ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A CONCEPTUAL FRAMEWORK FOR NETWORK TRAFFIC CONTROL AND MONITORING USING ARTIFICIAL NEURAL NETWORKS

UZOMA RITA ALO[1], SYLVESTER I. ELE[2], HENRY FRIDAY NWEKE[3]

[1]Mathematics/Computer Science Department, Alex Ekwueme Federal University, Ndufu-Alike, Ikwo, Ebonyi State, Nigeria

[1,2]Computer Science Department, University of Calabar, Etagbor, P.M.B 1115, Calabar, Cross River State, Nigeria

[3]Computer Science Department, Ebonyi State University, P.M.B, 053, Abakaliki, Ebonyi State, Nigeria

E-mail: [1]alo.uzoma@funai.edu.ng, [2]myyrs2015up@gmail.com, [3]henry.nweke@ebsu.edu.ng

ABSTRACT

Over the years, efforts have been made by various researchers to optimize Wireless area network enterprise to improve network performance with reduced cost. With necessary and appropriate network control and monitoring methods, reliable QoS of network traffic can be achieved which in turn would improve connections especially with high reliance of today businesses and commercial enterprises on fast internet. Moreover, the need for efficient network monitoring to improve quality of services have driven many companies to employ Multiprotocol Label Switching circuit for connectivity to see how to have control over traffic flow to and from branch offices in order to achieve QoS with optimize WAN enterprise. In this paper, machine learning algorithms with various backpropagation algorithms are analysed for effective network traffic control and monitoring. Specifically, the paper analyze the impact of neural network approach with various network parameters to improved network quality of service (QoS). In this case, ten different Back-propagation training algorithms were used to carryout ten different training attempts in order to determine the algorithms with the best performance. The result showed that there is a perfect correlation between the predicted values of the neural network model and the target output which implies that the model was successful in the prediction of the network traffic flow. The result also confirmed that the training algorithm of Back-Propagation was sufficient for predicting network traffic flow using the BR algorithms.

Keywords: *Network Traffic Control, Neural Network, Multilayer Perceptron, Multiprotocol Label Switching, Quality of Service*

1. INTRODUCTION

Bandwidth logjams have suddenly made it more urgent to come up with better ways of modelling and dealing with future traffic growth so that networks don't experience total meltdowns [1]. In today's business ventures that require efficient network traffic for effective business transaction, the traffic forecast is becoming the most important factor in providing the quality of experience (QoE) to the end users. Again, understanding where wireless bandwidth will come from can keep communication systems from overloading. A network traffic control and management is fast becoming a necessity in converged business Networks, and with the growth of networks, it is demanding to predict the development of network traffic [2]. The network applications use traffic prediction results to maintain its performance by adopting its behaviors. Furthermore, network service providers deploy the prediction values in ensuring the better Quality of Service (QoS) to network users by admission control and load balancing using inter or intra network handovers [3].

Managing and Controlling network traffic requires limiting bandwidth to certain applications, guaranteeing minimum bandwidth to others, and marking traffic with high or low priorities. Maximum bandwidth is limited by the slowest part between the source and intended destination [4]. In a packet-based network, such as the internet, the transmission of information is carried out in discrete packets. The path that a packet follows when travelling through the network is determined by the routing algorithm [5]. Besides network accounting and monitoring, system administrators can identify various problems that may occur in the network. With the help of traffic-flow, it is possible to analyze and optimize the overall work performance in a network. The applications and merits of traffic flow predication and analysis has led to various research in order to improve current methods. One of the most applied approaches is the use of supervised machine learning. Supervised machine learning approaches rely on labelled data from networks to model network flow. Typical examples of supervised machine learning approaches recently implemented for network flow include support vector machine, decision tree, Naïve Bayes, k-Nearest Neighbors, artificial neural networks etc. [6]. For instance, Jamuna [7] conducted a flow based traffic classification and comparison on the various Machine Learning (ML) techniques such as C4.5, Naïve Bayes, K-Nearest Neighbors, and RBF for IP traffic classification. From the result obtained from their findings, C4.5 Decision Tree gives 93.33% accuracy compare with other algorithms. Chabaa et al [8] implemented an artificial neural network (ANN) model based on the multi-layer perceptron (MLP) for analysis of internet traffic data over IP networks. They applied the ANN to analyze a time series of measured data for network response evaluation.

Artificial neural network is the most widely applied methods for traffic flow prediction, monitoring and control [9] [10]. Artificial neural network provides fast prediction of subsequent input values and is essential for online prediction of time series data. In addition, recent evaluation of artificial neural networks for computer network traffic prediction showed high performance accuracy [6]. To improve the performances and minimize error rate of artificial neural network, back propagation function (BP) is used to update the network weights and biases. However, back propagation (BP) function suffers from partial derivatives and inability to converge easily. Furthermore, BP is affected by initial weight generation during training. To solve the problem, various training functions have been implemented by various studies. These include resilient back propagation(RP), Levenberge-Marquardt (LM),BFGS Quasi-Newton (BFG), Bayesian Regularization (BR), Scaled-Conjugate Gradient (SCG), Fletcher-Powell Conjugate Gradient (CGF), Conjugate Gradient with polak-Ribiere updates (CGP), Gradient Descent (GD), Gradient Descent with momentum (GDM), Gradient Descent with adaptive learning rate(lr) (GDA), Levenberge Marquardt (LM) and One Step Secant (OSS) [11] [12] [6]. For instance, according to Quesada [13], the BFGS Quasi-Newton was an alternative approach developed to remedy the drawback of the computationally expensive Newton's method which requires many operations to evaluate the Hessian matrix and compute its inverse. Tiwari et al [14] reported that the gradient descent algorithm updates the weight and biases along the steepest descent direction, but it is usually associated with poor convergence when compare with conjugate Gradient Descent algorithm which has a faster convergence. The BR is the linear combination of Bayesian methods and ANN to automatically determine the optimal regularization parameters [15]. Here, regularization involve imposing certain prior distributions on the parameters of the model. LM algorithm, on the other hand, is a backpropagation algorithm specifically designed to minimized sum of square error functions. It is an effective and second order learning algorithm used for fast and efficient training [16] [17]. Conjugate Gradient with Fletcher Reeves (CGF) uses the Fletcher Reeves (FR) update, while Conjugate Gradient with Polak/Rebiere Restarts uses the Polak-Rebiere (PR) update.

However, based on computational testing, the PR performs better than FR [18]. With varieties of back propagation algorithms for training neural network based network traffic prediction, it is challenging to choose the right training functions to ensure improved performance. There is need to tackle challenging issues such as: which of the backpropagation algorithm produce optimal results for network traffic prediction? What are existing procedure for network traffic prediction using neural networks? Therefore, this paper comprehensively evaluated ten (10) back propagation algorithms for computer network traffic prediction. From existing literature, no studies have implemented and evaluated different back propagation algorithms for traffic prediction. Our evaluation would assist network traffic predication and monitoring researchers with accurate knowledge of different efficient implementation of artificial neural network training with back propagation functions. Consequently, the contributions of this paper are outlined below:

- Propose machine learning and artificial neural network methods for network traffic control and management using real time traffic data;

- Comprehensively evaluate combination of artificial neural network parameters to ensure efficient and improved quality of services in the network;
- Extensive review of various approaches and methods for network traffic control and management recently implemented.

The remainder of this paper is organized as follows: Section 2 discusses background and related works, section 3 presents the proposed network traffic control framework, section 4 discuss experimental setup and results obtained while section 5 concludes the paper and provide future research directions.

2. BACKGROUD AND LITERATURE REVIEW

This section explains various approaches for network traffic control and management. Moreover, computationally based artificial neural networks approaches for network traffic management are also explained.

2.1 Network Traffic Control and Management

In every computer Network, there are a lot of communication devices trying to access resources and at the same time getting requests to carry out some tasks for other device. Moreover, at the same time certain types of communication devices may be busy to respond to the request being made to them. Consequently, there are bundles of information exchange in the network in form of request, response and control data. These data are essentially in the form of a huge number of packets floating around in the network. This huge amount of data acts as a load on the Network, which results in slowing down the operations of other communication devices due to delay in communication activities in the network [19].

Traffic control, management, monitoring and analysis can be carried out for several different reasons. It may be to examine or investigate the usage of network resources; evaluate and assess the performance of network applications; regulate and adjust Quality of Service (QoS) policies in the network; log the traffic to comply with the law, or create pragmatic and realistic models of traffic for academic purposes [20]. Without effective traffic management and control, networks are vulnerable to possible congestion when the offered traffic exceeds the network capacity, leading to serious drop and deterioration of network performance. In their recent studies, Kalyvankar [19] defined network traffic as the density of data present in any Network while [21] refers to it as the set of traffic controls within the network that regulate traffic flows for the purpose of maintaining the usability of the network during conditions of congestion. Similar to the road traffic analogy, recent studies [22, 23] opined that in Network Traffic, when you send a request on the network, it is possible that due to some problem or other requests you have to wait for some time. If over a period of time a number of packets queue up and wait, then it results in traffic. As soon as traffic is created, you must wait till it is over, which can be for any length of time, depending on the situation. According to Rougha and Kalmanek [24], the resolution to this is Network Traffic Management and this process starts first with measuring the traffic on the network. Traffic management required even during overload, and also traffic management is required for continuous media [25].Traffic management has several objectives. It attempts to distinguish different types of traffic and handle each type in the appropriate way, e.g. is real-time traffic is forwarded with minimal delay while best-effort traffic can afford to wait longer for any unused bandwidth; traffic management responds to the onset of congestion, a good example is the TCP of a protocol that adapts the rate of TCP sources to avoid serious congestion; and lastly, traffic management seeks to maintain an acceptable level of network performance under heavy traffic conditions [26].

2.2 Neural Network Control and learning Model

Machine learning algorithms (learning models) are described as learning a target function (f) that best maps input variable (X) to an output variable (Y). In this case, Y = f(X). It is a learning task where predictions are made in the future (Y) given an input (X) [27].There are varieties of machine learning algorithms that have been proposed for network traffic managements and related applications. **Error! Reference source not found.** illustrates comparative analysis of these machine learning models, including their strengths and weaknesses and wide areas of industrial applications [28] [29] [27] [29] [30].

Among the machine learning model outline in

Table *1*, *artificial neural networks* are (Figure 1) widely used for traffic management due to its ability to effectively model raw data with good representational results [31] [32]. Artificial Neural networks are statistical learning models, inspired by

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Biological Neural Networks, that is the nervous system (e.g. the brain) and is used in machine learning [33] [32]. The algorithm is collection of neurons with synapses connecting them, organized into three main parts: *input*, *hidden* and *output* layers. Moreover, neural network models are machine learning frameworks that attempt to mimic the learning pattern of natural biological neural networks. Biological neural networks have interconnected neurons with dendrites that receive inputs, then based on these inputs produce an output signal through an axon to another neuron [34]. In their recent study, Jiang et al [35] characterized neural network model as a powerful learning tools employed in a wide range of applications, such as control of complex nonlinear systems, optimization, system identification, and patterns recognition. In addition, Fister et al [36] reported that neural network can approximate any unknown continuous non-linear function by overlapping the outputs of each neuron. Furthermore, Zhang et al [37] noted that, the approximation errors could be made arbitrarily small by the learning process in a neural network is called the training algorithm, in which there are many different algorithms, with different characteristics and performance. The artificial neurons are interconnected and communicate with each other. Each connection is weighted by previous learning events and with each new input of data more learning takes place. Ho [39] went further to say that one example of Neural network algorithm is Deep Learning, which is especially concerned with building much larger complex neural networks.

Moreover, neural network model provide means to learn the underlying relationship between input

- The neuron inputs to the networks, Pi
- The weights (that is the coefficient which can be adapted to the networks), W_i
- The propagation function, which enables the network to obtain from inputs and weights, the value of post-synaptic potential of the neuron h_i. The most common function being the addition of all inputs. The Activation or Transfer function, *f*. The result of the propagation function is transformed in the real output of the neuron through activation function.

3. PROPOSED NEURAL NETWORK FRAMEWORK FOR NETWORK TRAFFIC MANAGEMENT

In this study, a scenario of Multiprotocol Label Switching (MPLS) connectivity for a WAN connectivity and decentralized network access of a company was considered, in which each branch, including the head office is connected to each other (in a mesh) via MPLS circuit. The Leased Lines of the network is taken at the head office and all the individual branches can access the internet by accessing the head office network first via the MPLS. Our choice of the MPLS technique was motivated by the fact that WAN enterprises try to figure out how to optimize its use with less expensive connections such as the internet [40]. The architecture of the model network as depicted in Figure 2 is such that it gives more control to the head office over what is and what can be accessed over the entire network. The internet at the head office is shared between multiple branches, thereby minimizing the underutilization of available bandwidth at any point of time.

In the WAN connectivity, each branch has its own internet access using internet Lease Lines and broadband connections. In this case, the inter branch communication (ERP, VoIP, Video conferencing, etc.) travels in the MPLS circuits between the branches and the traffic go to the internet Leased Lines from branch itself. This type of technique gives the best performance for real time traffic, data traffic and internet traffic [40].

The MPLS is a transport technology and a data gathering network basically used for high performance network such as telecommunication network. The MPLS is a middleman protocol between layers 2 and 3 in the OSI model, providing additional features for transport of data across the network. It uses packet-forwarding technology known as Labels in order to make data forwarding decisions. To make MPLS work, a Label-switched paths (LSPs) is needed. The LSP is essentially a unidirectional tunnel of MPLS information exchange among routers in an MPLS network. In addition, MPLS router operates on preset paths for various sources to destination. Every router on the LSP must be able to switch the packet forward for efficiencies over typical Internet Protocol (IP) routing to be accomplished.

To justify the choice of MPLS technology as the development framework for this study, a comparative analysis of two popular WAN connectivity options, MPLS and Ethernet network connectivity [41] is established as shown in Error!

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Reference source not found., using factors such as scalability, WAN routing, cost, WAN protocol behaviour, common applications, and Quality of service (QoS). The table clearly shows why majority of organizations prefer MPLS WAN connectivity. Some of its pro include outsourced routing, any-to-any connectivity, built-in support for quality of service (QoS), and service level agreement.

Table 1. Different machine Learning Techniques, their Strengths, Weaknesses and Industrial Application

Learning Algorithms	**Strength**	**Weakness**	**Area of wide industrial Applications**
Decision Tree	It is flexible in terms of data type of input and output variables	Decision criteria only consider one input attributes at a time	It is use in finance for option pricing, Remote sensing for pattern recognition and in Medicine to identify at-risk patients and disease trends
Linear Regression	It has very high performance in both scoring and learning	Linear assumption of input features, which is mostly false	It is use in business, for sales forecasting based on the trends and Risk assessment
Neural networks	Have outperformed other approaches in a large datasets,	Requires large amounts of data, computationally expensive to train	Fingerprint classification, performance optimization in manufacturing systems, Air craft detection in SAR images, image recognition, image captioning, etc.
Bayesian Networks	It converges faster and requires relatively little training data	Categorical variable must be transformed into multiple binary variable	Document classification, Sentiment analysis and Email Spam filtering
Support Vector Machine	It can handle large number of dimensions and does not make any strong assumptions on data	They can be abysmally slow in test phase.	It is commonly used for stock market forecasting by various financial institutions
Nearest Neighbors (KNN)	The strength of K nearest Neighbors is its simplicity as no model needs to be trained	The weakness of KNN is it doesn't handle high number of dimensions well	Applied in the agricultural related field, and is used in search application.
Logistic Regression	Easier to inspect and less complex, and can also handle non-linear effects	May over fit sparse training data, it cannot predict continuous outcomes and not robust to outliers and missing values.	It is applied in the field of epidemiology to identify risk factors for diseases, in weather forecasting and in credit scoring systems for risk management.
Apriori Algorithm	It makes use of large item set properties and it is easy to implement.	Waste of time to hold a vast number of candidate sets with much frequent item sets,	Use in health care for detecting adverse drug reactions, in E-Commerce for market basket analysis and for Google auto-complete.

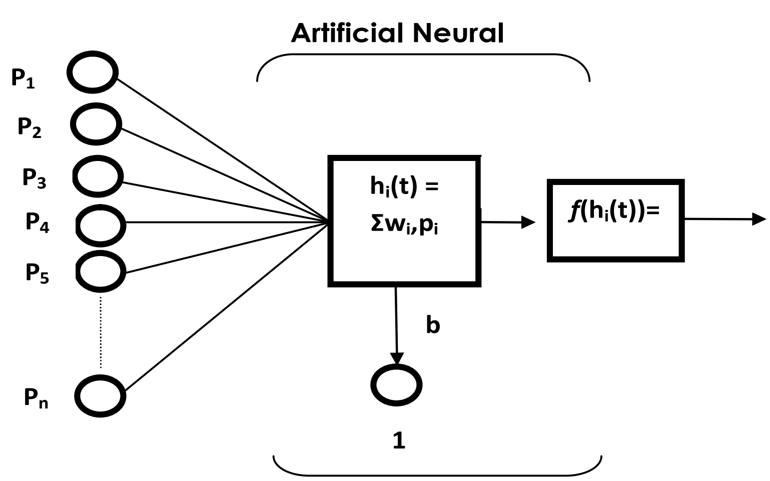

Figure 1. Typical Neural Network Learning Model [8]

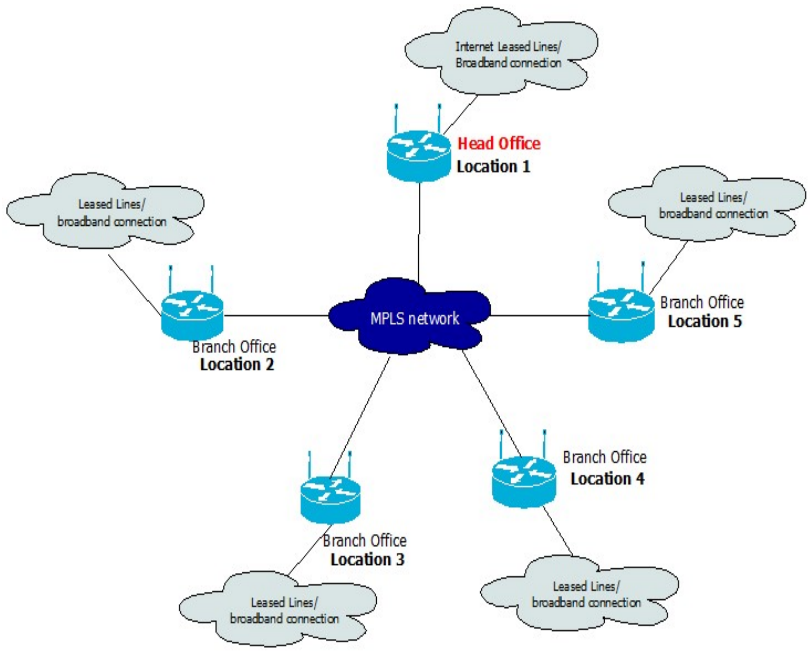

Figure 2. Diagram Of The MPLS Network Architecture

Table 2. Comparison Of MPLS And Ethernet Networks

Indicators	**MPLS**	**Ethernet**
Scalability	MPLS can scale to over thousands of sites.	Ethernet can scale up to hundreds of sites.
WAN routing	MPLS allows business to leave WAN routing to the service provider and keep fewer WAN engineers and staff.	Ethernet gives WAN engineers control and responsibility over routing.
Cost	MPLS typically cost more than Ethernet but less than T1 lines.	Ethernet is more affordable than MPLS.
WAN protocol behaviour	MPLS can handle any-to-any connectivity, including voice and video.	Ethernet has low latency and high throughput.
Common application	MPLS is best and most widely used to interconnect data centres with branches offices and branches to other branches.	Ethernet is best for interconnecting data centres.
Quality of service (QoS).	MPLS has QoS options to enable preferential treatment of latency-sensitive traffic like VoIP.	Network engineers bypass QoS complexity by hooking switches directly to Ethernet pipes.

Therefore, the goal of the study is to improve network performance especially the Multiprotocol Label Switching and make efficient use of network resources by better matching the resources with traffic demands. Consequently, network connectivity will be optimize by ensuring that all packets and loads in the network are balanced, allocation of resources for variable bit (VBR) streams, and increase utilization of network resources to fulfil quality of service (QoS) requirements. Instead of relying on the network hardware, which is expensive, we proposed a *neural network* base framework for the network control, using predictive scheme.

The predictive scheme was necessary due to its efficient prediction of traffic generated by multimedia sources. Moreover, it enables effective traffic and congestion control procedures at the network edges [42].With Multiprotocol Label Switching, the first time a packet enters the network, it is assigned to a specific forwarding equivalent class (FEC), indicated by a short bit sequence and the label to the packet. Each router in the network has to indicate how to handle packets of a specific FEC type, so once the packet has entered the network, routers do not need to perform header analysis.

In this framework, packet streams, ρ_i, was considered from a network source, S, to a destination node, d, as a sequence. Each pack

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

sequence, ρ_i is composed of chunks of packets, $\rho_1, \rho_2, \rho_{3_i}, \ldots \rho_n$. as presented in equation (1)–(3).

$$\rho \in \sum_{i=1}^{n} \rho_i \; \forall C_n \; \ldots\ldots\ldots \quad (1)$$

$$\rho_i = \sum \rho, \rho \subset C_n \; \ldots\ldots\ldots \quad (2)$$

$$C_n = \cup_i \rho_i; \; 0 < i \leq n \leq \ldots\ldots \quad (3)$$

Where

ρ is chunk of packet in the network connection

ρ_i is the packet sequence in the network

C_n is the network connection

ρ_g is the total packets in the network

Equation (4)–(6) shows that for any point to point connection, C_n from a source, S, to a destination, d, then

$$C_n \in S \rightarrow d, \; but \; C_n \notin d. \; \ldots\ldots\ldots \quad (4)$$

$$\Longrightarrow C_n(S) = C_n(d).\; C_n^s = C_n^d \ldots\ldots \quad (5)$$

$$\text{If } C_n(S) \neq C_n(d),\; C_n^s \neq C_n^d. \ldots\ldots\ldots \quad (6)$$

Equation (7)-(8) indicates that for any point to multi point connection, $\rho \in \rho_i;\; 0 < i \leq n.$

$$\rho_i = \sum \rho,\; \rho_i \in C_n \; \ldots\ldots \quad (7)$$

$$\rho_\theta = \sum_{i=1}^{n} \rho_i \, \rho_i \subset \rho_\theta \; \forall \; C_n \; \ldots\ldots \quad (8)$$

$$\text{for } C_n^i \in lS \rightarrow d_i$$

$$C_n^i \in S \Longrightarrow C_n^i \in d_i \; iff \; n > 0$$

Where

C_n^i is the entire network connections and,

d_i is the destinations in the network.

Our model uses a multi-layer perceptron (MLP) model which is a frequently used neural network with feed-forward (forward propagation). We adopted two hidden layers between the inputs and output layers; layer has four neurons with a fully connected network. Using notation representation, each node in the hidden layer is a perceptron with a collection of weight defined in the path. Because there are multiple nodes which uses double search rate with W_{ij} indicate the weight for node i associated with the feature j at the previous layer. This is diagrammatically represented in **Error! Reference source not found.**.

The sequence of packet ρ_i formed the input to the network. The packet sequence comprises of the chunks of packets from the source to a node (destination) in a particular connection. A weight on the network is of the form W_{jk}^l where i denotes the number of layer; j denotes the number of neuron from the i^{th} layer. k is the number of neuron form the $(i+1)^{th}$ layer. Because the Activation of neural network is basically measured on how the relevant weighted sum is, we represent the weight as a matrix. The number of rows correspond to the number of features (inputs) and the number of columns corresponds to the number of neurons in the layer number 2. The W^1 is a n x 4 matrix. The activities of the first hidden layer (the values of its neuron) will be stored in a matrix called A^2. But we first compute $Z^{(2)}$ as shown in equation (9).

$$Z^{(2)} = P_g . W^1 .. \quad (9)$$

With the total packets in the entire network represented as P_θ . This is computed as indicated in equation (10).

$$P_\theta = [P_i \quad P_{i+1} \quad P_{i+2} P_{i+3} \ldots P_{i+n}] \ldots \quad (10)$$

As we apply an activation function element by element to $Z^{(2)}$ we get $R^{(2)}$, which is the activities of the second hidden layer. The Hyperbolic

Tangent Activation function, *tanh* is used in this study. The *tanh* is an activation function with output values range is (-1, 1).

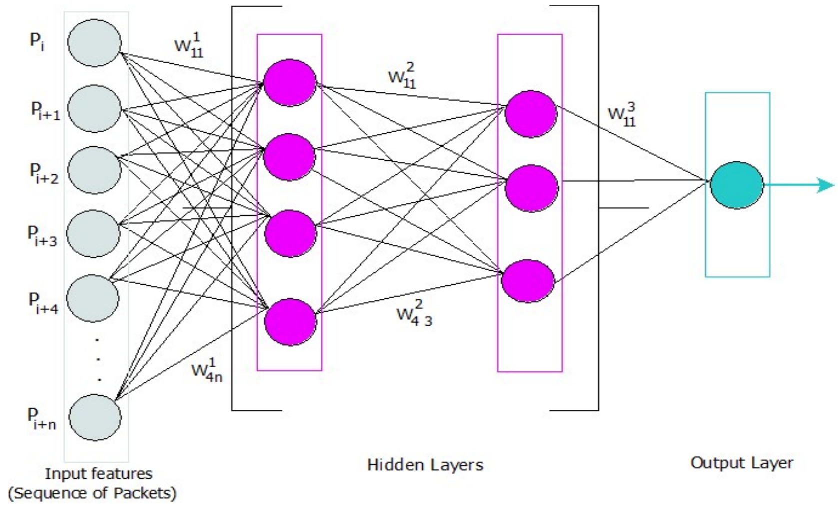

Figure 3. Typical architecture of Neural Network Training Model

The hyperbolic tangent function, function of the feed-forward, are defined in equations (11)-(13).

$$g(x) = \sigma(x) = \tanh(x)\frac{\mathrm{Sinh}\,x}{\mathrm{Cosh}\,x} \ldots\ldots$$

$$= \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \ldots\ldots \quad (11)$$

Computing the gradient for *tanh* is the function of a feed-forward activation evaluated at x.

$$g(x) = \tanh(x)\frac{\partial\,\mathrm{Sinh}(x)}{\partial\,\mathrm{Cosh}(x)}$$

$$= \frac{\frac{\partial}{\partial t}\mathrm{Sinh}(x)\,Cosh(x) - \frac{\partial}{\partial t}\mathrm{Sinh}(x)\,Cosh(x)}{\mathrm{Cosh}^2\,(x)}$$

$$\frac{Cosh^2\,(x) - Sinh^2\,(x)}{\mathrm{Cosh}^2\,(x)}$$

$$= 1 - \frac{\mathrm{Sinh}^2(x)}{\mathrm{Cosh}^2(x)}$$

$$= 1 - tanh^2(x). \ldots\ldots \quad (12)$$

Consequently, using equation (11), the neuron of the first hidden layer yields

$$\sigma(x) = \sigma(P_\theta) = \tanh(P_\theta)\frac{e^{P_\theta} - e^{-P_\theta}}{e^{P_\vartheta} + e^{-P_\vartheta}} \ldots\ldots \quad (13)$$

To obtain $A^{(2)}$, the activities of our second layer, we apply matrix multiplication of $Z^{(2)}$ element by element (see equation (14)).

$$A^{(2)} = \sigma(Z^{(2)})$$

$$A^{(2)} = \partial\left(Z^{(2)}\right) = \tanh\left(Z^{(2)}\right). \ldots\ldots \quad (14)$$

$Z^{(2)}$ is an intermediary matrix that holds the value of $P_\vartheta \,.\, W^1$.

Now, instead of using P_θ as input, we will use $A^{(2)}$ and repeat exactly the same steps above and add weight from the layer two to layer three. Again, we organized the weight as a matrix, $W^{(2)}$, such that, each row corresponds to the number of neurons in the layer 2 (hidden layer 1) and each columns corresponds to the number of neurons in the layer 3 (hidden layer 2).

Since for any given layer (for MLP), the input of any given layer is always the output of the previous, we trained out network such that our data for our second layer, that is $A^{(2)}$ becomes the input to our next layer. This is illustrated in equation (15).

$$Z^{(3)} = A^{(2)}\,.\,W^{(2)}Z^{(3)}. \ldots\ldots\ldots \quad (15)$$

Whatever the result of the $Z^{(3)}$ is, the output of the next layer which is $A^{(3)}$ becomes equation (16).

$$A^{(3)} = \partial(Z^{(3)}) = \tanh(Z^{(3)})$$

$$A^{(3)} = \tanh\left(Z^{(3)}\right). \ldots\ldots \quad (16)$$

Having obtained $A^{(3)}$, the next step is to connect the last layer, applying the same procedures as the previous. We calculate $Z^{(4)}$ using $A^{(3)}$ which is the result of the previous layer as our input features. We still organize the synapses (weights) into matrix, $W^{(3)}$. See equation(17).

$Z^{(4)} = A^{(3)} . W^{(3)} . . . .$ (17)

We have to add biases to our network, biases are learnable parameters and each neuron has a bias. The addition of biases will help us determine if our neurons are activated (fired) or not, and helps to increase the flexibility of our model. For instance, in our model, we consider our neuron to be fired when the output falls between 0 and 1. If output is less negative, we apply rectilinear activation function (RELU) and it becomes zero immediately, a situation that will decrease the ability of our model to train from the data properly. But when we add a bias, it automatically eliminates every negative values that may result in the final results.

We then introduce a bias for inactivity in order to be able to regulate the network to determine when the network is activating meaningfully; we organized all biases into a vector and added it to the matrix, our final matrix is seen in equation (18).

$$Z^{(4)} = A^{(3)} . W^{(3)} . b_i^{(3)} . . . . \quad (18)$$

Where i = 1, 2, 3, ..., n

4. EXPERIMENTS

4.1. Dataset Description

To analyze and evaluate the methods described for neural network based network control and monitoring, dataset collected at Universidad De Cauca was selected as experimental sample [43] In addition, we used version 2 of the dataset that can be downloaded from Kaggle (https://www.kaggle.com/jsrojas/ip-network-traffic-flows-labeled-with-87-apps). The dataset collected for six days (26, 27, 28 April and 9, 11, 15 May, 2017) were stored in comma separated format (csv) with 87 features. Some of the features include IP network information, inter-arrival time, layer 7 protocol etc. Moreover, the data contain mixed type composed of numeric, nominal and date. In this study, the dataset was divided into train and test set using 70% and 30% respectively. To improve the network prediction performance and prevent saturation of the artificial neural network, the dataset was normalized using the Z-score and limit the data point to certain ranges [0, 1], according to [44]. This has shown to be effective in similar task [12]. The normalization equation is shown in equation (19) below:

$$X_i = \left(\frac{X - X_{min}}{X_{max} - X_{min}}\right) . \quad (19)$$

Where:

X_i - is the normalized value;

X - is the original value;

X_{min} - is the minimum value among all the values for related variable;

X_{max} - is the maximum value among all the values for related variable.

4.2. Evaluation Criteria

To assess the prediction of the network using the dataset described earlier, two performance metrics: Mean Square Error (MSE) and correlation coefficient are used [45]. MSE is a scale – dependent metric that measures the differences between the predicted values and the actual values of the data being computed. Equation (20) explains how the MSE is computed.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2 . . . . \quad (20)$$

The coefficient of correlation (R) value provide alternative indicator between the predicted and the experimental value. The R is computed as shown in equation (21).

$$R = 1 - \left(\frac{\sum(ei - pi)}{\sum pi}\right)^2 . . . . \quad (21)$$

4.3. Results and Discussions

The aim of training artificial neural network is to minimize errors and maximize the performance through optimization of weights and biases of the network. Major studies use adaptive approach to the networks through performance functions. The most widely used artificial neural network function is the back propagation function [11]. Back propagation function helps to minimize the error function and improve performance during training. However, back propagation algorithm tends to get trapped in local minima and lack convergence speed due to complex nature of finding the minimum errors. In addition, the training function maybe affected by the process of initial weights generation during network training [12]. Other heuristic training functions have been utilized by various studies to training artificial neural networks. Other training algorithms include BFG (BFGS Quasi-Newton), BR (Bayesian Regularization), SCG (Scaled-Conjugate Gradient), CGF (Fletcher-Powell Conjugate Gradient), CGP (Conjugate Gradient with polak-Ribiere updates), GD (Gradient Descent), GDM (Gradient Descent with momentum), GDA (Gradient Descent with adaptive lr), LM (Levenberge Marquardt) and OSS (One Step Secant) [46]. Here, we comprehensively and iteratively evaluate the impact of these training algorithms for artificial neural network based methods for network control and monitoring.

Table 3. Values For Different Statistical Measures And Percentage Accuracy

Training attempt	NET SIZE (Architecture)	Algorithm	EPOCHS	MSE
1	9-10-1	BFG	500	0.078616
2	9-10-1	BR	500	0.001619
3	9-10-1	SCG	500	0.054526
4	9-10-1	CGF	500	0.099485
5	9-10-1	CGP	500	0.099336
6	9-10-1	GD	500	0.110570
7	9-10-1	GDM	500	0.090543
8	9-10-1	GDA	500	0.062869
9	9-10-1	LM	500	0.060097
10	9-10-1	OSS	500	0.986370

The experimental evaluations were implemented in MATLAB 2016 (R2016b) version (https://in.mathworks.com/) using the Neural network toolbox on system computer running on window 10 Operating systems. The system is using an Intel CoreTM I7-6700 CPU @ 3.400GHz with installed Random Access memory (RAM) capacity of 16GB.During the training, ten different back-propagation training algorithms (learning functions) as listed above were considered in ten different training attempts, using 500 Epochs for each algorithm, and the iteration for the performance was recorded. Three learning functions with better performances were further selected and retraining was carried out twice with different network sizes (Architectures) for each selected function, using the same number of epochs (500) and the iterations for best performance noted. The aim is to know which algorithm and at what architecture we can obtain the minimized MSE (best performance) and highest R value of the network.

4.3.1. Performance Evaluation of Impact of Backpropagation Algorithm in Network Traffic Control

This section discusses the impact of the learning algorithms on the network performance, using the mean Square Error (MSE) as the performance function. The Mean Square Error is the average Square error between the outputs and the targets. Lesser values are superior while zero implies no error. The network was trained using different algorithms in each training attempt. Uniform network architecture of 9-10-1was maintained and training carried out at training cycles (epochs) of 500. After the first training regime, the best Mean Square Error (MSE) were found to be (0.001619, 0.054526, and 0.060097) for Bayesian Regulation BR, SCG, and LM, respectively(

Table *3*). The performances (MSE) for these algorithms are shown in **Error! Reference source not found.**.

The aim of the study was to obtain a more accurate result (best possible performance) from the network, thus the three algorithms with the best MSE were retained and retrained in the second experimental setup, using three different network topology (architecture) for each retained algorithm. The result from the second experimental setup is illustrated in **Error! Reference source not found.**. As seen in the table, the best results for training and testing of BR (Trainbr) were obtained in terms of minimum MSE values of 0.000024, 0.001287 at network size of 9-20-1 and 9-10-1, respectively. The closest performance is the SCG with minimum MSE value of 0.002314 at network size of 9-15-1.

Error Histogram was viewed to obtain additional verification of the network performance. As indicated in **Error! Reference source not found.**, the blue bars in the error histogram represent training data, green represents the validation data, and the red bars represents the Testing data.

The results show that the training algorithm of Back-Propagation was sufficient for predicting network traffic flow for all packets. Also, from the result, the study concludes that the Bayesian Regularization (BR) and Scaled Conjugate Gradients (SCG) gives more precision and are more suitable to predict network traffic flow for all packets using artificial neural network algorithm. Researchers such as [18] [16], had in their separate studies of the applications of backpropagation algorithms network training concluded that Levenberg-Marquardt exhibits better prediction accuracy than any other algorithms.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

However, none of these researchers included Bayesian Regularization (BR) was not considered in their studies. Nevertheless, the result of the present study agreed with a study of various training algorithms on neural networks conducted by [47], which established that SCG back-propagation training function gives better prediction accuracy than the CG, BFGS, GDM, OSS, CGF, GDX and GDA.

Table 4. Values Of Different Statistical Indicators For The Retained (Or Restrained)

Training Attempt	Net Architecture (size)	Algorithm	Epochs	MSE
1	9-10-1	BR	500	0.001287
2	9-10-1	LM	500	0.060131
3	9-10-1	SCG	500	0.063485
4	9-15-1	BR	500	0.012320
5	9-15-1	LM	500	0.097130
6	9-15-1	SCG	500	0.002314
7	9-20-1	BR	500	0.000024
8	9-20-1	LM	500	0.070510
9	9-20-1	SCG	500	0.011384

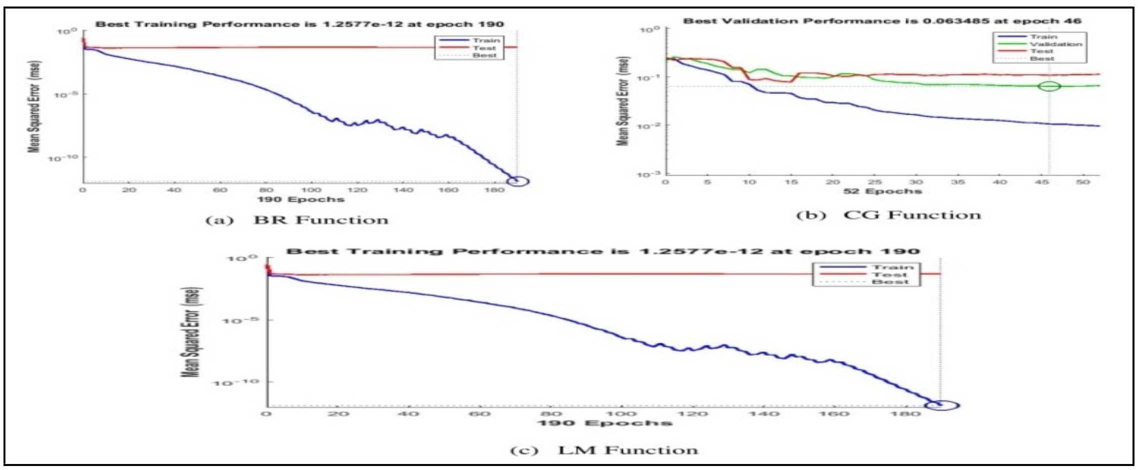

Figure 4. Performances (MSE) Of (A) BR, (B) CG And (C) LM

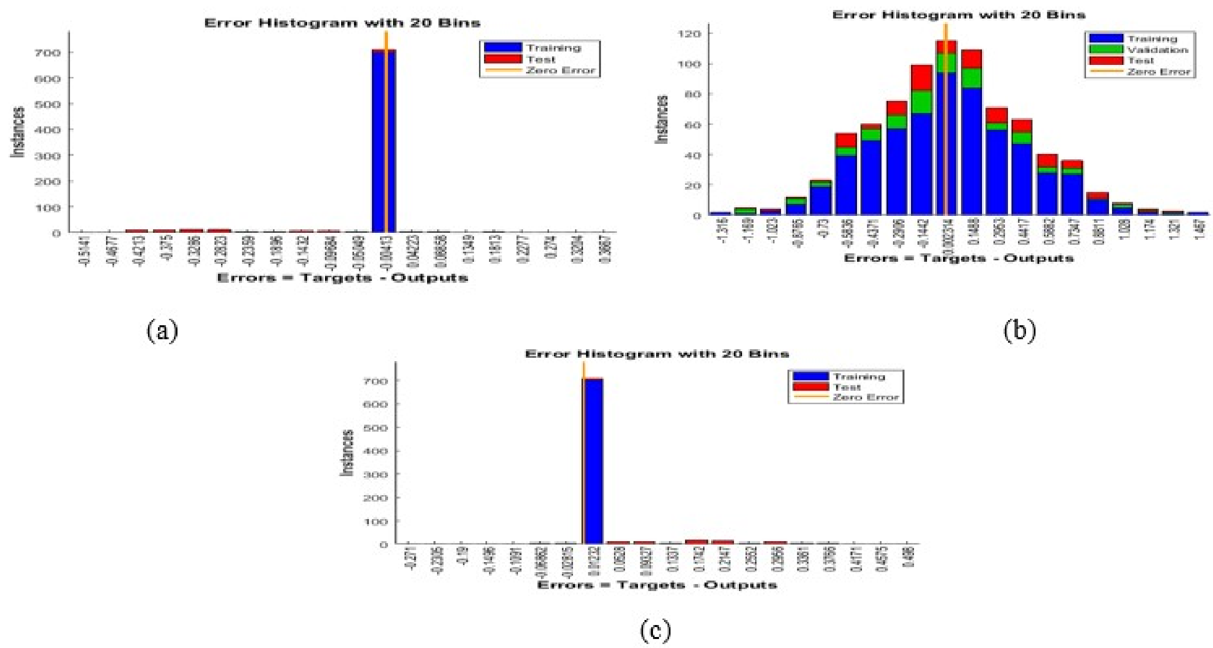

Figure 5. Error Histogram Using Different Back Propagation Function (A) BR, (B) SCG And (C) LM Functions

Therefore, this study can conclude that BR and SCG algorithms exhibited an efficient performance on network control and monitoring.

4.3.2. Coefficient of Correlation of Predicted Network Performance

This section discusses the performance in terms of correlation coefficient (R) of the predicted network. The correlation coefficient can vary between -1 and +1. R values closer to +1 represents the most accurate prediction. The coefficient of correlation (R) value provide alternative indicator between the predicted and the experimental value. For our first training setup**,** as illustrated in **Table 5**, the maximum R values of 1.00000, 0.83005 were obtained when the dataset were trained and tested in Bayesian Regularization (BR) and Scaled Conjugate

Gradients (SCG), respectively. Whereas, in the second experimental setup, after retraining the best three algorithms from the first training regime, the maximum R values of 1.00000, 0.99063 were found when the dataset was trained in BR and SCG, respectively. **Error! Reference source not found.** provide further statistical details of the analysis. The analysis of the results shows that there is a perfect correlation between the predicted values of the neural network model and the actual output values, implying that the model was successful in the prediction of network traffic flow. It further confirms that gives more precision when compare to other algorithms. The scattered diagrams (correlation Coefficient) of the measured and predicted output for BR, SCG and LM are illustrated in **Error! Reference source not found.**, **Error! Reference source not found.** and **Error! Reference source not found.**. **Error! Reference source not found.** shows the training state (gradients) for the model validation using the BR algorithm.

Table 5. Values Of Coefficient Of Correlation (R) For Different Training Algorithms

Training attempt	NET SIZE (Architecture)	Algorithm	Epochs	R
1	9-10-1	BFG	500	0.11168
2	9-10-1	BR	500	1.00000
3	9-10-1	SCG	500	0.83005
4	9-10-1	CGF	500	0.23002
5	9-10-1	CGP	500	0.12438
6	9-10-1	GD	500	0.16032
7	9-10-1	GDM	500	0.09054

8	9-10-1	GDA	500	0.06287
9	9-10-1	LM	500	0.08042
10	9-10-1	OSS	500	0.14549

Table 6. Values Of Different Statistical Indicators For The Retained (Or Restrained) Functions

Training Attempt	Net Architecture (size)	Algorithm	Epochs	R
1	9-10-1	BR	500	0.99063
2	9-10-1	LM	500	0.26882
3	9-10-1	SCG	500	0.57860
4	9-15-1	BR	500	1.00000
5	9-15-1	LM	500	0.14369
6	9-15-1	SCG	500	0.15889
7	9-20-1	BR	500	1.00000
8	9-20-1	LM	500	0.00163
9	9-20-1	SCG	500	0.88378

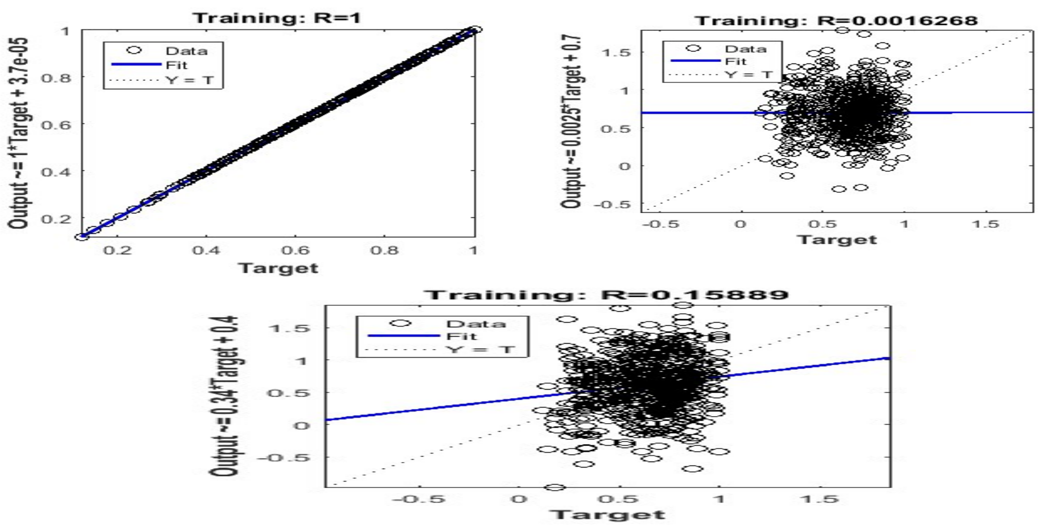

Figure 6. Correlation coefficient (R) for the network Performance using BR back propagation

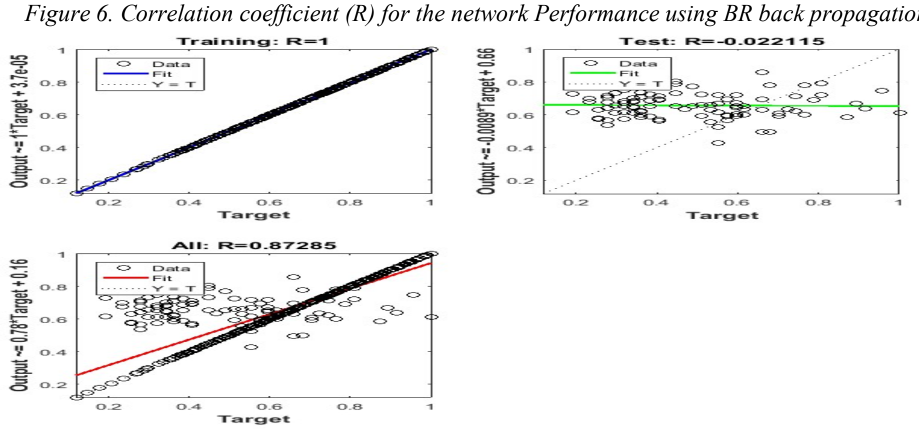

Figure 7. Correlation function (R) for network performance using CG back propagation function

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

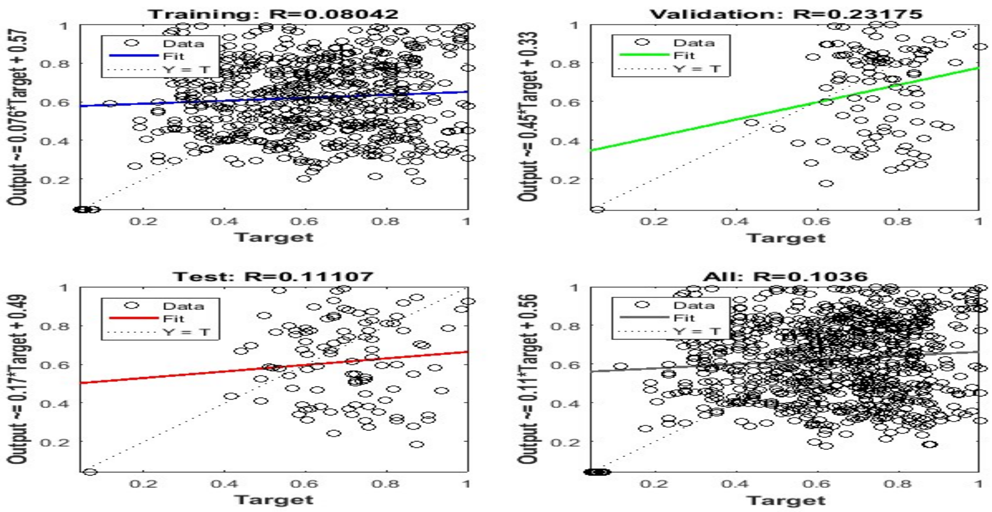

Figure 8. Correlations coefficient (R) for LM back propagation function

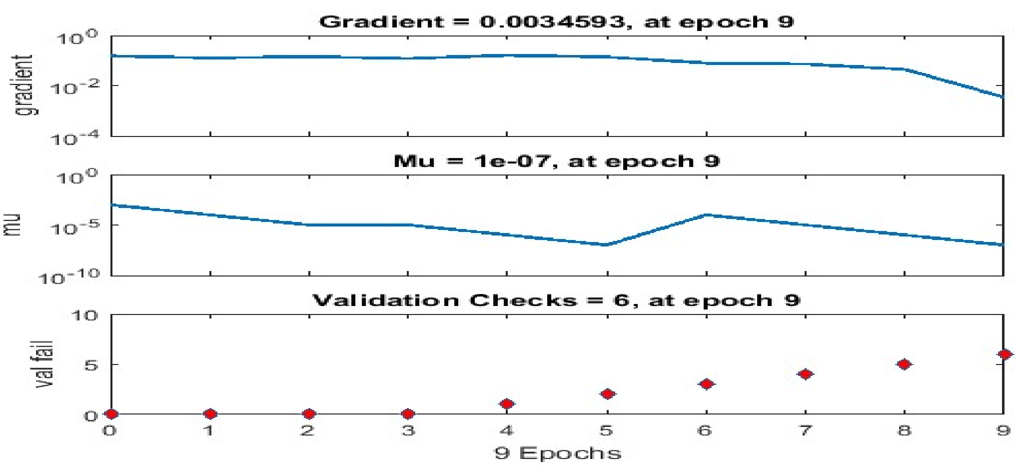

Figure 9. Training State For Model Validation Using BR Back Propagation Function

5. CONCLUSION

Well-organized and efficient prediction schemes are extremely significant in achieving required Quality of services of network traffic. The goal is to improve network performance and make efficient use of network resources by better matching the resources with traffic demands. This will optimize network connectivity by ensuring that all packets and loads in the network are balanced; ensure allocation of resources for variable bit (VBR) streams; and increase utilization of network resources to fulfill quality of service (QoS) requirements. We have presented a systematic and step by step procedure, using the hyperbolic tangent activation functions (*tanh*) on how the proposed framework handled the training session from the inputs features and make predictions.

In the study, the Mean Square Error (MSE) and the correlation coefficient (R) were selected as criteria to evaluate the networks find the optimum solution. From experimental evaluations, Bayesian Regularization algorithm (BR) that is, the Trainbr, showed a substantial rise in performance when the neurons in the hidden layer was raised to 20 (9-20-1 network architecture), with the MSE and R values found to be (0.000024) and (1.00000), respectively. The results showed that the training algorithm of Back-Propagation was sufficient for the prediction of network traffic flow.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1] Neil, S. (2010). Forecasting Network Traffic to Avoid Meltdowns. Business Report. Available at: https://www.technologyreview.com/s/422043/forecasting-network-traffic-to-avoid-meltdowns/

[2] Tian, H., Zhou, X. and Liu, J. (2017). A Hybrid Network Traffic Prediction Model Based on Optimized Neural Network. 2017 18th International Conference on Parallel and Distributed Computing, Application and Technology (PDCAT), 18-20 Dec. 2017. IEEE.

[3] Gowrishankar and Satyanarayana, P. S. (2008). Neural Network Based Traffic Prediction for Wireless Data Networks. International Journal of Computational Intelligence Systems, Vol.1, No. 4. pp. 379-389.

[4] Andrew, J. (2017). How to Increase Router Bandwidth. Available at: https://itstillworks.com/increase-router-bandwidth-25417.html. Retrieved: August 20, 2018:5.00pm.

[5] CISCO (2018). QoS: Classification Configuration Guide, Cisco IOS XE Release 3S. Cisco Systems, Inc. 170 West Tasman Drive. San Jose, CA 95134-1706, USA. http://www.cisco.com.

[6] Oliveira, T. P., Barbar, J. S., &Soares, A. S. (2016). Computer network traffic prediction: a comparison between traditional and deep learning neural networks. *International Journal of Big Data Intelligence*, *3*(1), 28-37.

[7] Jamuna .A. and Ewards S.E (2013). Efficient Flow based Network Traffic Classification using Machine Learning. International Journal of Engineering Research and Applications (IJERA). Vol. 3, Issue 2. Pp.1324-1328.

[8] Chabaa, S., Zeroual, A. and Antari, J. (2010). Identification and Prediction of Internet Traffic Using Artificial Neural Networks. In the Journal of Intelligent Learning Systems & Applications. Scientific Research. Vol.2, pp.147-155

[9] Ang, M. K.J., Valla, E., Neggatu, N. S. and Moore, A. W. (2017). Network Traffic Classification via Neural Networks. A Technical Report. University of Cambridge, Computer Laboratory (UCAM-CL-TR-912). Available at: http://www.cl.cam.ac.uk/Retrieved: August 23, 2018:2.40pm.

[10] Carro, B., Lopez-Martin, M., Esguevillas, A. S. and Lloret, J. (2017). Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things. Special Section on Big Data Analytics In Internet Of Things and Cyber-Physical Systems. IEEE Access. VOLUME 5, pp. 18042 – 18050.

[11] Schalkoff, R. J. (1997). Artificial Neural Networks. McGraw-Hill International Editions

[12] Chiroma, H., Abdulkareem, S., &Herawan, T. (2015). Evolutionary Neural Network model for West Texas Intermediate crude oil price prediction. *Applied Energy*, *142*, 266-273.

[13] Quesada, A. (2019). 5 Algorithms to Train a Network. Machine Learning blog. Neural designer. Available at: https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network. Retrieved: May 26, 2019:3.07pm.

[14] Tiwari, S. Naresh, R. and Jha, R. (2013). Comparative Study of Backpropagation Algorithms in Neural Network Based Identification of Power System. International Journal of Computer Science & Information Technology (IJCSIT). Vol. 5(4).

[15] Okut, H. (2016). Artificial Neural Networks – Model and Application. Intechopen. http://dx.doi.org/10.5772/63256.

[16] Sapna, S, Tamilarasi, A. and Kumar, M. P. (2012). Backpropagation Learning Algorithm Based on Levenberg Marquardt Algorithm. Journal of Computer Science and Information Technology. pp. 393-398. DOI : 10.5121/csit.2012.2438.

[17] Nawi, N. M., Khan, A. and Rehman, M. Z. (2013). A New Levenberg Marquardt Based Back Propagation Algorithm Trained with Cuckoo Search. Fourth International Conference on Electrical Engineering and Informatics (ICEEI, 2013), pp 18-23. Procedia Technology, ScienceDirect. Available at: www.sciencedirect.com. Retrieved: May 27, 2019:3.20pm.

[18] Comert, Z. and Kocamaz, A. F. (2017). A Study of Artificial Neural Network Algorithms for Classification of Cardiocotography Signals. BitlisEren University Journal of Science and Technology. Vol. 7(2), pp 93-103.

[19] Kalyankar, N. V. (2009). Network Traffic Management. Journal of Computing, Volume 1, Issue 1. pp. 191 – 194. https://Sites.Google.Com/Site/Journalofcomputing/

[20] Bujlow, T. (2017). Classification and Analysis ofComputer Network Traffic. Unpublished PhD Thesis, Networking & Security

Department of Electronic Systems Aalborg University.

[21] Chen, T. M. (2007). Network Traffic Management. *The Handbook of Computer Networks*, Hossein Bidgoli (ed.), Wiley, Dallas, Texas.

[22] Dresner, K. and P. Stone, P. (2005). Multi-agent traffic management: An improved intersection control mechanism. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and MultiAgent Systems, Utrecht, The Netherland.

[23] Yuksel, M., Sikdar, B., Vastola, K. S. and Szymanski, B. (2000). Workload Generation for ns Simulations of Wide Area Networks and the Internet. Proceedings of Communication Networks and Distributed SystemsModeling and Simulation Conference, pp 93-98, San Diego,CA, USA, 2000.

[24] Roughan, M. and Kalmanek, C. R. (2003). Pragmatic modeling of broadband access traffic". Computer Communications 26 (8) 804–816.

[25] Suryawanshi, P. (2004). Internet Traffic Management.

[26] Ganesh, A. J., Key, P. B., Polis, D., & Srikant, R. (2006). Congestion notification and probing mechanisms for endpoint admission control. *IEEE/ACM Transactions on Networking (TON)*, *14*(3), 568-578.

[27] Le, J. (2018). A Tour of the Top Ten Algorithms for Machine Learning Newbies. Towards Data Science. https://towardsdatascience.com/a-tour-of-the-top-10-algorithms-for-machine-learning-newbies-dde4edffae11. Retrieved: 13 - 12 – 2018. 00:09.

[28] Wakefield, K. (2018). A Guide to Machine Learning and their application. SAS, UK. Available at: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html. Retrieved: December 13, 2018:00.10 am.

[29] Qian, K. (2018). Comparing Supervised Learning Algorithms. Data school. Available at: https://www.dataschool.io/comparing-supervised-learning-algorithms. Retrieved: 13 - 12 – 2018. 00.00.

[30] Zaidi, F. (2018). Machine Learning Algorithms: How they work and Use cases for each. Towards Data Science. Available at: https://towardsdatascience.com/machine-learnings-algorithms-how-they-work-and-use-cases-for-each-type-part-i-of-iii-67997c6dda84. Retrieved: December 13, 2018:00.07 am.

[31] Nweke, H. F., Teh, Y. W., Algaradi, M. A and Alo, U. R. (2018). Deep Learning algorithms for human activity recognition using mobile and wearable sensor networks: state of the Art and Research Challenges. Expert Systems with Applications, Volume, 105. Pp 233-261.

[32] Pelk, H. (2017). Machine Learning, Neural Networks and Algorithms. Chatbots Magazines. Available at: https://chatbotsmagazine.com/machine-learning-neural-networks-and-algorithms-5c0711eb8f9a. Retrieved: 23rd August, 2018, @ 2.56pm.

[33] Miller, S. (2015). Mind: How to Build a Neural Network (Part One). Avalaible at: https://www.github.com/stevenmiller888/mind.

[34] Portilla, J. (2016). Beginners Guide to Neural Network with R. Data science and AI Consulting. Available at: https://www.indatalabs.com.

[35] Jiang, Y., Yang, C., Na, J., Li, G., Li, Y. and Zhong, J. (2017). A Brief Review of Neural Networks Based Learning and Control and Their Applications for Robots. In the Journal of Complexity. Hindawi and Wiley & Sons Inc.

[36] Fister, I. et al. (2016). "Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution," Nonlinear Dynamics, vol. 84, no. 2, pp. 895–914, 2016.

[37] Zhang, Y., Li, S. and Guo, H. (2017). A type of biased consensus-based distributed neural network for path planning, Nonlinear Dynamics, vol. 89, no. 3, pp. 1803–1815.

[38] Shi, X., Wang, Z. and Han, L. (2017). "Finite-time stochastic synchronization of time-delay neural networks with noise disturbance," Nonlinear Dynamics, vol. 88, no. 4, pp. 2747–2755, 2017.

[39] Ho, R. (2018). An Introduction to 6 Machine Learning Models. Dzone, available at: https://dzone.com/articles/introduction-6-machine. Retrieved: 13 - 12 – 2018. 00:08.

[40] Weinberg, N. and Johnson, J. T. (2018). MPLS Explained. Hetwork World from IDG. Available at: https://www.networkworld.com/article/2297171/sd-wan/network-security-mpls-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

explained.html. Retrieved: August, 28, 2018: 12.31pm.

[41] Parmenter, T. (2018). MPLS vs. Ethernet: Which WAN Connectivity option isbest? TechTarget. https://searchnetworking.techtarget.com/feature/MPLS-vs-Ethernet-Which-WAN-connectivity-option-is-best. Retrieved: 14 - 12 – 2018. 01.14.

[42] Oravec, M., Petráš, M. and Pilka, F. (2008). Video Traffic Prediction Using Neural Networks. ActaPolytechnicaHungarica. Vol. 5, No. 4. pp. 59 – 78.

[43] Rojas, J. S., Gallón, Á. R., & Corrales, J. C. (2018). Personalized Service Degradation Policies on OTT Applications Based on the Consumption Behavior of Users. In *International Conference on Computational*