

Network Traffic Prediction using ARIMA and Neural Networks Models

G. Rutka

*Faculty of Electronics and Telecommunication, Riga Technical University
 Riga, Azenes street 12-317, LV-1048, phone: +371 9627600, e-mail: gundegarutka@tvnet.lv*

Introduction

Network traffic exhibits degree of self- similarity at large time scale and high degree of multifractal at small time scale. Network performance is a vital to business for bringing a product to the consumers. With the growing number of network users, more acute is maintenance and reliability of these networks. The predictability of network traffic is of significant interest in many domains, including adaptive applications [1], congestion control [2], admission control [3], wireless and network management [4]. An accurate traffic prediction model should have the ability to capture the prominent traffic characteristics, e.g. short and long dependence, self similarity in large- time scale and multifractal in small- time scale. For these reasons time series models are introduced in network traffic simulation and prediction.

The today's network has a great number of applications (not just voice conversation), each with its own traffic characteristics, and new applications can arise at any time. There are many more varieties of network connectivity, architecture, and equipment, and, accordingly, a different type of traffic flows. There are no standard network topologies around which all design efforts can be based, and the topologies that exist are subject to constant change [5]. These all factors complicate the accuracy of traffic modelling and prediction.

There has been a large work focused on developing prediction models for computer data networks. The most popular model for prediction is used autoregressive integrated moving average – ARIMA (also fractional ARIMA), but this model often fails to perform correct prediction [6], [7]. The Auto Regressive Integrated Moving Average (ARIMA) with Generalized Auto Regressive Conditional Heteroscedasticity (GARCH) model is a non-linear time series model which combined the linear ARIMA with conditional variance GARCH. ARIMA/GARCH model provides flexibility when it is applied to model the network traffic [8], [9]. Neural networks also provide reliable prediction, but it depends on traffic characteristics [10], [11], [12].

The major open question is to understand the accuracy of made prediction using Internet traffic in different time scales and what are the benefits of imposed prediction models.

Self- similarity

For a self similar time series:

$$\{X\} = \{X_1, X_2, \dots, X_k\}. \tag{1}$$

The m-aggregate $\{X_k^{(m)}\}$ with its k-th term:

$$X_k^{(m)} = \frac{X_{km-m+1} + \dots + X_{km}}{m}, \text{ where } k=1,2,3,\dots \tag{2}$$

In particular, we assume that X has an autocorrelation function (ACF) of the form:

$$r(k) \sim k^{-(2-2H)}L(k) \text{ as } k \rightarrow \infty, \tag{3}$$

where H is called the Hurst parameter and L(k) is slowly varying at infinity.

Analyzing the ACF we can see whether the traffic data is short range dependent or long range dependent or exactly self similar or asymptotically self similar traffic.

The Hurst parameter H in (1) is in the range $0.5 < H < 1$ and it characterizes the process in terms of the degree of self-similarity and long time dependence. The degree of self-similarity and long-range dependence increases as $H \rightarrow 1$. In our experiments self-similarity will be estimated by the use of variance-time plot method. This is one of the easiest methods how to estimate Hurst's coefficient. In the process the variance of aggregate the self-similar process is defined:

$$\text{VAR}(X^{(m)}) = \text{VAR}(X)/m^\beta. \tag{4}$$

In the (4) β is calculated from the equation:

$$H = 1 - \beta/2. \tag{5}$$

The (4) can be rewritten is the following form:

$$\log\{\text{VAR}(X^{(m)})\} \sim \log\{\text{VAR}(X)\} - \beta \log\{m\}. \tag{6}$$

If VAR(X) and m are plotted on a log-log graph then by fitting a least square line through the resulting points we can obtain a straight line with the slope of $-\beta$ [13], [14], [15], [16].

ARIMA models

Multistep prediction can be achieved by using the predicted value as the real value or by aggregating the traffic into larger time interval. ARIMA(p; d; q) model is used for prediction. An ARIMA (p,d,q) model is composed of three elements:

p: Autoregression;

d: Integration or Differencing;

q: Moving Average.

A simple ARIMA (0,0,0) model without any of the three processes above is written as:

$$Y_t = a_t \quad (7)$$

The autoregression process [ARIMA (p,0,0)] refers to how important previous values are to the current one over time. A data value at t_1 may affect the data value of the series at t_2 and t_3 . But the data value at t_1 will decrease on an exponential basis as time passes so that the effect will decrease to near zero. It should be pointed out that f is constrained between -1 and 1 and as it becomes larger, the effects at all subsequent lags increase.

$$Y_t = \phi_1 Y_{t-1} + a_t \quad (8)$$

The integration process [ARIMA (0,d,0)] is differenced to remove the trend and drift of the data (i.e. makes non-stationary data stationary). The first observation is subtracted from the second and the second from the third and So the final form without AR or MA processes is the ARIMA (0,1,0) model:

$$Y_t = Y_{t-1} + a_t \quad (9)$$

The order of the process rarely exceeds one ($d < 2$ in most situations).

The moving average process [ARIMA (0,0,q)] is used for serial correlated data. The process is composed of the current random shock and portions of the q previous shocks. An ARIMA (0,0,1) model is described as:

$$Y_t = a_t - q_1 a_{t-1} \quad (10)$$

As with the integration process, the MA process rarely exceeds the first order.

Box-Jenkins forecasting methodology is used to establish the ARIMA (p,d,q) model for prediction at each scale. Box-Jenkins methodology involves four steps:

- The first step is the tentative identification of the model parameters.

This is done by examining the sample autocorrelation function:

$$r_k = \frac{\frac{1}{n-k} \sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X})}{\frac{1}{n} \sum_{t=1}^n (X_t - \bar{X})^2} \quad (20)$$

where

$$\bar{X} = \frac{\sum_{t=1}^n X_t}{n} \quad (21)$$

and the sample partial autocorrelation function:

$$r_{kk} = \begin{cases} r_1, & \text{if } k=1; \\ \frac{r_k - \sum_{j=1}^{k-1} r_{k-1,j} r_{k-j}}{1 - \sum_{j=1}^{k-1} r_{k-1,j} r_j}, & \text{if } k=2,3,\dots; \end{cases} \quad (22)$$

where

$$r_{kj} = r_{k-1,j} - r_{kk} r_{k-1,k-j}, \text{ for } j=1,2,\dots,k-1 \quad (23)$$

- Estimation step. Once the model is established, the model parameters can be estimated using either a maximum likelihood approach or a least mean square approach. In this paper both the maximum likelihood approach and the least mean square approach were tried and their results are almost exactly the same. Thus we stick to the least mean square approach to estimate the model parameters for its simplicity.

- Diagnostic check step. Diagnostic checks can be used to see whether or not the model that has been tentatively identified and estimated is adequate. This can be done by examining the sample autocorrelation function of the error signal, i.e. the difference between the predicted value and the real value. If the model is inadequate, it must be modified and improved.

- Final model is determined, it can be used to forecast future time series values.

Neural networks

Many authors have applied many different neural network (NN) architectures and algorithms to explore traffic modeling task [10], [11], [12].

In our research we use the following prediction steps:

1. Creation of NN;
2. NN has been initialized (this property defines the function used to initialize the network's weight matrices and bias vectors)
3. NN has been simulated.
4. NN has been trained.

Once the network weights and biases have been initialized, the network is ready for training. The training process requires a set of examples of proper network behavior - network inputs p and target outputs t . During training the weights and biases of the network are iteratively adjusted to minimize the network performance function.

5. NN inputs (also targets) and/or outputs (also errors) are adapted. This property defines the parameters and values of the current adapt (initialization, performance, training) function.

6. After NN model is created, simulated and trained it is used for prediction ok k steps ahead.

Research models (phases)

Our research is emphasized to self-similar traffic prediction using neural networks. Traffic data is taken from website <http://freestats.com/>. This data was collected for one year. Another data trace is collected using website access statistics of local area network users using access to the site www.fotoblog.lv.

For statistical analyses and neural network testing we use program package “MATLAB p6.5” and “STATISTICA neural networks”.

Firstly, we have deeply studied the character of the statistical material (traffic data). Accordingly to that we have calculated and proved that traffic data is self-similar. We have investigated the character of the autocorrelation function (ACF).

Secondly, we modulated and simulated different prediction models - ARIMA models and neural network models. Using different step prediction we have verified the prediction accuracy taking into account prediction steps ahead.

Review of studied cases (results)

In our experiments we analyze 2 types of traces:

Table 1. Summary of the traces used in the study

Name	Observations	Step	Duration
Freestats	8760	1 hour	365 days
Fotoblog	172800	1 sec	2 days

Analyzing the self similarity of these traces we have calculated the Hurst parameter. The results are shown below in the Fig. 1.

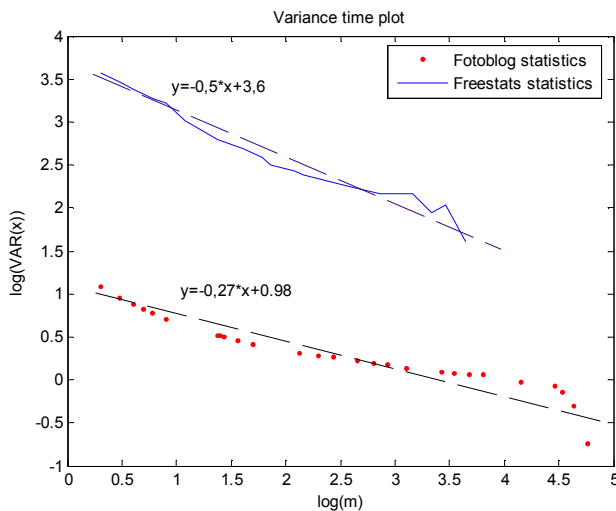


Fig. 1. The Hurst parameter estimation with the variance - time plot

The variance-time curve (Fig. 1) shows an asymptotic slope that is easily estimated to be about -0.50 for Freestats trace and -0.27 for Fotoblog trace, resulting in a practically identical estimate of the Hurst parameter H of about 0.75 for Freestats trace and 0.865 for Fotoblog trace.

It is important to understand the ACF role in the prediction model selection. If there is no autocorrelation

function present in the signal (traffic data), there is nothing to model, a linear approach is bound to fail, a nonlinear approach is likely to fail and the best predictor is probably the mean value of the signal. For this reason we have studied the autocorrelation structure of our traces. Fig. 2 and Fig. 3 shows the ACF of a representative Freestats trace.

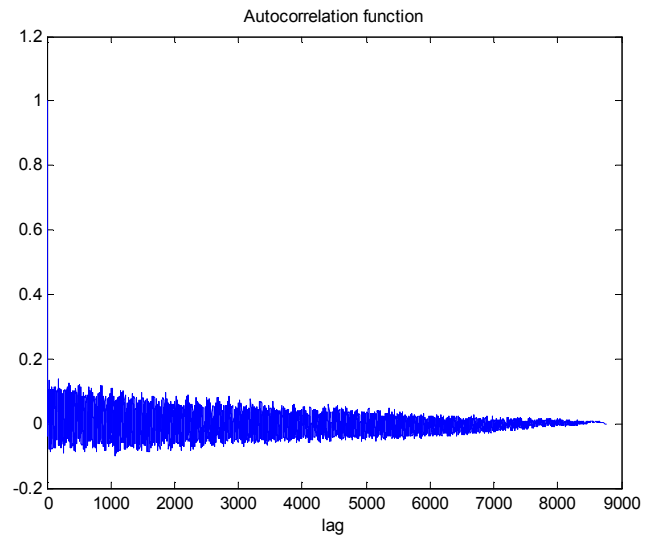


Fig. 2. Autocorrelation structure of Freestats (all possible lags)

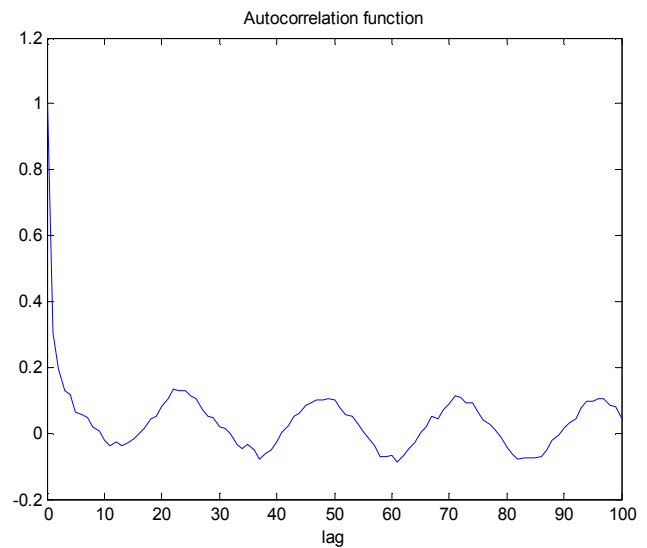


Fig. 3. Autocorrelation structure of Freestats (part of lags)

Analyzing the ACF for Freestats trace in the different lag window (Fig. 3) we can see that ACF has a sine-wave shape pattern which exponential decays. We expect that such trace will be quite predictable using linear models.

Fig. 4 and Fig. 5 shows the ACF of Fotoblog trace. It is clearly not a white noise and yet it doesn't have the strong behavior of Freestats trace. We would expect that such a trace is predictable to some extent linear models or Neural network models.

The Fig. 2-4 shows that the traffic data is long - range dependent process. It can be verified analyzing the Hurst parameter H which is a measure of the level of self-similarity of a time series that exhibits long-range dependence. The closer H is to 1, the greater the degree of persistence or long-range dependence.

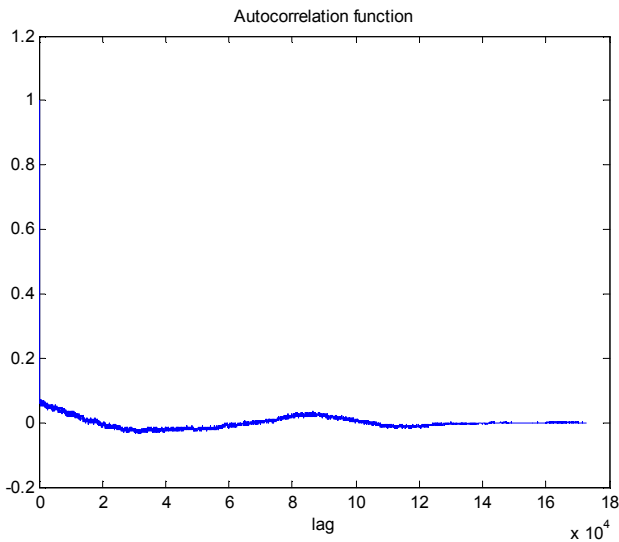


Fig. 3. Autocorrelation structure of Fotoblog (all possible lags)

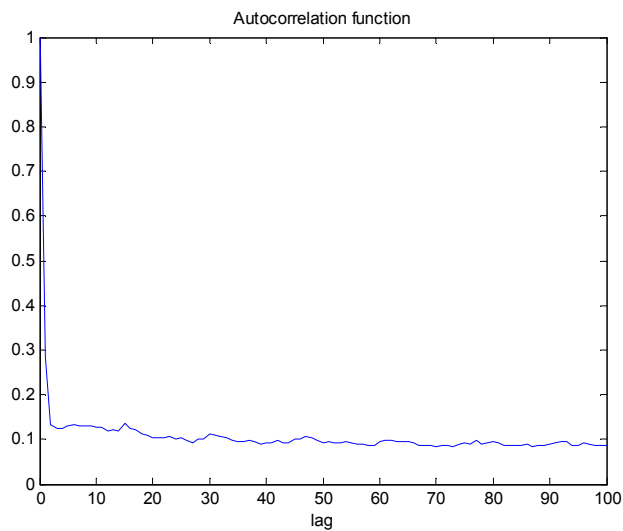


Fig. 4. Autocorrelation structure of Fotoblog (part of lags)

We have also analyzed the ACF for some m -aggregates of the traffic data (Fig. 5-6).

The nature of autocorrelations of m -aggregates shows a hyperbolic fall-off as in [17], [18]. We can find a notable self similarity when we compare the plots for different m -aggregates (Fig. 5-6). This requires a data set over a long duration.

Especially at the higher level m all the m -aggregated series give familiar autocorrelations. Thus we have a different indication of self similarity also useful for verification in a small data set.

For future value prediction modeling we used feed forward backpropagation neural network.

The graphical representation of the results of feed-forward backpropagation network use for traffic simulation and training are in Fig. 7- Fig. 9.

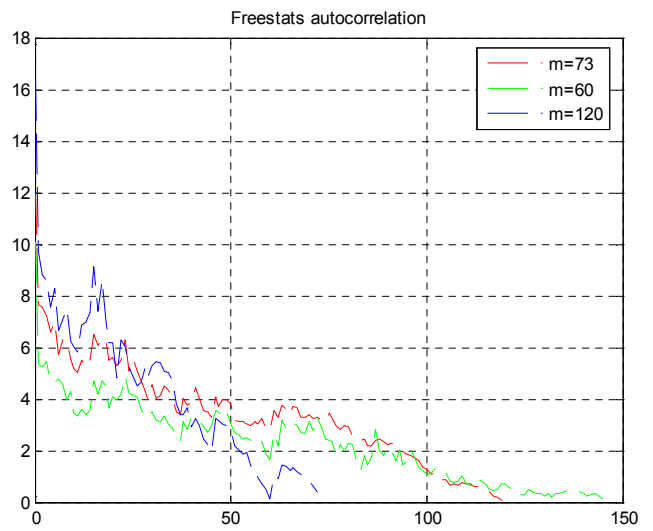


Fig. 5. Autocorrelation structure of Freestats at different aggregation level m

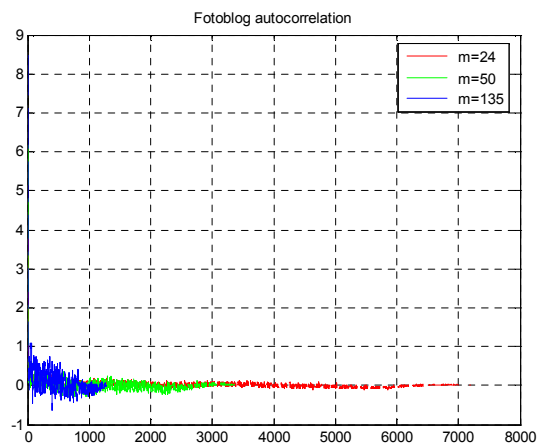


Fig. 6. Autocorrelation structure of Fotoblog at different aggregation level m

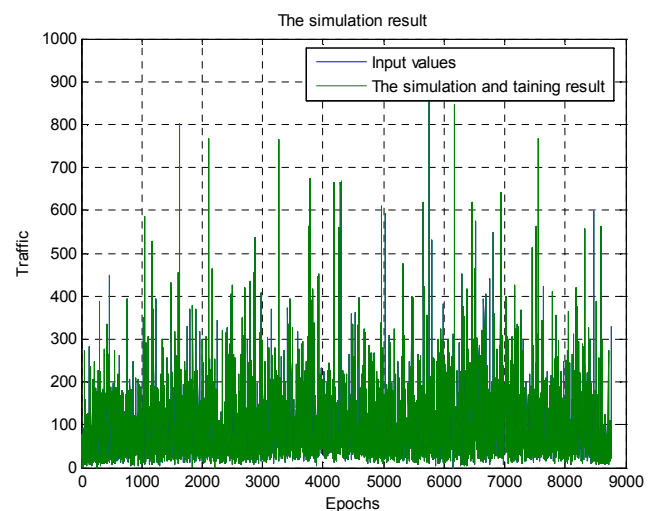


Fig. 7. The simulation and training result of Freestats statistics using feed forward NN

Fig. 7 shows that the NN model has 100% trained and simulated the traffic data as they were given in the inputs. The average error (mean square error) is $8.3e^{-29}$ what means that error is closed to zero - in Fig. 7 we can see that there are no differences between simulation and raining result.

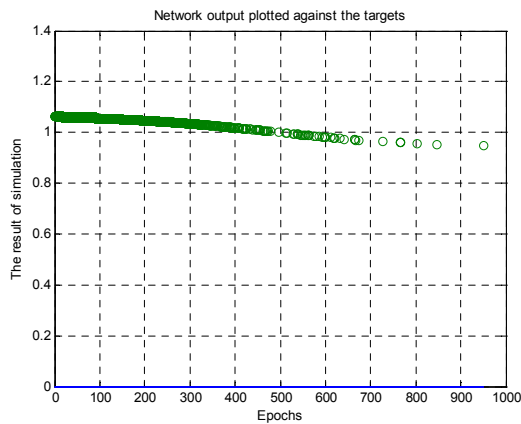


Fig. 8. Network output during the learning process of Freestats statistics using feed forward NN

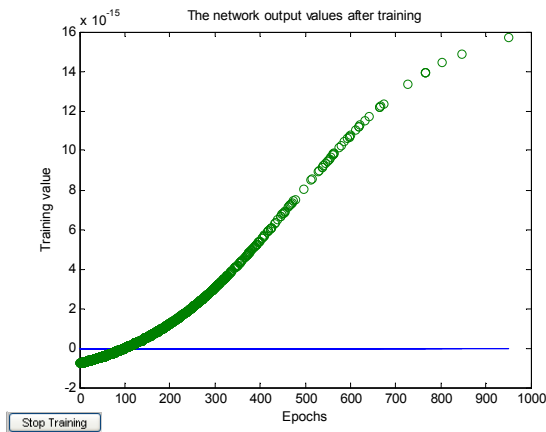


Fig. 9. The network output after training of Freestats statistics using feed forward NN

After NN model simulation and training we have used it for prediction ok k- steps ahead. The same prediction of k -steps ahead is done using ARIMA models. The results are summarized in Fig. 10 - Fig. 11.

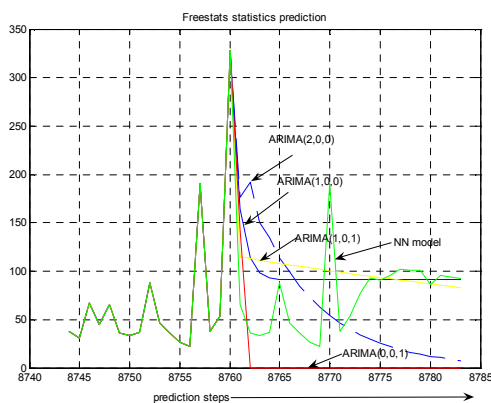


Fig. 10. Freestats statistics prediction using ARIMA and NN models

As we see in Fig. 10 the ARIMA models produce very familiar results - after some step prediction the prediction slope becomes changeless. In the picture we have shown 30 steps of previous known input values and 20 steps of prediction ahead. The prediction values of NN model differs from those of ARIMA.

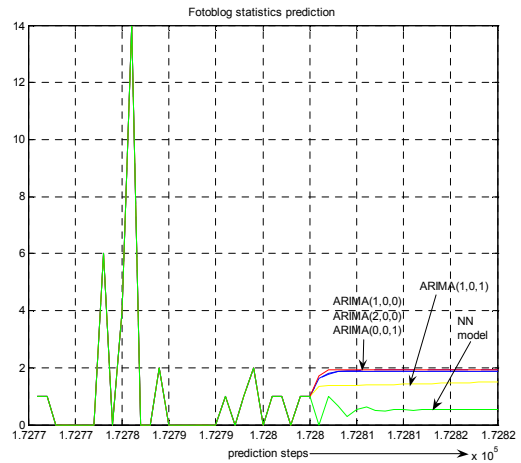


Fig. 11. Fotoblog statistics prediction using ARIMA and NN models

As we see in Fig. 11 we have gain almost the same results as in Fig. 10.

Conclusions

The traces used in our experiments can be considered as continuous-time stochastic process $X(t)$ to be statistical self-similar with parameter $H=0.75$ for Freestats trace and $H=0.865$ for Fotoblog trace ($0.5 < H < 1$). The self similar traffic estimation shows that as longer period is used for traffic data analyses (or the total volume of the traffic data observations) as faster the Hurst parameter becomes closer to 1. This trend is shown in the Fig. 1. This means that the infinite traffic data is the best, but that is not the real case.

We have analyzed ACF in different scales. The conclusion is that the traffic data is long – range dependent process in the both cases - for Freestats and Fotoblog traces. It can be verified analyzing the Hurst parameter H - the closer H is to 1, the greater the degree of persistence or long-range dependence.

Regarding ARIMA models for Freestats trace:

ARIMA(0,0,1) was able to predict only 2 steps ahead, ARIMA (1,0,0) – 6 steps ahead, ARIMA (2,0,0) - 8 steps ahead and ARIMA (1,0,1) - 50 steps ahead after what the parameter estimation process converged.

Regarding ARIMA models for Fotoblog trace:

ARIMA(0,0,1) was able to predict only 2 steps ahead, ARIMA (1,0,0) - 14 steps ahead, ARIMA (2,0,0) - 20 steps ahead and ARIMA (1,0,1) - 100 steps ahead after what the parameter estimation process converged.

Regarding NN models we focused only on one type of neural network, respectively, feed-forward backpropogation network, because this model is very easy to use (simulation and training process don't take a long time and simulation and training errors are close to zero).

Comparing the results of done prediction we can conclude that ARIMA models are easier to use for training and forecast, but the prediction result is not very accurate. Contrary NN models are quite complex while they are created simulated and trained and only after that they are used for prediction. As a result of such complex process the prediction result is much better comparing with ARIMA models.

Acknowledgement

This work has been partly supported by the European Social Fund within the National Program "Support for the carrying out doctoral study program's and post- doctoral researches" project "Support for the development of doctoral studies at Riga Technical University".

References

1. **Stemm M., Seshan S, Katz R.H.** „A network measurement architecture for adaptive applications”, Proceedings of INFOCOM 2000, vol.1, pp.285-294
2. **Jacobson V.** “Congestion avoidance and control”, Proceeding of the ACM SIGCOMM 1988, Symposium, pp.314-329
3. **Jamin S. and group of authors.** „A measurement based admission control algorithm for integrated services packet networks”, Proceedings of ACM SIGCOMM 1995, pp.56-70
4. **Kim M. and Noble B.** “Mobile network estimation”, Proceeding of 7th Annual International Conference on Mobile Computing and Networking, pp.298-309
5. **Dr.Thomas B.Fower.** “A short tutorial on fractals and internet traffic”, The telecommunications review, 1999
6. **Sang A.and Li S.** „A predictability analysis of network traffic”, Computer networks, vol.39, pp.329-345, 2002
7. **Shu Y., and group of authors.** “Traffic prediction using FARIMA models”, ICC, vol.2, pp.891-895, 1999
8. **Zhou B. and group of authors.** „Network traffic modeling and prediction with ARIMA/GARCH”, Proceeding of the third international working conference: performance modelling and evaluation of heterogeneous networks (HET-NETS'05), Ilkly, UK, July 2005
9. **Nakatsuma T. and Tsurumi H.** „ARMA-GARCH models: Bayes Estimation Versus MLE and Bayes Non-stationary Test”, Department of Economics, Retgurs University, 1996
10. **Tong H., Li C., He J.** “Internet traffic prediction based on boosting feedforward neural network”, 2004
11. **Li H., Kozma R.** “A dynamic Neural Network Method for Time Series Prediction Using the KIII Model”, Proceeding IJCNN, Portland, 2003.
12. **Jonckheere E.A., Silvester J.A.** “Utilizing neural networks to reduce packet loss in self similar teletraffic patterns”, 2003
13. **Peha J.M.** “Protocols can make traffic appear self-similar”, IEEE/AMC/SCS Communication networks and distributed systems modeling and simulation conf., pp.47-52, Jan.1997
14. **Robert S. and Boudee J.Le.** “A Markov modulated process for self-similar traffic”, Sept 23, 1995
15. **Jonckheere E.A., Silvester J.A.** “Neural network modeling of self-similar teletraffic patterns”, 2003
16. **Rose O.** “Estimation of the Hurst parameter of long range dependent time series”, University of Wurzburg, Research report series, Feb.1996
17. **Chakraborty D. and group of authors.** “Self-similar and fractal nature of internet traffic”, International Journal of Network Management, v.14 n.2, p.119-129, March 2004
18. **Wilinger W, Taqy and group of authors.** „Self similarity through high variability: statistical analysis of ethernet la traffic at the source level”, IEEE/ACM Transactions Networking, No.5, pp.71-86, 1997

Submitted for publication 2007 11 28

G. Rutka. Network Traffic Prediction using ARIMA and Neural Networks Models // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 4(84). – P. 47–52.

The network traffic prediction plays a fundamental role in network design, management, control and optimization. The self-similar and non-linear nature of traffic makes accurate prediction more difficult. This paper presents a view of models used for network traffic prediction using ARIMA and neural network models. Our experiments inspect the accuracy of k-step ahead prediction using ARIMA and neural network models. Ill. 11, bibl. 18 (in English; summaries in English, Russian and Lithuanian).

Г. Рутка. Прогнозирование нагрузки сети применяя модели ARIMA и нейронные сети // Электроника и электротехника. – Каунас: Технология, 2008. – № 4(84). – С. 47–52.

Прогнозирование нагрузки сети имеет большую роль в проектирование, управление, контроль и оптимизаций сети. Самопохожест и нелинейны характер нагрузки усложняет предсказание нагрузки. Представлены модели используемые для предсказания нагрузки в интернете применяя ARIMA и нейронные сети. Рассматривается точность прогнозирование модели ARIMA и нейронные сети. Ил. 11, библи. 18 (на английском языке; рефераты на английском, русском и литовском яз.).

G. Rutka. Tinklo apkrautumo prognozavimas naudojant ARIMA ir neuroninius tinklus // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2008. – Nr. 4(84). – P. 47–52.

Tinklo apkrautumo pranašavimas įtakoja tinklo projektavimo procesą, valdymą, kontrolę ir optimizavimą. Nenuspėjama ir apkrovimo tendencija, todėl tikslų pranašavimą labiau apsunkina. Čia pristatyti modeliai, panaudoti tinklo apkrautumi prognozuoti, naudojant ARIMA ir neuroninius tinklus. Atlikti bandymai su prognozuojamais modeliais įvertinant jų tikslumą naudojant ARIMA ir neuroninių tinklų modelius. Il. 11, bibl. 18 (anglų k.; santraukos anglų, rusų ir lietuvių k.).