Training Set Size for Generalization Ability of Artificial Neural Networks in Forecasting TCP/IP Traffic Trends

Vusumuzi Moyo
Department of Computer science,
University of Fort Hare
P.O Box X1314, Alice,
South Africa

Khulumani Sibanda
Department of Computer science,
University of Fort Hare
P.O Box X1314, Alice,
South Africa

ABSTRACT
In this paper we empirically investigate various sizes of training sets with the aim of determining the optimum training set size for generalization ability of an ANN trained on forecasting TCP/IP network traffic trends. We found from both the simulation experiments and literature that the best training set size can be obtained by selecting training samples randomly, between the interval $5 \times N_W$ and $10 \times N_W$ in number, depending on the difficulty of the problem under consideration.

General Terms
Pattern Recognition.

Keywords
Generalization ability, Artificial Neural Networks and Training set size.

1. INTRODUCTION
Artificial Neural Networks (ANNs) have been used in many fields for a variety of applications, and proven to be reliable. Inspired by biological systems, particularly the observation that biological learning systems are built of very complex webs of interconnected neurons, ANNs are able to learn and adapt from experience. They have demonstrated to be one of the most powerful tools in the domain of forecasting and analysis of various time series [1]. Time Series Forecasting (TSF) deals with the prediction of a chronologically ordered variable, and one of the most important application areas of TSF is in the domain of network engineering. As more applications vital to today's society migrate to TCP/IP networks it is essential to develop techniques that better understand and predict the behaviour of these systems.

TCP/IP network traffic forecasting is vital for the day to day running of large/medium scale organizations. By improving upon this task, network providers can optimize resources (e.g. adaptive congestion control and proactive network management), allowing an overall better Quality of Service (QoS). TCP/IP forecasting also helps to detect anomalies in the network. Security attacks like Denial-of-Service (DoS) or even an irregular amount of SPAM can be detected by comparing the real traffic with the values predicted by forecasting algorithms, resulting in economic gains from better resource management.

Literature from various authors has shown that unlike all other TSF methods, ANNs can approximate almost any function regardless of its degree of nonlinearity [2, 3]. This positions them as good candidates for modeling non linear and self similar time series such as TCP/IP network traffic. Inspite of this huge advantage, ANNs are not completely absolved from any problems. One major issue that limits the applicability of ANN models in forecasting tasks is the selection of the optimal training set size. The size of the training set is of great importance to ANNs as it responsible for adjusting weights during the ANN learning process. This has a profound influence on the generalization capabilities of the ANN [4]. Generalization is a measure that tells us how well the ANN performs on the actual problem once training is complete. Once the ANN can generalize well, it means that it is capable of dealing with new situations such as a new additional problem or a new point on the curve or surface.

Although individual studies have been conducted and some form of heuristics provided for the selection of the training set size, none have been universally accepted as the results are largely contradictory. Some researchers such as Leung and Zue (1989) [5] suggest that the larger the training set the better the ANN generalization whilst others such as Weigend et al (1990) [6] are of the opposite view. In any case most of these studies have been conducted on synthetic datasets e.g. (Glass-Mackey time series) making the solutions thereof difficult to apply to real world problems. In fact, until a number of experiments have been done, it is unknown which size of the training set will provide optimum solutions. Hence new users of ANNs particularly in the forecasting of TCP/IP network traffic domain, usually blindly employ trial-and-error strategies to determine the optimal values for this parameter without any prior substantive guidelines. This results in the addition of more time to the already slow process of training an ANN.

In this paper the effect of different training set sizes on the generalization ability of ANNs is empirically investigated. Although the results presented in this paper are for a particular case study, they provide a valuable guide for engineers and scientists who are currently using, or intend to use ANNs.

2. ARTIFICIAL NEURAL NETWORKS
Haykin (1999) [7] defines ANNs as "physical systems which can acquire, store and utilize experimental knowledge". The basic unit of an ANN is a neuron. An artificial neuron acts in the same way as a biological neuron; each has a set of inputs and produces an output based on the inputs. A biological neuron produces an output by comparing the sum of each input to a threshold value. Based on that comparison it produces an output. In addition, it is able to differently weigh each input according to the priority of the input. The inputs and outputs of a biological neuron are called synapses and these synapses may act as inputs to other neurons or as outputs such as muscles. Thus it creates an interconnected network of neurons which combined produce an output based on a number of weights, sums and comparisons. One motivation for ANN systems is to capture this kind of highly parallel computation based on distributed representations.

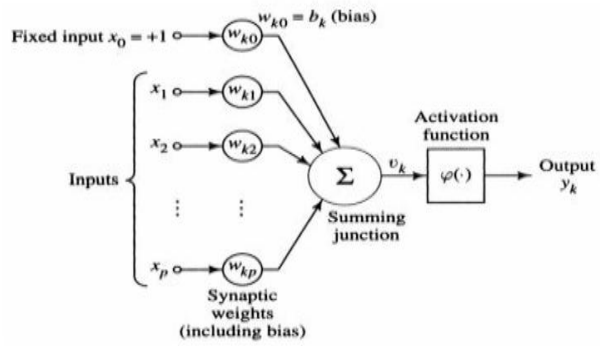

Fig 1: An artificial neuron (adapted from [7])

Fig 1 shows the typical structure of an artificial neuron, the inputs are denoted by $x_1, x_2 \ldots x_p$ and weights are denoted by $w_{ko}, w_{k1}, w_{k2} \ldots w_{kp}$. The neuron calculates the weighted sum w_k, x as:

$$w_k, x = \sum_{i=1}^{p} w_{ki} x_i \qquad (1)$$

The output of the neuron is governed by the activation function, which acts as a threshold. The output is given by:

$$y_k = f(\sum_{i=1}^{p} w_{ki} x_i + b_k) \qquad (2)$$

Where f *is* the activation function, (b_k) is the bias and y_k is the output signal.

Among the various types of ANN models, Multilayer perceptron (MLP) is the most extensively applied to a variety of problems. MLPs are formed by several neurons arranged in groups called layers. The most popular and the simplest MLP consist of three layers, an input layer, a hidden layer, and an output layer. The ANN thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases) being the free parameters of the model. The sliding time window approach is the most common MLP model for forecasting. It takes as inputs the time lags used to build a forecast and it is given by the overall formula:

$$X_{p,t} = w_{o,0} + \sum_{i=I+H}^{I+H} f \sum_{s=1}^{k} \sum_{r=1}^{w_s} X_{st-L_{sr}} w_{i,j} \qquad (3)$$

Where $w_{i,j}$ is the weight of the connection from node j to i (if j=0 then it is a bias connection), o denotes the output node and f is the Logistic sigmoidal activation function.

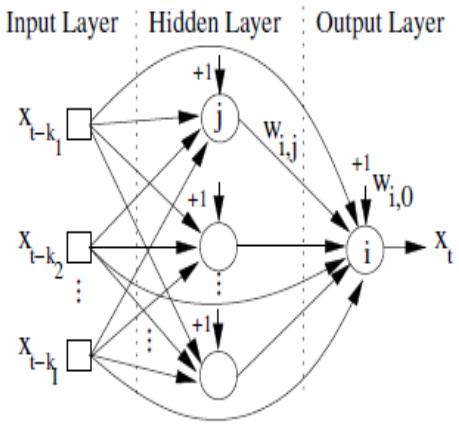

Fig 2: Sliding time window MLP (adapted from [7])

In the vast majority of papers that deal with the prediction and forecasting of TCP/IP traffic, Feedforward networks optimized with the aid of the Backpropagation (BP) algorithm have been used. According to Haykin (1998) [8], "*this is because BP is easy to implement and fast and efficient to operate*". The BP process is commenced by presenting the first example of the desired relationship to the network. The input signal flows through the network, producing an output signal. The output signal produced is then compared with the desired output signal and the errors propagated backwards in the network. In this work we have adopted the BP sliding time window approach for our ANN models.

3. DATA AND METHODS

In our approach for the study we used experimental method which is a proven method for testing and exploring cause and effect relationships. The benefit of using this method is that it allows the control of variables thereby enabling the isolation of a particular variable to observe the effects due to that variable alone. In this case our interest was on the effects of the size of the training set on ANN generalization. The software used for the purposes of this study is Matlab Version 7.4.0.287 (R2007a). Matlab is an application software and programming language with interfaces to Java, C/C++ and FORTRAN. In this study, Matlab provides an environment for creating programs with built-in functions for performance metrics and forecasting using its Neural Networks toolbox Version 5.0.2 (R2007a). The computer used to conduct this study is an Intel(R) Core(TM) 2CPU6300@1.86GHz. The data was collected from the South African Tertiary Institutions Network (TENET) website (www.TENET.ac.za). We analysed network traffic data which comprised inbound traffic in (bits/ sec) from the University of Fort Hare VC Alice Boardroom 101 – Fa 0/1 router. The data spanned from the 1st of March 2010 from 02:00 hours to the 21st of September 2013 02:00 hours in daily intervals, equating to 700 observations. As in all practical applications the data suffered from several deficiencies that needed to be remedied before use for ANN training. Preprocessing was done which included Linear interpolation to fill in missing values, which amounted to 7 such observations. Matlab Neural Network toolbox has a built-in function, *mapminmax* which scales the data down before training so that it has 0 mean and unity standard deviation and then scales it up again after training, so as to produce outputs with 0 mean and unity standard deviation. The data was partitioned into training and testing sets. 547 samples were allocated to the train set whilst 182 were allocated to the test set.

In order to investigate how the size of the training set affects the generalization ability of ANNs, 10 different training sets were generated as subsets chosen from the full training set of 547 samples. These subsets were arbitrarily created at 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of the full training set of 547 samples, as these were deemed to be appropriate sizes for subset selection. The choice of only 10 subsets was made mostly due to the fact that testing a larger number of subsets was highly unlikely to produce more telling results as Sontag (1992) [9], who conducted similar investigations recorded comprehensive results using only 10 subsets. A supervising script was written to compute the ANN inputs and targets. On visual inspection of the time series a sliding time window of size 150 was arbitrarily chosen. During training an input and output layer of 1 neuron corresponding to the forecasting horizon was used. The weights were randomly initialised in the range [-0.5, 0.5], the Back-propagation Levenberg- Marquardt training rule was

used to update the weights, the Logistic sigmoid and Linear activation functions were used in the hidden and output layers of the ANN models respectively. Training was stopped after 1000 epochs and the generalization performance of the ANNs tested by presenting the unknown test set to the ANNs and calculating the Root Mean Squared Error (RMSE) between the actual and predicted values. RMSE is a dimensionless value calculated to compare ANN performance. The RMSE on the test set (MSE_{te}) was calculated using the following equation:

$$RMSE_{te} = \sum_{p=1}^{P_{te}} (d_p - o_p)^2 \qquad (4)$$

where d_p is the desired output for each input pattern and o_p is the actual output produced by the ANN. In order to minimize the random effect of the initial weights on results, for each experiment conducted, 4 training runs were made and the results averaged. We also ensured that all other variables that could potentially affect the quality of results remain constant. Hence throughout the duration of our investigations the learning rate and momentum remained fixed at 0.

Two other performance evaluation criteria were used. The correlation statistic (R) selected to measure the linear correlation between the actual and the predicted traffic. The optimal R value is unity and a value smaller than 0.7 is assumed to be problematic. To estimate the efficiency of the fit, the Coefficient of determination also known as the R^2 criterion is used. The optimum R^2 value is unity and a value smaller than 0.7 corresponds to a very poor fit.

4. THE EXPERIMENTS

4.1 Single Hidden Layer

In the first phase of our investigations we, examined a single hidden layer ANN with an architecture of (1, 60, 1) i.e. 1 input neuron, 60 hidden neurons and 1 output neuron. We chose this architecture because it exhibited better generalization performance in preliminary experiments conducted prior to the main investigations. We trained the ANN using both the full training set of 547 samples and reduced variations of it for a number of training runs. The results of the experiments are shown in Fig 3.

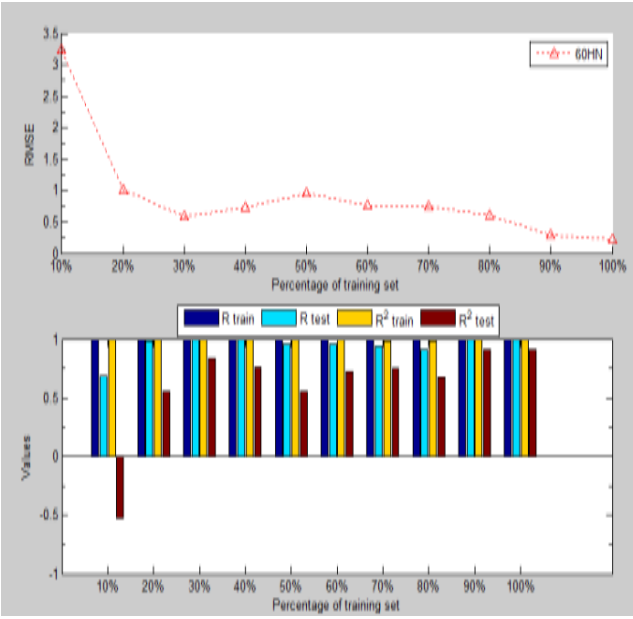

Fig 3: Generalization errors in RMSE (top); R and R^2 for train and test sets (bottom).

4.2 Two Hidden Layer

In the second phase of our investigations we examined the performance of a 2 hidden layer ANN of architecture (1, 5, 35, 1) i.e. 1 input neuron, 5 first hidden layer neurons, 35 second hidden layer neurons and 1 output neuron. As in the previous case, this was the architecture that exhibited substantially better generalization performance amongst the 2 hidden layer architectures examined in the preliminary investigations. Furthermore, Maier and Dandy (1997)[10] carried out a similar investigation using an almost similar architecture of (1, 8, 30, 1) with some measure of success. We trained the ANN using both the full training set of 547 samples and reduced variations of it for different training runs. The results of the experiments are shown in Fig 4.

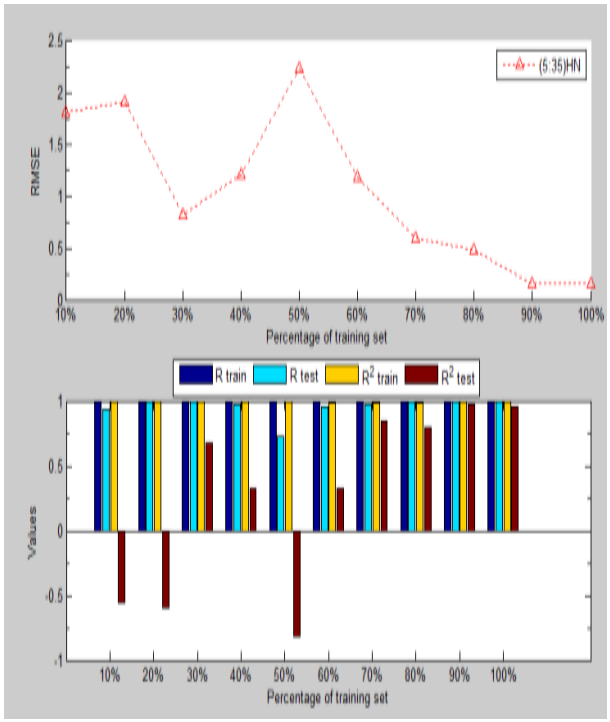

Fig 4: Generalization errors in RMSE (top); R and R^2 for train and test sets (bottom).

4.3 Different Heuristics

In the third phase of our investigations, we performed a comparative analysis of some of the heuristics on the selection of a sufficient training set size for a given problem given in literature, with respect to their effectiveness and reliability towards the determination of the optimum training set size for generalization ability of ANNs in the context of forecasting TCP/IP network traffic trends. For these investigations we used the same single hidden layer ANN of architecture (1, 60, 1) as used previously. The results of the investigation are shown in Fig 5.

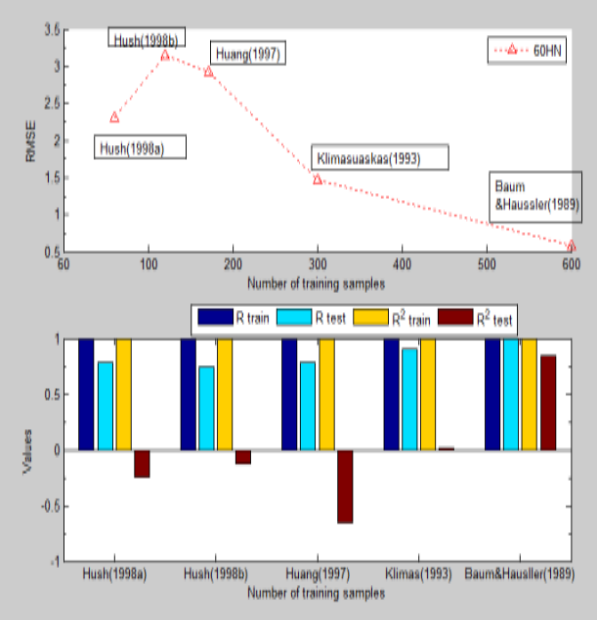

Fig 5: Generalization errors in RMSE (top); R and R^2 for train and test sets (bottom).

5. RESULTS AND DISCUSSIONS

To assess the generalization performance of ANNs trained on different training set sizes in the task of forecasting a real world TCP/IP network traffic time series, we conducted several experiments varying the size of the training set. We begin with the results depicted in Fig 3, which show the generalization errors versus the size of the training set for a single hidden layer ANN with 60 hidden neurons. On examining Fig 3, one cannot shy away from the exponential decrease in generalization errors as the size of the training set is enlarged. This trend remains apparent until a minimum RMSE in generalization errors is attained at the full training set of 547 samples. Our results are similar to those of Kavzoglu (1999) [11], who in empirically conducting experiments on ANNs in speech synthesis recorded an exponential decrease in generalization errors as the size of the training set increased. Quite evidently from the results in Fig 3, the generalization ability of a single hidden layer network trained on a reduced training set is significantly degraded juxtaposed with a network trained on the full training set of 547 samples. Perhaps one of the reason why this could be so is that the full training set is more representative of the problem space that the altered variations of it. This undoubtedly adds another dimension to the whole debate that of the quality of the training set.

Our findings are contrary to those of Zhang et al. (1998) [12], who suggest that a training set size of only 5% of the full dataset is sufficient for a good generalization ability of a single hidden layer ANN. Also, Lange et al. (1997)[13] introduced the idea of a critical training set size. Through experimentation they found that examples beyond this critical size do not improve generalization, pointing that an excess of training patterns have no real gain. They state that this critical training set size is problem dependent. Although we are not entirely sympathetic to this viewpoint it would appear that the argument raised by [13] was quite valid, judging by the nature of the generalization error curve in Fig 3. At the full training set of 547 samples, the generalization errors seem to plateau raising potential doubts that any increase in training set size beyond that point would have further decreased the errors, we are however reluctant to derive substantial conclusions in that regard due to the fact that we did not train the ANNs beyond the full allocated training set size of 547 samples in this research.

On analysis of the statistical measures in Fig 3, we note that at lower proportions of the training set, values of the R^2 on the test set were largely lower than at higher proportions of the training set. Infact at 10% of the training set a negative R^2 value on the test set was recorded indicating a very bad fit between the ANN activations and targets. Generally the ANNs exhibited poor performance on both R and R^2 values for training set sizes below 80% of the full training set of 547 samples.

The next sets of results to be discussed are those exhibited by a 2 hidden layer ANN with an architecture of (1, 5, 35, 1). Once again the trend in Fig 4 indicate a decrease in generalization errors as the size of the training set is increased. However this time around this trend is not as exponentially smooth as in the previous case. This is evidenced by the huge fluctuations in generalization errors witnessed from 10% to 50% of the full training set of 547 samples, at 60% of the full training set of 547 samples, the generalization errors gradually decline reaching a mimina at the full training set. Whilst we are not exactly certain as to the actualities resulting in the slightly erratic behaviour of the ANN between 10% to 50% of the full training set of 547 samples, one possible reason could be that within this range of the training set, particular instances within the ANN are a better representation of the problem space than others. If these instances are randomly spread throughout the data set then the likelihood of them being selected for training is equally random. We are certain though that the erratic behaviour of the ANNs was certainly not caused by the addition of an extra hidden layer, that could perhaps cause the ANN to be unstable at those particular sizes of the training set. Although this could be a distinct possibility one may ask the question, if that was so how does one then explain the unerratic behaviour expressed by the ANN between 50% of the training set to the full training set of 547 samples.

Analysis of statistical measures indicate that from 60% and below of the full training set of 547 samples, values of R^2 (on the test sets) range mostly between 0.6 to -0.8 revealing a poor fit between the ANN activations and targets. As largely expected the best performance in terms of R and R^2 were recorded at sample sizes larger than 70% of the full training set of 547 samples. On a comparative note, the generalization errors from Figs 3 and 4 show that the best generalization performance of a single hidden layer architecture in Fig 3 is recorded at a RMSE of approximately 0.25, whilst for a 2 hidden layer architecture in Fig 4, is attained at a comparably similar error level. An interesting observation from this is that both these generalization errors were attained at the full training set of 547 samples. What is perhaps most striking about these results is the comparative equivalency of the generalization results in Figs 3 and 4 despite the different ANN architectures. Our results contradict the empirical findings suggested by Baum and Haussler (1989) [14] that suggest that more training data are required to achieve optimal generalization ability for larger ANNs. In our study, generalization ability was not affected by using larger ANNs despite the size of the training set, meaning to say the size of the network architecture did not affect how our ANNs responded to the training set. This quite indelibly is also in stark contrast to the dataset size guideline of Haykin (1999) [7].

Moving on to the results illustrated by Fig 5, which show the comparative performance of different heuristics provided by various authors, on selecting the optimal training set size for generalization ability of ANNs in forecasting TCP/IP network traffic trends. It can be noted again that as the number of training samples increases, there is a gradual almost logarithmic decrease in generalization errors. For the forecasting of TCP/IP network traffic, the heuristic proposed by Hush 1998 [15] suggests an insufficiently small number of training samples, whilst on the other hand the heuristics recommended by Klimasauskas (1993)[16] and Baum and Haussler (1989)[14], indicate training set samples that are relatively close to the optimum solution. R and R^2 values at or above 0.8 are statistically significant, of the results in Fig 5.8, only Baum and Haussler (1989)'s heuristic satisfy that condition. None of the ANNs examined showed any signs of overfitting.

6. CONCLUSIONS AND FUTURE WORK

The experimental results regarding the relationship between the size of the training set and the generalization exhibited by an ANN have been discussed. The results indicate that altering the size of the dataset used to train an ANN has an obvious impact on its generalization ability. Evidence has been presented that contradicts some of the current ANN design schools of thought, which imply that smaller quantities of training data necessarily produce better-quality forecasting models. The empirical results from this research indicate that frequently a larger quantity of training data will produce a better-generalized Backpropagation ANN model. Stathakis (2009) [18] who arrived at similar conclusions to that of ours, proposes that this trend is apparent, mostly due to one of two reasons; 1) the network is suffering from overfitting when training on small data sets or 2) the ANN is really being affected by the size of the data.

After carefully observing the performance plot during training of our ANNs it was determined that the ANNs in our study were infact not suffering from the effects of overfitting and that it was instead the size of the training set that was actually causing the ANNs to perform poorly. We conclude therefore that the ANNs poor performance on small training sample sizes could be attributed to one of two possibilities: 1) the data set was so diverse that the ANNs could not represent it accurately without seeing 100% of the instances during training, 2) particular instances within the ANN are a better representation of the problem space than others, and if these instances are spread throughout the data set then the likelihood of them being selected for training is proportionate to the size of the subset.

Although it may not be possible to give a theoretical recommendation on the exact size of the training sample for all practical problems, our experimental results suggest that for forecasting TCP/IP trends, larger training samples sizes perform significantly better than smaller ones in as far as generalization ability is concerned. This however may introduce important practical implications as smaller ensembles require less computational efforts. We found from both simulation experiments and literature that the best training sample size can be obtained by selecting training samples randomly, between the interval $5 \times N_w$ and $10 \times N_w$ in number, depending on the difficulty of the problem under consideration.

With regards to the future, as with almost any area of research, progress leads toward more questions. Based on the research carried out in this study, our results suggest considerable potential for future work. We plan in extending our investigations to new self similar and chaotic time series. In addition, more testing is needed on other datasets to validate the effectiveness of the conclusions reached in this research. Further testing using a larger number of ANN architectures is a necessity, as in this work we only considered 2 network architectures for examination, perhaps testing on a variety of network architectures may yield different results.

Richards (1991) [4] suggests that although the size of the training set is of considerable importance, the characteristics and the distributions of the data as well as the sampling strategy used are crucial. He states "….*simply presenting more of the same insufficiently informative training examples will not guarantee that the system will exhibit good generalization*." From this statement we gather that one can have an extremely large number of training samples, but if those training examples are poor examples of the concepts that are to be learned, it is unlikely that good generalization will occur. It is important that any future investigations carried out on the training set should take into consideration that valuable fact, as the quality of the training set could be a potential game changer in as far as the generalization ability of ANNs is concerned. We intend to look into this area quite extensively in our future explorations. Lastly, as this study was mostly limited to Feed-forward ANN learning problems with the Backpropagation learning algorithm, it could be also beneficial to investigate the effects of the size of the training set on the performance and generalization ability of other ANN models, including Self Organizing Maps (SOM) and Learning Vector Quantization (LVQ), with the aim of deriving some general conclusions that can be used to construct some guidelines for users in design of these particular ANN models.

7. ACKNOWLEDGMENTS

This work is based on the research undertaken within the TELKOM Coe in ICTD supported in part by Telkom SA, Tellabs, SAAB Grintek Technologies, Eastell, Khula Holdings, THRIP, GMRDC and National Research Foundation of South Africa (UID: 86108). The opinions, findings and conclusions or recommendations expressed here are those of the authors and none of the above sponsors accepts liability whatsoever in this regard.

8. REFERENCES

[1] S. Chabaa, "Identification and Prediction of Internet Traffic Using Artificial Neural Networks," *J. Intell. Learn. Syst. Appl.*, vol. 02, no. 03, pp. 147–155, 2010.

[2] R. Aamodt, "Using Artificial Neural Networks To Forecast Financial Time Series," Norwegian university of science and technology, 2010.

[3] H. Tong, C. Li, J. He, and Y. Chen, "Internet Traffic Prediction by W-Boost: Classification and Regression," *Neural Comput.*, vol. 2, no. 973, pp. 397–402, 2005.

[4] E. Richards, "Generalization in Neural Networks, Experiments in Speech Recognition," University of Colarado, 1991.

[5] H. Leung and W. Zue, "On the Generalization Capability of Multi-Layered Networks in the Extraction of Speech Properties," in *International Conference on Acoustics, Speech and Signal Processing*. 1989, pp. 422–425.

[6] A. Weigend, D. Rumelhart and B. Huberman, "Predicting the future: a connectionist approach." *Int. J. Neural Syst.*, vol. 1, no. 3, pp. 193–209, 1990.

[7] S. Haykin, *Neural Networks: A comprehensive foundation*, Second. Pearson, 1999, pp. 2–3.

[8] T. Mitchell, *Machine learning*. McGraw Hill Publishers, 1997, pp. 100–150.

[9] E. Sontag, "Feedback stabilization using two hidden layer nets," *IEEE Trans. Neural Networks*, vol. 3, no. 6, pp. 34–60, 1992

[10] H. R. Maier and G. C. Dandy, "Determining Inputs for Neural Network Models of Multivariate Time Series," *Comput. Civ. Infrastruct. Eng.*, vol. 12, no. 5, pp. 353–368, Sep. 1997.

[11] T. Kavzoglu, "Determining Optimum Structure for Artificial Neural Networks," *In proceedings of the 25th Annual Technical Conference and Exhibition of the Remote Sensing Society*, 1999, no. September, pp. 675–682.

[12] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *Int. J. Forecast.*, vol. 14, no. July, pp. 35–62, 1998

[13] N. Lange, C. M. Bishop, and B. D. Ripley, "Neural Networks for Pattern Recognition.," *J. Am. Stat. Assoc.*, vol. 92, no. 440, p. 1642, Dec. 1997

[14] E. Baum and D. Haussler, "What size net gives valid generalization," *Neural Comput.*, vol. 1, no. 1, pp. 159–161, 1989.

[15] D. Hush, "Classification with neural networks," in *Proceedings of the IEEE International Conference on Systems Engineering*, 1989, pp. 50–57.

[16] C. Klimasauskas, *Applying neural networks*. 1993, pp. 47–72.