E 65 (2002) 056201.
[17] Q. Meng, Y. Peng, A new local linear prediction model for chaotic time series, Phys. Lett. A 370 (2007) 465–470.
[18] Q.-F. Meng, Y. Peng, H. Qu, M. Han, The neighbor point selection method for local prediction based on information criterion, Acta Phys. Sin. 57 (3) (2008) 1423–1430.
[19] H. Liu, D. Liu, L.-F. Deng, Chaotic time series prediction using fuzzy sigmoid kernel-based support vector machines, Chin. Phys. 15 (6) (2006) 1196–1200.
[20] H. Li, J. Zhang, X. Xiao, Neural Volterra filter for chaotic time series prediction, Chin. Phys. 14 (11) (2005) 2181–2188.
[21] Q. Meng, Q. Zhang, W. Mu, A novel multi-step adaptive prediction method for chaotic time series, Acta Phys. Sin. 55 (4) (2006) 1666–1671.
[22] P. Akritas, P.G. Akishin, I. Antoniou, A.Y. Bonushkina, I. Drossinos, et al., Nonlinear analysis of network traffic, Chaos Solitons Fractals 14 (2002) 595–606.
[23] Y. Chen, B. Yang, J. Dong, A. Abraham, Time-series forecasting using flexible neural tree model, Inf. Sci. 174 (2005) 219–235.
[24] Y. Chen, B. Yang, A. Abraham, Flexible neural trees ensemble for stock index modeling, Neurocomputing 70 (2007) 697–703.
[25] Y. Chen, B. Yang, J. Dong, Nonlinear system modeling via optimal design of neural trees, Int. J. Neural Syst. 14 (2) (2004) 125–137.
[26] K. Chellapilla, Fitness distributions in evolutionary computation, IEEE Trans. Evol. Comput. 1 (1997) 209–216.
[27] A.D. Doulamis, N.D. Doulamis, S.D. Kollias, An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources, IEEE Trans. Neural Netw. 14 (1) (2003) 150–165.
[28] L. Wang, B. Yang, Y. Chen, X. Zhao, J. Chang, Modeling early-age hydration kinetics of Portland cement using flexible neural tree, Neural Comput. Appl., doi:10.1007/s00521-010-0475-4, Online First, 3 November 2010.
[29] H.-L. Sun, Y.-H. Jin, Y.-D. Cui, S.-D. Cheng, Network traffic prediction by a wavelet-based combined model, Chin. Phys. B 18 (11) (2009) 47–60.
[30] F.H.T. Vieira, G.R. Bianchi, L.L. Lee, A network traffic prediction approach based on multifractal modeling, J. High Speed Netw. 17 (2) (2010) 83–96.
[31] Q. Meng, Y. Chen, Small-time scale network traffic prediction based on local support vector machine regression model, Chin. Phys. B 18 (2009) 2112–2194.
[32] W.K. Wonga, M. Xia, W.C. Chu, Adaptive neural network model for time-series forecasting, Eur. J. Oper. Res. 207 (2010) 807–816.