NIPES

Journal of Science and Technology Research

Journal homepage: www.nipesjournals.org.ng

Neural Network Prediction of Self-Similarity Network Traffic

[a]Ikharo A. B. and [b]Anyachebelu K. T.

[a]Dept. of Computer Engineering, Faculty of Engineering, Edo state University Uzairue, Edo State, Nigeria

[b]Dept. of Computer Science, Faculty of Natural and Applied Sciences, Nasarawa State University Keffi, Nasarawa State

Article Info

***Keywords*:**
Burstiness, Self-similar, Network traffic, Performance, Simulation, Artificial Neural Network, Packet

Received 29 August 2022
Revised 15 September 2022
Accepted 19 September 2022
Available online 2 Dec. 2022

ISSN-2682-5821/© 2022 NIPES Pub.

DOI:
https://doi.org/10.5281/zenodo.7390658

Abstract

Several factors are found to influence either short or long-term burstiness in Transmission Control Protocol (TCP) flow across many networking facilities and services. Predicting such self-similar traffic has become necessary to achieve better performance. In this study, ANN model was deployed to simulate College Campus network traffic. A Feed Forward Backpropagation Artificial Neural Network (ANN) and Wireshark tools were implemented to study the network Scenario. The predicted series were then compared with the corresponding real traffic series (Mobile Telephone-Network (MTN)-Nigeria). Suitable performance measurements of the Means Square Error (MSE) and the Regression Coefficient were used. Our results showed that burstiness is present in the network across many time scales. With the increasing number of data packet distributions thereby providing a steady flow of burst over the entire period of system load as the traffic network performance improves.

1. Introduction

Analysis of traffic in deployed computer networks is important for determining performance, reliability and scalability [1]. A number of factors, such as a slow start phase of the congestion window, packet losses, multi-switching of packets at the bottleneck rate and acknowledge-compression of Transmission Control Protocol (TCP) traffic, could influence either short or long-term burstiness in TCP flow [2]. A significant adverse impact on network performance is shown attributable to traffic self-similarity and, while throughput declines gradually as self-similarity increases, queueing delay increases more drastically [3, 4].

Self-similarity calls for greater network resources like link bandwidth and buffer space. Also, network performance goes down as seen by reduced throughput, greater packet loss rate and greater packet retransmissions. The overall network experiences reduced performance because of the Long-Range Dependence (LRD).

A self-similar object is said to be exactly similar to a part of itself [5] or it looks qualitatively the same with a sufficiently large scale of the time axis and exhibits a long-term dependence [6]; implying that self-similarity is a particular feature of a stochastic object, such as time series that remain identical in different scale of time and space. This is the case for computer network traffic. Neural Network exhibits a degree of self-similarity at large scale and high degree of multi-fractal at small time scale. Network traffic condition is a very complex system and nonlinear in nature, therefore, Artificial Neural Network (ANN) is highly suitable for the situations where the underlying processes exhibit chaotic features.

Given the ubiquity of the burstiness present across many networking facilities and services, predicting and managing self-similar traffic has become an important problem owing to new complexities associated with self-similarity which makes it difficult for the achievement of

high network performance and quality of service (QoS). Most institutions networks burst and many data packets are lost and often time data packets sent never get to their destinations.

[7] did the analysis of the network traffic over the IP network by developing an ANN model using multi-layer perceptron. For this network response was evaluated by using ANN and further analysing the time series of network data. The results so obtained led to the conclusion that the ANN model using the Levenberg-Marquardt (LM) algorithm can be very well used for network traffic prediction and can be applied as an excellent and fundamental tool for the management of the internet traffic at different times.

The work aims to simulate and predict network traffic with a view to diagnose its bursty nature and with the hope to optimise network performance for service delivery. In the face of Distributed Denial-of-Service (DDoS) attacks, network resources such as the bandwidth are near exhausted to the detriment of other vital resources and services. In some cases, disrupting routing operations thereby degrading the performance of the network. Therefore, it is extremely important in today's world to analyse network traffic processes, allowing us to detect anomalies in the network and model the processes occurring in the network traffic [8].

In [9], modelling of measured self-similar network traffic in OPNET simulation tool was implemented. [10] studied the application of Hurst Exponent (H) and the R/S Analysis in the classification of FOREX Securities. [11] modelled the workload intensity of the FIFA World Cup website using ANN. [12] worked on the analysis that non-interference with network traffic leads to network traffic congestion and paralysis.

2. Methodology

2.1 Data Capture and Sample Structure

Data was obtained from a monitored standard network traffic (MTN) and a customised network traffic using Wireshark software over a period of time. ANN performance was employed by using different learning methods, activation function, hidden layer, and neuron numbers to determine prediction of burst rate. ANNs was trained using experimental data obtained from the College Campus network. The ANN is a 2-39-1 Feed Forward Backpropagation network implemented to predict the bursty nature of network traffic. Wireshark tools that measure and capture data packets of network traffic were employed to obtain needed information about the network scenario. This tool was used to examine the data packets sent and received along with the communications protocol employed in the transmission. Wireshark was selected for use because of the following features [13]: live data capture, support for offline protocol analysis, its support for almost all network, transport and application protocols and its compatibility with other products such as standard. pcap format which makes it compatible for use with other network tools. A quantum number of packet sizes capture over 2 hours would suffice for this study. Data used for the study involve a five distinct non-successive days recursive approach. However, only the readings with the highest number of records were used for the purpose of this research to minimise statistical errors.

2.2 ANN Stochastic Self-similar Process Implementation

In Figure 1, W_{ij} is the weight vectors between input layer and the hidden layer, W_{jk} is the weight vectors between the hidden layer and Tang-sigmoid. Since the initial value of connection weights (W_{ij} and W_{jk}) in the neural network layers are arbitrary, the neural network must be trained to assure that the deviation between the desired output value and the actual value is as small as possible, neural network reversely send the error between the real output of sample to neurons in layers, it constantly adjusts the weights of layers using the *gradient descent method*

to reduce the error caused by the weight, so it can assure that the error between the real output of training samples and the output of neural network within a set range or the number of training is maximum. The values of the various W_{ij} and W_{jk} are then obtained.

The purpose of the training is to adapt the neural network with the desired characteristics. This is done by changing the weight and bias terms associated with each of the connections. Three stages of the training are implemented: *feedforward* of inputs, *backpropagation* of errors and updating of *weights* and *biases*. In the feedforward stage, a sigmoid function with output range between 0 and 1, is used to obtain activation for the weighted sum inputs. In ANN training, suitable stopping criteria are needed to determine when the training should be stopped. In this research, the training is stopped when the mean square error (MSE) of the output starts to increase. This is because further training will not increase the network performance. The number of hidden neurons is fixed based on new criteria. The proposed model is used for estimation and prediction. The key of the proposed method is to select the number of neurons in hidden layer. The architecture for fixing number of hidden neurons in ANN is shown in Figure 1. The selected criteria for ANN model is $(4n^2 + 3)/(n^2 - 8)$ using 39 numbers of hidden neurons and obtained a minimal MSE value [14].

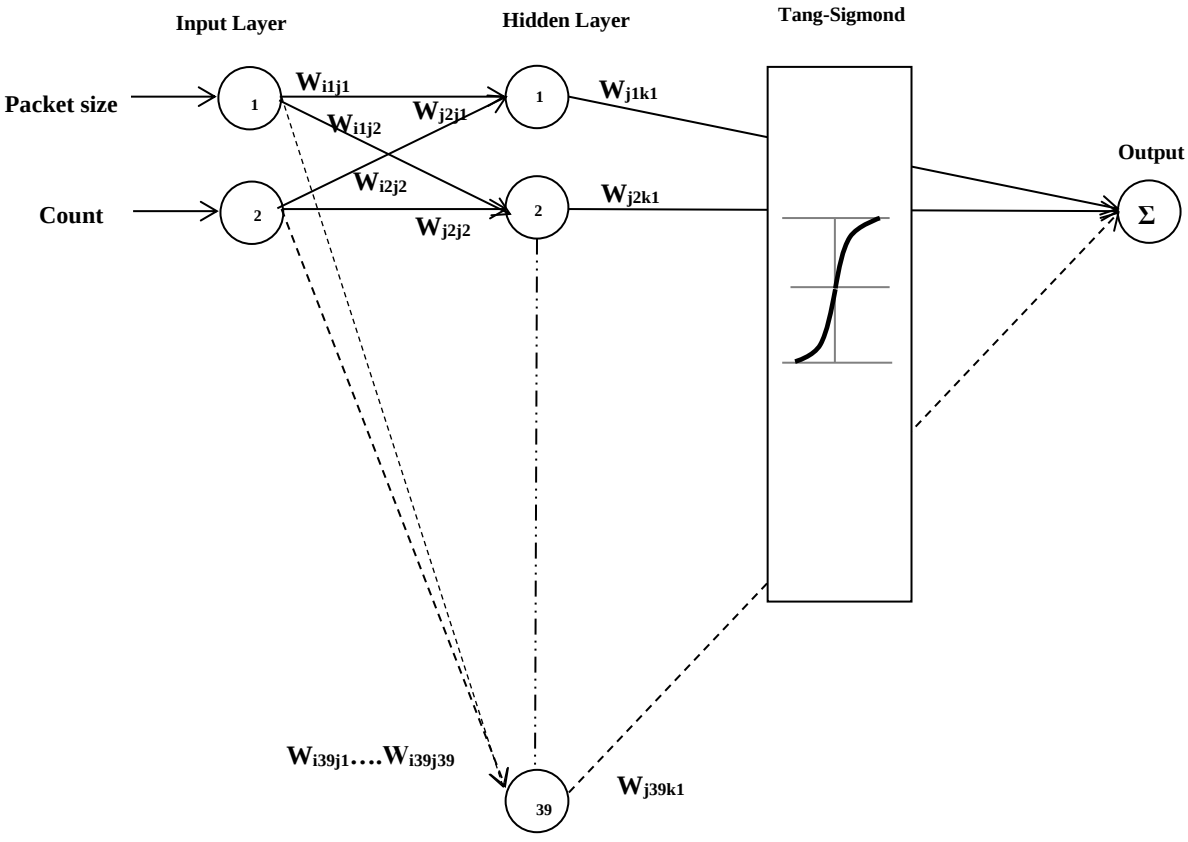
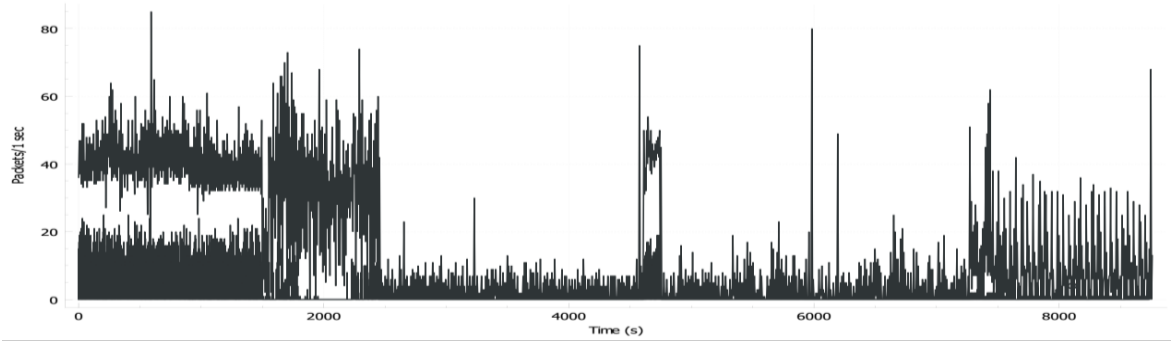


Figure 1: Feed Forward Artificial Neural Networks (ANNs) with Back Propagation for Supervised Learning

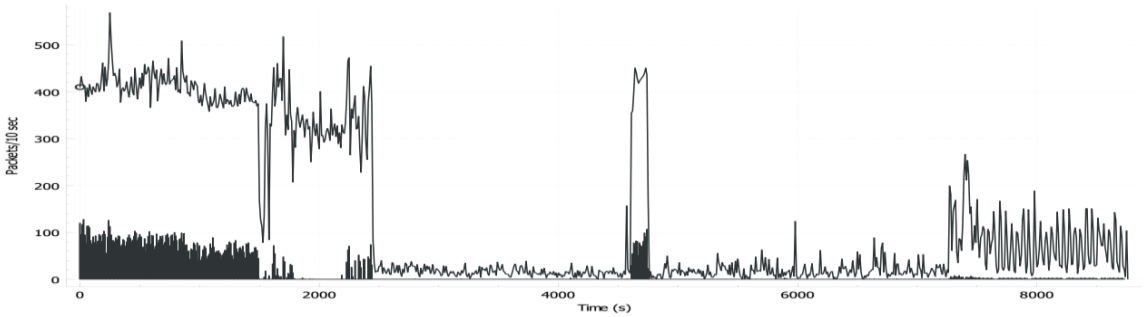
3. Results and Discussion

3.1 Network Simulation Outcome

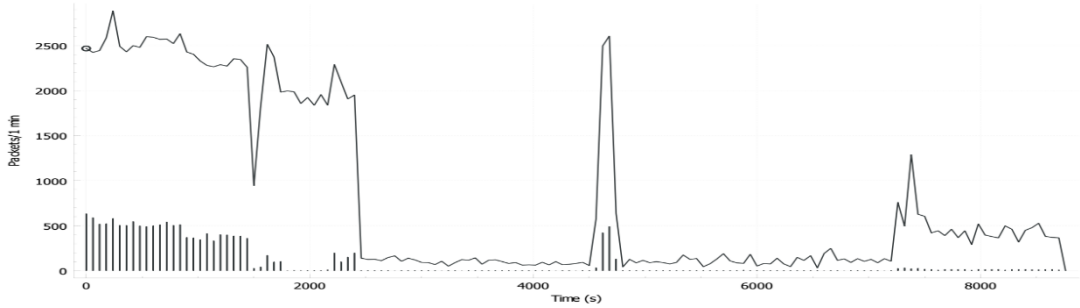
Captured packets of two (2) different network traffics (College Campus network and standard MTN network) using Wireshark Sniffer were made in real-time. These measured traffic were used for the study analysis, modelling and simulation. The captured traffic outcomes are shown in Figure 2 with depicted instances of exhibited layout at various time scales. Figure 2 (a) is at time scale of 1 second. The profile captures all packet data and TCP errors. It simply indicates the number of packets per unit second between transmitted and received. Figure 2 (b) is at time scale of 10 seconds and Figure 2 (c) is at time scale 1 minute. Figure 3 shows recursive pattern with peak values of data packets at 6 other points in the time axis. Figure 3 shows a pattern with three peak values between 600 and 800 seconds, 1500 and 1800, and between 2500 and 3000 seconds.



(a)

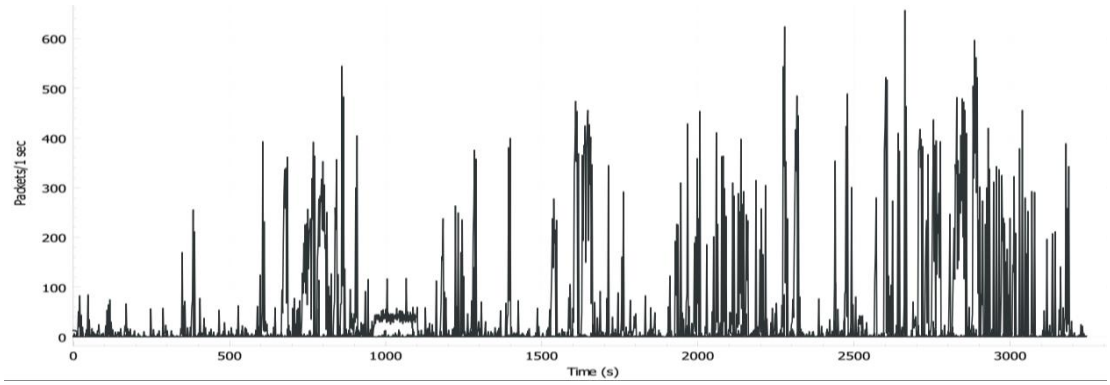


(b)

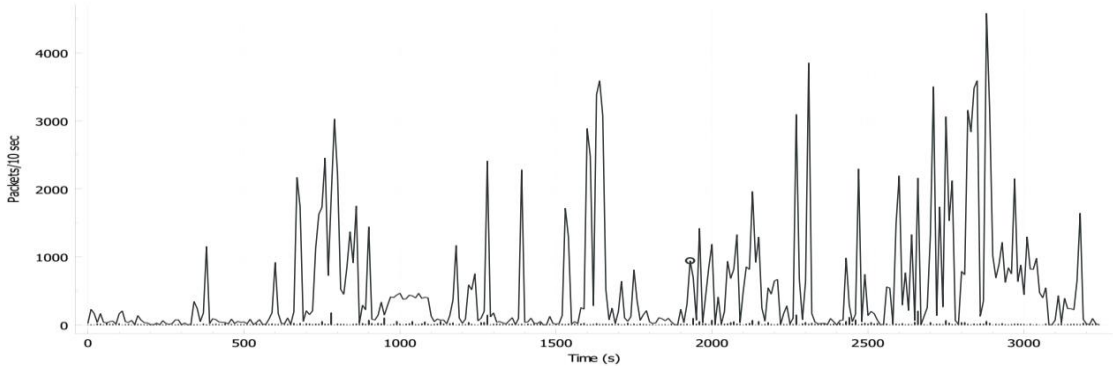


(c)

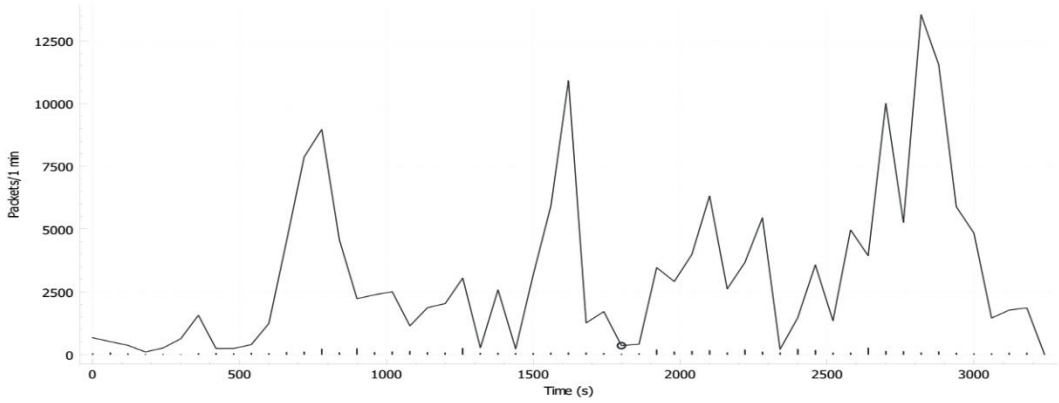
Figure 2: Measured Test traffic for Campus network – DataCap2



(a)



(b)



(c)

Figure 3: Measured Test traffic for MTN – DataCapMTN

These figures depicted sequences of plots of the data packet counts, that is, a number of packets per unit time for three different traffics. Starting with a time unit of zero second with subinterval of 100 seconds per slot. A surface observation of these profiles indicate that Figure 2 shows

ramping pattern at the start of the profile and peak values at two other points in the time axis. In a general sense, these profiles are self-similar in the same time scale – bursty in their characteristics. In particular, notice must be drawn to the fact that there is the absence of a natural length of a burst, but simply consist of bursty subperiods separated by less bursty subperiods. These patterns exhibited by these profiles suggest the use of self-similar stochastic processes for their traffic modellings.

The increase in packet count distribution in Figure 2 is due to aggregation. Apart from the scaling property, the figures appear to share identical distribution characteristics. In other words, if the tenfold multiplication of packet count dimensions (incurred at each aggregation step) were to be compensated by division, the sequences would be indistinguishable. That suggests the plots have similar statistical properties.

One major feature that is prominent and dominant throughout all the shown time-scales is burstiness. Bursts occur when traffic intensity peaks around the average traffic level, and are clearly observed at all time-scales in the entire Figures. Furthermore, bursts have no natural duration; they appear at all timescales and themselves consist of higher-time-resolution bursts.

3.2 Network Model Implementation and Evaluation

In a bid to ensure that ANN predicted traffic burst rates are in conformity with the desired outcomes, comparison of the measured traffic results is shown in Figures 4 and 5. The three traffic traces were simply selected for the duration of the captured data and the outcome was plotted using clustered column graphs to illustrate the comparison.

Figure 4 is a column chart view for dataCap2 data sets, indicating a close value for both the measured and the predicted values of the burst rate. More so, Figure 4 displayed marked presence of burst. The burst rate of 0.3, which is the highest recorded value at the start of the traffic is significantly low and its distribution is also small. It is observed here that as the distribution expands or gets larger, the burst rate is relatively more constant and has lower values of burst rates.

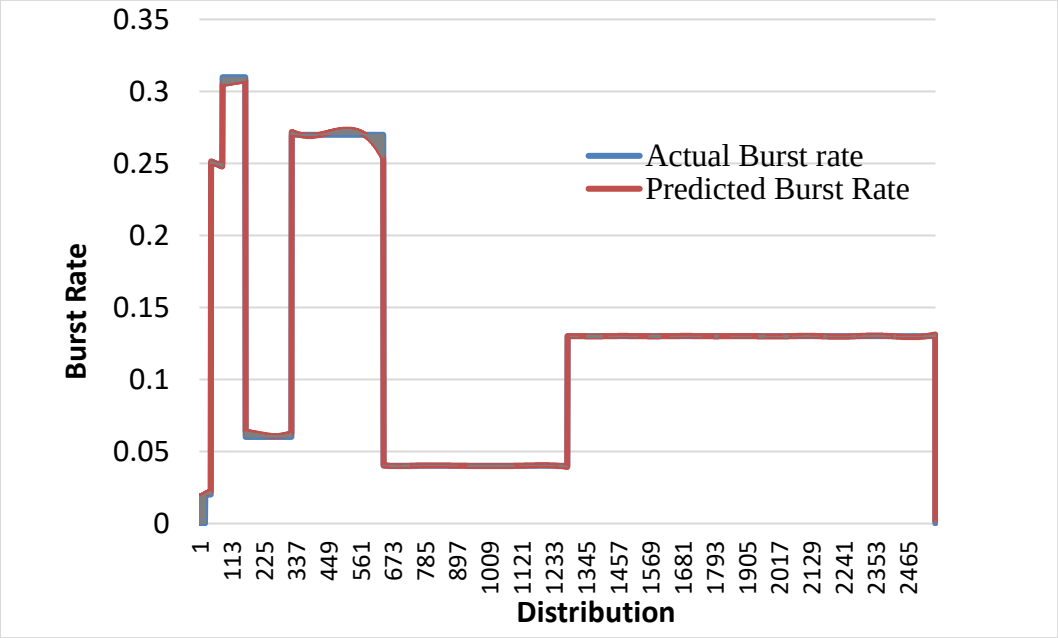


Figure 4: *Profile of DataCap2 Measured and Predicted Campus Network Traffic Compared*

3.3 Evaluating ANN Network Performance

The ANN model prediction performance indices are shown in Table 1. The various values for the epoch, mean square error (MSE), regression and net performances are given accordingly. The total number of data used in the ANN Network is 2561 in which 60% (1537) of these data is for the training phase with 20% (512) for validation and 20% (512) for testing. While it is found that the ANN model can make predictions with high degree of accuracy, the learning rate of the network is 0.015.

Figure 1 in section 2.2 is the simulated ANN model that provided the platform on which Figures 4 and 5 were realised and the performance indices were obtained as indicated in Table 1 for the predictions.

Table 1: Performance Indices for the ANN Prediction

Locations	Epochs	MSE	Regression	Performances
DataCap2	20	$7.72577e^{-3}$	0.98	$7.59195e^{-3}$
DataCapMTN	100	$5.59213e^{-3}$	0.99	$1.80957e^{-6}$

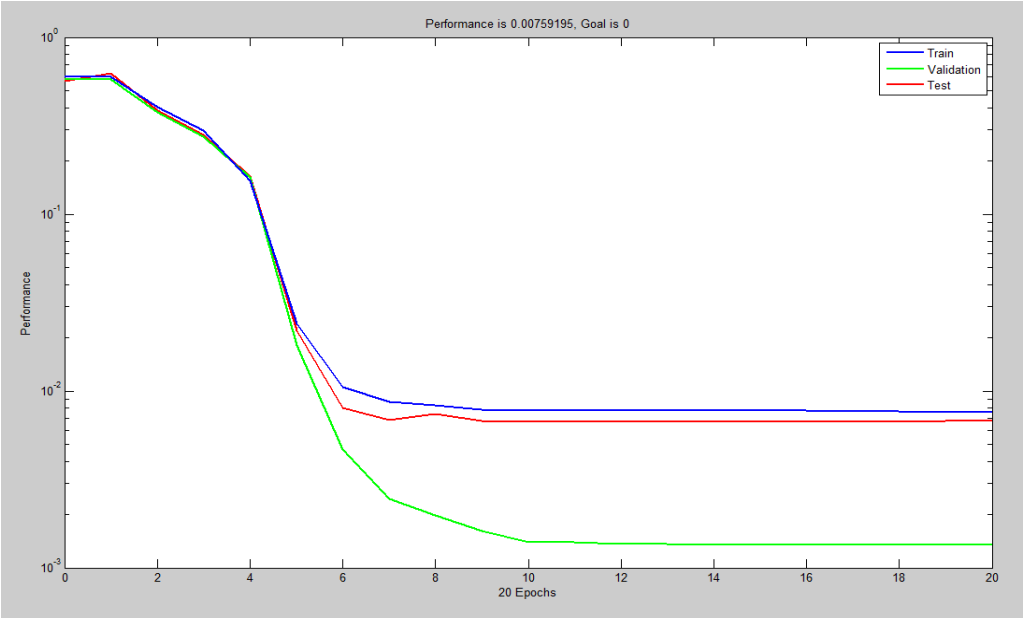


Figure 5: ANN Training Outcome for DataCap2

Figures 5 and 6 represent the ANN training outputs for DataCap2 and DataCapMTN as they converge. According to [14, 15] small training dataset (less than 5%) provides a high prediction error indicating an insufficient dataset to adapt the ANN model parameter to the network characteristics. Thus, subscribing to the use of higher percentage value of which 20% is an acceptable value. In this study, we used 20%, sufficient to improve our prediction accuracy or traffic predictability. Moreover, enlarging the training dataset does not really improve the prediction error because the model will be over trained. Our use of 20% for test are quite adequate and that ANN model has capture during training phase, the strong correlation of the traffic with Long-Range Dependence (LRD). The performance displayed by Figure 4 shows

the rapid convergence of the training, validation, test output of the ANN network dataset to correctly predict the traffic scenarios.

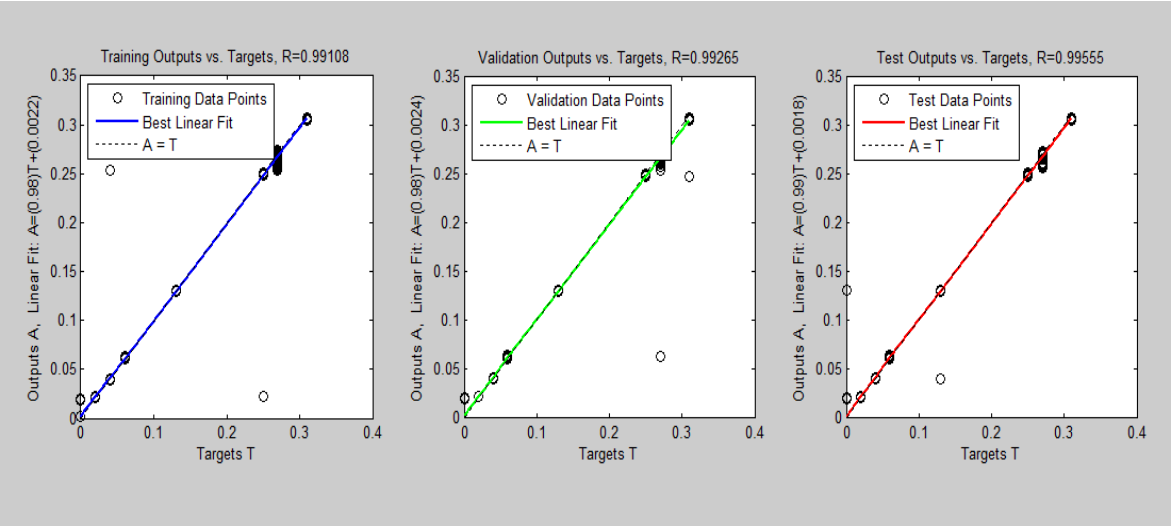


Figure 6: Regression Analysis of Outputs and Targets for DataCap2

3.4 Discussion of Findings

Results obtained showed that with real traffic, the ANN model has established network traffic that have similar statistical characteristic, such as self-similarity and long-range dependency. The ANN model was implemented in MATLAB software and the neural network module of this software was used to implement backpropagation part of model. Comparative evaluations of self-similar traffic have been found to usually concentrate on two aspects [5]: how accurately self-similar processes can be generated and how quickly the methods generated follows self-similar sequences.

The captured data packets were fed to the ANN model and corresponding results indicate that the model has high accuracy. In establishing self-similarity, the values obtained showed that both the simulated and the real traffic analysed were both self-similar. The degree of self-similarity is dependent on the volume of traffic and the type of service.

ANN model was deployed to predict network traffic behaviour of Campus network traffic. The model was put in place to test the traffic network self-similarity behaviour and established the fact that the burst nature of the networks is such as provide leverage for better performance and enhanced quality of service. There were mild discrepancies between the capture and the simulated network traffics in the sense of packet-rate, bursts intensity, and variances. This could be attributed to the failure on the part of signal quality during data capture for analysis purpose. Even though, efforts were put in place to carefully eliminate some of these traffic errors.

4. Conclusion

In this study, we have presented our research in the area of modelling and simulations of College Campus network traffic. And that by extension, the ANN predictions for Campus network traffic in the chosen situations have behaved similarly. The study has been able to establish a trained ANN that was capable of predicting the self-similarity in the Campus network traffic with great accuracy with LRD characteristics. We have tried to minimise

discrepancies between the captured and the simulated network traffic in the sense of packet-rate, bursts intensity, and variances. The system network performance improves with the increasing number of data packet distribution but degrades at the start of connections signifying the possible presence of burst arrival rate connection and load range of the network during contention resolution for the network traffic.

References

[1] Ikharo A. B., Anyachebelu K. T., Blamah N. V., Abanihi V. K. (2020). Optimising self-similarity network traffic for better performance. International Journal of Scientific Research in Science and Technology, 7(4): 164-176

[2] Wu, C. and Irwin, J. D. (2013). Introduction to computer networks and cybersecurity. CRC Press, New York.

[3] Rezaul, K. M. and Grout, V. (2010). An overview of long-range dependent network traffic engineering and analysis: Characteristics, simulation, modelling and control. 2nd International ICST Conference on Performance Evaluation Methodologies and Tools http://dx.doi.org/10.4108/valuetools.2007.1892.

[4] Panarello, C., Lombardo, A., Schembra, G., Meo, M., Mellia, M. and Marsan, M. A. (2016). Network interface power management and TCP congestion control: a troubled marriage. *Australian Journal of Electrical and Electronics Engineering*. 13(1): https://doi.org/10.1080/1448837X.2015.1093678

[5] Mohamed, M. M., Kiarash, M. and Mehdi, R. (2014). Modeling of self-similar network traffic using artificial neural networks.

[6] Kotenko, I., Saenko, I., Lauta, O. and Kribel. A. (2020). An approach to detecting cyber attacks against smart power grids based on the analysis of network traffic self-similarity. *Energies 2020*, 13, 5031; doi:10.3390/en13195031

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

[8] Dymora, P.; Mazurek, M. (2021). Influence of model and traffic pattern on determining the self-similarity in IP networks. Appl. Sci. 2021, 11, 190. https://dx.doi.org/10.3390/app 11010190

[9] Fras, M., Mohorko, J. and Učej, Z. C. (2010). Modeling of measured self-similar network traffic in OPNET simulation tool. *Informacije*, Ljubljana, 40(3): 224-231

[10] Raimundo, M.S.; Okamoto, J., (2018). Application of hurst exponent (H) and the R/S analysis in the classification of FOREX securities. Int. J. Model. Optim. 2018, 8, 116–124.

[11] Yasir, S. and Olivia, D. (2015). Modeling website workload using neural networks. *arXiv:1507.07204v1* [cs.DC],: 1 - 25.

[12] Song, H. and Gan, L. (2015). The research on the prediction of the network traffic based on the improved iac-gray method, *CHEMICAL ENGINEERING TRANSACTIONS*, **46**: 1297 - 132.

[13] Computerweekly.com (2018). Quick and dirty wireshark tutorial. https://www.computerweekly .com/tutorial/Quick-and-dirty-wireshark-tutorial

[14] Sheela, K. G and Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering.* Vol. 2013, Article ID 425740, 11 pages. Hindawi Publishing Corporation. http://dx.doi.org/10.1155/2013 /425740.

[15] Zhani, M. F. and Elbiaze, H. (2009). Analysis and prediction of real network traffic. *Journal of networks*. 4(9): 855 – 865.