

Modelling the Multi-Layer Artificial Neural Network for Internet Traffic Forecasting: The Model Selection Design Issues

Mba O. Odim
Computer Science Department
Redeemer's University
Ede, Nigeria
odimm@run.edu.ng

Jacob A. Gbadeyan
Mathematics Department
University of Ilorin
Ilorin, Nigeria
jagbadeyan@unilorin.edu.ng

Joseph S. Sadiku
Computer Science Department
University of Ilorin
Ilorin, Nigeria
jssadiku@unilorin.edu.ng

ABSTRACT

Internet traffic forecasting models with learning ability, such as the artificial neural network (ANN), have been growing in popularity in recent time due to their impressive performance in modelling the high degree of variability and nonlinearity of internet traffic. This study examined the impacts of some design issues on performance of the multi-layer artificial neural network for internet traffic forecasting. The traffic forecasting was modelled as a standard time series problem and the multilayer artificial neural network designed to performs the time series function mapping. The input lags were varied from 1 to 24. The training epoch values of 200, 500, and 1000 on one and two hidden layered networks were used. The learning algorithm was *backpropagation* with 0.1 learning rate and 0.9 momentum on logistic sigmoid activation function. The model was implemented in Visual Basic and validated with four categories of classified time series internet traffic of a branch residential network of one of a firm in Nigeria. Various predictive performances without consistent pattern were observed on the issues considered, however, input lag one gave the worst performance in all cases for the HOURLY traffic; three of the four traffic categories demonstrated the superiority of two hidden layers to one hidden layer. Although the epoch values of 200, 500 and 1000 showed no consistent performance variations, epoch value 200 outperformed the others on the model selections. The study revealed that input lags, number of hidden layers and epoch values could impact on the traffic forecasting performance of multilayer perceptron and that performance could be considerably improved by careful selection of those parameters through experimentations.

CCS Concepts

• **Computing Methodologies** → **Artificial Intelligence** → **Machine learning** → **Machine learning approaches** → **Neural Networks**

Keywords

Internet Traffic, Times Series Forecasting, Machine learning, Multi-layer Artificial Neural Network, Design issues

CoRI'16, Sept 7–9, 2016, Ibadan, Nigeria.

1. INTRODUCTION

Accurate information about offered traffic is required for efficient resource provisioning and general capacity planning of an Internet service. The inability of most statistical methods in modelling the high variability of internet traffic accurately, and their lack of reasoning capabilities have triggered an increased number of studies that employ non-traditional statistical methods including machine learning. Furthermore, traditional summary statistics, particularly the sample mean and variance are instable metrics for working with the high variability of internet traffic, as such the sample means and sample variances are not reliable statistics for summarising traffic properties [1]. Machine learning techniques, such as the Artificial Neural Network (ANN) employ mechanisms that allow computers to evolve behaviour based on knowledge gained from dynamic observations. Machine learning technique based on nonlinear elements is often referred to as Neural Network. Neural networks are networks of nonlinear elements interconnected through adjustable weights and they play a prominent role in machine learning. Artificial (ANN) emerged with the aim of imitating the information processing process of human brain. Through learning, ANNs can determine nonlinear relationship in a data set by associating the corresponding output to input patterns. The multilayer artificial neural network, among other machine learning models, has shown impressive results in forecasting studies [2, 3, 4, 5, 6, 7, 8, 9]. However, applying an ANN to a given forecasting endeavour is a hard task, as basic modelling issues must be carefully considered for enhanced precision. The issues include the network architecture, learning parameters and data pre-processing methods [6, 8]). The inconsistencies in performance reports on the design issues in the literature was noted also in [8]. In [9] it was argued that ANN technique should not be applied arbitrarily as has been sometimes suggested and even used in the internet forecasting domain [10, 11].

The paper examined the impacts of number of input lags, hidden neurons and training epochs on the precision of the multilayer artificial neural network in forecasting internet traffic.

1. RELATED WORK

Quite a number of research efforts has been reported in the literature on seeking appropriate models for forecasting Internet traffic.

2.1 Internet Traffic Forecasting: Statistical Methods

In [12] a comparative study on suitable statistical methods for network traffic estimation was conducted. In the paper, several estimation methods for IP network traffic were studied. The study

showed that non-linear time series models could model and forecast better than the classical linear time series models. Anand in [13] investigated a non-linear Time series model, the Generalised Autoregressive Heteroskedasticity (GARCH) in internet traffic modelling. The model showed that the forecasting algorithm was accurate compared with actual traffic. Although nonlinear statistical models can capture the burstiness of network's traffic, the models are parametric in nature and therefore require the knowledge of the distribution of the traffic. In addition, they are analytical and therefore require explicit programming to clearly specify the algorithmic steps. To take the advantages of machine learning paradigms, applying machine learning techniques to internet traffic forecasting has been on the increase.

2.2 Machine Learning and Artificial Neural Network for Internet Traffic Forecasting

A vast number of research efforts have been on going in exploring machine learning techniques to internet traffic predictions, the results of which have demonstrated their superiority to statistical forecasting methods. A concurrent neuro-fuzzy model to discover and analyse useful knowledge from available Web log data was proposed in [14]. The study used self-organizing map for pattern analysis and a fuzzy inference system to capture the chaotic trend to provide short term (hourly) and long term (daily) web traffic trend predictions. Empirical results demonstrated that the proposed approach was efficient for mining and predicting web traffic. A study in [15] presented a neural network ensemble (NNE) for the prediction of TCP/IP traffic using time series forecasting (TSF) point of view. The NNE approach was compared with TSF methods (Holt -winter and ARIMA) and the NNE was found to compete favourably with the TSF methods. In [16] the least square support vector machines was applied to solve the problem of accurately predicting non-peak traffic and the method had a good generalization ability and guaranteed global minima. [17] Presented a neural network ensemble approach and two adapted time series methods (ARIMA and Holt-Winters) for forecasting the amount of traffic in TCP/IP based networks. The experiments with the neural ensemble achieved the best results for 5 min and hourly data, while the Holt-Winters was the best option for the daily forecasts. The study in [10] investigated the ensembles of artificial neural networks in predicting long-term internet traffic. The proposed prediction models were compared with the classic method of Holt-Winters. Prangchumpol in [18] presented a description approach to predicting incoming and outgoing data rate in network system by using a data (machine learning) mining techniques, the association rule discover. The result of the study showed that the technique could predict future network traffic.

2.3 Design Issues with Forecasting with Artificial Neural Network

A detailed state of the art presentation on forecasting with artificial neural networks was made in [8]. The study showed that overall; ANNs gave satisfactory performance in forecasting, but went on to indicate the inconsistencies in performance reports of design issues in the literature. The inconsistencies were attributed to trial and error methodology adopted in most studies. Faraway and Chatfield [9] argued that it was unwise to apply ANN models blindly in black box mode as had sometimes been suggested. Shamsuddin, *et al.* in [7] investigated the effect of applying

different number of input nodes, activation functions and pre-processing techniques on the performance of backpropagation network in time series revenue forecasting. The findings showed that the performance of ANN model could be considerably improved by careful selection of those parameters. In [19], the performance of two learning algorithms: the linear regression and Neural Network Standard Back propagation were compared on the prediction of four major stock market indexes. The comparison showed that the neural network approach resulted in better prediction accuracy than the Linear Regression model. Chabaa *et al* in [20] presented an ANN based on the multi-layer perceptron for analysing a time series measured internet traffic data over IP networks. The comparison between some training algorithms demonstrated the efficiency and accuracy of the Levenberg Maquardt and the Resilient back propagation algorithms. Chukwuchekwa in [21] compared the performance of the back propagation gradient descent technique and genetic algorithm on some pattern recognition problems. The backpropagation (BP) algorithm was found to outperform the genetic algorithm in that instance. The study suggested that caution should be applied before using other algorithms as substitutes for the BP algorithm, more especially in classification problems. In [2], an evaluation of several learning rules for adjusting ANN weights was carried out on the popular airline passenger data set. The Levenberg-Marquardt backpropagation algorithm showed the best performance among other learning rule. Various degrees of performances were observed in [22] on examining the impact of input lags of the multilayer perceptron in forecasting internet traffic on a two layered network. In [23] a survey of research and application issues on Web usage mining based on various mining technique was conducted to provide some understanding in designing algorithms suitable for mining data.

This review demonstrated the impressive results of applying machine learning technique, such as the artificial neural networks, in forecasting Internet traffic as well as raising concerns over the little or no consideration given by researchers on the design issues. The paper therefore presents results from the study on the impacts of some multi-layer perceptron design issues on internet traffic forecasting.

3. METHODOLOGY

The traffic forecasting was modelled as a standard time series problem and the multilayer artificial neural network designed to performs the time series function mapping.

3.1 Time series for Traffic Forecasting

Traffic forecasting is a standard time series prediction task. The goal is to approximate the function that relates the future values of a variable of the previous observations of that variable [24]. In some situations, such as internet traffic, data are non-stationary and chaotic. In such situation, one general assumption is that historical data incorporate all behaviour required to capture the dependency between the future traffic and that of the past. Therefore, the historical data is the major player in the forecasting process. The second assumption to model and forecast the dynamic of the traffic is that its values are expressed by discrete time series [2, 3]. A discrete time series is a vector $\{y_t\}$ of observations made at regular intervals, $t=1, 2, 3, \dots, N$. For the time series forecasting problem, the inputs are typically the past observation of the data series and the output is the future value.

Suppose y_1, y_2, \dots, y_N denote an observed time series of the traffic loads, then the basic problem is to estimate future traffic value such as y_{N+k} , where the integer k is called the *lead time* or *the forecasting horizon* [25]. For the univariate method, forecasts of a given traffic load are based on a model fitted only to the past observations of the given time series, so that $\hat{y}_t(N, k)$ depends only on y_1, y_2, \dots, y_{N-1} . The estimate of y_{N+1} is computed as a weighted sum of the past observations:

$$\hat{y}_{N+1} = w_0 y_N + w_1 y_{N-1} + w_2 y_{N-2} + \dots \quad (1)$$

where the $\{w_i\}$ are weights.

The Multi-Layer Perceptron performs the following function mapping [3, 8]:

$$\hat{y}_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-n}) \quad (2)$$

where \hat{y}_t is the estimated traffic at time t , $(y_{t-1}, y_{t-2}, \dots, y_{t-n})$ denotes the training pattern composed of a fixed number (n) of lagged observations of the series.

The weight to be used in the ANN model are estimated from the data by minimizing the sum of squares of the within-sample one-step ahead forecast errors, namely

$$S = \sum_t (\hat{y}_t - y_t)^2 \quad (3)$$

over the first part of the time series, called the *training set*. The last part of the time series called *the test set*, is kept in reserve so that genuine out of sample (*ex ante*) forecasts can be made and compared with the actual observations. Equations (1) and (2) give a *one-step-ahead forecast* as it uses the actual observed values of all lagged variables as inputs. If multistep-ahead-forecasts are required, then it is possible to proceed in one of two ways. Firstly, construct a new architecture with several outputs, giving $\hat{y}_t, \hat{y}_{t+1}, \hat{y}_{t+2}, \dots$, where each output would have separate weights for each connection to the neurons. Secondly, ‘feedback’ the one-step-ahead forecast to replace the lag 1 value as one of the input variables, and the same architecture could then be used to construct the two-step-ahead forecast, and so on [16]. This study adopted the latter iterative approach because of its numerical simplicity and because it requires fewer weights to be estimated.

3.2 The Multilayer Neural Network

Neural network is a powerful model for solving complex problems because it has natural potential of solving nonlinear problems and can easily achieve the input-out mapping, it is good for solving predicting problems [26]. The basic features of the multilayer perceptrons include:

- i. The model of each neuron in the network includes a nonlinear activation function that is differentiable.
- ii. The network contains one or more layers that are hidden from both input and output nodes.

- iii. The network exhibits a high degree of *connectivity*, the extent of which is determined by synaptic weights of the network.

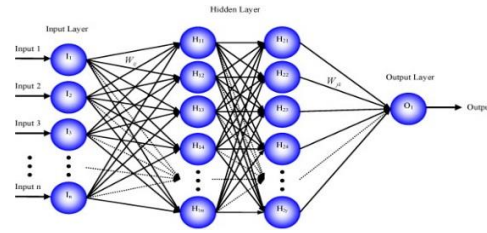


Figure. 1. Architecture of Multilayer Perceptron with two hidden layers

3.2.1 A Neural Model

The node is the basic unit of the Artificial Neural Network. Each node is able to sum many inputs x_1, x_2, \dots, x_n from the environment or from other nodes, with each input modified by an adjusted node weight (Figure 2). The sum of these weighted inputs is added to an adjustable threshold for the node and then passed through a modifying (activation) function that determines the final output.

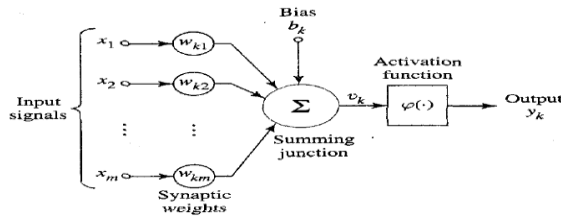


Figure. 2. Nonlinear model of a neuron [26]

The neural model in Figure 2 includes an externally applied bias, denoted by b_k . The bias has the effect of increasing or lowering the net effect of the activation function, depending on whether it is positive or negative, respectively. Mathematically, we may describe the neuron k depicted in Fig. 2 by the following equations:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (4)$$

and

$$y_k = \varphi(u_k + b_k) \quad (5)$$

where x_1, x_2, \dots, x_m are the input signals; $w_{k1}, w_{k2}, \dots, w_{km}$ are the respective synaptic weights of neuron k . u_k is the *linear combiner* output due to the input signals, b_k is the ‘‘bias’’, $\varphi(\cdot)$ is the *activation function*, and y_k is the output signal of the neuron. The use of the bias b_k has the effect of applying *affine transformation* to the output v_k of the linear combiner in the model this is shown by

$$v_k = u_k + b_k \quad (6)$$

The bias b_k is an external parameter of neuron k .

The activation function, denoted by $\varphi(v)$ defines the output of a neuron in terms of induced local field v . It is this function (also

called, *the transfer function*) that determines the relationship between inputs and outputs of a node and a network. In general, the activation function introduces a degree of nonlinearity that is valuable for most ANN applications. Among these functions, sigmoid function is very popular. It is a strictly increasing function that exhibits a graceful balance between linear and nonlinear behaviour. The Logistic Sigmoid is defined as in (5)

$$\varphi(v) = \frac{1}{1 + \exp(-v)} \quad (7)$$

A logistic sigmoid function assumes a continuous range of values from 0 to 1. Additional types of activation functions can be found in [8]. Among these functions, logistic transfer function is the most popular choice [8].

3.2.1 Training of artificial neural networks

ANN has to be trained before it can be put to use. The goal of the training is to find the logical relationship from the given input/output. There are two strategies of the learning: **supervised** and **unsupervised**. This study employs the supervised learning strategy. Supervised learning typically operates in two phases – training and test set. The training set is used for estimating the arc weights while the test set is used for measuring the generalization ability of the network. Training is used to gain generalised knowledge about the system under consideration and testing is used to predict (forecast) the system behaviour using the knowledge gained. On the other hand, unsupervised techniques such as the reinforcement learning is independent of training data and operate by directly interacting with the environment.

The training algorithm employed is the Backpropagation. It is a supervised training strategy and popular method for training the multilayer perceptron. The training proceeds in two phases [26]:

1. In the *forward phase*, the synaptic weights of the network are fixed and the input signal is propagated through the network, layer by layer, until it reaches the output. Thus, in this phase, changes are confined to the activation potentials and outputs of the neurons in the network.
2. In the *backward phase*, an error signal is produced by computing the output of the network with desired response. The resulting error signal is propagated through the network, again layer by layer, but this time the propagation is performed in the backward direction. In this second phase, successive adjustment is made to the synaptic weights of the network.

In [5] it is also reported that the backpropagation is the most computationally straightforward algorithm for training the multi-layer perceptron. They summarized the algorithms steps as

1. Obtain a set of training patterns
2. Set up ANN model that consist of number of input neurons, hidden neurons, and output neurons
3. Set learning rate (h) and momentum rate (a)
4. Initialize all connections (W_{ij} and W_{jk}) and bias weights (q_k and q_j) to random values.
5. Set the minimum error E_{min} /number of epochs
6. Start training by applying input pattern one at a time and propagate through the layers then calculate total error

7. Back-propagate error through output and hidden layers and adapt W_{ij} and q_j .
8. Back-propagate error through hidden and input layer and adapt weights W_{ij} and q_j .
9. Check if Error < E_{min} or max epoch reached. If not, repeat steps 6 – 9, otherwise, stop training.

3.3 Data collection and Description

Internet traffic data was collected in hourly average kilo bit/s of TCP/IP traffic of a company's resident network from January 1 2010 to September 30 2010 (making up 6552 data points each for IN and OUT traffic data), daily traffic data from January 1 to December 31, 2010 (making up 365 data points each for IN and OUT traffic data), using PRTG ([Paessler Router Traffic Grapher](#)), a network monitoring and bandwidth usage tool from a company called PAESSLER. 20Mbps bandwidth was allocated for upload (Traffic IN) and 20Mbps for download (traffic out) statically for the period under consideration.

3.3.1 Data Pre-processing/Normalisation

Nonlinear activation functions such as the logistic function typically restricts the possible output from a node to, typically, (0, 1) or (-1, 1). This is to avoid computational problems, to meet algorithm requirement and to facilitate network learning. Four methods for input normalization are summarized in [8]. This study employs, the Linear transformation to [0, 1], defined as

$$y_n = (y_0 - y_{min}) / (y_{max} - y_{min}) \quad (8)$$

where y_n and y_0 represent the normalized and original data: y_{min} , y_{max} , are the minimum, maximum of the column or rows respectively.

3.3.2 Training and Testing set

Eighty percent (80%) of the data, that is, 5241.6 approximated to 5242 was used for training the network, while twenty per cent (20%), that is, 1310.4, approximated to 1310, was used for testing the generalisation predictive capability of the network each for the HOURLY_IN and HOURLY_OUT flow traffic. Also, a training set of 80% and testing set of 20% were used for each of the DAILY traffic, that is. 292 data points for training and 73 for testing.

3.4. Finding the appropriate complexity of the Network

For times series forecasting problem, a training pattern consists of a fixed number of lagged observations of the series [7]. The inputs (number of lag observations) were varied from 1 to 24, excluding the bias. One and two hidden layers were considered. The number of hidden nodes were equalled to the number of input nodes. In several studies, networks with the number of hidden nodes being equal to the number of input nodes are reported to have better forecasting [8]. One output node was used, one look-ahead. So the model of our network is $k, k, k, 1$, where k represents the number of lag observations (input variables). The epochs were based on 200, 500, and 1000. The best model according to [18] is the one that gives the best result in the test set. The logistic sigmoid activation function was used [8]. The Error correction backpropagation algorithm with learning rate: 0.1; momentum: 0.9 was used to train the network.

3.5 Stopping and Evaluation Criteria

The training stop after each epoch respectively. Typically, as SSE based objective function or cost function to be minimized during the training process is defined in (10). The measure of accuracy employed is the Root Mean Square error (RMSE) defined as

$$RMSE = \sqrt{\frac{1}{n} \sum_i (\hat{y}_i - y_i)^2} \quad (9)$$

where n is the total number of sample group observations, \hat{y}_t is the predicted (computed) value while y_t is the target value at time t . RMSE is one of the most commonly used measure of forecast error to examine how close the forecast is to the actual value [5]. The best model is the one that gives the best result in the test set, that is, the model that has the least RMSE in the testing set [27].

4. RESULTS AND DISCUSSION

The system was implemented in visual basic. The RMSE of various models were recorded and compared based on the design issues considered. The results are presented and discussed in this section.

4.1 HOURLY_IN traffic

The RMSE of the testing (prediction) results of the various models based on the number of input lags, number of hidden layers and training epochs on one and two hidden layers network respectively were compared for the HOURLY_IN traffic. Figure 3 depicts these results.

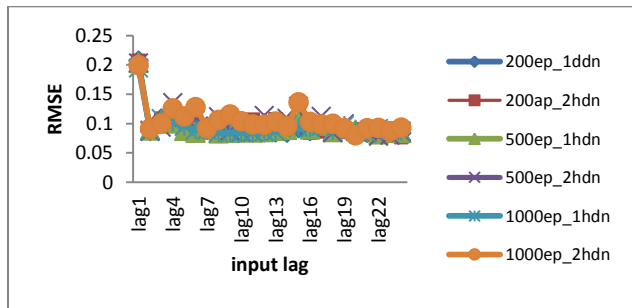


Figure 3. ANN model selection for the HOURLY_IN traffic

There were varying degrees of performance with no regular patterns of performance among the input lags, between the one and two hidden layers networks, and among the various epochs used. Nevertheless, the worst performance for all the cases is input lag 1. The least RMSE with the value 0.0766984 of this experiment occurred at input lag 24 with 200 training epochs on two hidden layer network. Therefore, the best model for forecasting the HOURLY_IN traffic is input lag 24, 200 training epochs using two hidden layers.

4.2 HOURLY_OUT traffic

The RMSE of the testing (prediction) results of the various models were compared for the HOURLY_OUT traffic. The results are shown in Figure 4.

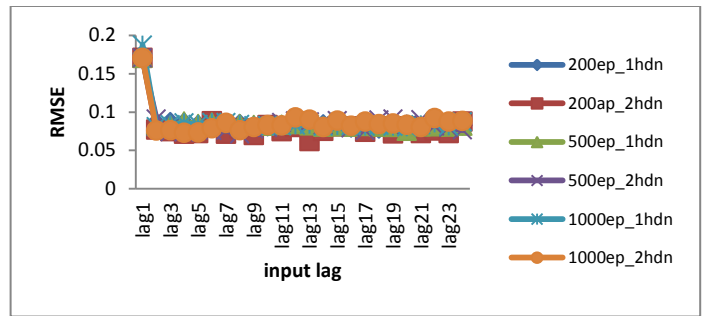


Figure 4. ANN model selection for the HOURLY_OUT traffic

There also various values of the performance measure with no particular patterns on the issues for the HOURLY_OUT traffic. As in the HOURLY_IN, the worst performance for was recorded at input lag 1 in all the cases. The least RMSE with the value 0.0621992 occurred at input lag 13 with 200 training epochs on two hidden layer network. Therefore, the best model for forecasting the HOURLY_OUT traffic is input lag 13, 200 training epochs using two hidden layers.

4.3 DAILY_IN traffic

Figure 5 presents the prediction RMSE of the various models for the DAILY_IN traffic.

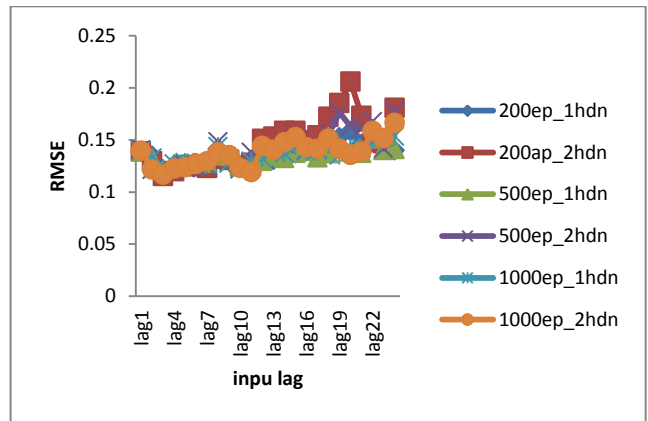


Figure 5. ANN model selection for the DAILY_IN traffic

Different performance values were also observed with no particular patterns on the various prediction models. The least RMSE with the value 0.116691 of this experiment occurred on input lag 3 with 1000 training epochs on two hidden layer network. Therefore, the best model for forecasting the DAILY_IN traffic is input lag 3, 1000 training epochs using two hidden layers.

4.4 The DAILY_OUT traffic.

Figure 6 presents the RMSE of the various prediction models for the DAILY_OUT traffic.

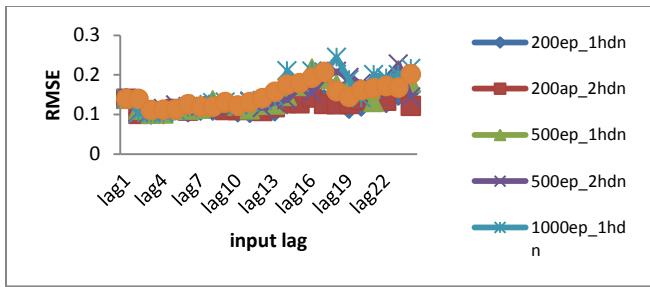


Figure 6. ANN model selection for the DAILY_OUT traffic

No particular patterns of performance was observed among the various models, although there were different values of the performance measure. The least RMSE with the value 0.099416 of this experiment occurred on input lag 3 with 200 training epochs on one hidden layer network. Therefore, the best model for forecasting the DAILY_OUT traffic is input lag 3, 200 training epochs using one hidden layer.

Table 1 presents a summary of the model selection for forecasting the various traffic categories.

Table 1: Summary of the Forecasting Model Selection for the traffic Categories

	Hidden layers	Epochs	Learning rate	momentum	Input lags	Least RMSE
HOURLY_IN	2	200	0.1	0.9	24	0.07669
HOURLY_OUT	2	200	0.1	0.9	13	0.06219
DAILY_IN	2	200	0.1	0.9	3	0.11521
DAILY_OUT	1	200	0.1	0.9	3	0.09941

For the HOURLY_IN traffic the traffic computed (predictive) values based on 24 input lags on two hidden layer network using 200 training epochs was deployed, and for the HOURLY_OUT traffic, the study used 13 input lags of the traffic computed values of the testing set on 200 training epochs on two hidden layers network designed to perform. The study deployed 3 input lags, two hidden layers of 3 neurons each using 200 training epochs for predicting the DAILY_IN traffic. For the DAILY_OUT traffic, 3 input lags, one hidden layer of three neurons with 200 training epochs were selected.

Figure 7 compares the predicted models selected for the traffic categories.

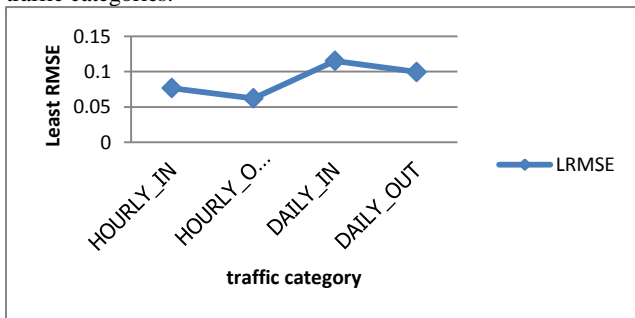


Figure 7. Summary least RMSE for model selection of the traffic categories

The HOURLY traffic categories had a better prediction performance than the DAILY traffic counterparts. This could have been attributed to the very large sample size used for the HOURLY traffic. It has been reported that the ANN for forecasting perform better with large sample size than with small sample size (Zhang *et al.* 1998 [8] and Zhang, *et al.* [26]). In addition, figure 7 revealed that various forecasting models may exist for different traffic categories, even if the traffic categories are all from the same network operator.

This study has observed different forecasting models for the various traffic categories based on the issues. The findings suggest that carefully consideration of the design issues is indispensable for improving the predictive performance of a multi-layer artificial neural network rather than applying it to internet traffic forecasting blindly. However, there are no generally acceptable techniques for determining the optimal design parameter but by experimentations, an improved predictive performance model is feasible.

5. CONCLUSION

This study examined the impacts of some important design issues in modelling a multilayer perceptron artificial neural network for Internet traffic forecasting. The traffic forecasting was modelled as a standard time series problem and a multilayer artificial neural network designed to performs the time series function mapping. The mechanism was implemented in a Visual Basic programming environment and tested with real Internet traffic data through experimentation with the various design issues considered. Although no particular pattern of performance was observed the study showed that the forecasting performance can be affected by the number of input lags, hidden layers and training epochs,. Despite that the study did not make any attempt to determine an optimal values for the various factors considered, it has shown that careful experimentation is required to choose appropriate values for each of the design issues. Therefore, the multilayer perceptron should not be applied blindly to Internet traffic forecasting.

6. REFERENCES

- [1] Crovella, M. and Krishnamurthy B. 2006. *Internet Measurement*. John Wiley & Sons, Ltd., England.
- [2] Benkacha, S., Benhra J., and El Hassani, H. 2015. Seasonal Time Series Forecasting Models on Artificial Neural Network. *International Journal of Computer Applications*. 116, 20, 0975-8887, DOI=: 10.5120/20451-2805
- [3] Benkacha, S., Benhra, J. and El Hassani, H. 2013. Causal Method and Time Series Forecasting Model based on Artificial Neural Network. *International Journal of Computer Applications*. 75, 7, 0975 – 8887.
- [4] Islam, S., Keung, J., Lee K. and Liu, A. 2012. Empirical prediction models for adaptive resource provisioning in the cloud. *Future Generation Computer Systems*. 28, 155-162. DOI= 10.1016/1.future2011.05.027
- [5] Chabaa, S. Zeroual, A. and Antari, J. 2010. Identification and prediction of internet traffic using artificial neural networks. *Journal of Intelligent Learning Systems & Applications*. 2, 147-155. DOI=1.4236/jilsa.2010.23018
- [6] Shamsuddin, S. M., Sallehuddin R., and Yusof, N. M. 2012. Artificial neural network time Series modelling

- for revenue forecasting. *Chiang Mai J. Sci.* 35, 3, 411-426.
- [7] Cortez, P., Rio, M. Sousa, P. and Rocha M.. 2007. Topology aware internet forecasting using neural networks. In *Proceedings of the 17th International Conference on Artificial Neural Networks* (Porto, Portugal), *Lecture Notes in Computer Science* 4669, 445-452, Springer.
- [8] 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, 35 – 62.
- [9] Faraway, J. and Chattfield, C. 1998. Times series forecasting with neural networks: a comparative study using the airline data. *Journal of Appl. Statist.* 47, 231 – 250.
- [10] Miguel, M. L. F., Penna, M. C., Nievola, J. C. Pellenz and M. E. 2012. .New models for long-term internet traffic forecasting using artificial neural networks and flow based information. In *Proceedings of 2012 IEEE Network Operations and Management Symposium*, 1082-1088. DOI= 10.1109/NOMS.2012.6212033,
- [11] 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, 143–155.
- [12] Mariam I., Dadarlat, V. and Iancu, B. 2009. A Comparative Study of the statistical Methods suitable for Network Traffic Estimation. In *Proceedings of the 13th WSEAS International Conference on Communications.* 99-104.
- [13] Anand, C. N. 2009. *Internet traffic modeling and forecasting using non-linear time series model Garch.* M.Sc. Thesis, Department of Electrical and Computing Engineering, College of Engineering, Kansas State University.
- [14] Wang, X., Abraham, A. and Smith, K. A. 2005. Intelligent web traffic mining and analysis, *Journal of Network and Computer Applications.* 28, 147-165.
- [15] Cortez, P., Rio, M., Rocha, M. and Sousa, P. 2006. Internet Forecasting using Neural Networks. In *Proceeding of the International Joint Conference on Neural Network* (Vancouver), 2635 – 2642.
- [16] Zhang, Y. and Liu, Y. 2009. Comparison of parametric and nonparametric techniques for non-peak traffic forecasting, *World Academy of Science, engineering and Technology.* 51,
- [17] 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, 143–155.
- [18] Prangchumpol, D. A. 2013. Network traffic prediction algorithm based on data mining technique. *World Academy of Science, Engineering and Technology.* International Science Index, <http://www.waset.org>.
- [19] Fok, W. W. T, Tam V. W. L. and Ng, H. 2008. Computational neural network for global stock Indexes Prediction, In *Proceedings of World Congress on Engineering* (London, UK, July 2 -4, 2008).
- [20] Chabaa S., Zeroual, A., and Antari, J. 2010. Identification and Prediction of Internet traffic using artificial neural networks. *Journal of Intelligent Learning Systems & Applications.* DOI=1.4236/jilsa.2010.23018.
- [21] Chukwuchekwa, U. J. 2011. Comparing the performance of backpropagation algorithm and genetic algorithms in pattern recognition problems. *International Journal of Computer Information Systems.* 2, 5, 7-12.
- [22] Odum, M. O., Gbadeyan J. A. and Sadiku J. S. 2014. A neural network model for improved internet service resource provisioning. *British Journal of Mathematics & Computer Science.* 4, 17, 2418-2434,
- [23] Dogne, V., Jain, A. and Jain, S.. 2015. Evolving trends and its application in web usage mining: a survey. *International Journal of soft computing and engineering.* 4, 6, 98-101,
- [24] Rutka, G., and Lauks, G. 2007. Study on internet traffic prediction models. *Electronics and Electrical Engineering. – Kaunas: Technologija,* 6, 78, 47–50,
- [25] Chatfield, C. 1992. *The analysis of time series: An introduction (4th ed.)*. Chapman & Hall, London:.
- [26] Haykin, S. 2009. *Neural networks and learning machines (3rd ed.)*. Pearson Education, Inc, New Jersey.
- [27] Zhang, G. P. Patuwo, B. E, and Hu, M. Y. A. 2001. Simulation Study of Artificial Neural Networks for Nonlinear Time-series Forecasting, *Computer & Operations research,* 128, 381-396.