Prediction of Internet traffic using time series and neural networks

Conference Paper · June 2014

CITATIONS
4

READS
410

2 authors:

Christos Katris
Athens University of Economics and Business
11 PUBLICATIONS **38** CITATIONS
SEE PROFILE

Sophia Daskalaki
University of Patras
40 PUBLICATIONS **924** CITATIONS
SEE PROFILE

Prediction of Internet Traffic using Time Series and Neural Networks

Christos Katris, Sophia Daskalaki

Department of Electrical and Computer Engineering, University of Patras[1]

Abstract. The purpose of this work is to compare forecasting model building procedures with Internet traffic data. More specifically we compare ARIMA, FARIMA and Artificial Neural Networks (ANN) models built for several datasets from public domain. Since most Internet traffic data carry properties like self-similarity and long-range dependence it is expected that FARIMA will dominate over the ARIMA. However, predictions may further be improved if we recognize the non-linear structure, existing sometimes in the time series, with the help of ANN. The networks that are built here are multilayer perceptrons and their topology is determined using techniques like False Nearest Neighbors and mutual information from dynamical systems. The forecasting performance of the models is measured through RMSE and MAE.

Keywords: Internet traffic, self-similarity, long-range dependence, FARIMA, Neural Networks

1 Introduction

Internet traffic, as it is well known, carries certain statistical properties, such as self-similarity and Long Range Dependence (LRD) (Leland et al, 1994; Beran et al., 1995). Because of these properties the autocorrelation function of the resulting time series decays hyperbolically, instead of exponentially which would have indicated Short Range Dependence (SRD). Forecasting models such as AR and ARIMA (Asteriou and Hall, 2011) can capture SRD dependencies, but not necessarily LRD. Despite of this weak point their use in modeling Ethernet traffic or video traffic is extensive, for example in (Zhani and Elbiase, 2009; Won and Ahn, 2005).

To overcome this problem self-similar models such as Fractional Gaussian Noise (Paxson, 1997) and FARIMA$(0,d,0)$ models have been used. Recent traffic measurements, however, unveiled the co-existence of LRD and SRD. A particular category of models that can describe both dependencies are FARIMA (p,d,q) (Hosking, 1981; Granger and Joyeux, 1980), which have also been used for modeling various types of Internet traffic. For example, in (Corradi et al, 2001) for Ethernet and video data and in (Shu et al, 1999) for broadband traffic. The main issue with all FARIMA models is the estimation of the fractional difference parameter *d*, while one of the suggested approaches is the Geweke-Porter-Hudak estimator (Geweke and Porter-Hudak, 1983).

A more recent alternative approach in forecasting is based on Artificial Neural Networks (ANN). Examples of ANN models used for time series prediction can be found in (Balkin and Ord, 2000; Frank et al., 2001) and for Internet traffic prediction in (Wang et al., 2008; Cortez et al., 2012). The strength of ANN is mainly displayed when the time series carries a nonlinear structure. Since ARIMA and FARIMA models assume linearity in mean their predictive ability is limited in cases of nonlinearity. Therefore, a statistical test that can suggest whether the data support the assumption of linearity is considered to be quite valuable. Specifically for this study, the White Neural Network test (Lee et al, 1993) has been used; and it becomes clear from our results that selection of a forecasting model according to the White Neural Network test leads to improved predictions. More recently hybrid methods which combine time series and neural networks have been proposed (Zhang, 2003; Aladag et al, 2012) in order to capture both linearities and non-linearities in time series.

The remaining of the paper is as follows. Section 2 reviews the LRD concept and the estimation of Hurst parameter through the R/S method. Moreover, it summarizes the procedure for constructing a FARIMA process, i.e. it goes over the order selection, the estimation of the fractional difference parameter *d* through the Geweke Porter-Hudak estimator, as well as the estimation of the remaining parameters. Section 3 discusses the concept of non-linearity and its detection through the White Noise test. Moreover, it reviews Artificial Neural Networks, their use in time series prediction and their building procedure using concepts from dynamic systems. Finally, sections 4 and 5 present the application of all previously discussed concepts to actual Internet traffic data and proceeds to a comparison of the models using the measures RMSE and MAE.

2 Long Range Dependence and Hurst Exponent

When measured at smaller timescales (from microseconds to minutes), Internet traffic displays burstiness, which means large variability even after several aggregations over time. This characteristic can be described with the help of self-similarity. Moreover, a self-similar process exhibits long memory while the reverse is not always true. Hurst exponent on the other hand, is a common tool to characterize time series and measure LRD in them.

The concept of LRD in time series data is related to the decay of autocorrelation function, which in the case of long memory is slower than exponential. For a stationary stochastic process X_t the autocorrelation function is defined as $\rho(k) = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma^2}$, where μ and σ are the mean and standard deviation of X_t, respectively and provides a way to characterize the type of dependence encompassed in the process. Moreover, it is viewed as a measure of similarity between the series X_t and X_{t+k}. If the autocorrelation function decays to zero with an exponential rate, then the stochastic process exhibits short range dependence. On the contrary, if it decays to zero very slowly so that $\sum_{k=1}^{\infty} \rho(k) = \infty$, then X_t exhibits long memory.

In practice this implies that for large values of *k*, the autocorrelation function has approximately power-law shape $\rho(k) = k^{-\beta},\ 0 < \beta < 1$. The presence of correlation even after very large number of lags in LRD processes furnishes them with quite different properties compared to others. Specifically, for LRD processes there is no timescale where the assumption of independence may hold even approximately and the main measure for long memory in time series is the Hurst exponent. More detailed analysis of these notions can be found at (Crovella and Krishnamurthy, 2006).

Hurst exponent (*H*) was introduced in 1951 by Edwin Hurst in the field of hydrology, and since then it has been used in a variety of fields. When $0.5 < H < 1$, the series exhibits long memory. The closer the value is to 1, the stronger the long memory. The value $H = 0.5$ indicates absence of long memory (uncorrelated series or series with autocorrelation function that decays exponentially to zero). A value in the range of $0 < H < 0.5$ indicates antipersistence (e.g. high values are more likely to be followed by low values and vice versa). For values $H > 1$ the only conclusion one can make is the probable non stationarity of the series.

Most studies related to Internet traffic modeling use Hurst exponent to measure the strength of LRD in the series. The exponent may be estimated with a number of different methods; however for this work the well-known R/S method is used. Details for this estimation method can be found in (Peters, 1994).

2.1 ARIMA vs FARIMA Model Building

Time series models such as ARIMA have been extensively used to describe Internet traffic. However, the LRD characteristic cannot be captured by ARIMA, therefore FARIMA models, which can describe both types of dependence (SRD and LRD), is the indicated alternative solution. FARIMA are extensions of ARIMA (*p*, *d*, *q*) models, where the parameter *d* may take real (not only integer) values.

A FARIMA time series model is formulated as:

$\Phi_p(L)\,(1-L)^d(X_t) = \Theta_q(L)\varepsilon_t$, where L is the lag operator, $\Phi_p(L) = 1 - \varphi_1 L - \ldots - \varphi_p L^p$ and $\Theta_q(L) = 1 + \theta_1 L + \ldots + \theta_q L^q$.

Moreover, $(1-L)^d = \sum_{j=0}^{\infty} \binom{d}{j} (-1)^j L^j$, $\qquad \binom{d}{j}(-1) = \frac{\Gamma(-d+j)}{\Gamma(-d)\Gamma(j+1)}$

and the error terms ε_t are Normal White Noise (0, σ^2)

The procedure of fitting a FARIMA model to a traffic trace as described in (Liu et al, 1999) comprises the following steps:

***Step 1. Convert data to a zero-mean series*:**

The simpler transformation is to subtract the sample mean from the traffic trace.

Step 2. Specify the order of the model

In this step the order of the model is determined. For our implementation presented in the next section we restrict the order of autoregressive and moving average order to be less than or equal to 5 (i.e. $0 \le p \le 5,\ 0 \le q \le 5$). Since there would be 25 possible ARMA models we use the lowest Bayes Information Criterion (BIC) for selecting one of them. The order of the selected model would be the order for the proposed FARIMA model. The BIC criterion, which is also called Schwartz Bayesian criterion, is given by:

$$BIC = \ln(s^2) + n\ln(T)$$

Where s^2 is the variance estimate of the residuals, n the number of parameters to be estimated and T the number of observations.

Step 3. Estimate the parameters of the model

After the order of the model is fixed, the rest of the parameters (*d*, φ_i and θ_j) are estimated using the Geweke and Porter-Hudak(GPH) estimator for *d* and a Maximum Likelihood (ML) methodology for the others, under the assumption that the stationary, fractionally integrated series follows normal distribution.

a. Estimate the fractional parameter *d*

Geweke and Porter-Hudak proposed a non-parametric estimator for the fractional parameter *d* only, without specifying the other coefficients of the model (Geweke and Porter-Hudak, 1983). When the long-memory effect is present, the estimator is expected to take values between 0 and 0.5. However, in some cases the GPH estimator fails to do and then the parameter *d* can be estimated together with the remaining parameters, using the ML estimation methodology that follows.

b. Estimate the remaining parameters for the FARIMA model

The estimation of the coefficients of a FARIMA model can be realized using the recursive ML procedure suggested by Sowell (1992). This method is a one-step procedure where all parameters (including *d*) can be estimated; however the procedure is somewhat simpler if *d* is already known. The strategy in that case is to employ the exact ML for all parameters (including *d*). The evaluation of the likelihood function

$$f(X_N, \Sigma) = (2\pi)^{-N/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2} X_N' \Sigma^{-1} X_N\right\}$$

is performed through a recursive procedure, while the covariance matrix has to be written in terms of the model parameters. The ML estimation goes through nonlinear optimization using nlminb optimizer or augmented Lagrange method with r package rugarch (Ghalanos , 2012).

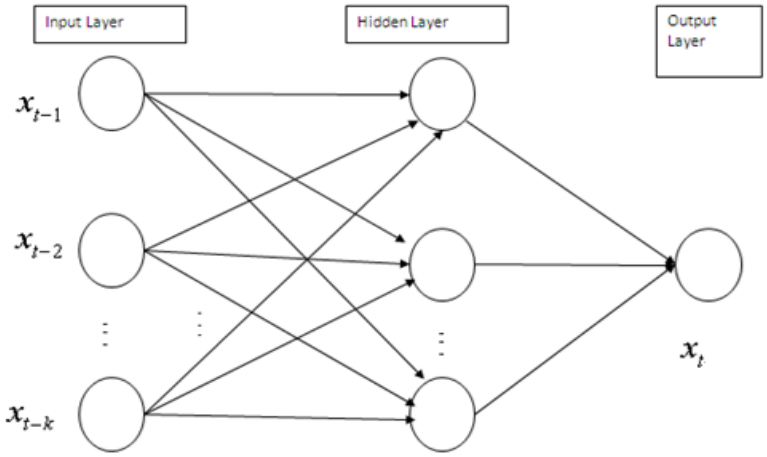
3 Non-Linearity and Neural Networks Framework

Both ARIMA and FARIMA models considered previously assume there is linearity in mean. In case this is not true, the predictability of these models is limited. In such occasions ANN can be found very useful. Moreover, there are tests based on ANN which detect non-linearity.

Neural networks are flexible structures based on brain information processing and appear quite suitable for time series forecasting. An introduction to neural nets can be found in (Lippmann, 1987) and an overview of bibliography in (Zhang et al., 1998).For our purpose the main advantage of NN is that they can model non-linear phenomena, while this is not possible with ARIMA or FARIMA. The most widely used architecture for NN is that of a multilayer perceptron and studies such as (Ding et al., 1995; Cortez et al., 2012) use this approach. Such a network is a feed-forward NN that contains the input layer, one or more hidden layers and the output layer. For our study here, a monotone multilayer perceptron network with 1 hidden layer will be used (Fig. 1).

In order to make predictions the basic ingredients are the input variables, the number of hidden layers and the number of nodes to each layer. The inputs and the number of nodes have to be identified and for that reason, we have borrowed concepts and tools from the dynamical systems. Based on those a sliding window with time lagged variables to be used for each forecast will be defined. Another issue will be to determine the relation between inputs and outputs in the NN and by this induct non-linearity. This role is undertaken by the activation function, with the most popular being a special case of sigmoid, the logistic activation function (Zhang et al., 1998). On the contrary, for the output, a linear function is a popular choice. Lastly, the training algorithm to be used is the adaptive gradient descent method (Haykin, 1999).

Fig.1. The Neural Network Architecture



3.1 Concepts from Dynamical Systems and relation with Time-Series

A time series can be considered as a dynamical system, for which the next state is expressed as a function of its current state. Especially for a discrete time series, this is expressed as: $\boldsymbol{x}(t+1) = F(\mathbf{x}(t))$. From this point of view, a time series may change over time with a regular and simple shape based on previous values or may evolve to a chaotic attractor (non-predictable behavior). In that case, there would be no periodicities, the behavior would be more complex and its evolution cannot be described in practical way. The goal is to identify a simpler system from which the data could have come from. The reconstruction of an original phase-space (time series) $\boldsymbol{x}(t)$, is performed with the use of a transformation Φ which depicts each point of the original space to the reconstructed one. Moreover, the dimension of the reconstructed phase-space must have the same dimension as the original space.

The embedding theorem (Takens, 1981; Mañé, 1981) claims that the space of time lagged vectors with sufficiently large dimension will capture the original time series. This dimension is called *embedding dimension* and gives the number of lagged variables that are needed to reconstruct satisfactorily the original time series. The transformations Φ and Φ^{-1} have to be continuous and 1-1 and in order to achieve that, the embedding dimension *m* has to be $m > 2d$, where *d* is the original dimension. In reality, however there is no way to identify the actual variables that contribute to the state vector (Frank et al, 2001). Thus, there are heuristics that decide on an embedding dimension, such as the False Nearest Neighbors (FNN) method, which is used later for our study. For more details on this subject, one may look into (Abarbanel and Kennel, 1993). In practice, if for a time series the embedding dimension is decided to be $m = 3$, then it is assumed that the value of X_t depends on X_{t-1}, X_{t-2} and X_{t-3}.

Another important issue to phase-space reconstruction is the determination of the time lag *k*. When *k*>1, the time series is considered as over-sampled, and a sampling with a sampling rate *k* must take place. This step is the first that has to be performed. The time lag has to be chosen in a way that keeps a balance between small value and great independence. There are many methods that may help us decide about the time lag, but in this work the concept of mutual information will be used. The procedure with the mutual information can be seen as the analog of the correlation function but in a non-linear setting. The scope is to have the lowest average mutual information between variables. More details about this concept can be found in sources such as (Kantz and Schreiber, 2004). Practically, if the minimum of $I_\varepsilon(t)$ has been found in lag 10 for example, then the series will have to be resampled every 10 points to obtain more accurate forecasts. However, if the 1st lag for example leads to great decrease of $I_\varepsilon(t)$ but not to a minimum, then it is preferred to deem a resampling rate of 1 and not to resample the data. This step results in obtaining maximal information from the data and more accurate forecasts.

3.2 Construction of the Neural Network

As we present in the next section the following steps will be followed to construct an appropriate ANN for each dataset:

Step 1. ***Reconstruction of the time series based on Mutual Information***.

We use the Mutual Information concept to decide whether resampling is needed and on a resampling rate for the dataset. We then reconstruct the time series.

Step 2. ***Selection of Input Variables based on the Embedding Dimension***.

Using the FNN procedure, we decide about the embedding dimension *m* that each dataset carries. This step suggests the lagged variables that should be introduced into the model, so we decide about the number of nodes in the input layer.

Step 3. ***Decide about nodes in the hidden layer and training epochs***

We consider one hidden layer and decide for 1, 2, 10, 20, or 50 nodes based on a minimum RMSE during training, calculated for the 100 most recent observations of the training sample. The training is performed using back-propagation with the adaptive gradient descent algorithm and for 500 epochs of training. Moreover, the activation function is sigmoid for the hidden layer and linear for the output.

The final topology of the ANN will consist of an input layer with I nodes, one hidden layer with H nodes and an output layer with one node, denoted as (I , H ,1).

3.3 Testing for Non-Linearity

The existence of non-linearity in time series is the main reason for using ANN structures instead of the classical forecasting models. Given a dataset however, it is not obvious whether it carries non-linear dependence, therefore a statistical test that can support the assumption of linearity or non-linearity is considered as a valuable tool. There have been developed statistical tests towards this direction and for our study we will use the White Neural Network test for non-linearity (Lee et al, 1993).

This test is based on an ANN structure and its null hypothesis is that the time series model that creates the dataset is linear. This type of test is consistent against arbitrary nonlinearity in mean. The statistic of the test is assumed to follow Chi-square distribution (NIST,2006).

4 Data Analysis and Traffic Modeling

4.1 Overview, Exploratory Analysis of Data and Statistical Characteristics

The procedure discussed previously is now applied to model traces from LAN or WAN traffic and also VBR or MPEG-4 video traffic at different levels of aggregation. All traces are publicly available, either from the Internet Traffic archive at

Bellcore(Leland and Wilson, 1991; Paxson and Floyd, 1995) or from TU-Berlin (Frank et al., 2004). Table 1 gives overview information for each trace and specifically its source, type, level of aggregation (unit of measure and time scale) and volumes of data for the training and test set respectively. Moreover, it provides information regarding the split we used to create training and test sets. Table 2, on the other hand, displays the descriptive statistics calculated for all traces.

Table 1. Overview of traces

Trace	Source	Type	Unit	Time Scale	Training (Test) Set
Aug89	Bellcore	LAN	Mbytes	10 seconds	264 (50)
Oct89	Bellcore	LAN	Mbytes	seconds	1260 (500)
LBL_PK4	Bellcore	WAN	Mbytes	10 seconds	300 (60)
From Dusk till Down	TU Berlin	MPEG4	Kbytes	frames	4000 (1000)
Die Hard III	TU Berlin	MPEG4	Kbytes	frames	4000 (1000)
Jurassic Park	TU Berlin	MPEG4	Mbytes	seconds	3000 (600)
Star Wars IV	TU Berlin	MPEG4	Mbytes	seconds	3000 (600)
The Firm	TU Berlin	VBR	Kbytes	frames	4000 (1000)
Mr. Bean	TU Berlin	VBR	Kbytes	frames	4000 (1000)

Table 2. Descriptive statistics of data sets

Trace	Mean	StDev	Skewness	Kurtosis	CV
Aug89	1.3174	0.5657	1.0613	4.1318	0.4294
Oct89	0.2990	0.1173	0.5133	3.5531	0.3925
LBL_PK4	0.3637	0.2139	1.5893	5.5847	0.5882
From Dusk till Down	3.1838	2.0555	1.4073	6.6582	0.6456
Die Hard III	3.1248	2.3684	1.4953	5.3769	0.7579
Jurassic Park	0.1002	0.0452	0.9637	4.2010	0.4507
Star Wars IV	0.0335	0.0128	1.1006	6.4343	0.3823
The Firm	1.6140	1.2616	1.8510	6.8871	0.7817
Mr. Bean	2.2049	1.0725	2.3069	13.8533	0.4864

All traces exhibit right skewness and leptokurtosis. Furthermore, looking at the Coefficient of Variation (CV) we may conclude that Star Wars IV and Jurassic Park traces have less dispersion than the other video traces, while Oct89 trace has less dispersion than the other Ethernet traces.

4.2 Characteristics of Data over Time

The statistical characteristics of each dataset over time are crucial for building the appropriate forecasting model. For example, stationarity would imply that the parameters of the probability distribution that describe the data can be assumed constant over time. Using the Augmented Dickey-Fuller (ADF) unit root test (Fuller, 1976) all datasets was found to be stationary. Moreover, the data was checked for randomness

and autocorrelation using the runs test and the Ljung-Box test (Ljung and Box, 1978), respectively. We concluded that all datasets display auto-correlation and are clearly not random.

Moreover, to obtain evidence about the existence of long-memory in the time series the Hurst exponent (H) was estimated using the R/S method. Finally, the White Neural Network test was performed to test for non-linearity in mean. The null hypothesis is that the time series is linear. Table 3 displays the results.

Table 3. Existence of Long-Memory and Non-linearity Tests

Trace	**H (R/S Method)**	**White NN Test*(p-value)**
Aug89	**0.6032113**	1.2628 (**0.5318**)
Oct89	0.9104218	7.0662 (0.0292)
LBL_PK4	0.8415698	18.2919 (<0.01)
From Dusk till Down	**0.5771654**	177.2802 (<0.01)
Die Hard III	0.6978816	351.3466 (<0.01)
Jurassic Park	0.9746704	0.4978 (**0.7796**)
Star Wars IV	0.8147262	4.8191 (**0.0899**)
The Firm	0.8252875	3.1556 (**0.2064**)
Mr. Bean	0.7888182	53.4466 (<0.01)

* Performed using a chi square distribution and the model order (# of lags) is 1.

The estimation of Hurst exponent indicates that there is long memory for all datasets, while specifically for *Aug89* and *From Dusk till Down* traces the presence of long memory is not very strong. Testing for non-linearity, on the other hand, concludes that for traces *Aug89*, *Jurassic Park*, *Star Wars IV* and *The Firm* linearity cannot be rejected at the 0.05 level of significance.

4.3 Alternative Models and Forecasting Performance

Next using the methodologies of sections 2.1 and 3.2 for every trace we select the "best" ARIMA and FARIMA model as well as the "best" topology for an ANN model. Table 4 displays the selected parameters for the three alternative models while Table 5 gives their one-step ahead forecasting performance measured by RMSE and MAE on the test set.

Table 4. Selected forecasting models

Trace	ARIMA	FARIMA	ANN
Aug89	(1, 1, 1)	(1, **0.3961873**, 1)	**(4, 20, 1)**
Oct89	(1, 1, 1)	(1, **0.440625**, 1)	**(9, 50, 1)**
LBL_PK4	(0, 1, 4)	(0, **0.3955196**, 4)	**(7, 50, 1)**
From Dusk till Down	(4, 1, 2)	(4, **0.00000001**, 2)	**(6, 10, 1)**
Die Hard III	(5, 1, 2)	(5, **0.00000001,** 2)	**(10, 20, 1)**
Jurassic Park	(1, 0, 2)	(1, **0.2732444**, 2)	**(10, 20, 1)**
Star Wars IV	(2, 1, 3)	(2, **0.1785253**, 3)	**(10, 50, 1)**
The Firm	(0, 1, 1)	(0, **0.5**, 1)	**(6, 10, 1)**
Mr. Bean	(2, 1, 2)	(2, **0.3777264**, 2)	**(10, 10, 1)**

,

Table 5. Forecasting performance of models

Trace	ARIMA RMSE MAE	FARIMA RMSE MAE	ANN RMSE MAE
Aug89	0.5677 0.4540	**0.4275** **0.3211**	0.4460 0.3418
Oct89	**0.1062** 0.0856	0.1087 0.0882	0.1077 **0.0852**
LBL_PK4	0.2131 0.1478	0.2059 0.1468	**0.1894** **0.1421**
From Dusk till Down	1.1634 0.6785	1.0259 0.7024	**1.0146** **0.5899**
Die Hard III	0.7834 0.5414	0.7544 0.5290	**0.5135** **0.3045**
Jurassic Park	0.0177 0.0126	**0.0175** **0.0123**	0.0427 0.0374
Star Wars IV	0.0411 0.0384	**0.0087** **0.0053**	0.0123 0.0092
The Firm	0.3994 0.1804	0.3962 0.1870	**0.3712** **0.1751**
Mr. Bean	0.9195 0.4611	0.9156 0.4632	**0.8937** **0.4248**

According to the results in Table 5 the ARIMA models give the worst performance for all traces except from Oct89 trace, where the RMSE is slightly smaller. FARIMA models outperform all others at August 89, Jurassic park and Star Wars traces, while the ANN forecasting models are more accurate at the rest of the cases.

Combining the results in Tables 3 and 5 it is easy to see that when it comes to model selection we can extract a rule that seems to improve performance: Apply the White Neural Network test and in case of significant deviation from linearity select the ANN model, otherwise use the "best" FARIMA model. Applying such an approach for our nine datasets, we will end up with the best model at all cases except from The Firm trace. It is interesting to point out here that according to Table 2 the Firm trace displayed also the greatest dispersion (according to CV) compared to the

other traces. So it is possible that large deviations from the mean value indicate that a linear model may not be as good recommendation as a non-linear one.

5 Conclusions

In this work, we compare methodologies that build ARIMA, FARIMA or ANN forecasting models for Internet traffic. The datasets used for the study give either LAN, WAN or Video traffic and come from publicly available sources. The FARIMA model is popular for capturing long-memory when it exists in a time series. The ANN is recommended when there is non-linearity in the mean and the ARIMA model is used mainly for comparison and due to its popularity. According to our study the test for non-linearity should drive our model selection process. If the test suggests significant deviation from linearity then the ANN model should be used, while if this is not the case the FARIMA model is the best choice. Applying such a rule for our datasets would lead to the best models for 8 out of 9 traces.

References

1. Abarbanel H. and Kennel M.(1993). *Local false nearest neighbors and dynamical dimensions from observed chaotic data*, Physical Review E, **47(5):** pp. 3057– 3068
2. Aladag, C.H., Egrioglu, E., and Kadilar C. (2012). *Improvement in Forecasting Accuracy Using the Hybrid Model of ARFIMA and Feed Forward Neural Network*, American Journal of Intelligence Systems, **2(2)**:pp. 12–17
3. Asteriou D. and Hall S.G. (2011). *ARIMA Models and the Box–Jenkins Methodology*. Applied Econometrics (Second ed.). Palgrave MacMillan. pp. 265–286.
4. Beran J., Sherman R., Taqqu M.S. and Willinger W. (1995).*Variable bit-rate video traffic and long range dependence*, IEEE Transactions on Communications, **43(2/3/4)**: pp. 1566-1579
5. Balkin S.D. and Ord J.K. (2000). *Automatic neural network modeling for univariate time series*, International Journal of Forecasting, **16**: pp. 509–515
6. Corradi M., GarroppoR.G., Giordano S. and Pagano M. (2001).*Analysis of f-ARIMA processes in the modeling of broadband traffic*, ICC'01, **3**: pp. 964-968
7. Cortez P., Rio M., Rocha M. and Sousa P. (2012).*Multi-scale Internet traffic forecasting using neural networks and time series methods*, Expert Systems, **29** (2): pp. 143-155
8. Crovella M. and Krishnamurthy B. (2006). Internet Measurement: Infrastructure, traffic and applications,John Wiley and Sons Ltd.
9. Ding X., Canu S. and DENOEUX T. (1995). *Neural network based models for forecasting*, In Proceedings of Applied Decision Technologies Conference (ADT'95). Uxbridge, UK: pp. 243–252
10. Frank R.J., Davey N. and Hunt S.P. (2001). *Time Series Predictions and Neural Networks*, Journal of Intelligent and Robotic Systems, **31(1-3)**:pp. 91-103
11. Frank H.P., Fitzek and Martin Reisslein. (2004). *MPEG-4 and H.263 Video Traces for Network Performance Evaluation, IEEE Network*, **15(6)**: pp. 40-54.
12. Fuller W.A. (1976). *Introduction to Statistical Time Series*, New York: J Wiley & Sons.
13. Ghalanos A.(2012). rugarch: Univariate GARCH models. R package version 1.2-7.
14. Granger, C. W. J. and Joyeux, R. (1980).*An introduction to long-memory time series models and fractional differencing*, Journal of Time Series Analysis**1**: pp.15–30

15. Geweke G., and Porter-Hudak S. (1983). *The Estimation and Application of Long Memory Time Series Models*, Journal of Time Series Analysis, **4 (4)**: pp. 221-238
16. Hosking J.R.M. (1981). *Fractional differencing*, Biometrika, **68**: pp. 165-176.
17. Haykin S. (1999). Neural Networks – a Comprehensive Foundation. Prentice Hall, New Jersey, 2nd edition
18. Kantz H. and Schreiber T. (2004). *Nonlinear Time Series Analysis*, Cambridge University Press, 2nd edition
19. Lee T.H., White H., Granger C.W.J. (1993).*Testing for neglected nonlinearity in time seriesmodels*, Journal of Econometrics, **56**: pp. 269–290.
20. Leland W.E., Taqqu M.S., Willinger W. and Wilson D. (1994).*On the self-similar nature of Ethernet traffic (extended version)*, IEEE/ACM Transactions on Networking, **2**: pp.1-15
21. Lippmann, R. P. (1987). *An introduction to computing with neural nets*.IEEE ASSP Mag. **4**: pp. 4–22, April.
22. Liu J., Shu Y. Zhang L., Xue F., Oliver W. and Yang W. (1999).*Traffic Modeling Based on FARIMA Models*, Canadian Conference on Electrical and Computer Engineering - CCECE, **1**: pp. 162-167
23. Ljung G.M., Box G. E. P. (1978). *On a Measure of a Lack of Fit in Time Series Models*, Biometrika**, 65 (2**): pp. 297–303
24. Mañé R. (1981). *On the dimension of the compact invariant sets of certain nonlinear maps*, In D. A. Rand and L.-S. Young.Dynamical Systems and Turbulence, Lecture Notes in Mathematics, **vol. 898**. Springer-Verlag. pp. 230–242
25. NIST/SEMATECH (2006). *Engineering Statistics Handbook - Chi-Squared Distribution*
26. Paxson V. and Floyd S. (1995). *Wide-Area Traffic: The Failure of Poisson Modeling*, IEEE/ACM Transactions on Networking, **3(3)**: pp. 226-244
27. Paxson V. (1997). *Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic*, Computer Communication Review, **27(5)**: pp. 5-18
28. Peters, E.E. 1994. Fractal Market Analysis: Applying Chaos Theory to Investment and Economics. Brisbane: John Wiley and Sons Inc.
29. Shu Y., Jin Z., Zhang L., Wang L., Oliver W. and Yang W. (1999).*Traffic prediction using FARIMA models*, IEEE Int.Conf. on Commun., **2**: pp.891–895
30. Sowell F. (1992).*Maximum likelihood estimation of stationary univariate fractionally integrated time series models*. J. Econometrics**, 53 (1-3**): pp.165-188.
31. Takens F. (1981). *Detecting strange attractors in turbulence*, In D. A. Rand and L.-S.Young.Dynamical Systems and Turbulence, Lecture Notes in Mathematics, vol. 898. Springer-Verlag. pp. 366–381.
32. Won Y., Ahn S., (2005). *GOP ARIMA: Modeling the nonstationarity of VBR processes*, Multimedia Systems, **10 (5)**: pp.359-378
33. Wang C., Zhang X., 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). Adelaide, Australia, pp. 367–370
34. Zhang G., Patuwo B. E., and Hu, M.Y. (1998). *Forecasting with artificial neural networks: the state of the art*, International Journal of Forecasting**, 14(1)**: pp. 35–62.
35. Zhang G. (2003). *Time series forecasting using a hybrid ARIMA and neural network model*, Neurocomputing, **50**: pp. 159-175.
36. Zhani M.F. and Elbiase H. (2009).*Analysis and Prediction of Real Network Traffic*, Journal of Networks,**4 (9)**: pp. 855-865