British Journal of Mathematics & Computer Science
4(17): 2418-2434, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

A Neural Network Model for Improved Internet Service Resource Provisioning

M. O. Odim[1*], J. A. Gbadeyan[2] and J. S. Sadiku[3]

[1]*Department of Mathematical Sciences, Redeemer's University, Nigeria.*
[2]*Department of Mathematics, University of Ilorin, Nigeria.*
[3]*Department of Computer Science, University of Ilorin, Nigeria.*

Original Research Article

Received: 14 July 2013
Accepted: 15 September 2013
Published: 26 June 2014

Abstract

The paper seeks for a good forecast model that can accurately represent the inherent characteristics of Internet traffic and forecast the desired traffic load to satisfy the performance target using Artificial Neural network technology. We developed a computational tool in Visual Basic 6.0 for this purpose, based on a Multi-layer network. By making use of an empirical study to examine the effect of some ANN model design issues such as impact of lag observation and the number of neurons of a two hidden layer network on Internet traffic prediction, the results shows that ANN is a powerful forecast modeling tool that can accurately capture the inherent traffic characteristics and forecast the desired traffic load and that these factors have greater impact on the performance ANN forecaster.

Keywords: Internet traffic, times series forecasting, multi-layer artificial neural network, resource provisioning, quality of service.

1 Introduction

The Internet has become increasingly an indispensable part of all aspects of our modern society. It is now the hob driving various forms of human activities. It is the backbone for the emergence of modern computing platforms such as cloud computing, on-line services, e-commerce, among others. But the Quality of service (QoS) requirement of Internet services is of increasingly important. Depending on the type of system considered, a failure to meet the QoS requirements can lead to serious financial losses, customers and reputation, and in some cases even to loss of human lives [1]. The perceived performance and user satisfaction of an Internet service depends to a large extent on the availability and sufficiency of critical resources to accommodate the smooth and efficient flow of traffic load from customers. Hence, to ensure that such resources do not constitute bottlenecks, ideally, an Internet service should be assigned necessary and sufficient amount of resources to handle its current load.

**Corresponding author: odimm@run.edu.ng;*

A quick solution is to increase the capacity of the resources, but this approach is not sustainable because the resource requirement is not constant and the budget of resources is limited. In addition traffic data experiences high degree of variability. So during some period there is low level of activities and at some other period the activities are high. Meent in [2] reported that traditional bandwidth provisioning formulae were based merely on a rule of thumb; for instance, "the mean traffic rate, plus a margin of 30%"; and that such a fixed margin is not universally applicable, nor does it guarantee no needless waste of resources on the one hand, and availability of sufficient resources on the other hand. There is therefore a problem of under forecasting as well as over forecasting. Under forecasting leads to under-provisioning and this results to restricting resources and hence loss of information and poor QoS. Over forecasting on the other hand, leads to over-provisioning which results to wastage of resources. Therefore a proactive prediction-based resource provisioning mechanism that can accurately capture the inherent traffic characteristics and forecast the desired traffic load is required in order to cope with the ever-fluctuating resource usage pattern of Internet applications.

The paper explores a machine learning approach, particularly, an Artificial Neural Network (ANN) predictive model for Internet resources provisioning problem. 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. However, designing an ANN for forecasting is not a trivial matter, several design issues must be careful taken into consideration to obtain an optimum model for prediction. Some of these issues are considered in the paper.

The paper seeks for a good forecast model that can accurately represent the inherent traffic characteristics and forecast the desired traffic load to satisfy the performance target.

1.1 Related Work

1.1.1 Traffic forecasting and resource provisioning

Analytical performance models were used in [3] to design controllers that dynamically switch servers from one service environment to another as needed. In [4] the problem of appropriate resource allocation for various service environments in generic large-scale utility computing infrastructures was addressed. [5] propose a middleware for controlling performance and availability of cluster-based multi-tier systems, using analytical model. [6] designed a queuing-theoretical methods to provision servers in the service tier with profit optimization model. [7] propose a model driven server switching policy to dynamically allocate server resources in enterprise network. [8] presented a self-managing technique that jointly addresses the resource and admission control optimization problems in virtualized servers. A Joint cross layer approach between application layer and MAC layer for enhancing Quality of Service (QoS) for multimedia applications was proposed in [9]. [10] developed prediction based resource measurement and provisioning strategies using Neural Network and linear Regression. Their scheme was experimented on CPU usage flow.

1.1.2 Traffic modelling and forecasting

In [11] a neural network ensemble (NNE) for the prediction of TCP/IP traffic using time series forecasting (TSF) point of view was presented. The NNE approach was compared with TSF methods (e.g. Holt -Winter and ARIMA) and the NNE was found to compete favorably with the TSF methods. [12] reports that Internet traffic forecasting is now receiving attention from forecasting domain and that applying Artificial Neural Network to forecasting is becoming promising forecasting alternative. [13] compare the Levenberg Marquardt and the Resilient back propagation and observed that the two algorithms can be successfully used for analyzing internet traffic over an IP network. [14] Applied least square support vector machines to solve the problem of accurately predicting non-peak traffic and concluded that the method has a good generalization ability and guarantees global minima. [15] proposed a concurrent neuro-fuzzy model to discover and analyze useful knowledge from available Web log data.

1.1.3 Traffic forecasting and bandwidth provisioning

Markov model-based bandwidth estimation for future time interval was proposed in [16] to address the time-varying traffic in ATM virtual circuit. The adaptive bandwidth control schemes is studied in [17] with respect to time dependent Poisson traffic using a point-wise stationary fluid-flow approximation technique. [18] proposed a dynamic provisioning architecture for intra-domain Quality of service model, by developing an efficient bandwidth algorithm that takes explicitly into account traffic statistics to increase the user's benefit and the network revenue simultaneously. A new service model that provides quantitative per-flow bandwidth guarantees was proposed in [19]. Meent [2] developed a link dimensioning approach, which explicitly incorporates the offered traffic in terms of both the average rate as well as its fluctuations at small time scale, and desired level of performance. He proposed some mathematical formulas that obtain the required bandwidth capacity. The ARCH model was employed in [20] for forecasting and dynamically bandwidth provisioning. To address the limitation of the ARCH model, [21] proposed the GARCH model for traffic forecasting.

Most of the researches are based on the assumption that the nature of the traffic is known and those based on artificial neural networks do not explicitly study the impact of design issues in the selection of a model for Internet traffic forecasting. In addition, the models were implemented on third party software, in which the designers have little or no control of.

We developed a computational platform, based on a Multi-layer network. By making use of an empirical study, we examine the impact of lag observation and the number of neurons of a two hidden layer network on Internet traffic prediction. We believe that, the results of this study will effectively contribute to enhancing several Internet management related tasks such as general resource management procedures. In addition, researchers and practitioners can employ this tool to monitoring and administering network traffic.

2. Methodology

The multi-layer perceprton (MLP) neural network was used for the study because of their impressive results in forecasting [10,22,23,24,12]. 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 (25). The basic features of the multilayer perceptrons (Fig. 1.), according to [25] are:

- The model of each neuron in the network includes a nonlinear activation function that is differentiable.

- The network contains one or more layers that are hidden from both input and output nodes.
- The network exhibits a high degree of connectivity, the extent of which is determined by synaptic weights of the network.

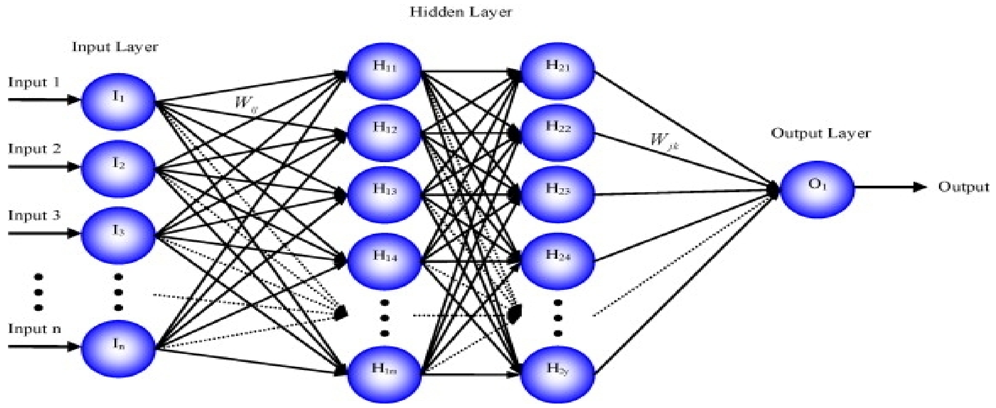

Fig. 1. An architecture of a multilayer network with two hidden layers and one output neuron

2.1 A Neural Model

The basic unit of the Artificial Neural Network is the node. Each node is able to sum many inputs x_1, x_2, …,x_n form the environment or from other nodes, with each input modified by an adjusted node weight (Fig. 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.

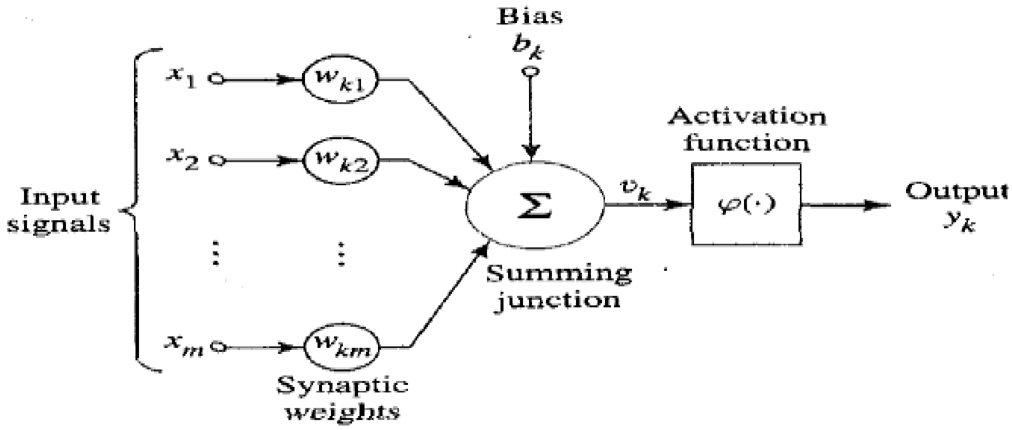

Fig. 2. Nonlinear model of a neuron [25]

The neural model in Fig. 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:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

and

$$y_k = \varphi(u_k + b_k) \tag{2}$$

where x_1, x_2, ..., x_m are the input signals; w_{k1}, w_{k2}, ..., 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", φ (.) 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

$$v_k = u_k + b_k \tag{3}$$

The bias b_k is an external parameter of neuron k. We may formulate the combination of equations (1) to (3) as follows:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \tag{4}$$

and

$$y_k = \varphi(v_k) \tag{5}$$

Therefore the model may be reformulated as in Fig. 3

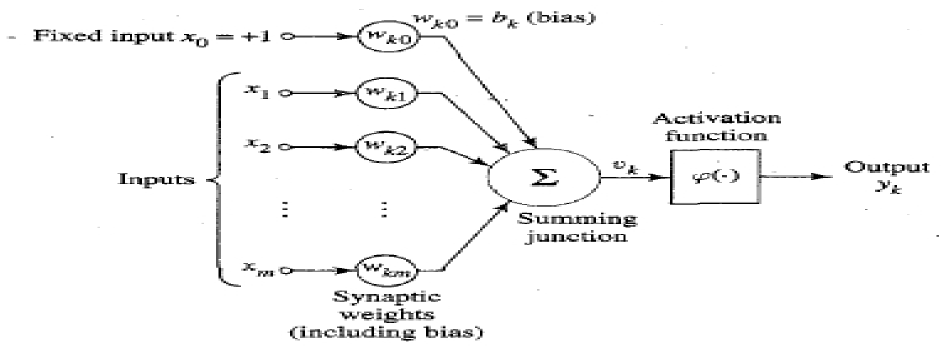

Fig. 3. A nonlinear model of a neuron w_{k0} accounts for the bias [25]

Although the models of Fig. 2 and Fig. 3 are different in appearance, they are mathematically equivalent.

The activation function, denoted by φ *(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 behavior. The sigmoid come in two basic forms: The logistic and the hyperbolic tangent function.

- The Logistic Sigmoid is defined as in (5)

$$\varphi(v) = \frac{1}{1+\exp(-v)} \tag{6}$$

A logistic sigmoid function assumes a continuous range of values from 0 to 1.

- The hyperbolic tangent function takes values that range from -1 to +1. For the corresponding form of a sigmoid function, we use the hyperbolic tangent function, defined by

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

Additional types of activation functions can be found in [16]. Among these functions, logistic transfer function is the most popular choice [16].

2.2 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 two strategies of the learning: supervised and unsupervised. We shall employ 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 Back propagation. It is a supervised training strategy and popular method for training the multilayer perceptron. The training proceeds in two phases [25]:

1. In the f*orward 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.

[24] also reported that the backprobagation 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. Backpropagate error through output and hidden layers and adapt W_{ij} and q_j.
8. Backpropagate 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.

2.3 Designing an ANN for Forecasting

Accurate traffic prediction may be used to optimally smooth delay sensitive traffic or dynamically locate bandwidth to traffic streams. The problem of traffic prediction is a standard time series prediction task, the goal of which is to approximate the function that relates the future values of a variable of the previous observations of that variable. For time series forecasting problem, the inputs are typically the past observations of the data series and the output is the future value. The ANN performs the following function mapping

$$\hat{y}_t = f(\hat{y}_{t-1}, \hat{y}_{t-2}, \ldots \hat{y}_{t-p}) \tag{8}$$

where $\hat{y}_t$ is the observation at time.

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

$$S = \Sigma_t (\hat{y}_t - y_t)^2 \tag{9}$$

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. Equation (7) gives a *one-step-ahead forecast* as it uses the actual observed values of all lagged variables as inputs. If a *multistep-ahead-forecasts* are required, then it is possible to proceed in one of two ways. Firstly, we could construct a new architecture with several outputs, giving $\hat{y}_t$, $\hat{y}_{t+1}$, $\hat{y}_{t+2}$, . . . , where each output would have separate weights for each connection to the neurons. Secondly, we could '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. We adopted the latter iterative approach because of its numerical simplicity and because it requires fewer weights to be estimated.

For times series forecasting problem, a training pattern consists of a fixed number of lagged observations of the series [22]. Suppose we have N observations: y_1, y_2, y_3, ...,y_N in the training set and we need 1-step ahead forecasting, then using an ANN with n input nodes, we have N-n training patterns. The first training pattern will be composed of y_1, y_2, y_3, ...,y_n as inputs and y_{n+1} as the target output. The second training pattern will contain y_2, y_3, ..., y_{n+1} as inputs and y_{n+2} as the desired output. Finally, the last training pattern will be y_{N-n}, y_{N-n+1}..., y_{N-1} as inputs and y_N for the target.

2.4 Experimental Set Up

The section presents the preliminary study we have conducted and the next section shows the results of the investigation.

2.4.1 Variable selection

Variations in traffic load measured in hourly average kilo bit/second were selected for this study. This is because bandwidth capacity is measured in kilo bit/s.

2.4.2 Data collection and description

We collected hourly average kbit/s data 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". 20Mpbs bandwidth was allocated for upload (Traffic IN) and 20Mbps for download (traffic out) statically for the period under consideration.

Fig. 4 depicts the bandwidth (hourly average bit/second) traffic utilisation for the period. Except for few cases, the bandwidth was excessively being overprovisioned, particularly for the OUT traffic.

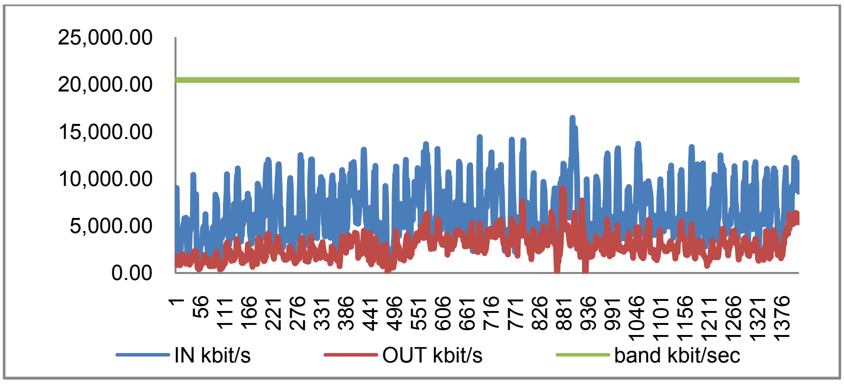

Fig. 4. A segment of Bandwidth utilization for the period January 2010 to September 2010

2.4.3 Data pre-processing/data normalization

Nonlinear activation functions such as the logistic function typically have the squashing role in restricting or squashing the possible output from a node to, typically, (0, 1) or (-1,1). Data normalization is often performed before training process begins. When nonlinear transfer functions are used at the output nodes, the desired output values must be transformed to the range of the actual outputs of the network. Even if a linear output transfer function is not used, it may still be advantageous to standardize the output as well as the inputs to avoid computational problems, to meet algorithm requirement and to facilitate network learning.

Four methods for input normalization are summarized in [22]

For the experiment, we employ Linear transformation to [0, 1], defined as

$$y_n = (y_0 - y_{min})/(y_{max} - y_{min}) \tag{10}$$

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

2.4.4 Training and testing set

For the hourly traffic average kbit/s (for both IN and OUT, respectively), 6552 data points, representing traffic for the months of January to July were used for training (about 78% of the sampled data points), and the data for the months of August and September were used for testing, representing about 22% of the sampled data points. For the daily traffic in k byte (for both IN and OUT, respectively), 292 data points were used for the training, representing 80% of the sampled data, and the remaining 73 data points were used for testing, representing a 20% of the sampled data.

2.4.5 ANN paradigm: 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 [16] Number of inputs: we vary the inputs from 1 lag observation to 21 lags observations, excluding the bias.

1. Number of hidden layers: We used a two layered network for the study.
2. Number of hidden nodes shall be equal 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 [22].
3. Number of output nodes: 1, for one look-ahead.

So the model of our network is k, k, k, 1, where k is the number of lag observations (input variables). The best model according to [18] is the one that gives the best result in the test set.

2.2.6 Activation functions

The activation function used is the logistic sigmoid. Logistic activation (transfer) function is the most popular choice [22].

2.4.7 Evaluation criteria

The training stop after 1000epochs.Typically, as SSE based objective function or cost function to be minimized during the training process is defined in (8).

2.4.8 Training

a. Training algorithm: Error correction back propagation
b. Number of epochs: 1000
c. Learning rate: 0.8; momentum: 0.5

2.4.9 Implementation

a. Implementation tool: Visual Basic 6.0 environment, to take advantage of the visual capability of the environment.
b. Measure of Accuracy employed is the Root Mean Square error (RMSE) defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{t}(y_t - \hat{y}_t)^2} \tag{11}$$

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 k. RMSE is one of the most commonly used measure of forecast error to examine the how close of the forecast to the actual value [24]. The best model according to [23] is the one that gives the best result in the test set, i.e. model that has the least RMSE in the testing set

3. Results and Discussion

We present and discuss the preliminary results obtained from our experiments under the following headings:

3.1 The ANN Forecaster

We developed a simple and easy to use multi criteria multi-layer ANN tool in Visual Basic 6.0, with good interfaces, for the forecasting processes. The interfaces provide users with an ease to use platform, such that they do not have to bother about the complexities associated with coding while using this tool. From the result that follows, the ANN traffic forecaster can learn traffic pattern well, make one-look-ahead forecast (Figs. 5 and 6 for sample training and test patterns respectively). In addition the users' interface incorporated provides an ease to use platform. These include, "automated data pre-processing, data embedding, choice of number of epochs, learning rate and momentum". All these can be done at the interface level.

3.2 Categories of Network Traffic

We present and discuss the results obtained from the empirical study of the various traffic categories studied.

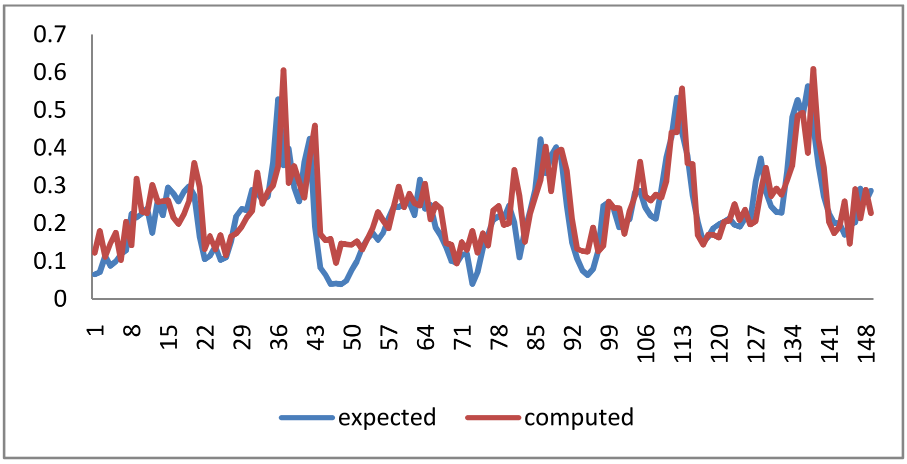

Fig. 5. A segment of the training pattern for the hourly IN traffic with 6 lag observations at different time intervals

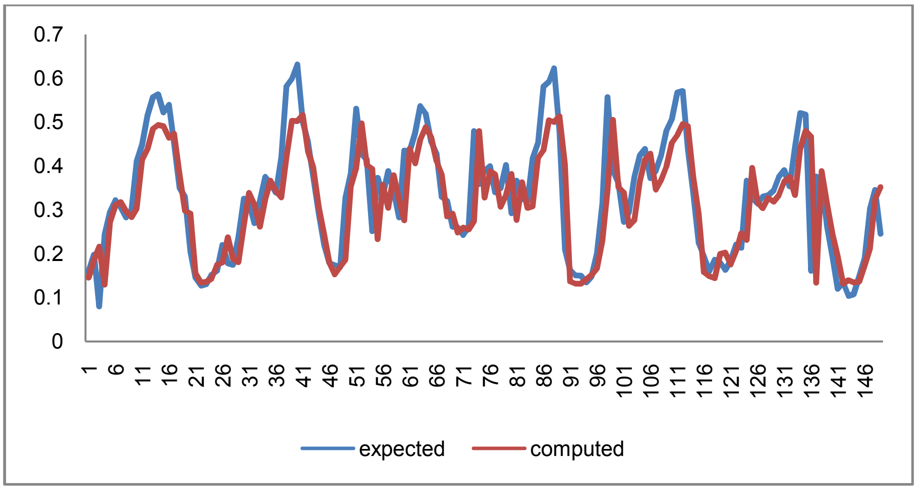

Fig. 6. A segment of the test pattern for the hourly IN traffic with 6 lag observations at different time intervals

3.2.1 Hourly IN traffic

Fig. 7 shows the Root Mean Square Error of the training and test set. The least RMSE for the training and the test occurred at 6 lag observations, with the RMSE of 0.081839 for the training set and 0.087332 for the test set. Therefore, we select the model 6, 6, 6, 1, for the Hourly IN traffic. The RMSE of the training set is lower in most cases than the RMSE *of the test set.* So the system learns the pattern better than it generalizes it.

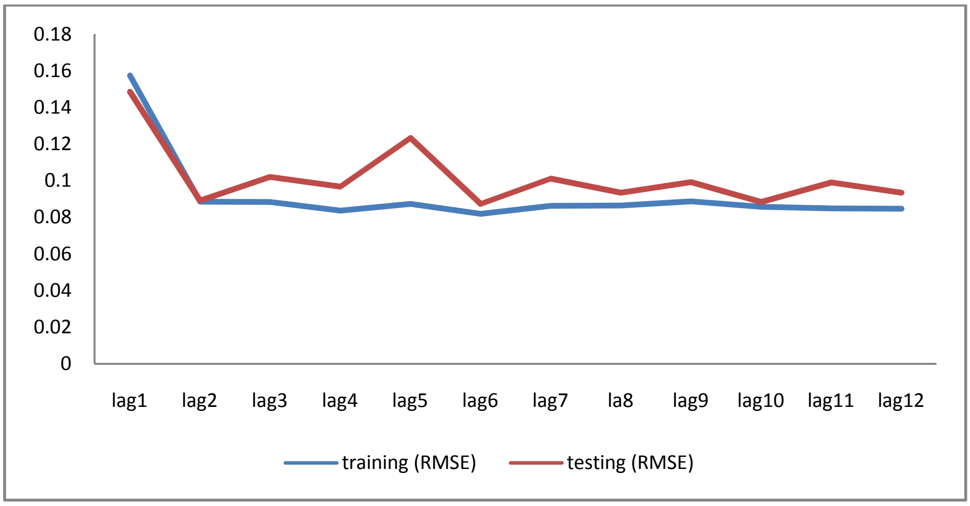

Fig. 7. The root mean square error of the training and test set of the hourly IN traffic at different lag observations

3.2.2 Hourly OUT traffic

Fig. 8 shows the RMSE for the training and testing sets respectively of the Hourly OUT traffic data at the various lags. From the figure, the least RMSE for the training set occurs at lag 10, with RMSE of 0.064096; while the least RMSE for the test set occurs at lag 5, with RMSE of 0.068722. The network with lag 5 observations produces the best result, so choose 5, 5, 5, 1, for the hourly OUT traffic. Again the RMSE of the training is lower than of the test set at the different lags. So the system learns the pattern better than it generalizes it.

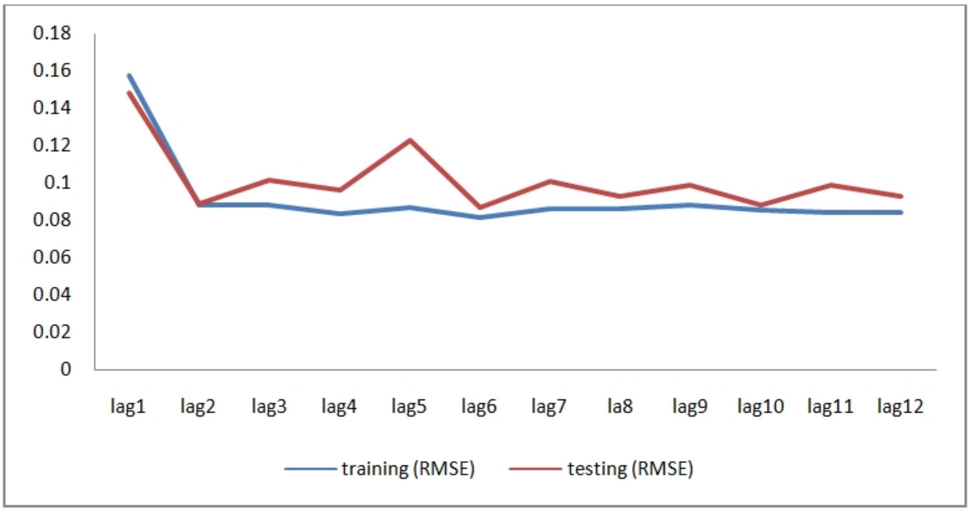

Fig. 8. The root mean square error of the training and test set of the Hourly OUT traffic at different lag observations

3.2.3 Daily in traffic

The RMSE accuracy measure for the models is shown in Fig. 9 for the training and testing sets. The least RMSE for the training set occurs at lag 10, with RMSE of 0.124375; while the least RMSE for the testing set occurs at lag 7, with RMSE of 0.549301. So we select 7, 7, 7, 1 as the best model for the Daily IN traffic. Also, the pattern learning is better than the generalization.

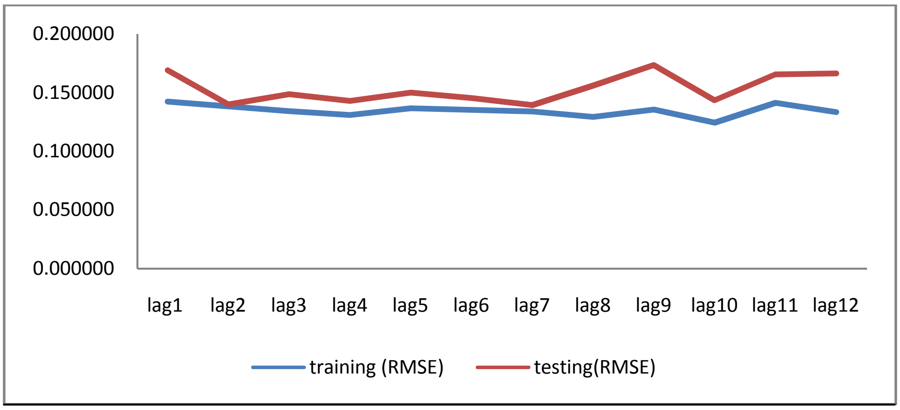

Fig. 9. The root mean square error of the training and test set of the daily IN traffic at different lag observations

3.2.4 Daily out traffic

The RMSE measures for the models are presented in Fig. 10for the training and test sets. The least RMSE for the training set occurs at lag 9, with RMSE of 0.10076486; while the least RMSE for the test set occurs at lag 10, with RMSE of 0.27998710. So we choose the model 10, 10, 10, 1 for the daily OUT traffic. Here the pattern learning is better in most cases, but in quite number of cases, the generalization was better.

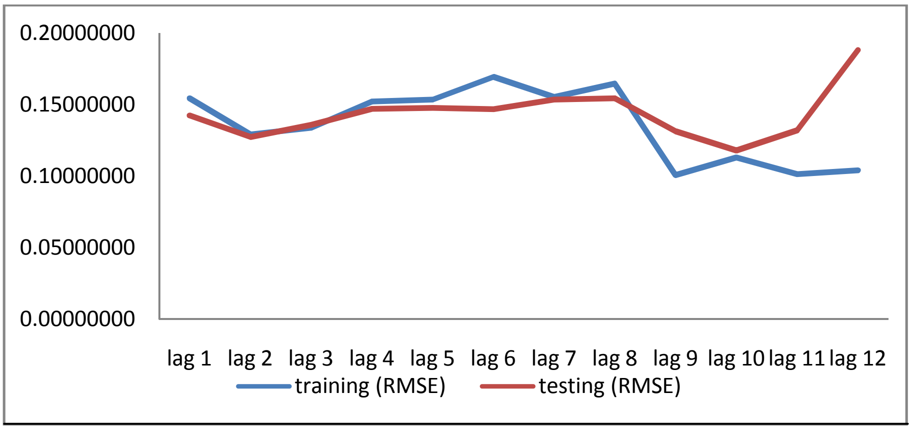

Fig. 10. The root mean square error of the training and test set of the Daily OUT traffic at different lag observations

3.2.5 The four traffic categories compared

The RMSE for the selected models for the four traffic categories are depicted in Fig. 11. The figure shows that, for the selected models, the training ability of the network is better than the generalization. In addition, the hourly traffic learning and generalization appear better than of the daily traffic. This could have been attributed to the sample sizes used. The hourly traffic each used a sample size of 6552 data points. While we used 365 data points only for the daily traffic.

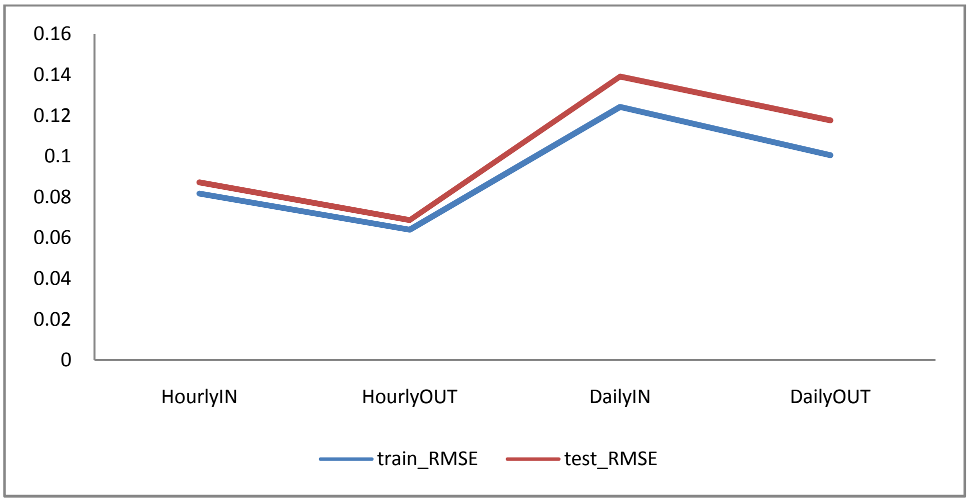

Fig. 11. The least root mean square error of the training and test set of the different network categories

4 Conclusion

A proactive prediction-based resource provisioning mechanism that can accurately capture the inherent traffic characteristics and forecast the desired traffic load is required in order to cope with the ever-fluctuating resource usage pattern of Internet applications. To this extent, we proposed and developed an Artificial Neural Network based predictive model. The result of the study using empirical Internet traffic data show that ANN is a powerful forecast modeling tool that can accurately capture the inherent traffic characteristics and forecast the desired traffic load. The following can be closely observed from the result presented.

- The study has shown that the factors considered have greater impact on the performance ANN forecaster. We have observed various degree of performance among the different architectural models.
- The performance of the Hourly traffic appears better than that of the daily traffic. This could be as a result of the sample size used. For the Hourly traffic, we used 6552 data points each for the IN and OUT traffic, and for the Daily traffic we used only 365 data points each for the IN and OUT traffic. We may therefore need larger data sample size to improve on the performance of traffic forecasting with ANN. However, the cost of using very large sample is that it takes a longer time to train the network.
- The architecture with the best training ability may not be the best for generalization.

Therefore it advisable to always divide the data sets into training and test sets.

- There are different models for various problem domains. For instance, we have observed different models for the Hourly IN traffic, Hourly OUT traffic, Daily IN traffic and Daily OUT traffic.
- From the above, it is important to highlight that not all ANN recommendations may apply to all data domain.

Hence designing An ANN for forecast, the factors need to be duly taken into consideration; the decision on the number of input variables (lags) and number of neurons should not be made blindly. We intend to investigate the impacts of some other factors, such as the choice of training algorithm, activation function, learning rate, and momentum, number of hidden layers and sample size on the performance on Internet traffic. Besides, we shall obtain a resource provisioning method based on our traffic prediction. We believe that, the results of this study will effectively contribute to enhancing several Internet management related tasks such as general resource management procedures. In addition, researchers and practitioners can employ this tool to monitoring and administering network traffic.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Konev S. Performance engineering of distributed component-based systems – benchmarking, modelling and performance prediction. A PhD. dissertation submitted to University of Technology, Darmstad; 2005.

[2] Meent RV. Network link dimensioning: a measurement and modelling based approach. A PhD. Thesis Submitted to Centre for Telematics and Information Technology, University of Twente, The Netherland; 2006

[3] Bennani MND, Menasce A. Resource allocation for autonomic data centers using analytical performance models. In Proc. IEEE Int'l Conf. on Autonomic Computing (ICAC); 2005.

[4] Costa P, Napper J, Pierre GM, Steen V. Autonomous resource selection for decentralised utility computing. In Proc. of Int'l Conference on Distributed Computing Systems; 2009.

[5] Arnaud J, Bouchenak S. Adaptive internet services through performance and availability control. In Proc. of ACM Symposium on Applied Computing. 2010;444-451.

[6] Villela D, Pradhan P, Rubenstein D. Provisioning servers in the application tier for e-commerce systems. ACM Trans on Internet Technology.2007;7(1):1-23.

[7] Xue JW, Chester AP, He L, Jarvis SA. Model-driven server allocation in distributed enterprise systems. In Proc. of Int'l Conference on Adaptive Business Information Systems (ABIS); 2009.

[8] Almeida J, Almeida V, Ardagna D, Francalanci C. Trubian M. Resource management in the autonomic service-oriented architecture. In Proc. IEEE Int'L Conf. On Autonomic Computing (ICAC); 2006.

[9] Bhuvaneswari M, Seethalakshmi P. Application based utility adaptation with prioritised weight assignment strategy for minimizing delay in multimedia applications. International Journal on Computer science and Engineering. 2011;3(4):1672-1680.

[10] Islam S, Keung J, Lee K, Liu A. Empirical prediction models for adaptive resource provisioning in the cloud. Future Generation Computer Systems.2011;8(2012):155-162.

[11] Cortez P, Rio M, Rocha M, Sousa P. Internet Forecasting using Neural Networks. IJCNN. International Joint Conference on Neural Network, Vancouver. 2006;2635–2642.

[12] Cortez P, Rio M, Rocha M, Sousa P. Topology aware internet forecasting using neural networks. Artificial Neural Networks – ICANN, 17th International Conference, Lecture Notes in Computer Science. Porto, Portugal. 2007;4669:445-452.

[13] Chabaa S, Zeroual A, Antari J. Identification and prediction of internet traffic using artificial neural networks. Journal of Intelligent Learning Systems & Applications. 2010;2:147-155.

[14] Zhang Y, Liu Y. Comparison of parametric and nonparametric techniques for non-Peak traffic forecasting. World Academy of Science, Engineering and Technology. 2009;51: 8-14.

[15] Wang X, Abraham A, Smith KA. Intelligent web traffic mining and analysis. Journal of Network and Computer Applications. 2005;28:147-165.

[16] Afek Y, Cohen M, Haalman E, Mansour Y. Dynamic bandwidth allocation polices. INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE. 1996;2:880 –887.

[17] Doulamis AD, Doulamis ND, Kollias SD. An adaptive Neural Network Model for Recursive Nonlinear Traffic Prediction and Modelling of MPEG Video Sources. IEEE Transactions on Neural Networks.2003;14:150-166.

[18] Elias J, Martignon F, Capone A. An efficient dynamic bandwidth allocation algorithm for quality of service networks. LNCS IFIP International Federation for Information Processing. 2006;4195:132-145.

[19] Elias J, Martignon F, Capone A, Pujolle G. Distributed algorithms for dynamic bandwidth provisioning in communication networks. Journal of communications. 2006;1(7):47-56.

[20] Krithikaivasan B, Zeng Y, Deka K, Medhi D. ARCH-based traffic forecasting and dynamic bandwidth provisioning for periodically measured nonstationary traffic. IEEE/ACM Transactions on Networking. 2006;X(Y):1-13. Accessed 11 July 2013. Available: http://sce.umkc.edu/~dmedhi/papers/kzdm-ton-06.pdf.

[21] Anand CN. Internet Traffic Modelling and forecasting using non-linear time Series Model Garch. An M.Sc. thesis submitted to the Department of Electrical and Computing Engineering, College of Engineering, Kansas State University; 2009.

[22] Zhang G, Patuwo BE, Hu MY. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting. 1998;14:35-62.

[23] Zhang G, Patuwo BE, Hu MY. A simulation study of artificial neural networks for nonlinear tine-series forecasting. Computer & Operations research. 2001;28:381-396.

[24] Shamsuddin SM, Sallehuddinand R, Yusof NM. Artificial neural network time series modeling for revenue forecasting. Chiang Mai J. Sci. 2008;35(3):411-426.

[25] Haykin S. Neural networks and learning machines.3rded.New Jersey: Pearson Education Inc; 2009.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=583&id=6&aid=5075