Topology Aware Internet Traffic Forecasting using Neural Networks

Paulo Cortez[1], Miguel Rio[2], Pedro Sousa[3], and Miguel Rocha[3]

[1] Department of Information Systems/R&D Algoritmi Centre, University of Minho, 4800-058 Guimarães, Portugal,
pcortez@dsi.uminho.pt
WWW home page: http://www.dsi.uminho.pt/~pcortez
[2] Department of Electronic and Electrical Engineering, University College London, Torrington Place, WC1E 7JE, London, UK,
m.rio@ee.ucl.ac.uk
[3] Department of Informatics, University of Minho, 4710-059 Braga, Portugal,
{pns, mrocha }@di.uminho.pt

Abstract. Forecasting Internet traffic is receiving an increasing attention from the computer networks domain. Indeed, by improving this task efficient traffic engineering and anomaly detection tools can be developed, leading to economic gains due to better resource management. This paper presents a Neural Network (NN) approach to predict TCP/IP traffic for all links of a backbone network, using both univariate and multivariate strategies. The former uses only past values of the forecasted link, while the latter is based on the neighbor links of the backbone topology. Several experiments were held by considering real-world data from the UK education and research network. Also, different time scales (e.g. every ten minutes and hourly) were analyzed. Overall, the proposed NN approach outperformed other forecasting methods (e.g. Holt-Winters).

Keywords: Link Mining, Multilayer Perceptrons, Multivariate Time Series, Network Monitoring, Traffic Engineering.

1 Introduction

Nowadays, more and more applications are migrating into TCP/IP networks (e.g. VoIP, IPTV). Hence, it is important to develop techniques to better understand and forecast the behavior of these networks. In effect, TCP/IP traffic prediction is gaining more attention from the computer networks community [18, 12, 1, 2]. By improving this task, network providers can optimize resources, allowing a better quality of service. Also, traffic forecasting can help to detect anomalies (e.g. security attacks, viruses or an irregular amount of SPAM) by comparing the real traffic with the forecasts [8, 7].

Often, TCP/IP traffic prediction is done intuitively by network administrators, with the help of marketing information on the future number of costumers and their behaviors [12]. Yet, this may not be suited for serious day-to-day network administration. Developments from the areas of Operational Research and Computer Science as lead to solid forecasting methods that replaced intuition based ones. In particular, the field of Time Series Forecasting (TSF), deals with the prediction of a chronologically ordered variable, where the goal is to model a complex system as a black-box, predicting its behavior based in historical data [10]. The TSF approaches can be divided into univariate and multivariate, depending if one or more variables are used. Multivariate methods are likely to produce better results, provided that the variables are correlated [14].

Several TSF methods have been proposed, such as the Holt-Winters [10] and Neural Networks (NN) [9, 16, 2]. Holt-Winters was developed for series with trended and seasonal factors and more recently a double seasonal version has been proposed [17]. In contrast with the conventional TSF methods (e.g. Holt-Winters), NNs can predict nonlinear series. In the past, several studies have proved the predictability of network traffic by using similar methods. For instance, the Holt-Winters was used in [8, 6] and NNs have also been proposed [18, 7, 2].

Recently, there has been an increasing interest in Link Mining, which aims at the discovery of useful patterns in graph structured datasets [4]. For a given goal (e.g. prediction), the idea is to use models that learn from data extracted from correlated links. The Internet backbone, which is made up of core routers that transport data through countries or continents, is a fertile ground for Link Mining.

This work will use recent real-world data from the United Kingdom Education and Research Network (UKERNA) backbone. NNs will be used to predict the traffic for all 18 links of this backbone network, under univariate and multivariate approaches. The former is based on the previous traffic from the current link, while a heuristic rule is proposed for the latter, where the NNs are fed with data from current and the direct neighbor links. Furthermore, the predictions will be analyzed at different time scales (e.g. every ten minutes, hourly) and compared with other methods (e.g. Holt-Winters).

2 Time Series Data

The data collection was based in the Simple Network Management Protocol (SNMP), which quantifies the traffic passing through every network interface with reasonable accuracy [15]. SNMP is widely deployed by every Internet Service Provider/network and the collection of this data does not induce any extra traffic on the network. This work will analyze traffic data (in Mbit/s) from all links of the UK academic network backbone (UKERNA). This backbone contains a total of eight core routers and 18 links. Figure 1 plots the respective direct graph. The data was recorded into two datasets (every 10 minutes and every hour), between 12 AM of 14th June 2006 and 12 AM of 23th July 2006. The obtained multivariate series included 2 missing periods for the 10 minute data, which were replaced with a linear interpolation. The missing values are explained by the fact that the SNMP scripts are not 100% reliable, since the SNMP messages may be lost or the router may not reply on time. Yet, this occurs very rarely and it is statistically insignificant. The hourly multivariate series contains 936 observations for each link, while the 10 minute data encompasses a total of 5613 time records.

As an example, the hourly traffic of two neighbor links, London-Cosham (LC) and Cosham-Bristol (CB), is plotted in Figure 2. In the first case (LC), it is clear the influence of two seasonal components due to the the intraday and intraweek cycles. The weekly pattern is less visible in the second example (CB).

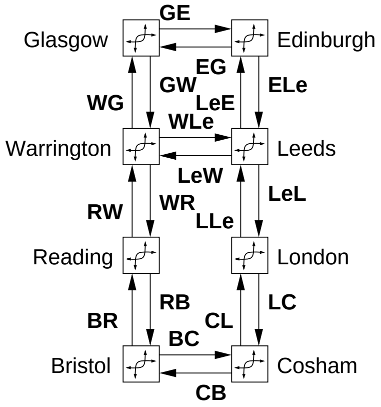

Fig. 1. The schematic of the UK academic Internet backbone

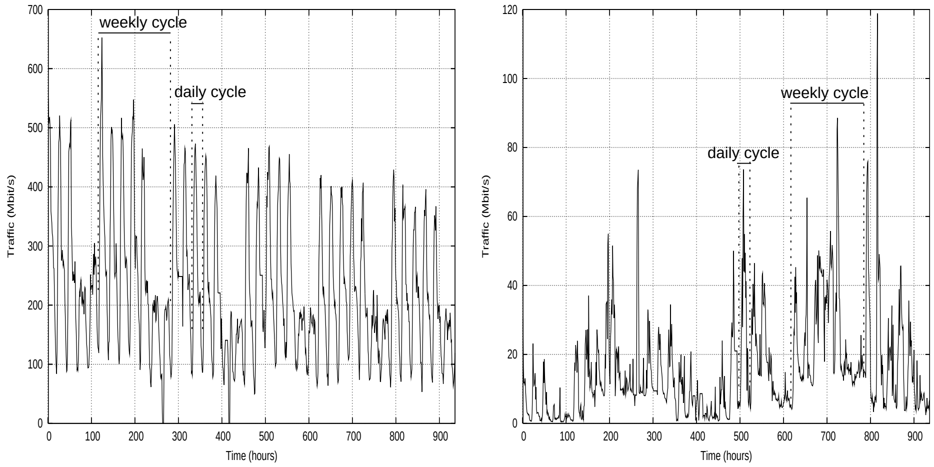

Fig. 2. The hourly IP traffic rate for the London-Cosham (left) and Cosham-Bristol (right) links

3 Forecasting Methods

A Time Series Forecasting (TSF) model assumes that past patterns will occur in the future. Let $y_t = (y_{1t}, \ldots, y_{kt})$ denote a multivariate series, where y_{ij} is the jth chronological observation on variable i and k is the number of distinct time variables ($r = 1$ when a univariate setting is used). Then [14]:

$$\begin{array}{l} \widehat{y}_{pt} = f(y_{1t-1}, \ldots, y_{1t-n}, \ldots, y_{rt-1}, \ldots, y_{rt-n}) \\ e_t = y_{p,t} - \widehat{y}_{pt} \end{array} \qquad (1)$$

where $\widehat{y}_{pt}$ denotes the estimated value for the pth variable and time t; f the underlying function of the forecasting model; and e_t is the error (or residual).

The overall performance of a model is evaluated by an global accuracy measure, namely the Root Mean Squared Error (RMSE) and Relative RMSE (RRMSE), given in the form [19]:

$$\begin{array}{l} RMSE = \sqrt{\sum_{i=P+1}^{P+N} e_i^2 / N} \\ RRMSE = RMSE/RMSE_{\overline{y}_{pt}} \times 100\,(\%) \end{array} \tag{2}$$

where P is the present time; N is the number of forecasts; and $RMSE_{\overline{y}_{pt}}$ is the $RMSE$ given by the simple mean prediction. The last metric ($RRMSE$) will be adopted in this work, since it has the advantage of being scale independent, where 100% denotes an error similar to the mean predictor ($\overline{y}_{pt}$).

Due to the temporal nature of this domain, a sequential holdout will be adopted for the forecasting evaluation. Hence, the first $TR = 2/3$ of the series will be used to adjust (train) the forecasting models and the remaining last $1/3$ to evaluate (test) the forecasting accuracies. Also, an internal holdout procedure will be used for model selection, where the training data will be further divided into training ($2/3$ of TR) and validation sets (1/3 of TR). The former will be used to fit the candidate models, while the latter will be used to select the models with the lowest error ($RMSE$). After this selection phase, the final model is readjusted using all training data.

3.1 Neural Networks

Neural Networks (NNs) are innate candidates for forecasting due to their nonlinear and noise tolerance capabilities. Indeed, the use of NNs for TSF began in the late eighties with encouraging results and the field has been growing since [9, 16, 18, 2].

The multilayer perceptron is the most popular NN used within the forecasting domain [9, 16, 18]. When adopting this architecture, TSF is achieved by using a *sliding time window*, in a combination also named *Time Lagged Feedforward Network* in the literature. A sliding window is defined by the set of time lags used to build a forecast. For instance, given the univariate time series 1,2,3,4,5,6 and sliding window $\{1, 2, 4\}$, the following training examples can be built: $1, 3, 4 \rightarrow 5$ and $2, 4, 5 \rightarrow 6$. In a multivariate setting, k sliding windows are used: $\{L_{11}, \ldots, L_{1W_1}\}, \ldots, \{L_{k1}, \ldots, L_{kW_k}\}$, where L_{ij} denotes a time lag for the ith variable.

In this work, a fully connected multilayer network with one hidden layer of H hidden nodes and bias connections will be adopted (Figure 3). The logistic activation function is applied on the hidden nodes and the output node uses a linear function [5]. The overall model is given in the form:

$$\widehat{y}_{p,t} = w_{o,0} + \sum_{i=I+1}^{I+H} f\left(\sum_{s=1}^{k} \sum_{r=1}^{W_s} y_{st-L_{sr}} w_{i,j}\right) \tag{3}$$

where $w_{d,s}$ is the weight from node s to d; (if $d = 0$ then it is a bias connection); $j \in \{1, \ldots, I\}$ is an input node; o is the output node; and f the logistic function ($\frac{1}{1+e^{-x}}$).

Before training, all variables are scaled with a zero mean and one standard deviation. Then, the initial NN weights are randomly set within $[-0.7, +0.7]$. Next, the training algorithm is applied and stopped when the error slope approaches zero or after a maximum of E epochs. Since the NN cost function is nonconvex (with multiple minima), NR runs are applied to each neural setup, being selected the NN with the lowest error [5]. After training, the NN outputs are rescaled to the original domain.

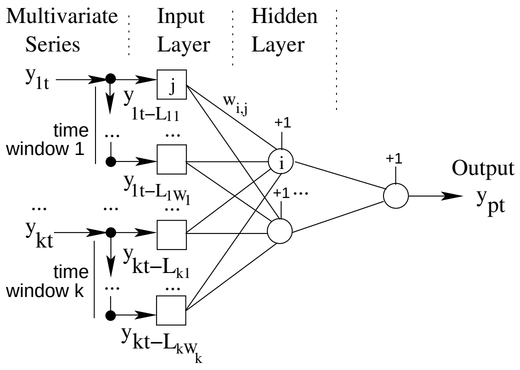

Fig. 3. The multilayer perceptron architecture for multivariate time series forecasting

Under this setting, the NN performance will depend on the number of hidden nodes (H), the selection of the k variables used in the multivariate model and the time window used for each variable. All these parameters can have a crucial effect in the forecasting performance. Feeding a NN with uncorrelated variables or time lags with affect the learning process due to the increase of noise. In addition, a network with few hidden nodes will have limited learning capabilities, while an excess of hidden nodes will lead to overfitting or generalization loss. Since the search space for these parameters is high, an heuristic procedure will be used for the model selection step (see Section 4).

3.2 Naive and Holt-Winters Methods

Two TSF methods will be used as a baseline comparison with the proposed NNs. The most common naive forecasting method is to predict the future as the present value. This setup will be termed NV1. Other possibility is to use a seasonal variant, where the forecast will be given by the observed value for the same period related to the previous daily (NVD) or weekly (NVW) cycles [17].

The *Holt-Winters* [10] is another important univariate forecasting technique from the family of Exponential Smoothing methods. The predictive model is based on some underlying patterns such as a trend or a seasonal cycle (K_1), which are distinguished from random noise by averaging the historical values. Its popularity is due to advantages such as the simplicity of use, the reduced computational demand and the accuracy of the forecasts, specially with seasonal series. More recently, this method has been extended to encompass two seasonal cycles (K_1 and K_2) [17].

4 Experiments and Results

All forecasting methods were implemented in the **R** environment, an open source and high-level programming language for data analysis [13]. The NNs were trained with the $E = 100$ epochs of the BFGS algorithm [11], from the family of quasi-Newton methods and available at the **nnet R** function, while the number of NN runs was set to $NR = 3$. The number of tested hidden nodes (H) was within the range {0,2,4,6} [2].

Two configurations are used for the variable selection. The first is the simple univariate model. The second multivariate setup will use topology information from the backbone (Figure 1), where the predicted traffic is based on the past values of the current link (p) plus the previous traffic observed in the closest neighbor links that are expected to influence the predicted link (p). For instance, the link Londom-Cosham (LC) presents only one direct neighbor (LeL)[4], while the connection Leeds-London (LeL) contains two (WLe and ELe). Several sliding windows were heuristically set for each series based on their characteristics. It should be noted that in previous univariate IP traffic forecasting work [2], this sliding window setup obtained high quality results. For the single variable model, the tested time window (W_p) was within the range {1}, {1,2,3,4,5,6}, {1,2,3,72,73} and {1,2,3,144,145} (10 minute scale); and {1}, {1,24,25}, {1,168,169} and {1,24,25,168,169} (hourly data). Under the multivariate setting, similar sliding windows were used for the target variable (p). Regarding the other variables, the same window (W_n) will be applied to all neighbor links. For these links, the tested windows were {1}, {1,2,3,4,5,6} and {1,2,3,144,145} (10 minute data) and {1}, {1,24,25}, {1,168,169} (1 hour scale).

The forecasting neural models appear in Table 1. Interestingly, the multivariate neighborhood heuristic is the best option to forecast 11 (10 minute series) and 10 (hourly data) of the 18 links. In general, the multivariate model uses a similar or even higher number of time lags for the predicted variable p than the neighbor links (the exception is link CB for the hourly series). Moreover, only seven models denote nonlinearity ($H > 0$): BC, ELe, LeL and RW, for the 10 minute data; and WG, GW and GE for the hourly series. These results confirm the notion that real/short time Internet traffic can be modeled by small networks.

The three naive methods (NV1, NVD and NVW) were tested on the model selection step. For all cases, the best model was NV1, which will be adopted as the naive benchmark. Turning to the Holt-Winters (HW) models, the internal parameters were optimized using a 0.05 grid search for the best training error ($RMSE$), which is a common procedure within the forecasting field. For the hourly series, non seasonal, seasonal ($K_1 = 24$ or $K1 = 168$) and double seasonal variants ($K_1 = 24$ and $K_2 = 168$) were tested. Within the selection stage, the weekly seasonal variant ($K1 = 168$) presented the lowest errors. The exception were the links BC, CB, LW (non seasonal model) and LC (daily seasonal with $K1 = 24$). Regarding the 10 minute series, only non seasonal and daily seasonal ($K_1 = 144$) models were tested, since trended effects should be higher than seasonal components at this scale. In effect, the non seasonal version was the best option for all links except BC, CB and GE.

The forecasts with the selected models were performed on the test sets (with 1871 values for the 10 minute series and 312 elements for the hourly data). Table 2 shows the forecasting errors ($RRMSE$) for each method. Thirty runs were applied for the NNs and the results are shown as the mean with the respective 95% t-student confidence intervals. The type of forecasting model is also shown for the NN method: univariate

[4] The link CL is not considered, since its origin (Cosham) matches the LC connection destination.

Table 1. The best neural forecasting models

Link	**Scale**					
	10 minutes			**1 hour**		
	H	W_p	W_n	H	W_p	W_n
BR	6	{1,2,3,144,145}	–	0	{1,24,25,168,169}	–
BC	0	{1,2,3,4,5,6}	{1}	0	{1,24,25}	–
LC	0	{1,2,3,4,5,6}	{1,2,3,4,5,6}	0	{1,24,25,168,169}	{1,24,25}
LLe	0	{1,2,3,4,5,6}	–	0	{1,24,25,168,169}	–
WR	0	{1,2,3,4,5,6}	–	0	{1,24,25,168,169}	–
WL	0	{1,2,3,4,5,6}	–	0	{1,24,25,168,169}	–
WG	0	{1,2,3,144,145}	{1,2,3,144,145}	2	{1,24,25}	–
ELe	4	{1,2,3,72,73}	{1,2,3,4,5,6}	0	{1,24,25,168,169}	{1,168,169}
EG	0	{1,2,3,4,5,6}	{1,2,3,4,5,6}	0	{1,168,169}	{1,168,169}
RB	0	{1,2,3,144,145}	–	0	{1,24,25,168,169}	{1,24,25}
CB	0	{1,2,3,4,5,6}	{1,2,3,144,145}	0	{1}	{1,168,169}
CL	0	{1,2,3,4,5,6}	{1,2,3,144,145}	0	{1,168,169}	{1,24,25}
LeL	4	{1,2,3,144,145}	–	0	{1,24,25}	{1,168,169}
RW	4	{1,2,3,144,145}	{1,2,3,144,145}	0	{1,24,25,168,169}	{1}
LW	0	{1,2,3,4,5,6}	{1}	0	{1,24,25}	–
GW	0	{1,2,3,4,5,6}	{1}	2	{1,168,169}	–
LeE	0	{1,2,3,144,145}	–	0	{1,24,25,168,169}	{1}
GE	0	{1}	{1}	2	{1,24,25,168,169}	{1,24,25}

(U) or Multivariate (M). Finally, the global performance is presented in the last row in terms of the mean error.

The analysis will start with the 10 minute data. As expected, the naive method gets the worst performance. The NV is only the best option for the last link (GE), presenting the highest mean error. The HW comes in second place. When compared with NV, the mean error decreases 2 percentage points. Moreover, it is the best method for 3 series (WL, WG and GE). Nevertheless, the proposed approach (NN) is clearly the best solution, outperforming (with statistical significance) other methods in 15 (of 18) links and presenting the lowest mean error. Also, it should be noted that the multivariate heuristic (M) is highly relevant, exceeding the NV/HW models in 10 of 11 cases.

The hourly scale is harder to predict, since the $RRMSE$ values are around five times higher than those obtained for the 10 minute series. NV is still the worst strategy, although it is now the best choice for 3 links (EG, CL and GE). Next comes the HW, which presents the lowest errors in 4 cases (WR, LeL, RW and GW). Again, the NNs obtain the best forecasts, presenting an overall performance 1.3/5.0 percentage points below the HW/NV errors and being the best option for 11 links. At this time scale, the multivariate model outperforms the other methods in half the cases (5 of 10).

For demonstrative purposes, Figure 4 presents the traffic forecasts for the first day of the 10 minute RW data (top) and the first week of the hourly LC series (bottom). In both cases, a high quality fit is achieved by the NN forecasts, which are close to the real values. Another relevant issue is related with the computational complexity. The

Table 2. Comparison of the forecasting models ($RRMSE$ values, in percentage)

Link	Scale					
	10 minutes			1 hour		
	NV	**HW**	**NN**	**NV**	**HW**	**NN**
BR	7.1	4.8	**4.5**±0.0 U	35.3	25.2	**24.8**±0.0 U
BC	19.1	18.0	**16.2**±0.0 M	69.8	68.7	**63.2**±0.0 U
LC	7.5	5.4	**5.2**±0.0 M	37.1	27.8	**22.4**±0.0 M
LLe	6.7	4.0	**3.8**±0.0 U	35.6	25.2	**21.3**±0.0 U
WR	8.9	6.8	**6.4**±0.1 U	40.5	**34.0**	34.1±0.0 U
WL	12.9	**10.5**	**10.5**±0.0 U	59.1	69.0	**58.4**±0.0 U
WG	7.1	**4.6**	**4.6**±0.0 M	36.0	25.1	**24.8**±0.1 U
ELe	9.6	8.5	**8.3**±0.0 M	44.3	44.3	**40.6**±0.0 M
EG	13.1	10.8	**10.2**±0.0 M	**56.3**	67.5	57.5±0.0 M
RB	6.6	2.9	**2.6**±0.0 U	36.5	15.0	**14.6**±0.0 M
CB	13.4	11.1	**10.2**±0.0 M	57.8	58.0	**53.8**±0.0 M
CL	10.9	9.7	**8.8**±0.0 M	**42.3**	45.4	57.2±0.0 M
LeL	7.4	4.6	**4.4**±0.0 U	37.6	**31.7**	34.9±0.0 M
RW	6.9	4.0	**3.7**±0.0 M	36.5	**19.0**	19.5±0.0 M
LW	21.6	21.5	**18.8**±0.0 M	87.0	87.0	**80.8**±0.0 U
GW	9.3	7.4	**6.7**±0.0 M	41.9	**39.4**	41.7±0.1 U
LeE	7.5	4.6	**4.3**±0.0 U	38.7	30.1	**29.2**±0.0 M
GE	**11.5**	**11.5**	11.8±0.0 M	**54.8**	80.8	90.7±1.6 M
Mean	10.4	8.4	**7.8**	47.8	44.1	**42.8**

proposed solution is very fast and can be used in real-time. For instance, with a Pentium Dual Core 3GHz processor, the thirty runs of the NN training and testing required only 15.8 (10 minute RW link data) and 3 (the hourly LC series) seconds.

5 Conclusions

In this work a Neural Network (NN) is proposed to forecast the Internet traffic for all 18 links of the UK academic network backbone. In particular, univariate and multivariate strategies were tested. The former used past data from the predicted link, while the latter used topology information, i.e. the direct neighbor links were also fed into the predictive model. Recent data, collected from the United Kingdom Education and Research Network (UKERNA), was analyzed using two forecasting types (or scales): real-time (every 10 minutes) and short-term (hourly values). Also, a comparison was made with two baseline benchmarks, the naive (NV) and Holt-Winters (HW) methods.

The NN multivariate strategy outperformed the univariate approach in 61% (real-time forecasts) and 56% (short-term predictions) of the links considered. Overall, the NN results are promising, with a global Relative Root Mean Square Error (RRMSE) of 7.8% (10 minute series) and 42.8% (hourly data). Indeed, the proposed NN solution produces the best forecasts, surpassing other methods in 83% (10 minute scale) and 61% (hourly series) of the cases. Moreover, the NNs are very fast and can be applied

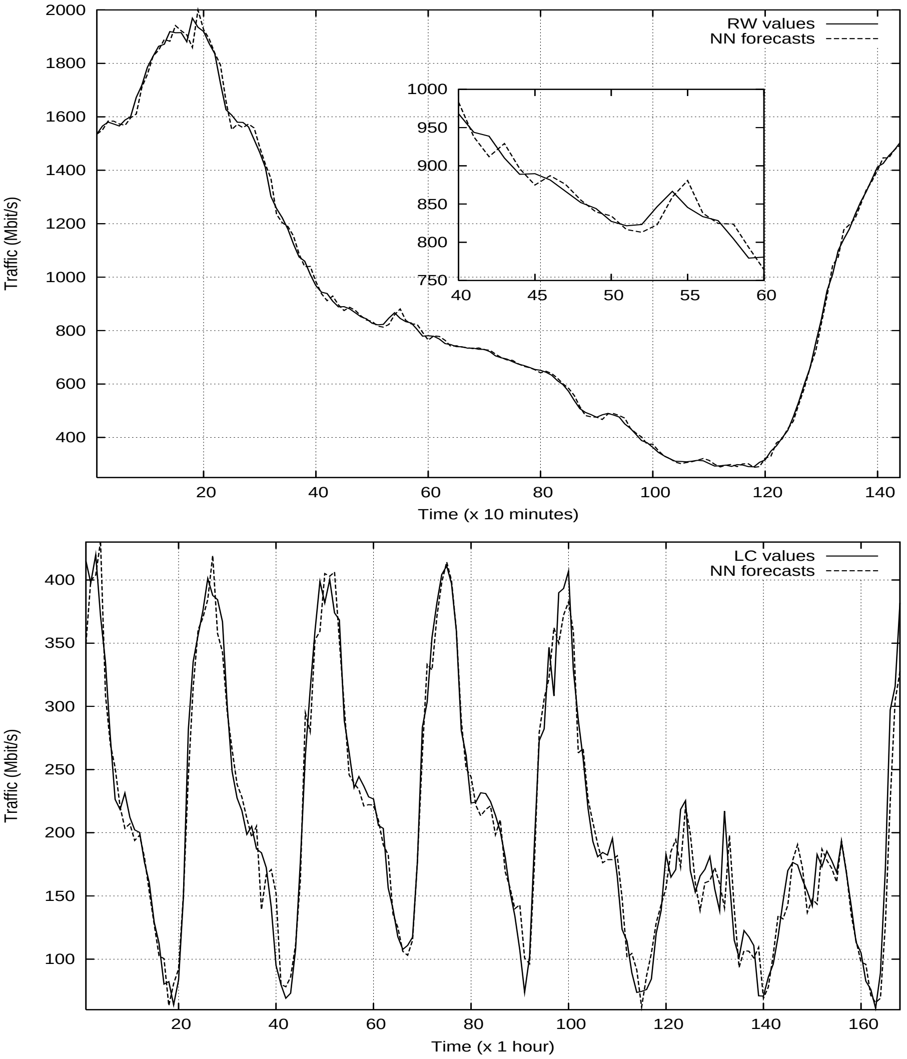

Fig. 4. Neural forecasts for the first day RW (top) and first week LC (bottom) links

in real-time. Therefore, the proposed approach opens room for producing better traffic engineering tools and methods to detect anomalies in the traffic patterns. This can be achieved with minimal use of computation resources and without producing any extra traffic in the network, since a passive monitoring system was adopted.

In future work, the comparison will be extended to other forecasting techniques (e.g. ARMA models [3]); the proposed approach will be applyed to traffic demands of specific Internet applications (e.g. VoIP); and distinct forecasting horizons will be tested, i.e. from one to several lookaheads. For this last option, several models could be used, where each NN is trained for a specific n-ahead forecast. As an alternative, the one step-ahead forecasts could be used iteratively as inputs. Under a multivariate setting, this would require the simultaneous forecasting of the predicted and direct neighbor NNs.

References

1. R. Babiarz and J. Bedo. Internet Traffic Mid-term Forecasting: A Pragmatic Approach Using Statistical Analysis Tools. *Lecture Notes on Computer Science*, 3976:111–121, 2006.
2. P. Cortez, M. Rio, M. Rocha, and P. Sousa. Internet Traffic Forecasting using Neural Networks. In *Proceedings of the IEEE 2006 International Joint Conference on Neural Networks*, pages 4942–4949, Vancouver, Canada, 2006.
3. P. Cortez, M. Rocha, and J. Neves. Evolving Time Series Forecasting ARMA Models. *Journal of Heuristics*, 10(4):415–429, 2004.
4. L. Getoor and C.P. Diehl. Link Mining: A Survey. *SIGKDD Explorations*, 7(2):3–13, 2005.
5. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, NY, USA, 2001.
6. Q. He, C. Dovrolis, and M. Ammar. On the Predictability of Large Transfer TCP Throughput. In *Proc. of SIGCOMM'05*, Philadelphia, USA, August 2005. ACM.
7. J. Jiang and S. Papavassiliou. Detecting Network Attacks in the Internet via Statistical Network Traffic Normality Prediction. *Journal of Network and Systems Management*, 12:51–72, 2004.
8. B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based Change Detection: Methods, Evaluation, and Applications. In *Proc. of Internet Measurment Conference (IMC'03)*, Miami, USA, October 2003. ACM.
9. A. Lapedes and R. Farber. Non-Linear Signal Processing Using Neural Networks: Prediction and System Modelling. Tech. Rep. LA-UR-87-2662, Los Alamos National Laboratory, USA, 1987.
10. S. Makridakis, S. Weelwright, and R. Hyndman. *Forecasting: Methods and Applications*. John Wiley & Sons, New York, USA, 1998.
11. M. Moller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.
12. K. Papagiannaki, N. Taft, Z. Zhang, and C. Diot. Long-Term Forecasting of Internet Backbone Traffic. *IEEE Trans. on Neural Networks*, 16(5):1110–1124, September 2005.
13. R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-00-3, http://www.R-project.org, 2006.
14. G. Reinsel. *Elements of Multivariate Time Series Analysis*. Springer, San Francisco, CA, second edition, 2003.
15. W. Stallings. *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*. Addison Wesley, 1999.
16. Z. Tang and F. Fishwick. Feed-forward Neural Nets as Models for Time Series Forecasting. *ORSA Journal of Computing*, 5(4):374–386, 1993.
17. J. Taylor, L. Menezes, and P. McSharry. A Comparison of Univariate Methods for Forecasting Electricity Demand Up to a Day Ahead. *Int. Journal of Forecasting*, 21(1):1–16, 2006.
18. H. Tong, C. Li, and J. He. Boosting Feed-Forward Neural Network for Internet Traffic Prediction. In *Proc. of the IEEE 3rd Int. Conf. on Machine Learning and Cybernetics*, pages 3129–3134, Shanghai, China, August 2004.
19. I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, CA, second edition, 2005.