Internet Traffic Forecasting using Neural Networks

Paulo Cortez, Miguel Rio, Miguel Rocha, and Pedro Sousa

Abstract— The forecast of Internet traffic is an important is-sue that has received few attention from the computer networks field. By improving this task, efficient traffic engineering and anomaly detection tools can be created, resulting in economic gains from better resource management. This paper presents a *Neural Network Ensemble (NNE)* for the prediction of TCP/IP traffic using a *Time Series Forecasting (TSF)* point of view. Several experiments were devised by considering real-world data from two large Internet Service Providers. In addition, different time scales (e.g. every five minutes and hourly) and forecasting horizons were analyzed. Overall, the *NNE* approach is competitive when compared with other *TSF* methods (e.g. Holt-Winters and ARIMA).

I. INTRODUCTION

TCP/IP traffic forecasting is a crucial task for any medium/large network provider that has received little atten-tion from the computer networks community [1]. However, there is a significant amount of tasks that have to be done by network administrators that would gain in using forecasting methods.

For example, several *Internet Service Providers (ISP)* use *Multiprotocol Label Switching (MPLS)* to implement traffic engineering by establishing a full mesh of MPLS virtual circuits between all pairs of routers in the network [2]. In theory, this would optimize the bandwidth resources in the network, allowing for better quality of service overall. Even if MPLS is not used, traditional allocation of the routing protocol weights can be done much more efficiently with the knowledge of the future demand traffic matrix between all end-points of the network. Hence, it is easy to envision automatic traffic engineering tools that adapt to future conditions of the network based on accurate traffic forecasting algorithms. Although not a direct concern of this work, long-term traffic forecasting can also be used for network planning and provision and as an input to the business/management departments.

Traffic forecasting can also help to detect anomalies in the networks [3][4]. Security attacks like Denial-of-Service, viruses, or even an irregular amount of SPAM can in theory be detected by comparing the real traffic with the values predicted by forecasting algorithms. This can result in an earlier detection of problems which will conduct to a more reliable service. It is also easy to envision automatic traffic

engineering fall-back procedures that react instantaneously to anomalies and reduce the probability of service disruption.

Nowadays, this task is often done intuitively by experi-enced network administrators with the help of marketing information on the future number of costumers and their usual behaviors [1]. This produces only a rough idea of what the traffic will look like with little use for serious day-to-day network administration. On the other hand, contri-butions from the areas of Operational Research, Statistics, and Computer Science as lead to solid forecasting methods that replaced intuition based ones. In particular, the field of *Time Series Forecasting (TSF)*, also termed *univariate forecasting*, deals with the prediction of a chronologically ordered variable [5][6]. The goal of TSF is to model a complex system as a black-box, predicting its behavior based in historical data, and not how it works.

Due to its importance, several TSF methods have been pro-posed, such as the Holt-Winters [7], the ARIMA methodol-ogy [8] and *Neural Networks (NN)* [9][10][11]. Holt-Winters was devised for series with trended and seasonal factors. More recently, a double seasonal version has been proposed [12]. The ARIMA is a more complex approach, requiring steps such as model identification, estimation and validation. Each ARIMA model is based on a linear combination of past values and/or errors. NNs are connectionist models inspired in the behavior of central nervous system, and in contrast with the previous methods, they can predict nonlinear series. In the past, several studies have proved the predictability of network traffic by using similar methods, such as Holt-Winters [3][13] and ARIMA [3][14][1]. Following the evi-dence of nonlinear network traffic [15], NNs have also been proposed [16][4].

Our approach uses already available information provided by *Simple Network Management Protocol (SNMP)* that quan-tifies the traffic passing through every network interface with reasonable accuracy [17]. SNMP is widely deployed by every ISP/network so the collection of this data does not induce any extra traffic on the network.

Based on recent data provided by two distinct ISPs, several experiments will be carried out in order to provide network engineers with useful feedback regarding the effectiveness of such techniques. The main contributions of this work are: i) Internet traffic is predicted using a pure TSF approach (i.e., only past values are used as inputs), in opposition to [1][3][13] and allowing its use in wider contexts; ii) in contrast with previous studies [1][15][16][4], the predictions are analyzed at different time scales (e.g. five minutes and hourly), and considering distinct lookahead horizons; iii) sev-eral forecasting methods are tested and compared, including the Holt-Winters (both traditional and recent double seasonal

Paulo Cortez is with the Department of Information Systems, University of Minho, 4800-058 Guimarães, PORTUGAL (phone: +351-253-510313; fax: +351-253-510300; email: pcortez@dsi.uminho.pt).

Miguel Rio is with the Department of Electronic and Electrical Engineer-ing, University College London, Torrington Place, WC1E 7JE, London, UK (email: m.rio@ee.ucl.ac.uk).

Miguel Rocha and Pedro Sousa are with the Department Infor-matics, University of Minho, 4710-059 Braga, PORTUGAL (email: {mrocha,pns}@di.uminho.pt).

versions), the ARIMA methodology and a NN ensemble based approach. As a result, the research work presented in this paper is expected to be an effective contribution for the development of intelligent network traffic forecasting engines.

The paper is organized as follows. Firstly, the Internet traffic data is presented and analyzed in Section II. The adopted forecasting methods are given in Section III, while the results are presented and discussed in the Section IV. Finally, closing conclusions are drawn (Section V).

II. TIME SERIES ANALYSIS

A *time series* is a collection of time ordered observations $(y_1, y_2, \ldots, y_t)$, each one being recorded at a specific time t (period), appearing in a wide set of domains such as Finance, Production and Control [6]. A time series model ($\widehat{y}_t$) assumes that past patterns will occur in the future. Another relevant concept is the *horizon* or *lead time* (h), which is defined by the time in advance that a forecast is issued.

The overall performance of a forecasting model is evaluated by an accuracy measure, namely the *Sum Squared Error (SSE)* and *Mean Absolute Percentage Error (MAPE)*, given in the form [5]:

$$
\begin{aligned}
e_t &= y_t - \widehat{y}_{t,t-h} \\
SSE_h &= \sum_{i=P+1}^{P+N} e_i^2 \\
MAPE_h &= \sum_{i=P+1}^{P+N} \frac{|y_i - \widehat{y}_{i,i-h}|}{y_i \times N} \times 100\%
\end{aligned}
\tag{1}
$$

where e_t denotes the forecasting error at time t; y_t the desired value; $\widehat{y}_{t,p}$ the predicted value for period t and computed at period p; P is the present time and N the number of forecasts. [18]. The $MAPE$ is a common metric in forecasting applications, such as electricity demand [18], and it measures the proportionality between the forecasting error and the actual value. This metric will be adopted in this work, since it is easier to interpret by the TCP/IP network administrators. In addition, it presents the advantage of being scale independent. It should be noted that the SSE values were also calculated but the results will not be reported here since the relative forecasting performances are similar to the ones obtained by the $MAPE$.

This work will analyze traffic data (in bits) from two different ISPs, denoted here as **A** and **B**. The **A** dataset belongs to a private ISP with centres in 11 European cities. The data corresponds to a transatlantic link and was collected from 6:57 AM on 7th June 2005. Dataset **B** comes from UKERNA[1] and represents aggregated traffic in the Janet[2] (the UK academic network) backbone. It was collected between 19th November 2004, at 9:30 AM, and 27th January 2005, at 11:11 AM. The **A** time series was registered every 30 seconds, while the **B** data was recorded at a five minute period. The first series (**A**) included 8 missing values, which were replaced by using a regression imputation (e.g. linear interpolation) [19]. The missing data is 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.

Within the forecasting community, the following forecasting types can be defined, depending on the time scale [10]:

- *Real-time*, which concerns samples not exceeding a few minutes and requires an on-line forecasting system;
- *Short-term*, from one to several hours, crucial for optimal control or detection of abnormal situations;
- *Middle-term*, typically from one to several days, used to plan resources; and
- *Long-term*, often issued several months/years in advance and needed for strategic decisions, such as investments.

Due to the characteristics of the Internet traffic collected, this study will only consider the first two types. Therefore, two new time series were created for each ISP by aggregating the original values; i.e. summing all data samples within a given period of time. The selected time scales were (Figure 1): every five minutes (series **A5M** and **B5M**) and every hour (**A1H** and **B1H**). For each series, the first $2/3$ of the data will be used to create (train) the forecasting models and the remaining last $1/3$ to evaluate (test) the forecasting accuracies (Table I). Under this scheme, the number of forecasts is equal to $N = NT - h + 1$, where h is the lead time period and NT is the number of samples used for testing.

TABLE I
THE SCALE AND LENGTH OF INTERNET TRAFFIC TIME SERIES

Series	Time Scale	Train Size	Test Size	Total Size
A5M	5 minutes	9848	4924	14772
A1H	1 hour	821	410	1231
B5M	5 minutes	13259	6629	19888
B1H	1 hour	1105	552	1657

The autocorrelation coefficient, a statistical measure of the correlation between a series and itself, lagged of k periods, is a powerful tool for time series analysis [8]:

$$
r_k = \frac{\sum_{t=1}^{P-k}(y_t - \overline{y})(y_{t+k} - \overline{y})}{\sum_{t=1}^{P}(y_t - \overline{y})}
\tag{2}
$$

where $y_1, y_2, ..., y_P$ stands for the time series and $\overline{y}$ for the series' average. Autocorrelations are an efficient way to perform decomposition, in particular for the detection of seasonal components [5]. As an example, the autocorrelations for the **A** data are plotted (Figure 2), which shows two seasonal effects in the five minute and hourly data, due to the intraday ($K_1 = 288/24$) and intraweek cycles ($K_2 = 168$).

III. FORECASTING METHODS

In the following subsections, the adopted forecasting methods will be explained in detail, including a description of the model selection steps and implementation details.

A. Naive Benchmark Method

The most commonly used naive forecasting method is to predict the future as the present value. Yet, this method will
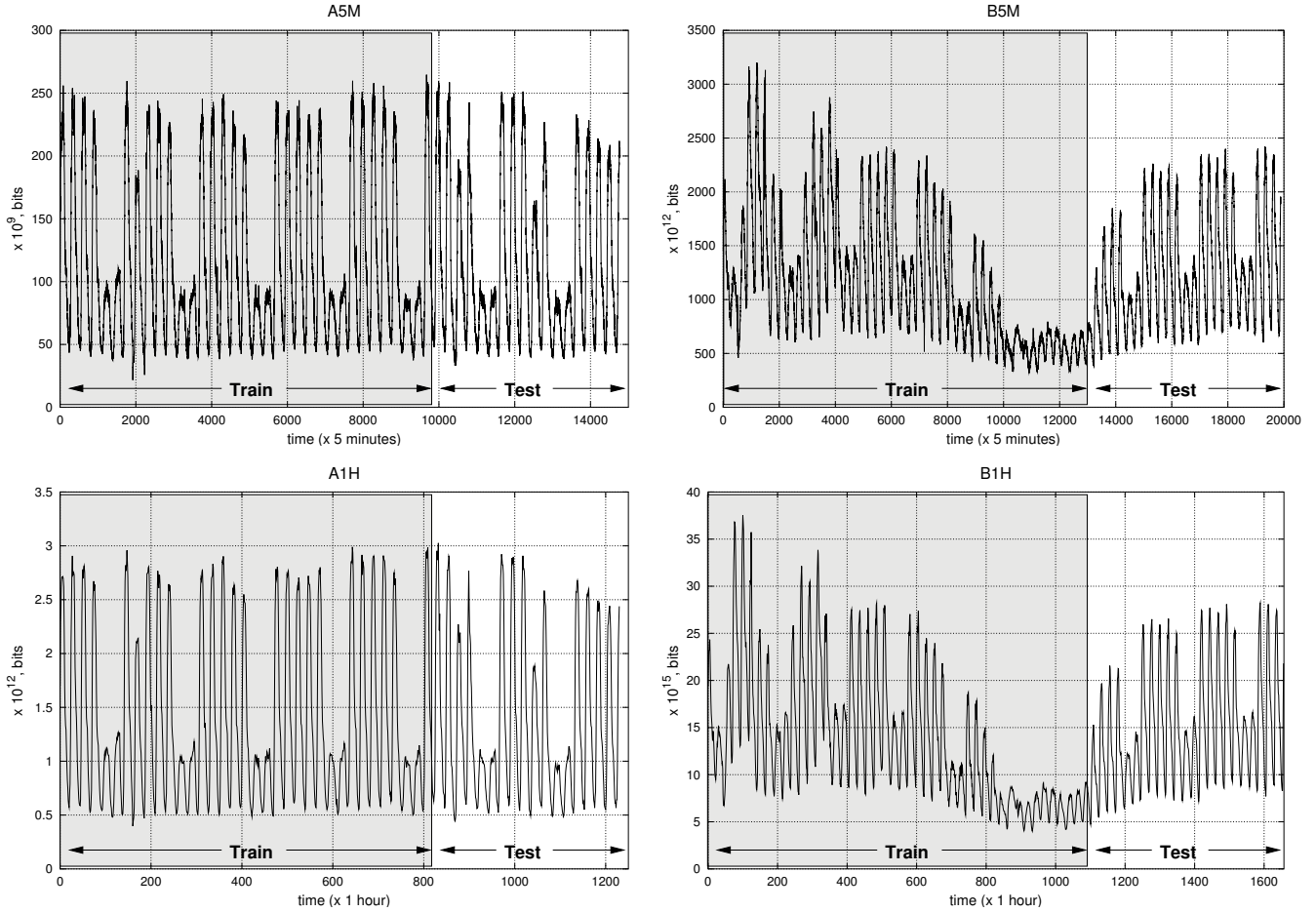
Fig. 1. The Internet traffic time series considered in this study (**A5M**, **B5M**, **A1H** and **B1H**)

perform poorly in seasonal data. Thus, a better and more natural alternative is to use a seasonal version, where a forecast will be given by the observed value for the same period related to the previous seasonal cycle [18]:

$$\widehat{y}_{t+h,t} = y_{t+h-K} \qquad (3)$$

where K is the seasonal period. In this work, K will be set to the weekly cycle. This *Naive* method, which can be easily adopted by the network administrators, will be used as a benchmark for the comparison with other forecasting approaches.

B. Holt-Winters

The *Holt-Winters* is an important forecasting technique from the family of Exponential Smoothing methods. The predictive model is based on some underlying patterns (e.g., trended and seasonable) that are distinguished from random noise by averaging the historical values [7]. 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. The general model is defined

by the equations [6]:

$$
\begin{aligned}
\text{Level} \quad & S_t = \alpha \frac{y_t}{D_{t-K_1}} + (1-\alpha)(S_{t-1} + T_{t-1}) \\
\text{Trend} \quad & T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1} \\
\text{Seasonality} \quad & D_t = \gamma \frac{y_t}{S_t} + (1-\gamma)D_{t-K_1} \\
& \widehat{y}_{t+h,t} = (S_t + hT_t) \times D_{t-K_1+h}
\end{aligned}
$$
(4)

where S_t, T_t and D_t stand for the level, trend and seasonal estimates, K_1 for the seasonal period, and α, β and γ for the model parameters. When there is no seasonal component, the γ is discarded and the D_{t-K_1+h} factor in the last equation is replaced by the unity.

More recently, this method has been extended to encompass two seasonal cycles, by using the expressions [12]:

$$
\begin{aligned}
\text{Level} \quad & S_t = \alpha \frac{y_t}{D_{t-K_1} W_{t-K_2}} + (1-\alpha)(S_{t-1} + T_{t-1}) \\
\text{Trend} \quad & T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1} \\
\text{Season. 1} \quad & D_t = \gamma \frac{y_t}{S_t W_{t-K_2}} + (1-\gamma)D_{t-K_1} \\
\text{Season. 2} \quad & W_t = \omega \frac{y_t}{S_t D_{t-K_1}} + (1-\omega)W_{t-K_2} \\
& \widehat{y}_{t+h,t} = (S_t + hT_t) \times D_{t-K_1+h} W_{t-K_2+h}
\end{aligned}
$$
(5)

where W_t is the second seasonal estimate, K_1 and K_2 are the first and second seasonal periods; and ω is the second seasonal parameter.
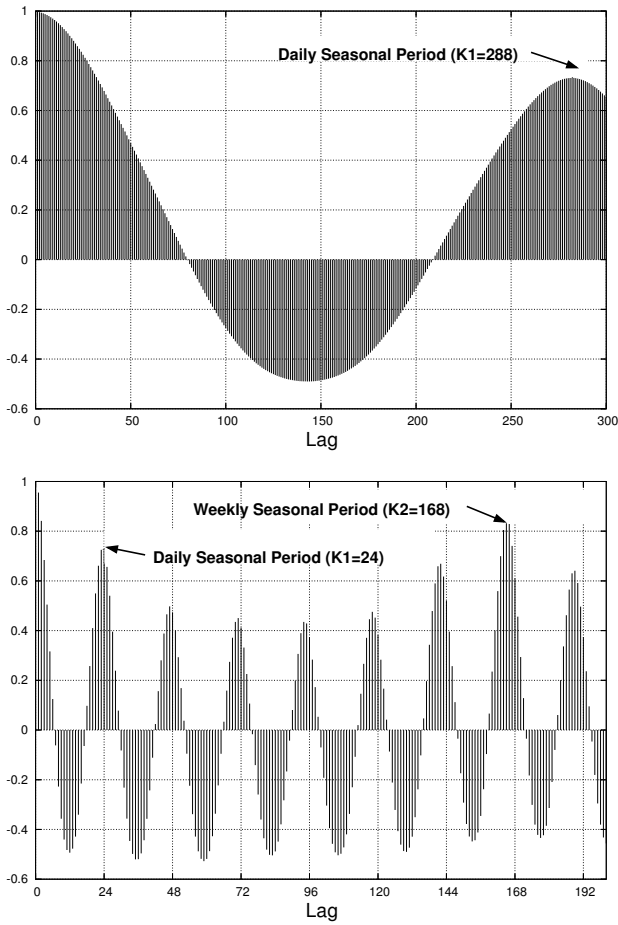
Fig. 2. The autocorrelations for the series **A5M** (top) and **A1H** (bottom)

The initial values for the level, trend and seasonal estimates will be set by averaging the early observations, as defined in [12]. The Holt-Winters parameters will be optimized using a grid search for the best training error (SSE_1), which is a common procedure within the forecasting field.

C. ARIMA Methodology

The Box-Jenkins methodology is another important forecasting approach, going over model identification, parameter estimation, and model validation [8]. The main advantage of this method relies on the accuracy over a wider domain of series, despite being more complex, in terms of usability and computational effort, than Holt-Winters. The global model is based on a linear combination of past values (AR components) and errors (MA components), being named *AutoRegressive Integrated Moving-Average (ARIMA)*.

The non seasonal model is denoted by the form $ARIMA(p, d, q)$ and is defined by the equation:

$$\phi_p(L)(1 - L)^d y_t = \theta_q(L)e_t \qquad (6)$$

where y_t is the series; e_t is the error; L is the lag or backshift operator (e.g. $L^3 y_t = y_{t-3}$); $\phi_p = 1 - \phi_1 L - \phi_2 L^2 - \ldots - \phi_p L^p$ is the AR polynomial of order p; d is the differencing

order; and $\theta_p = 1 - \theta_1 L - \theta_2 L^2 - \ldots - \theta_q L^q$ is the MA polynomial of order q. When the series has a non zero average through time, the model may also contemplate a constant term μ in the right side of the equation. For demonstrative purposes, the full time series model is presented for $ARIMA(1, 1, 1)$: $\widehat{y}_{t,t-1} = \mu + (1 + \phi_1)y_{t-1} - \phi_1 y_{t-2} - \theta_1 e_{t-1}$. To create multi-step predictions, the one step-ahead forecasts are used iteratively as inputs [18].

There is also a multiplicative seasonal version, often called SARIMA and denoted by the term $ARIMA(p, d, q)(P_1, D_1, Q_1)$. It can be written as:

$$\phi_p(L)\Phi_{P_1}(L^{K_1})(1 - L)^d(1 - L)^{D_1}y_t = \theta_q(L)\Theta_{Q_1}(L^{K_1})e_t \qquad (7)$$

where K_1 is the seasonal period; Φ_{P_1} and Θ_{Q_1} are polynomial functions of orders P_1 and Q_1. Finally, the double seasonal $ARIMA(p, d, q)(P_1, D_1, Q_1)(P_2, D_2, Q_2)$ is defined by [18]:

$$\phi_p(L)\Phi_{P_1}(L^{K_1})\Omega_{P_2}(L^{K_2})(1 - L)^d(1 - L)^{D_1}(1 - L)^{D_2}y_t = \theta_q(L)\Theta_{Q_1}(L^{K_1})\Psi_{Q_2}(L^{K_2})e_t \qquad (8)$$

where K_2 is the second seasonal period; Ω_{P_2} and Ψ_{Q_2} are the polynomials of orders P_2 and Q_2.

The constant and the coefficients of the model are usually estimated by using statistical approaches (e.g., least squares methods). It was decided to use the forecasting package X-12-ARIMA [20] (from the U.S. Bureau of the Census) for the parameter estimation of a given model. For each series, several $ARIMA$ models will be tested and the BIC statistic, which penalizes model complexity and is evaluated over the training data, will be the criterion for the model selection, as advised by the X-12-ARIMA manual [21].

D. Artificial Neural Networks

Neural models are innate candidates for forecasting due to their nonlinear and noise tolerance capabilities. The basic idea is to train a NN with past data and then use this network to predict future values. The use of NNs for TSF began in the late eighties with encouraging results and the field has been consistently growing since [9][10][22][11].
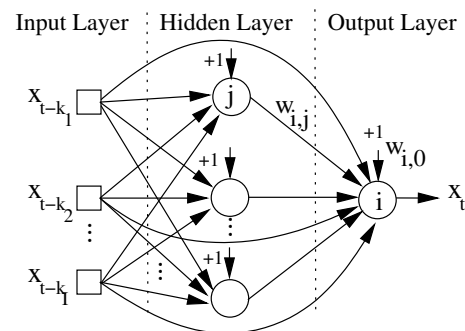

Fig. 3. The Multilayer Perceptron architecture

Although different types of NNs have been applied in the forecasting literature (e.g. Recurrent Networks [22]), the majority of the studies uses the Multilayer Perceptron

network [9][23][10][16][11]. 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 $\{k_1, k_2, \ldots, k_I\}$ used to build a forecast, for a network with I inputs. In this work, fully connected multilayer perceptrons, with one hidden layer of H hidden nodes, bias and direct shortcut connections will be adopted (Figure 3). To enhance nonlinearity, the logistic activation function was applied on the hidden nodes, while in the output node, the linear function was used instead, to scale the range of the outputs (the logistic function has a $[0,1]$ co-domain) [11]. The overall model is given in the form:

$$\widehat{y}_{t,t-1} = w_{o,0} \quad + \sum_{i=1}^{I} y_{t-k_i} w_{o,i} \\ + \sum_{j=I+1}^{o-1} f(\sum_{i=1}^{I} y_{t-k_i} w_{j,i} + w_{j,0}) w_{o_j} \tag{9}$$

where $w_{i,j}$ denotes the weight of the connection from node j to i (if $j = 0$ then it is a bias connection), o denotes the output node and f the logistic function ($\frac{1}{1+e^{-x}}$). Similar to the $ARIMA$ methodology, multi-step forecasts are built by iteratively using 1-ahead predictions as inputs [18].

In the training stage, the NN initial weights are randomly set within the range $[-1.0; 1.0]$. Then, the RPROP algorithm [24] was adopted, since it presents a faster training when compared with other algorithms (e.g. backpropagation) [24][25]. The training is stopped when the error slope approaches zero or after a maximum of 1000 epochs.

The quality of the trained network will depend on the choice of the starting weights, since the error function is non convex and the training may fall into local minima. To solve this issue, the solution adopted is to use an *Neural Network Ensemble (NNE)* where R different networks are trained (here set to $R = 5$) and the final prediction is given by the average of the individual predictions [26]. In the literature, this ensemble construction method is known by the term *Injecting Randomness*. In general, ensembles are better than individual learners, provided that the errors made by the individual models are uncorrelated, a condition easily met with NNs, since the training algorithms are stochastic in nature [27].

Under this setup, the NNE based forecasting method will depend solely on two parameters: the choice of the input time lags and number of hidden nodes (H). Both parameters have a crucial effect in the forecasting performance. A small time window will provide insufficient information, while a high number of lags will increase the probability of having irrelevant inputs. On the other hand, a network with few hidden nodes will have limited learning capabilities, while an excess of hidden nodes will lead to overfitting or generalization loss.

Guided by the experience obtained in previous work [23][28][11], both parameters will be set by a heuristic model selection step, where several configurations are tested. Therefore, the training data ($2/3$ of the series' length) will be divided into training and validation sets. The former, with $2/3$ of the training data, will be used to train the NNE.

The latter, with the remaining $1/3$, will be used to estimate the network generalization capabilities. Hence, the neural configuration with the lowest validation error (average of all $MAPE_h$ values) will be selected. After this model selection phase, the final NNE is retrained using all training data.

IV. EXPERIMENTS AND RESULTS

The Holt-Winters and NNs were implemented in an object oriented programming environment developed in the Java language by the authors. Regarding the ARIMA methodology, the different models will be estimated using the **X-12-ARIMA** package [21]. The best model (with the lowest BIC values) will be selected and then the forecasts are produced in the Java environment.

The *Holt-Winters (HW)* models were adapted to the series characteristics. The double seasonal variant ($K_1 = 24$ and $K_2 = 168$) was applied on the hourly series. Both seasonal ($K_1 = 288$) and non seasonal versions were tested for the five minute scale data, since it was suspected that the seasonal effect could be less relevant in this case. Indeed, SSE errors obtained in the training data backed this claim. To optimize the parameters of the selected models (results are shown in Table II), the grid-search used a step of 0.01 for the five minute data. The grid step was increased to 0.05 in the hourly series, due to the higher computational effort required by double seasonal models.

TABLE II
THE HOLT-WINTERS FORECASTING MODELS

Series	K_1	K_2	α	β	γ	ω
A5M	–	–	0.76	0.09	–	–
A1H	24	168	0.70	0.00	1.00	1.00
B5M	–	–	1.00	0.07	–	–
B1H	24	1105	0.95	0.00	0.75	1.00

Regarding the ARIMA methodology, an extensive range of different ARIMA models were tested for each series. In all cases, the μ constant was set to zero by the **X-12-ARIMA** package, used for model selection. In case of the hourly data, no differencing factors were used, since the series seem stationary and the Holt-Winters models provided no evidence for trended factors (with very low β values). A total of eight double seasonal $ARIMA$ models were tested, by using combinations of the p, P_1, P_2, q, Q_1 and Q_2 values up to a maximum order of 2. Finally, for the five minute datasets, 3 single seasonal (maximum order of 1) and 25 non seasonal (maximum order of 5) models were explored. Similar to the Holt-Winters case, for these series only non seasonal ARIMA models were selected. The best ARIMA models are shown in Table III.

For the NNE, the number of tested hidden nodes (H) was within the range $\{0,2,4,6,8\}$, since in previous work [11] it has been shown that even complex series can be modeled by small neural structures. Based on the series autocorrelations and seasonal traits, three different sets of time lags were explored for each time scale:

TABLE III
THE ARIMA FORECASTING MODELS

Series	Model	Parameters
A5M	(5 0 5)	$\phi_1 = 2.81$, $\phi_2 = -3.49$, $\phi_3 = 2.40$, $\phi_4 = -0.58$, $\phi_5 = -0.13$
		$\theta_1 = 1.98$, $\theta_2 = -1.91$, $\theta_3 = 0.75$, $\theta_4 = -0.26$, $\theta_5 = -0.20$
A1H	(2 0 0)(2 0 0)(2 0 0)	$\phi_1 = 1.70$, $\phi_2 = -0.74$, $\Phi_1 = 0.60$, $\Phi_2 = 0.06$
		$\Omega_1 = -0.08$, $\Omega_2 = 0.28$
B5M	(5 0 5)	$\phi_1 = 1.58$, $\phi_2 = -0.59$, $\phi_3 = 1.00$, $\phi_4 = -1.58$, $\phi_5 = 0.59$
		$\theta_1 = 0.74$, $\theta_2 = -0.08$, $\theta_3 = 0.97$, $\theta_4 = -0.77$, $\theta_5 = 0.06$
B1H	(2 0 1)(1 0 1)(1 0 1)	$\phi_1 = 1.59$, $\phi_2 = -0.62$, $\Phi_1 = 0.93$, $\Omega_1 = 0.82$,
		$\theta_1 = 0.36$, $\Theta_1 = 0.72$, $\Psi_1 = 0.44$

- $\{1,2,3,24,25,26,168,167,169\}$, $\{1,2,3,11,12,13,24,25,26\}$ and $\{1,2,3,24,25,26\}$ for the hourly data; and
- $\{1,2,3,5,6,7,287,288,289\}$, $\{1,2,3,5,6,7,11,12,13\}$ and $\{1,2,3,4,5,6,7\}$ for the five minute scale series.

The best forecasting neural models appear in Table IV. The number of hidden nodes suggest that the **A** datasets are nonlinear while the data from the ISP **B** are linear. Regarding the selected time lags, it is interesting to notice that there are two models that contrast with the previous methods: the **B5M** model includes seasonal information ($K_1 = 288$), while the **A1H** does not use the second seasonal factor ($K_2 = 168$).

TABLE IV
THE NEURAL NETWORK ENSEMBLE FORECASTING MODELS

Series	Hidden Nodes (H)	Input Time Lags
A5M	6	$\{1,2,3,5,6,7,11,12,13\}$
A1H	8	$\{1,2,3,24,25,26\}$
B5M	0	$\{1,2,3,5,6,7,287,288,289\}$
B1H	0	$\{1,2,3,24,25,26,168,167,169\}$

TABLE V
COMPARISON OF THE FORECASTING ERRORS ($MAPE_h$ VALUES, IN PERCENTAGE) FOR A LEAD TIME OF $h = 1, 12, 24$ AND $\overline{h}$ (AVERAGE FOR ALL $h \in \{1, \ldots, 24\}$ VALUES)

Series	h	Naive	Holt-Winters	ARIMA	NNE
A5M	1	34.79	2.98	2.95	**2.91**±0.00⋆
	12	34.79	10.99	10.06	**8.99**±0.06⋆
	24	34.83	21.65	18.08	**16.30**±0.21⋆
	$\overline{h}$	34.80	11.98	10.68	**9.59**±0.08⋆
B5M	1	20.10	1.44	1.74	**1.43**±0.01
	12	20.05	7.07	6.22	**6.09**±0.10⋆
	24	19.99	14.36	11.32	**10.92**±0.24⋆
	$\overline{h}$	20.05	7.65	6.60	**6.34**±0.11⋆
A1H	1	65.19	12.96	7.37	**5.23**±0.03⋆
	12	65.54	70.71	30.51	**28.37**±0.80⋆
	24	65.89	33.95	28.18	**25.11**±0.59⋆
	$\overline{h}$	65.67	50.60	26.96	**23.48**±0.49⋆
B1H	1	34.82	3.30	**3.13**	3.25±0.01
	12	35.12	15.11	**14.18**	14.88±0.06
	24	35.54	17.31	15.15	**12.20**±0.07⋆
	$\overline{h}$	35.18	13.69	12.69	**12.26**±0.03⋆

⋆ - Statistically significant when compared with other methods

After the model selection stage, the forecasts were performed for each method, testing a lead time from $h = 1$ to 24. In case of the NNE, 20 runs were applied to each configuration in order to present the results in terms of the average and respective t-student 95% confidence intervals [29]. Table V shows the forecasting errors (over the test data) for each method and lookaheads of $h \in \{1, 12, 24\}$. The global performance is presented in terms of the average error of all $h \in \{1, \ldots, 24\}$ values ($\overline{h}$). In addition, the $MAPE$ values for all forecasting lookaheads are plotted in Figure 4 (the NNE curves plot the average of the 20 runs).

As expected, the Naive benchmark reveals a constant performance at all lead times for the five minute series and it was greatly outperformed by the other forecasting approaches. Indeed, the remaining three methods obtain quite similar and very good forecasts ($MAPE$ values ranging from 1.4% to 3%) for a 5 minute lead. As the horizon is increased, the results decay slowly and in a linear fashion, although the Holt-Winters method presents a higher slope for both ISPs. At this time scale, the best approach is given by the NNE (Table V).

Turning to the hourly scale, the Naive method keeps the expected low forecasting performance. As before, the other methods present the lowest errors for the 1-ahead forecasts. However, the error curves are not linear and after a given horizon, the error decreases, in a behavior that may be explained by the seasonal effects (Figure 4). The differences between the methods are higher for the first provider (**A**) than the second one. Nevertheless, in both cases the ARIMA and NNE outperform the Holt-Winters method. Overall, the neural approach is the best model with a 3.5% global difference to ARIMA in dataset **A1H** and a 0.4% improvement in the second series (**B1H**). The higher relative NNE performance for the **A** ISP may be explained by the presence of nonlinear effects (as suggested in Table IV).

For demonstrative purposes, Figure 5 presents an example of 100 forecasts given by the NNE method for the series **A1H** and horizons of 1 and 24. The figure shows a good fit by the forecasts, which follow the series. Another relevant issue is related with the computational complexity. With a Pentium IV 1.6GHz processor, the NNE training (including the 5 different RPROP trainings) and testing for this series required
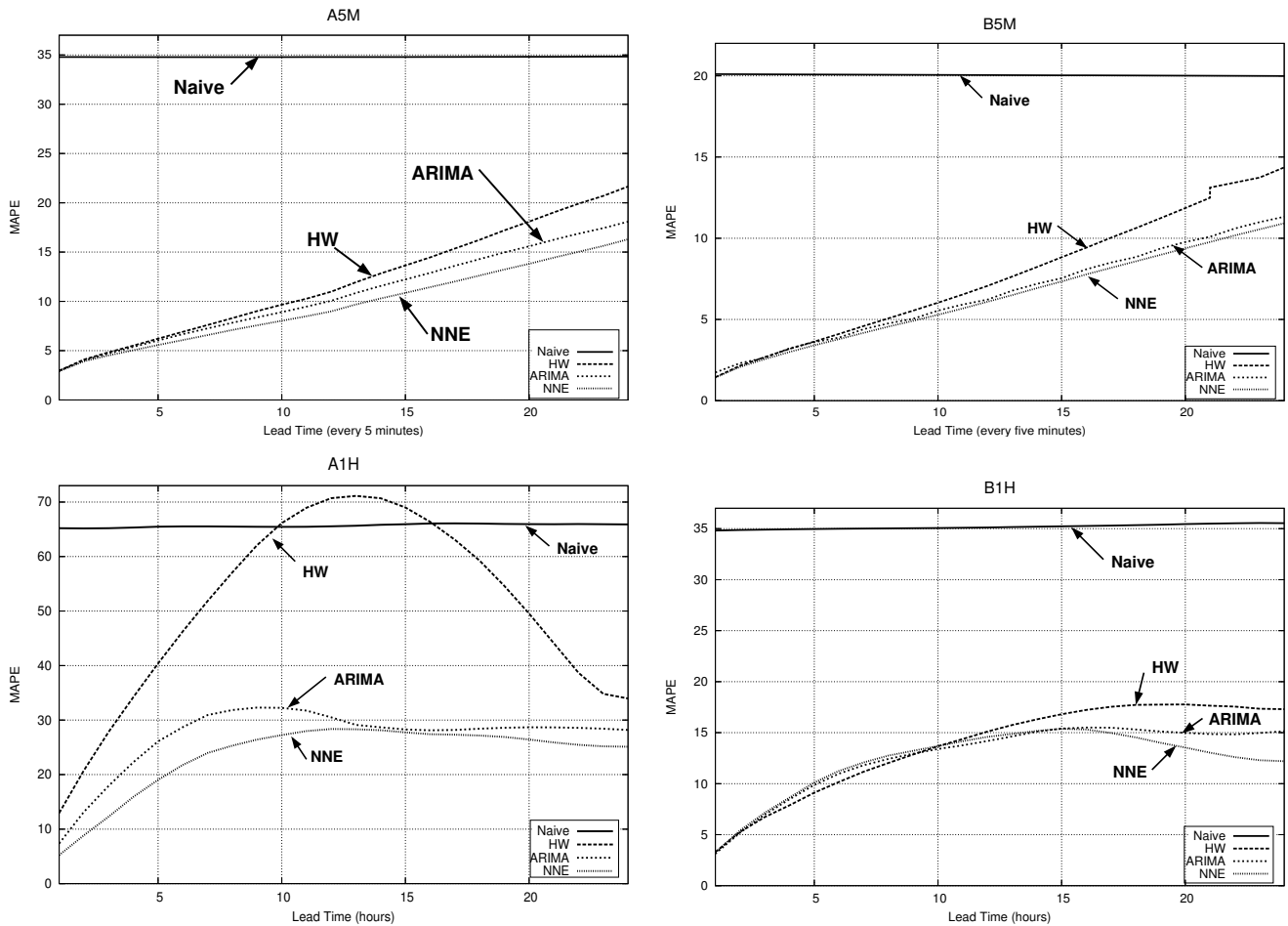
Fig. 4. The forecasting error results ($MAPE$) plotted against lead time (h) for the Internet Traffic series (**A5M**, **B5M**, **A1H** and **B1H**)

only 41 seconds. In this case, the computational demand for Holt-Winters increases around a factor of three, since the 0.05 grid-search required 137 seconds. For the double seasonal series, the highest effort is given by the ARIMA model, where the estimation of the parameters by the X12-ARIMA program, which was written in Fortran, took more than two hours of processing time.

V. CONCLUSIONS

In this work we analyzed the efficiency of several forecasting approaches when applied to TCP/IP traffic. These include three different *Time Series Forecasting (TSF)* methods: the Holt-Winters, the ARIMA methodology and a *Neural Network Ensemble (NNE)* approach. Recent data, collected from two large *Internet Source Providers (ISP)*, was analyzed using different forecasting types (or scales): real-time (every five minutes) and short-term (hourly aggregated values). Furthermore, each method was tested under several forecasting horizons, from one to twenty four periods ahead.

A comparison among the TSF methods shows that in general the NNE produces the lowest errors. When forecasting real-time or short-term data, the computational effort also needs to be considered. As shown in the previous section, and also argued by Taylor et al. [18], the ARIMA methodology
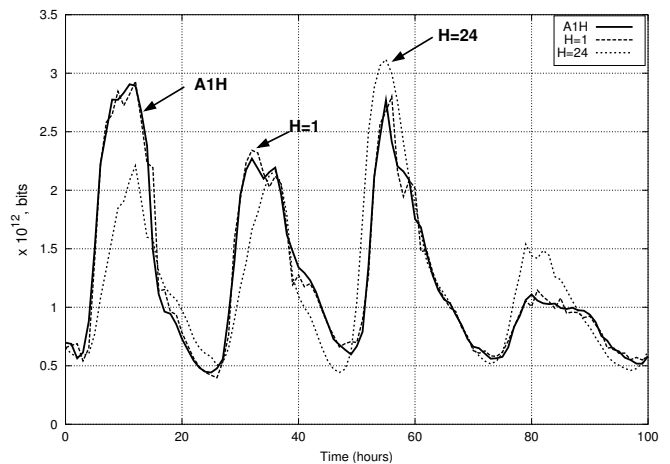

Fig. 5. Example of the Neural Network Ensemble forecasts for series **A1H** and lead times of $h = 1$ and $h = 24$

is rather impractical for on-line forecasting systems. Under this factor, the NNE is also the advised method, since it can be used in real-time.

The NNE results reveal promising performances. For the real-time forecasts, only a 1-3% error was obtained for the

five minute lookahead forecasts, a value that increases to 11–17% when the forecasts are issued two hours in advance (Table 4). Turning to the short-term predictions, the error goes from 3–5% (one hour ahead) to 12–23% (24 hour lookahead). Thus, the proposed approach opens room for producing better traffic engineering tools and methods to detect anomalies in the traffic patterns. This can be achieved without producing any extra traffic in the network and with minimal use of computation resources, since this work was designed assuming a passive monitoring system.

We plan, as future work, to apply similar methods to active measurement scenarios in which real-time packet level information is fed into the forecasting engine. We also intend to apply the forecasting techniques to traffic demands associated with specific Internet applications since this might benefit some management operations performed by ISPs such as traffic prioritization and network resource allocation. Moreover, a number of optimization methods can be used to improve the model selection step in the tested TSF models. For example, Evolutionary Computation can be used to to set the values of the ARIMA parameters [30] or to optimize both the weights and the topologies of the Neural Networks [11]. In this case, the model optimization can be conducted in parallel with the real-time use of the forecasting model, performing a model substitution when better results are expected.

REFERENCES

[1] K. Papagiannaki, N. Taft, Z. Zhang, and C. Diot, "Long-Term Forecasting of Internet Backbone Traffic," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1110–1124, Sept. 2005.

[2] B. Davie and Y. Rekhter, *MPLS: Multiprotocol Label Switching Technology and Applications*. USA: Morgan Kaufmann, 2000.

[3] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based Change Detection: Methods, Evaluation, and Applications," in *Proceedings of Internet Measurment Conference (IMC'03)*. Miami, USA: ACM, Oct. 2003.

[4] J. Jiang and S. Papavassiliou, "Detecting Network Attacks in the Internet via Statistical Network Traffic Normality Prediction," *Journal of Network and Systems Management*, vol. 12, pp. 51–72, 2004.

[5] J. Hanke and A. Reitsch, *Business Forecasting*. Allyn and Bancon Publishing, Massachussetts, USA, 1989.

[6] S. Makridakis, S. Weelwright, and R. Hyndman, *Forecasting: Methods and Applications*. John Wiley & Sons, New York, USA, 1998.

[7] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management Science*, vol. 6, pp. 324–342, 1960.

[8] G. Box and G. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden Day, San Francisco, USA, 1976.

[9] A. Lapedes and R. Farber, "Non-Linear Signal Processing Using Neural Networks: Prediction and System Modelling," Los Alamos National Laboratory, USA, Technical Report LA-UR-87-2662, 1987.

[10] X. Ding, S. Canu, and T. Denoeux, "Neural network based models for forecasting," in *Proceedings of Applied Decision Technologies Conf. (ADT'95)*, Uxbridge, UK, 1995, pp. 243–252.

[11] P. Cortez, M. Rocha, and J. Neves, "Time Series Forecasting by Evolutionary Neural Networks," chapter III: *Artificial Neural Networks in Real-Life Applications*, Idea Group Publishing, USA, pages 47–70, 2005.

[12] J. Taylor, "Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing," *Journal of Operational Research Society*, vol. 54, pp. 799–805, 2003.

[13] Q. He, C. Dovrolis, and M. Ammar, "On the Predictability of Large Transfer TCP Throughput," in *Proceedings of SIGCOMM'05*. Philadelphia, USA: ACM, Aug. 2005.

[14] A. Sang and S. Li, "A predictability analysis of network traffic," *Computer Networks*, vol. 39, no. 4, pp. 329–345, 2002.

[15] M. Hansegawa, G. Wu, and M. Mizuno, "Applications of Nonlinear Prediction Methods to the Internet Traffic," in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 3, Sydney, Australia, May 2001, pp. 169–172.

[16] H. Tong, C. Li, and J. He, "Boosting Feed-Forward Neural Network for Internet Traffic Prediction," in *Proceedings of the IEEE 3rd International Conference on Machine Learning and Cybernetics*, Shanghai, China, Aug. 2004, pp. 3129–3134.

[17] W. Stallings, *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*. Addison Wesley, 1999.

[18] J. Taylor, L. Menezes, and P. McSharry, "A Comparison of Univariate Methods for Forecasting Electricity Demand Up to a Day Ahead," *International Journal of Forecasting*, vol. In press, 2006.

[19] M. Brown and J. Kros, "Data mining and the impact of missing data," *Industrial Management & Data Systems*, vol. 103, no. 8, pp. 611–621, 2003.

[20] D. Findley, B. Monsell, W. Bell, M. Otto, and B. Chen, "New Capabilities and Methods of the X-12 Seasonal Adjustment Program," *Journal of Business and Economic Statistics*, vol. 16, no. 2, pp. 127–157, 1996.

[21] T. S. Staff, "X-12-ARIMA Reference Manual," http://www.census.gov/srd/www/x12a/, U. S. Census Bureau, Washington, USA, July, 2002.

[22] M. Hallas and G. Dorffner, "A comparative study on feedforward and recurrent neural networks in time series prediction using gradient descent learning," in *Proceedings of the European Meetings on Cybernetics and Systems Research (EMCSR'98)*, Vienna, Austria, 1998.

[23] P. Cortez, M. Rocha, J. Machado, and J. Neves, "A Neural Network Based Forecasting System," in *Proceedings of IEEE ICNN'95*, vol. 5, Perth, Australia, Nov. 1995, pp. 2689–2693.

[24] M. Riedmiller, "Advanced Supervised Learning in Multilayer Perceptrons - from Backpropagation to Adaptive Learning Techniques," *International Journal of Computer Standards and Interfaces*, vol. 16, pp. 265–278, 1994.

[25] R. Mendes, P. Cortez, M. Rocha, and J. Neves, "Particle Swarms for Feedforward Neural Network Training," in *Proceedings of The 2002 International Joint Conference on Neural Networks (IJCNN 2002)*, Honolulu, Havai, USA, IEEE Computer Society, May 2002, pp. 1895–1899.

[26] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. NY, USA: Springer-Verlag, 2001.

[27] T. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems, Lecture Notes in Computer Science 1857*, J. Kittler and F. Roli, Eds. Springer-Verlag, 2000, pp. 1–15.

[28] J. Neves and P. Cortez, "An Artificial Neural-Network Genetic Based Approach for Time Series Forecasting," in *Proceedings of IV Brazilian Symposium on Neural Networks*, D. L. Borges and W. Martins, Eds. Goiania, Brazil: IEEE Computer Society, Dec. 1997, pp. 9–13.

[29] A. Flexer, "Statistical evaluation of neural networks experiments: Minimum requirements and current practice," in *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, vol. 2, Vienna, Austria, 1996, pp. 1005–1008.

[30] P. Cortez, M. Rocha, and J. Neves, "Evolving Time Series Forecasting ARMA Models," *Journal of Heuristics*, vol. 10, no. 4, pp. 415–429, 2004.