Multiscale Internet traffic forecasting using neural networks and time series methods

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

[1] Dep. of Information Systems/Algoritmi, University of Minho, 4800-058 Guimarães, Portugal
[2] Dep. of Electronic and Electrical Engineering, University College London, Torrington Place, WC1E 7JE, London, UK
[3] Dep. of Informatics/CCTC, University of Minho, 4710-059 Braga, Portugal.

Abstract

This article presents three methods to forecast accurately the amount of traffic in TCP/IP based networks: a novel neural network ensemble approach and two important adapted time series methods (ARIMA and Holt-Winters). In order to assess their accuracy, several experiments were held using real-world data from two large Internet service providers. In addition, different time scales (five minutes, one hour and one day) and distinct forecasting look-aheads were analyzed. The experiments with the neural ensemble achieved the best results for five minutes and hourly data, while the Holt-Winters is the best option for the daily forecasts. This research opens possibilities for the development of more efficient traffic engineering and anomaly detection tools, which will result in financial gains from better network resource management.

Keywords: network monitoring; multilayer perceptron; time series; traffic engineering

1 Introduction

As more applications vital to today's society migrate to TCP/IP networks, it is crucial to develop techniques to better understand and forecast the behavior of these networks. In effect, TCP/IP traffic prediction is an important issue for any medium/large network provider and it is gaining more attention from the computer networks community (Papagiannaki et al., 2005; Babiarz and Bedo, 2006). By improving this task's performance, network providers can optimize resources (e.g. adaptive congestion control and proactive network management), allowing a better quality of service (Alarcon-Aquino and Barria, 2006). Moreover, traffic forecasting can also help to detect anomalies in the data networks. Security attacks like denial-of-service or even an irregular amount of SPAM can in theory be detected

by comparing the real traffic with the values predicted by forecasting algorithms (Krishnamurthy et al., 2003; Jiang and Papavassiliou, 2004). The earlier detection of these problems would conduct to a more reliable service.

Nowadays, TCP/IP traffic prediction is often done intuitively by experienced network administrators, with the help of marketing information on the future number of costumers and their behaviors (Papagiannaki et al., 2005). Yet, this produces only a rough idea of the real traffic. On the other hand, contributions from the areas of Operational Research and Computer Science has lead to solid forecasting methods that replaced intuition based ones in other fields. In particular, the field of time series forecasting (TSF) deals with the prediction of a chronologically ordered variable (Makridakis et al., 1998). 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 proposed, such as the Holt-Winters (Makridakis et al., 1998), the ARIMA methodology (Box and Jenkins, 1976) and artificial neural networks (NN) (Lapedes and Farber, 1987; Ding et al., 1995; Malki et al., 2004; Cortez et al., 2005). Holt-Winters was devised for series with trended and seasonal factors. More recently, a double seasonal version has been proposed (Taylor, 2003). The ARIMA is a more complex approach, requiring steps such as model identification, estimation and validation. 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 demonstrated the predictability of network traffic by using similar methods, such as Holt-Winters (Krishnamurthy et al., 2003) and ARIMA (Sang and Li, 2002; Papagiannaki et al., 2005). Following the evidence of nonlinear network traffic (Hansegawa et al., 2001), NNs have also been proposed (Jiang and Papavassiliou, 2004; Alarcon-Aquino and Barria, 2006; Wang et al., 2008).

In this work, several experiments are carried out, based on recent real-world data provided by two ISPs, in order to provide network engineers with a useful feedback. 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 contrast to (Krishnamurthy et al., 2003), which uses compact summaries of traffic data, and (Papagiannaki et al., 2005), which uses wavelets to smooth the signal, allowing its use in wider contexts;

ii) several forecasting methods are tested and compared, including a novel NN ensemble based on fast heuristic procedures for time window and model selection, and adaptations of the Holt-Winters, both traditional and recent double seasonal versions, and the ARIMA methodology;

iii) in contrast with previous studies (Hansegawa et al., 2001; Jiang and Papavassiliou, 2004; Papagiannaki et al., 2005; Babiarz and Bedo, 2006; Alarcon-Aquino and Barria, 2006; Wang et al., 2008), two distinct ISPs are considered, the predictions are analyzed at different time scales (i.e. five minutes, hourly, daily) and distinct ahead forecasts are performed.

The result of this research is expected to allow the development of intelligent TCP/IP traffic forecasting engines.

The article is organized in four sections. Firstly, the Internet traffic data is presented and analyzed. The forecasting methods are given in section 3, while the results are presented and discussed in the section 4. Finally, in the last section closing conclusions are drawn.

2 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 (Makridakis et al., 1998). 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 performance of a forecasting model is evaluated by an accuracy measure, such as the sum squared error (SSE) and mean absolute percentage error (MAPE) (Makridakis et al., 1998):

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

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.

The $MAPE$ is a common metric in forecasting applications, such as electricity demand (Malki et al., 2004; Taylor et al., 2006), 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 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 since the relative forecasting performances are similar.

Our approach uses already available information provided by the Simple Network Management Protocol (SNMP) that quantifies the traffic passing through every network interface with reasonable accuracy (Stallings, 1999). SNMP is widely deployed by every ISP/network, so the collection of this data does not induce any extra traffic on the network.

This work analyzes 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 to 11:17 AM on 29 July, 2005. Dataset **B** comes from UKERNA[1] and represents aggregated traffic in the 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 every five minutes. The first series (**A**) included 8 missing values, which were replaced using a linear interpolation (Hastie et al., 2001). The missing

[1]United Kingdom education and research networking association

data is explained by the fact that the SNMP scripts are not 100% reliable, since the SNMP messages may be lost. Yet, this occurs very rarely and it is statistically insignificant. Finally, it should be mentioned that within this domain it is difficult to collect more than 2/3 months of data, since network servers often reboot or pass through update/maintenance changes.

Depending on the time scale, the following forecasting types can be defined (Ding et al., 1995):

- 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 financial investments.

Due to the characteristics of the Internet traffic collected, this study will only consider the first three types. Therefore, three 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**), every hour (**A1H** and **B1H**) and every day (**A1D** and **B1D**)[2]. Due to the temporal nature of this domain, a sequential holdout (i.e. train/test split) will be adopted for the forecasting evaluation. Hence, the first 2/3 of the series will be used to fit (train) the forecasting models and the remaining last 1/3 to evaluate (test) the forecasting accuracies (Table 1). 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 1: The scale and length of Internet traffic time series

Series	**Time scale**	**Train length**	**Test length**	**Total length**
A5M	5 min.	9848	4924	14772
A1H	1 hour	821	410	1231
A1D	1 day	34	17	51
B5M	5 min.	13259	6629	19888
B1H	1 hour	1105	552	1657
B1D	1 day	46	23	69

The autocorrelation coefficient is a statistic that measures the correlation between a series and itself, lagged of k periods (Box and Jenkins, 1976):

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

[2]The datasets are available at: `http://www3.dsi.uminho.pt/pcortez/series/`

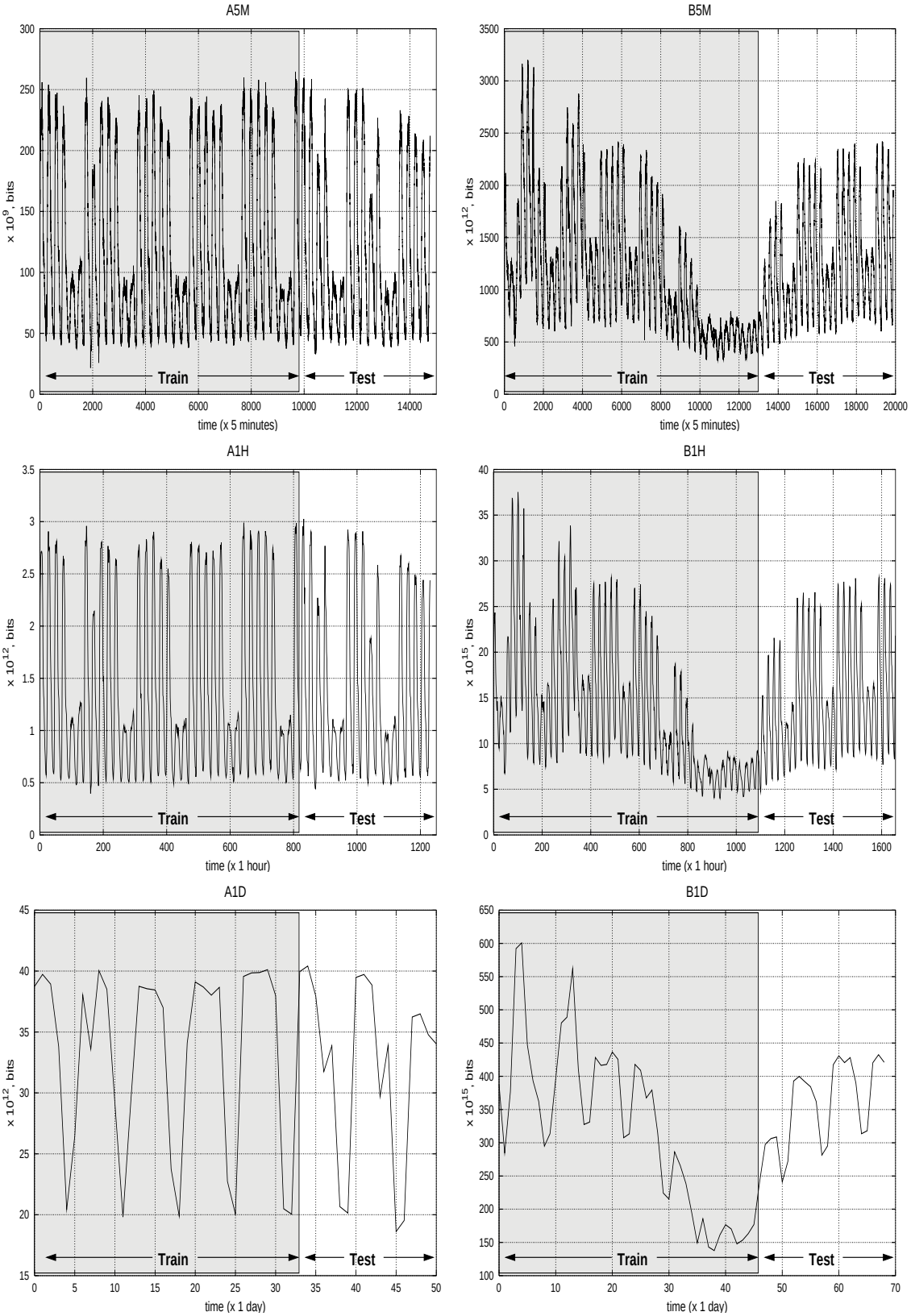

Figure 1: The Internet traffic time series (**A5M**, **B5M**, **A1H**, **B1H**, **A1D** and **B1D**) with several daily, weekly and seasonal patterns (e.g. traffic decrease in late December in **B** datasets is due to Christmas season).

where $y_1, \ldots, y_T$ stands for the time series and $\overline{y}$ for the series' average. Autocorrelations are useful for the detection of seasonal components (Makridakis et al., 1998). For example, the autocorrelations for the **A5M** and **A1H** series are plotted in Figure 2. The daily seasonal effect ($K_1 = 288$) is visible for the five minute data, while two seasonal components appear at the hourly scale, due to the the intraday ($K_1 = 24$) and intraweek cycles ($K_2 = 168$).

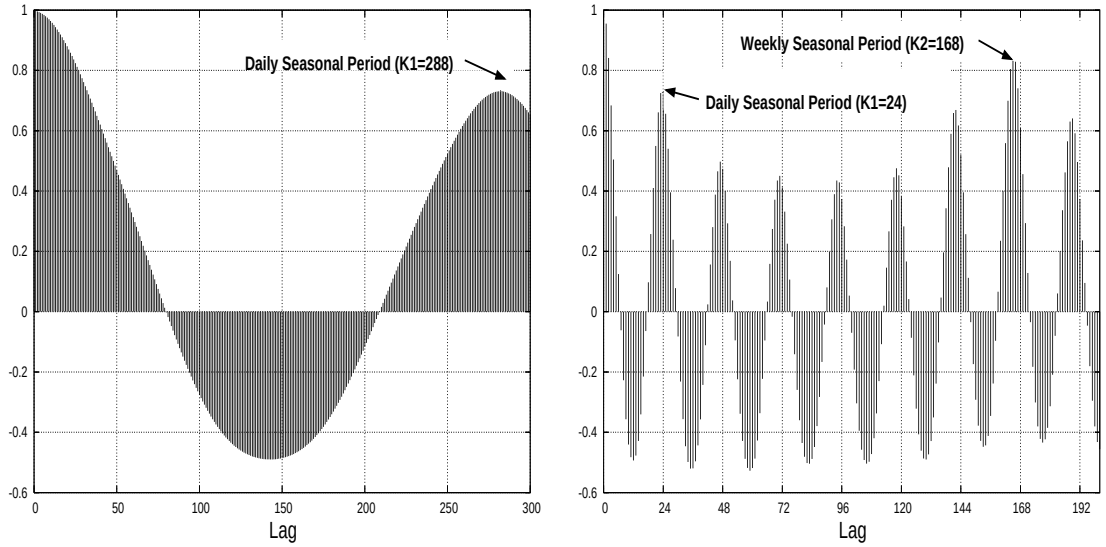

Figure 2: The autocorrelations for the series **A5M** (left) and **A1H** (right)

3 Forecasting methods

3.1 Naive benchmark

A common naive forecasting method is to predict the future as the present value. Yet, this method will perform poorly in seasonal data. Thus, a better 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 (Taylor et al., 2006):

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

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.

3.2 Holt-Winters

The Holt-Winters is an important forecasting technique where the predictive model is based on trended and seasonable patterns that are distinguished from noise by averaging the historical values. It presents advantages such as simplicity of use,

reduced computational demand and accuracy for seasonal series. The model is defined by the equations (Makridakis et al., 1998):

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

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 (Taylor, 2003):

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

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.

The initial values for the level, trend and seasonal estimates will be set by averaging the early observations (Taylor, 2003). The parameters (α, β, γ and ω) will be optimized by a grid search, which works by testing all combinations of a discrete set of values for each parameter. The aim is to get the lowest training error (SSE_1), which is a common procedure within the forecasting community.

3.3 ARIMA methodology

The ARIMA is another important forecasting approach that goes through model identification, parameter estimation and model validation (Box and Jenkins, 1976). The main advantage of this method relies on the accuracy over a wider domain of series, despite being more complex than the Holt-Winters. The 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:

$$\phi_p(L)(1-L)^d y_t = \theta_q(L) e_t \tag{6}$$

where y_t is the series; e_t is the error; L is the lag 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 (Taylor et al., 2006).

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:

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

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 (Taylor et al., 2006):

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

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 from the U.S. Bureau of the Census (Time-Series-Staff, 2002), 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.

3.4 Artificial neural networks

Neural models are innate candidates for forecasting due to their flexibility (i.e. there is no a priori restrictions on the type of relationship to be modeled) and nonlinear learning capabilities. Indeed, the use of NNs for TSF began in the late eighties with encouraging results and the field has been consistently growing since (Lapedes and Farber, 1987; Ding et al., 1995; Malki et al., 2004; Cortez et al., 2005).

Although there are other neural architectures, the majority of the NN studies use the multilayer perceptron network (Lapedes and Farber, 1987; Cortez et al., 1995; Ding et al., 1995; Tong et al., 2004; Cortez et al., 2005). With this network, TSF is achieved by using a sliding time window (Wang et al., 2008), defined by the set of time lags $\{k_1, k_2, \ldots, k_I\}$ used to build a forecast. For a given time period t, the NN inputs are $y_{t-k_I}, \ldots, y_{t-k_2}, y_{t-k_1}$ and the desired output is y_t. For example, let us consider the series $5_1, 10_2, 14_3, 19_4, 23_5$ (y_t values). If the $\{1,3\}$ window is adopted, then two training examples can be created: $5, 14 \rightarrow 19$ and $10, 19 \rightarrow 23$.

In this work, fully connected multilayer perceptrons, with one hidden layer of H hidden nodes, bias and shortcut connections will be adopted (Figure 3). To introduce nonlinearity, the logistic activation function was applied on the hidden nodes. The linear function was used in the output node, in order to scale the range of the outputs (Cortez et al., 2005). The final model is given by:

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

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 $ARIMA$, multi-step forecasts are built by iteratively using 1-ahead predictions as inputs (Taylor et al., 2006).

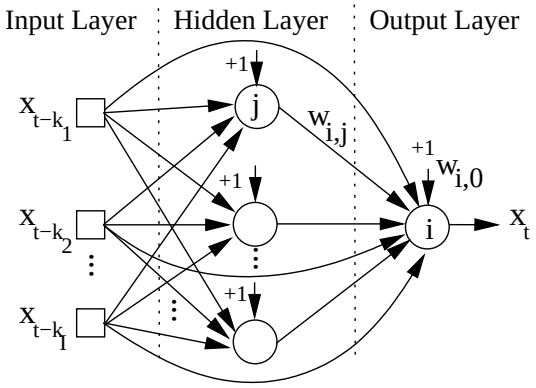

Figure 3: The neural network architecture

In the training stage, the NN initial weights are randomly set within the range $[-1.0; 1.0]$. Then, the RPROP algorithm was adopted, since it presents a faster training when compared with other algorithms such as the backpropagation (Riedmiller, 1994). 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 a 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 (Hastie et al., 2001). 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 (Dietterich, 2000).

Under this setup, the NNE performance will depend on two crucial parameters: the choice of the input time lags and number of hidden nodes (H). Feeding a NN with uncorrelated variables or time lags will affect the learning process due to the increase of noise. A NN with 0 hidden neurons can only learn linear relationships and it is equivalent to the classic Auto-Regressive (AR) model. By increasing the number of hidden neurons, more complex functions can be learned but also it increases the probability of overfitting to the data and thus losing the generalization capability.

Since the search space for these parameters is high, heuristic procedures will be proposed in the next section to reduce the computational effort, limiting the search to a few time window/hidden node combinations during the model selection step. In this stage, the training data (2/3 of the series' length) will be further 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. The NNE with the lowest validation error (average of all $MAPE_h$ values) will be selected. After the model selection, the final

NNE is retrained using all training data.

4 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 (Time-Series-Staff, 2002). 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 seasonal version ($K_1 = 7$) was used for the daily values, while 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 (Table 2), the grid-search used a step of 0.01 for the five minute and daily data. The grid step was increased to 0.05 in the hourly series, due to the higher computational effort required by the double seasonal models.

Table 2: The selected 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
A1D	7	–	0.00	0.00	1.00	–
B5M	–	–	1.00	0.07	–	–
B1H	24	1105	0.95	0.00	0.75	1.00
B1D	7	–	1.00	0.01	0.01	–

Regarding the ARIMA, an extensive range of models were tested. In all cases, the μ constant was set to zero by the X-12-ARIMA package. For the daily series, the p, P_1, q and Q_1 orders ranged from 0 to 2; and the d and D_1 orders were set to 0 and 1, in a total of 35 models. In case of the hourly data, no differencing factors were used, since the series seems 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. Table 3 shows the best ARIMA models.

The NNE heuristic rules for model selection were set as follows. The number of tested hidden nodes (H) was within the range {0,2,4,6,8}, since in previous work (Cortez et al., 2005) it has been shown that complex series can be modeled by small

Table 3: The selected 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)	$\phi_1 = 1.70$, $\phi_2 = -0.74$, $\Phi_1 = 0.60$, $\Phi_2 = 0.06$
		$\Omega_1 = -0.08$, $\Omega_2 = 0.28$
A1D	(2 1 0)(0 1 0)	$\phi_1 = -0.46$, $\phi_2 = -0.35$
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)	$\phi_1 = 1.59$, $\phi_2 = -0.62$, $\Phi_1 = 0.93$, $\Omega_1 = 0.82$,
		$\theta_1 = 0.36$, $\Theta_1 = 0.72$, $\Psi_1 = 0.44$
B1D	(1 1 2)(0 1 1)	$\phi_1 = 0.41$, $\theta_1 = 0.45$, $\theta_2 = 0.36$, $\Theta_1 = 0.53$

neural structures. Based on the seasonal traits, three different sliding windows were explored in each time scale:

- {1,2,3,4,5,6,7,8}, {1,2,3,6,7,8} and {1,7,8} for the daily series;
- {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.

Table 4 presents the selected NNEs. 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 4: The selected neural 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}
A1D	0	{1,7,8}
B5M	0	{1,2,3,5,6,7,287,288,289}
B1H	0	{1,2,3,24,25,26,168,167,169}
B1D	0	{1,7,8}

After the model selection stage, the forecasts were performed for each method by testing a lead time from h =1 to 24, for the five minute and hourly data, and an horizon of h =1 to 7 for the daily series. In case of the NNE, 20 runs were applied to each configuration in order to present the results in terms of the average and

t-student 95% confidence intervals (Flexer, 1996). Table 5 shows the forecasting errors for each method, when using the smallest and largest lookaheads. The global performance is presented as the average error ($\overline{h}$) for all h values. The overall view is given in Figure 4, where the $MAPE$ is plotted for all horizons.

Table 5: Comparison between the forecasting methods ($MAPE_h$ values, in percentage)

Series	**Horizon** (h)	**naive**	**Holt-Winters**	**ARIMA**	**NNE**
A5M	1	34.79	2.98	2.95	**2.91**±0.00*
	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
	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*
	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
	24	35.54	17.31	15.15	**12.20**±0.07*
	$\overline{h}$	35.18	13.69	12.69	**12.26**±0.03*
A1D	1	**6.77**	**6.77**	8.49	8.76±0.00
	7	**6.25**	**6.25**	7.23	7.99±0.00
	$\overline{h}$	**6.34**	**6.34**	8.12	8.48±0.00
B1D	1	20.81	**7.00**	9.79	12.99±0.01
	7	**13.65**	18.38	21.11	31.04±0.01
	$\overline{h}$	17.62	**13.43**	18.18	24.89±0.01

$\star$ - Statistically significant when compared with other methods

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 within the range 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 5).

Turning to the hourly scale, the naive method is still the worst method. 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

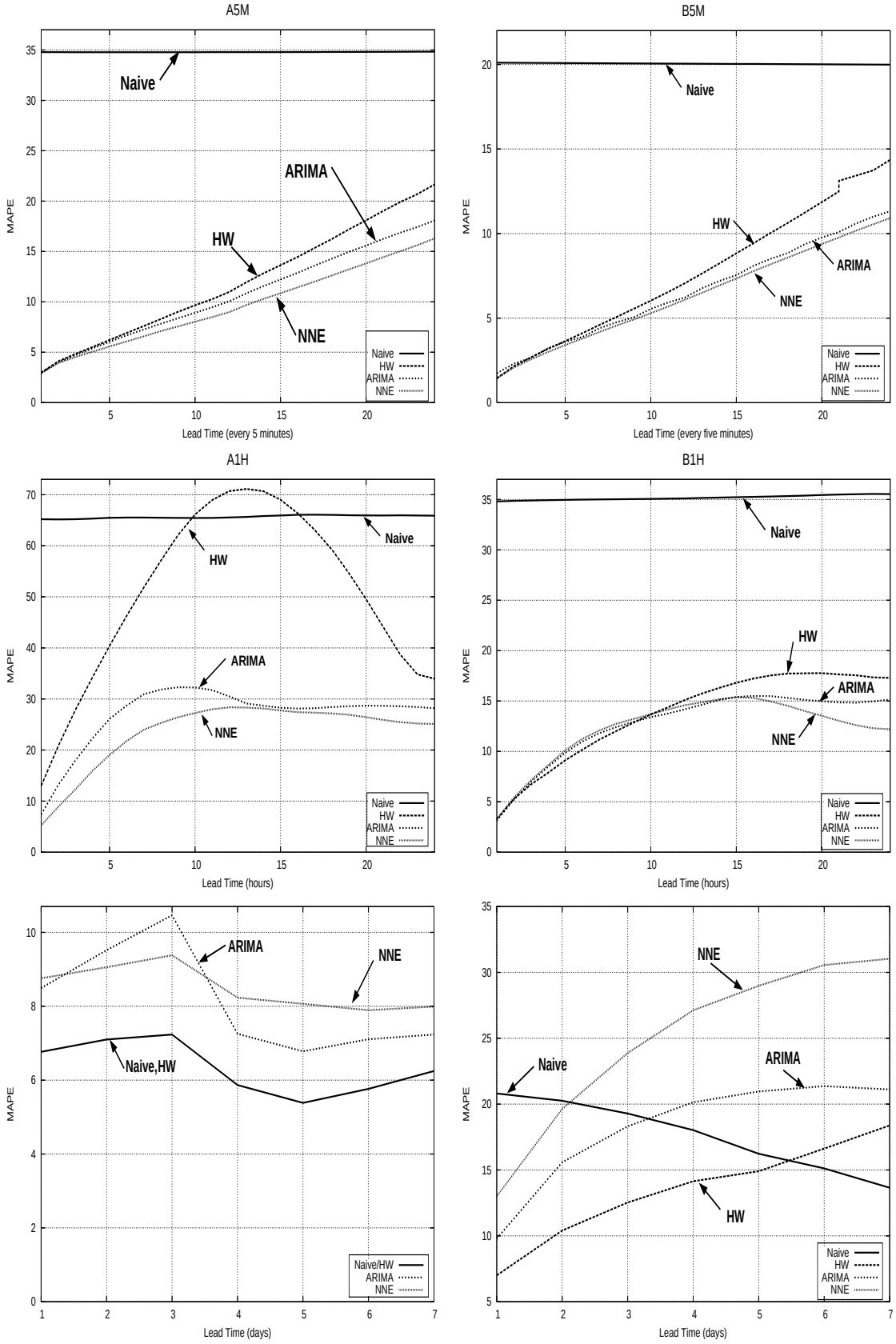

Figure 4: The forecasting error results ($MAPE$) plotted against the lead time (h)

the presence of nonlinear effects (as suggested in Table 4).

The analysis of the daily results shows a different behavior. The naive approach is one of the best options for the **A1D** data. This effect also occurs for series **B1D**, although only after a lead time of $h \geq 3$ for NNE, $h \geq 4$ for ARIMA and $h \geq 6$ for Holt-Winters. In both series, the best choice is the Holt-Winters method, which is equivalent to the naive method for the **A** series. These results are not surprising, since the Holt-Winters can be quite accurate even when few historic values are present (Makridakis et al., 1998). It should be noticed that the training data for **A1D** contains only 34 elements, while **B1D** contains 46. In contrast, NNs tend to give bad results when less than 50 observations are used (Makridakis, 1982).

For demonstrative purposes, Figure 5 presents 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 5 runs of the RPROP algorithm) and testing for this series required 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 X-12-ARIMA estimation took more than two hours of processing time.

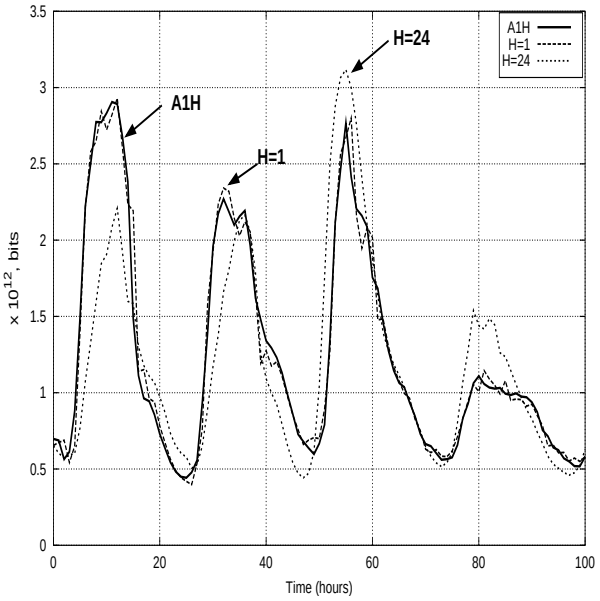

Figure 5: Example of the neural forecasts for series **A1H** and lead times of $h = 1$ and $h = 24$

5 Conclusions

In this article, three time series methods were presented to forecast the amount of traffic in TCP/IP based networks. A neural network ensemble (NNE) was developed

and the both the Holt-Winters and the ARIMA methods were adapted. Recent real-world data collected from two large Internet source providers (ISPs) was analyzed using different ahead predictions and time scales (e.g. every five minutes, hour and day).

A comparison among the time series methods shows that both ARIMA and NNE produce the lowest errors for the five minute and hourly data, with the latter method presenting the best overall performance. As shown in the previous section, and also argued in (Taylor et al., 2006), the ARIMA methodology is impractical for on-line forecasting systems because it requires more computation. Although the search space for NNE is high (i.e. selecting the best neural architecture and set of time lags), the heuristics proposed here for feature/model selection reduce substantially the computational effort and are easy to implement, while still providing competitive forecasts. Hence, the NNE is the recommended approach, since it can be used in real-time and this is crucial for dynamic resource allocation. At the daily scale, the Holt-Winters provided the best forecasts since our datasets contained few observations. However, in a on-line setting, an ISP could easily store hundreds of daily aggregated data. Thus, we believe that the proposed NNE would also lead to accurate forecasts in such scenario.

The experimental results reveal promising performances. Only a 1% to 3% error was obtained for the five minute forecasts. This value increased from 11% to 17% when the forecasts were issued two hours in advance. For the short-term predictions, the error goes from 3% to 5% (one hour in advance) until 13% to 22% (24 hour lookahead). Finally, the daily forecasts gave rise to error rates of 7% (one day horizon) and 6% to 13% (one week lookahead). Moreover, once this work was designed assuming a passive monitoring system, no extra traffic is required in the network. Hence, the recommended approach opens room for the development of better traffic engineering tools and methods to detect anomalies in the traffic patterns.

In the future, similar methods will be applied to forecast traffic demands associated with specific Internet applications, since this might benefit management operations performed by ISPs, such as traffic prioritization. Another interesting possibility, would be the exploration of similar forecasting approaches to other domains (e.g. electricity demand or road traffic).

Acknowledgements

This work is supported by the FCT (Portuguese science foundation) project PTDC-/EIA/64541/2006. We would also like to thank Steve Williams from UKERNA for providing us with part of the data used in this work.

References

Alarcon-Aquino, V. and Barria, J. (2006). Multiresolution FIR Neural-Network-Based Learning Algorithm Applied to Network Traffic Prediction. *IEEE Trans. on Systems, Man and Cybernetics - Part C*, 36(2):208–220.

Babiarz, R. and Bedo, J. (2006). Internet Traffic Mid-term Forecasting: A Pragmatic Approach Using Statistical Analysis Tools. *Lecture Notes on Computer Science*, 3976:111–121.

Box, G. and Jenkins, G. (1976). *Time Series Analysis: Forecasting and Control*. Holden Day, USA.

Cortez, P., Rocha, M., Machado, J., and Neves, J. (1995). A Neural Network Based Forecasting System. In *Proc. of IEEE ICNN'95*, volume 5, pages 2689–2693, Perth, Australia.

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

Dietterich, T. (2000). Ensemble methods in machine learning. In Kittler, J. and Roli, F., editors, *Multiple Classifier Systems, Lecture Notes in Computer Science 1857*, pages 1–15. Springer.

Ding, X., Canu, S., and Denoeux, T. (1995). Neural network based models for forecasting. In *Proc. of Applied Decision Technologies Conf. (ADT'95)*, pages 243–252, Uxbridge, UK.

Flexer, A. (1996). Statistical evaluation of neural networks experiments: Minimum requirements and current practice. In *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, volume 2, pages 1005–1008, Vienna, Austria.

Hansegawa, M., Wu, G., and Mizuno, M. (2001). Applications of Nonlinear Prediction Methods to the Internet Traffic. In *Proc. of IEEE Int. Symp. on Circuits and Systems*, volume 3, pages 169–172, Sydney, Australia.

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

Jiang, J. and Papavassiliou, S. (2004). Detecting Network Attacks in the Internet via Statistical Network Traffic Normality Prediction. *Journal of Network and Systems Management*, 12:51–72.

Krishnamurthy, B., Sen, S., Zhang, Y., and Chen, Y. (2003). Sketch-based Change Detection: Methods, Evaluation, and Applications. In *Proc. of Internet Measurment Conference (IMC'03)*, Miami, USA. ACM.

Lapedes, A. and Farber, R. (1987). Non-Linear Signal Processing Using Neural Networks: Prediction and System Modelling. Tech. Rep. LA-UR-87-2662, Los Alamos National Laboratory, USA.

Makridakis, S. (1982). The accuracy of extrapolation (times series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1:111–153.

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

Malki, H., Karayiannis, N., Nicolaos, B., and Balasubramanian, M. (2004). Short-term electric power load forecasting using feedforward neural networks. *Expert Systems*, 21(3):157–167.

Papagiannaki, K., Taft, N., Zhang, Z., and Diot, C. (2005). Long-Term Forecasting of Internet Backbone Traffic. *IEEE Trans. on Neural Networks*, 16(5):1110–1124.

Riedmiller, M. (1994). Advanced Supervised Learning in Multilayer Perceptrons - from Backpropagation to Adaptive Learning Techniques. *Int. Journal of Computer Standards and Interfaces*, 16:265–278.

Sang, A. and Li, S. (2002). A predictability analysis of network traffic. *Computer Networks*, 39(4):329–345.

Stallings, W. (1999). *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*. Addison Wesley.

Taylor, J. (2003). Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing. *Journal of Operational Research Society*, 54:799–805.

Taylor, J., Menezes, L., and McSharry, P. (2006). A Comparison of Univariate Methods for Forecasting Electricity Demand Up to a Day Ahead. *Int. Journal of Forecasting*, 21(1):1–16.

Time-Series-Staff (2002). X-12-ARIMA Reference Manual. http://www.census.gov/srd/www/x12a/, U. S. Census Bureau, Washington, USA, July.

Tong, H., Li, C., and He, J. (2004). 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.

Wang, C., x. Zhang, Yan, H., and Zheng, L. (2008). An Internet Traffic Forecasting Model Adopting Radical Based on Function Neural Network Optimized by Genetic Algorithm. In *Proceedings of IEEE Workshop on Knowledge Discovery and Data Mining (WKDD08)*, pages 367–370, Adelaide, Australia.

The authors

Paulo Cortez

Paulo Cortez received a MSc degree (1998) and a PhD (2002), both in Computer Science, University of Minho, where he works since 2001 as an Assistant Professor in the Department of Information Systems. He is also researcher at the Algoritmi centre, with interests in the fields of: business intelligence, data mining, neural networks, evolutionary computation and forecasting. Currently, he is associate editor of the *Neural Processing Letters* journal and he participated in 7 R&D projects (principal investigator in 2). He is co-author of more than sixty publications in international peer reviewed journals and conferences. Web-page: `http://www3.dsi.uminho.pt/pcortez`

Miguel Rio

Miguel Rio received the PhD from the University of Kent at Canterbury where he worked on Multicast distribution with Quality of Service. He has been the Principal Investigator of several UK and EU funded research project in areas of Telecommunications and Future Internet. Currently, he is Senior Lecturer in the Department of Electrical and Electronic Engineering, University College London. His research interests include peer-to-peer real-time delivery, routing, congestion control, and network traffic analysis. Web-page: `http://www.ee.ucl.ac.uk/~mrio`

Miguel Rocha

Miguel Rocha obtained a MSc degree (1998) and a PhD (2004), both in Computer Science, University of Minho. Since 1998, he is an Assistant Professor in the Artificial Intelligence group at the Department of Informatics at the same institution. His research interests include bioinformatics and systems biology, evolutionary computation and neural networks, where he coordinates funded projects and has a number of refereed publications in journals and international conferences (see `http://www.di.uminho.pt/~mpr`).

Pedro Sousa

Pedro Sousa received a MSc degree (1997) and a PhD (2005), both in Computer Science, University of Minho. In 1996, he joined the Computer Communications Group of the Department of Informatics at University of Minho, where he is a Assistant Professor and performs his research activities within the CCTC R&D Center. His main research interests include computer networks technologies and protocols, network simulation, TCP/IP protocols, quality of service, traffic scheduling and mobile networks. Web-page: `http://marco.uminho.pt/~pns`