ARTIFICIAL NEURAL NETWORKS IN FORECASTING THE QUARTERLY MALARIA CASES IN NIGERIA

BY

***Bukar, A.B. and ** Dikko, H.G.**
***Dept. of Mathematics, Nigerian Defence Academy, Kaduna.**
****Dept. of Statistics, Ahmadu Bello University, Zaria**

ABSTRACT

The objective of the study is to train artificial neural networks (AMORE SOFTWARE) with a suitable data obtained from the WHO record collected from the National Surveillance System (NSS). The architecture of the neural networks in use has two inputs nodes Y_{t-1} and Y_{t-4} with one hidden layer and one output layer Y_t. They were used to forecast quarterly malaria cases in Nigeria. The data used were from 1990 to 2003. A total of 32 data series was used in the training and ends up with $R^2 = 0.9946$, and 24 used in the validation of the neural networks with $R^2 = 0.5759$. The forecasting accuracy was found to be $R^2 = 0.9665$, which validate the networks.

Keywords: Artificial Neural Network, Training Data, Nodes, World Health Organization, Malaria,

1.0 INTRODUCTION

1.1 Brief on Malaria

Malaria is a serious infectious disease spread by certain mosquitoes call anopheles seeking for blood to produce their eggs. It is most common in tropical climates. It is characterized by recurrent symptoms of chills, fever, and an enlarged spleen.

In Nigeria, malaria is caused by anopheles bite through four different species of a parasite belonging to genus *Plasmodium*: *Plasmodium falciparum* (the most deadly), *Plasmodium vivax*, *Plasmodium malaria*, and *Plasmodium ovale*. The last two are fairly uncommon. Falciparum malaria is far more severe than other types of malaria because the parasite attacks all red blood cells, not just the young or old cells, as do other types. It causes the red blood cells to become very "sticky." A patient with this type of malaria can die within hours of the first symptoms. The fever is prolonged. So many red blood cells are destroyed that they block the blood vessels in vital organs (especially the brain and kidneys), and the spleen becomes enlarged. There may be brain damage, leading to coma and convulsions. The kidneys and liver may fail. It was soon observed that the use of insecticide- treated bed nets (henceforth termed ITNs) provided adequate protection against malaria infections, particularly in children. The World Health Organization has adopted the use of ITNs as one of the main strategies for malaria control in their Roll Back Malaria programme (RBM). At present ITNs are being applied in many malaria-endemic regions worldwide and their use has replaced the use of indoor house spraying with insecticides in many countries. The World Health Assembly advocated the large-scale use of insecticides for malaria control in 1955, and programmes were carried out to spray as many houses as possible with a residual deposit of insecticide [mostly organ chlorine compounds such as dichlorodiphenyl-trichloroethane (DDT) and dieldrin].

1.2 Neural Networks

An Artificial Neural Networks (ANNs) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain process information. The key elements of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Artificial neural networks, like people, learn by example. Artificial neural networks share is configured for a specific application, such as pattern recognition or data classification through a learning process. Learning in biological systems involves adjustments to the synaptic connection that exist between the neurons. This is true of artificial neural networks as well.

1.3 Characteristics of Neural Networks

An ANN is a model composed of several highly interconnected computational units called neurons or nodes. Each node performs a simple operation on an input to generate an output that is forwarded to the next nodes in the sequence. This parallel processing allows for great advantages in data analysis. There are three essential features of a neural network (Stern, 1991):

(i) the network topology,

(ii) the computational functions of its elements, and

(iii) the training of a network.

1.3.1 Network Topology

The network topology refers to the number and organization of the computing units the types of connections between neurons, and the direction of information flow in the network. The node is the basic organizational unit of a neural network, and nodes are arranged in a series of layers to create the artificial neural network. According to their location and function within the network, nodes are classified as input, output, or hidden layer nodes (Stern, 1991). Input layer nodes receive information from sources external to the neural network, and output layer nodes transmit information out of the neural network (Duliba, 1991). Hidden layer neurons act as the computational nodes in the neural network, communicating between input nodes and other hidden layer or output nodes.

1.3.2 Neural Network Computational Structure

The connections between neurons are classified as either excitatory or inhibitory, according to whether the neuron weights are negative or positive, respectively. The direction of information movement also distinguishes ANNs as either feed forward or feedback networks (Duliba, 1991). Feed forward networks permit data transfer in only one direction through the network, while feedback networks allow the data to cycle through the network (Stern, 1991).

1.3.3 Training the Network

Once the network layout and computational characteristics of the network are established, the network's adaptive learning strategy must be determined. Neural networks must "learn" to generate values consistent with the patterns in a given data set to make accurate projections. The learning process is when network weights change in response to a training data set. ANNs learn in two ways: supervised and unsupervised learning, which differ according to whether or not known answers are used to train the network.

Several studies have been carried out based on ANN application on time series data for the purpose of forecasting future events. For example, Moreno *et al.* (2011) used Multilayer Perception (MLP), Radial Base Function (RBF), Generalized Regression Neural Network (GRNN), and Recurrent Neural Network (RNN) models to analyze time series data made up of 244 time points. Niaki and Hoseinzade (2013) used ANN to forecast the daily direction of Standard and Poor's 500 (S&P 500) index in order to select the most influential features (factors) of the proposed ANN that affect the daily direction of S&P 500. More recently, Chaudhuri and Ghosh (2016) used Artificial Neural Network approach for time series modeling to forecast the exchange rate value of the Indian rupee vis a vis the US Dollars in a multivariate framework.

ANN based literature to analyze malaria data is increasing recently. Andrade *et al.* (2010) studied malaria diagnosis in the Brazilian Amazon using ANNs. Pandit and Anand (2016) studied Artificial Neural Networks for diagnosis of malaria in Red Blood Cells.

1.4 Source of Data

The data used reflect aggregation of malaria cases at the national level and are presented by gender, age and sub national level as submitted to W.H.O by National Surveillance Systems in 2005. It's arranged by month and year in the WHO reported cases of malaria in Nigeria. Malaria reporting from national surveillance systems varies in quality and reporting completeness and may have limited value in understanding the actual malaria burden, but may be useful for understanding trends in the relative burden of malaria in the public health sector.

2.0 MATERIAL AND METHODS

2.1 The neural networks models

A two input nodes, one hidden nodes, and one output node ($2 \times 1 \times 1$) artificial neural network is used to forecast the quarterly malaria cases.

Input Nodes: Because Malaria cases time series is affected by seasonality, two input nodes were chosen, one for Y_{t-1} while the other one for Y_{t-4}, and one bias node.

Hidden Nodes: There is only one hidden node since seasonality is the strong pattern in malaria cases time series and one bias node.

Output node: There is also one output node. Here Y_t is being forecasted as a function of Y_{t-1} and Y_{t-4}. A second output node for Y_{t+1} could easily be added to the artificial neural networks. In that case, the goal is to minimize one and two period ahead errors.

With respect to neural networks, the performance criterion is the minimization of squared error. Therefore, the total system error is expressed as follows:

$$E = \sum_{p,i}(t_{ip} - y_{ip})^2 \qquad \textbf{(2.1)}$$

where i indexes units of output; p indexes the input-output pairs to be learned; t_{ip}refers to the desired output, and $y_{ip}$is the network's calculated output. This function is minimized, and if the output functions are differentiable, the task of blame assignment is simplified. We begin with a set of arbitrarily chosen weights W and improve them by implementing the formula

$h(X,W)$ =output function of the network

$$W_t = W_{t-1} + \eta\Delta_j\left(X_t, W_{t-1}\right)\left(Y_t - h(X_t, W_{t-1})\right), t = 1,2,3,.... \qquad (2.2)$$

where

η= learning rate

Δ_j= vector containing partial derivatives of h with respect to Y_t

X_t= vector of inputs

W = weights

t = time index (Garson, 1998)

- **Dividing the data into training and validation set**

It is important to train and validate the ANNs using two different data sets. The first data set is used to train the network and the second data set is used to validate the model. In this work, we use 32 data series of the data to train the ANN and 24 data series for the validation set.

- **Scale all input variables**

Scaling input variables is a necessary step because of the type of nonlinear transformation and logic built into an artificial neural network at which the formula is given by

$$Scaledvalue = \frac{Actual - Minimum}{Maximum - Minimum} \qquad (2.3)$$

where

Maximum=Expected maximum value in the application

Minimum=Expected minimum value in the application

- **Initial weights and start a training epoch**

The initial values of the training weight can influence the required number of epochs and the final result. Thus, when good starting values are unknown, the randomization of the initial training weights is often recommended, typically randomly distributed between -1 and 1.

- **Input scaled variables**

In this step each neurode in the input layer has its value and distributes it to every node in the hidden layer. Thus, each neurode of the hidden layer receives all input values except that the signals are weighted to yield the input to the hidden neurodes. In this work,there is only one hidden layer neurodes, however, as mentioned; each signal to a neurode will have a different weight associated with it. To understand it better lets start the forecasting of the first quarter

$$I_0 = O_0$$

$$I_1 = O_1$$

$$I_2 = O_2$$

- **Weight and the sum output at the receiving nodes**

The weights were determined by many iterations of the training process. In a very real sense, the weights in an ANN contain its memory and intelligence. For this work 500 training epochs were performed to yield the weight.At node 4, the inputs are summed using:

$$I_4 = W_{24}O_2 + W_{14}O_1 + W_{04}O_0 \quad \textbf{(2.4)}$$

In general, at each neurode, a weighted average of all inputs is calculated.

- **Transform hidden input to outputs**

The relationship between the input and the output of the node is expressed by a transfer function. Because the sigmoid transfer function (also called logistic function) is used, the input value result in a neurode output with a value of 0 through 1.However, the logistic transfer function transforms the input to an output level in the range of 0 to 1. Consider the output of node 4, given input value I_4

$$O_4 = \frac{1}{1+e^{-I_4}} \quad \textbf{(2.5)}$$

During the transformation of hidden node inputs to their output, some hidden neurodes will be activated (i.e. non zero) and others will not (i.e., will equal zero). More importantly, as the network learns, the weights leading to the hidden layer neurodes are adjusted through back propagation so as to minimize the ANN error.

- **Weight and sum hidden node outputs at the output nodes**

This next step is to sum inputs at the final nodes, the output node. We see that two nodes send signals to the output node, nodes 4 and 3

$$I_5 = W_{45}O_4 + W_{35}O_3. \quad \textbf{(2.6)}$$

Node 4 is a hidden node and node 3 is a bias node. The bias node 3 is used to change I_5 and therefore, position the sigmoid function at a different location on the x-axis so that output O_5 has different values. It is a constant term where the added constant is determined by the weight from the bias node to the next node. Thus W_{35} is trained like the other weights.

- **Transform input at the ANN output nodes**

The input value I_5 is transformed to the output value, O_5

$$O_5 = \frac{1}{1+e^{-I_5}} \quad \textbf{(2.7)}$$

- **Calculate output error**

The output error is simply the difference between O_5 and the desired value.

$$E_5 = O_5 - O_4 \quad \textbf{(2.8)}$$

Back propagate errors to adjust weights

During this step, the weights are adjusted through a training process by the back propagation so as to yield the correct output variable.

- **Continue the epoch process all input in the training data set once.**
- **Calculate the epoch RMS**

Completing one epoch where all observations in the training data set are input to the ANN.

$$RMS = \sqrt{\sum \frac{e_i^{\,2}}{nt}} \quad \textbf{(2.9)}$$

- **Judge out-of –sample validity**

This out of sample test is an essential step in an actual application, it consist of running the remaining data (half data) to validate the ANN.

- **Use the model in forecasting**

To predict future output, simply input the scaled values of the input variables and calculate the output O_5 using the previously fitted weights. This is about as easy as using other forecasting methods.

- **The coefficient of determination R^2**

$$R^2 = 1 - \frac{target\ RMS^2}{s_d^2} \quad \textbf{(2.1.0)}$$

3.0 RESULTS

Table 1 shows the scaled inputs and output variable of the training set of the neural networks.

Table 1: Scaled input and output

		Input 1	Input 2	Output
Period	Quarter	Y_{t-1}	Y_{t-4}	Y_t
1	1			0.002
2	2	0.002		0.1312
3	3	0.1312		0.4479
4	4	0.4479		0.2165
5	1	0.2165	0.002	0.1114
6	2	0.1114	0.1312	0.1383
7	3	0.1383	0.4479	0.3065
8	4	0.3065	0.2165	0.0108
9	1	0.0108	0.1114	0.1369
10	2	0.1369	0.1383	0.1473
11	3	0.1473	0.3065	0.459
12	4	0.459	0.0108	0.1682
13	1	0.1682	0.1369	0.0605
14	2	0.0605	0.1473	0.1237
15	3	0.1237	0.459	0.3128
16	4	0.3128	0.1682	0.1503
17	1	0.1503	0.0605	0.1046
18	2	0.1046	0.1237	0.1133
19	3	0.1133	0.3128	0.5019
20	4	0.5019	0.1503	0.1423
21	1	0.1423	0.1046	0.0973
22	2	0.0973	0.1133	0.1145
23	3	0.1145	0.5019	0.4441
24	4	0.4441	0.1423	0.1605

25	1	0.1605	0.0973	0.1062	
26	2	0.1062	0.1145	0.1578	
27	3	0.1578	0.4441	0.443	
28	4	0.443	0.1605	0.1267	
29	1	0.1267	0.1062	0.1063	
30	2	0.1063	0.1578	0.1276	
31	3	0.1276	0.443	0.4564	
32	4	0.4564	0.1267	0.1423	
Series	Observation	Mean	std Error	minimum	Maximum
Y_t	32	0.1958	0.1422	0.002	0.5019

3.1 Result ofthe Training Output

Table 2: Output and desired output

Period	O_t	D_t	e_t
5	0.1113	0.1114	5E-05
6	0.1627	0.1383	-0.024
7	0.2131	0.229	0.0159
8	0.0936	0.0856	-0.008
9	0.1504	0.1369	-0.013
10	0.1676	0.1663	-0.001
11	0.4125	0.421	0.0085
12	0.1829	0.1682	-0.015
13	0.0617	0.0605	-0.001
14	0.1305	0.1237	-0.007
15	0.2193	0.235	0.0157
16	0.1797	0.1503	-0.029
17	0.1165	0.1046	-0.012
18	0.1445	0.1428	-0.002
19	0.2201	0.2315	0.0114
20	0.1545	0.1423	-0.012
21	0.0766	0.08	0.0034
22	0.1488	0.1466	-0.002
23	0.2198	0.2314	0.0116
24	0.1717	0.1605	-0.011
25	0.1383	0.1353	-0.003
26	0.159	0.1578	-0.001
27	0.2198	0.2193	-6E-04
28	0.156	0.1475	-0.008
29	0.186	0.1769	-0.009

30	0.172	0.1685	-0.003
31	0.2195	0.2215	0.002
32	0.1615	0.1541	-0.007
RMS			0.0104

The table 2 is showing the output giving by the neural networks and the desired output after the training set

$$R^2 = 1 - \frac{target\ RMS^2}{s_d^2}$$

$$R^2 = 1 - \frac{0.0104^2}{0.1422^2}$$

$$R^2 = 0.9946$$

3.2 Validation of the Neural Network

Table 3: Validation of the artificial neural networks I

		Input 1	Input 2	Output
Period	Quarter	Y_{t-1}	Y_{t-4}	Y_t
33	1	0.674829	0.14233	0.21432
34	2	0.217044	0.224492	0.54123
35	3	0.369425	0.379393	0.6523
36	4	0.58415	0.637486	0.20166
37	1	0.570744	0.674829	0.3725
38	2	0.335729	0.217044	0.6123
39	3	0.40288	0.369425	0.7859
40	4	0.668758	0.58415	0.20822
41	1	0.902543	0.570744	0.51264
42	2	0.348072	0.335729	0.54123
43	3	0.446373	0.40288	0.8967
44	4	0.780034	0.668758	0.5123
45	1	0.487277	0.902543	0.4236
46	2	0.286083	0.348072	0.78954
47	3	0.411707	0.446373	0.3125
48	4	0.780061	0.780034	0.21612
49	1	0.8745	0.7854	0.51268
50	2	0.287185	0.286083	0.89456
51	3	0.98546	0.98755	0.456
52	4	0.9875	0.7801	0.21267
53	1	0.98754	0.78006	0.54621
54	2	0.9856	0.987	0.6548

55	3	0.781285	0.62541	0.8569
56	4	0.986318	0.86952	0.714562
Series	Observation	Mean	std Error	
Y_t	24	0.5203	0.2306	

Table 4: Validation of the artificial neural networks II

period	O_t	D_t	e_t
33	0.21432	0.379393	0.0963
34	0.54123	0.637486	0.0225
35	0.6523	0.674829	0.0154
36	0.20166	0.217044	-0.003
37	0.3725	0.369425	-0.028
38	0.6123	0.58415	-0.215
39	0.7859	0.570744	0.1275
40	0.20822	0.335729	-0.11
41	0.51264	0.40288	0.1275
42	0.54123	0.668758	0.0058
43	0.8967	0.902543	-0.164
44	0.5123	0.348072	0.0228
45	0.4236	0.446373	-0.01
46	0.78954	0.780034	0.1748
47	0.3125	0.487277	0.07
48	0.21612	0.286083	-0.101
49	0.51268	0.411707	-0.115
50	0.89456	0.780061	0.5193
51	0.456	0.975297	0.0745
52	0.21267	0.287185	-0.144
53	0.54621	0.401808	0.1265
54	0.6548	0.781285	0.126485
55	0.8569	0.986318	0.129418

56	0.714562	0.6235	-0.09106
RMS			0.15016

$$R^2 = 1 - \frac{0.15016^2}{0.2306^2}$$

$$R^2 = 0.5759$$

In Table 3 and Table 4,the first four observations are missing because y_{t-4} is used. The Tables are showing the validation of the artificial neural networks, the second part of the data was used in the forecast in other to validate the neural networks.

The autoregressive model of order given by the neural networks is given below;

$$Y_t = 0.232 + 0.23Y_{t-4} - 0.073Y_{t-1} + e_t$$

$$Y_t = 0.047 - 0.026Y_{t-4} + 0.792Y_{t-1} + e_t$$

$$Y_{t+1} = 0.047 - 0.026Y_{t-3} + 0.792Y_t + e_{t+1}$$

$$Y_{t+2} = 0.047 - 0.026Y_{t-2} + 0.792Y_{t+1} + e_{t+2}$$

$$Y_{t+3} = 0.047 - 0.026Y_{t-1} + 0.792Y_{t+2} + e_{t+3}$$

$$Y_{t+4} = 0.047 - 0.026Y_t + 0.792Y_{t+3} + e_{t+4}$$

$$Y_{t+5} = 0.047 - 0.026Y_{t+1} + 0.792Y_{t+4} + e_{t+5}$$

$$Y_{t+6} = 0.047 - 0.026Y_{t+2} + 0.792Y_{t+5} + e_{t+6}$$

$$R^2 = 1 - \frac{0.01059}{0.05789}$$

$$R^2 = 0.9665$$

Figure 1is showing the number of malaria cases from 1990 to 2004 as submitted to W.H.O by National Surveillance Systems in 2005. It is being observed that the seasonal aspect of the cases is observed to be moderate from 1990 to 1998, and which starts increasing considerably from 1999 to 2004 where it reached its pick.

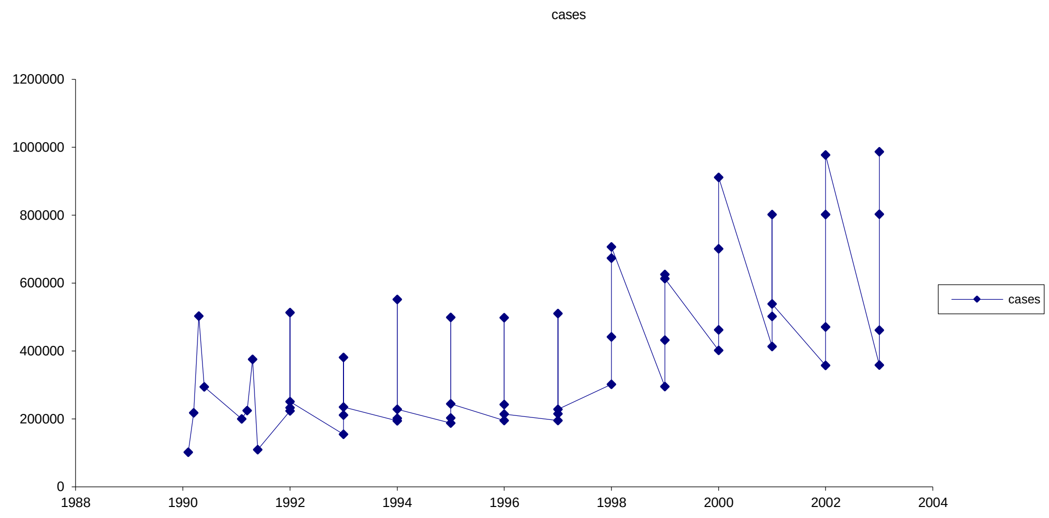

Figure 1: Number of Malaria Cases from 1990 to 2004 as submitted to W.H.O by National Surveillance Systems in 2005.

The figure 2 is a two line graphs showing the difference between the output value and the desired output value of the first training set of the neural networks.

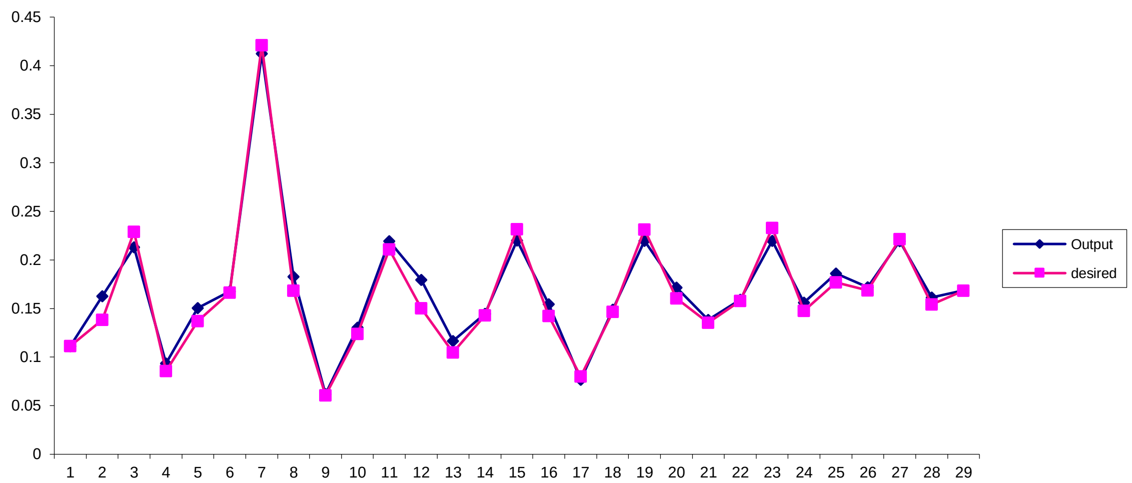

Figure 2: The output value and the desired output

Figure 3 is showing the final result of the training, it is observed that, after the 34 th there is much difference between the output and the desired output of the trained neural networks.

This is due to the fact that the seasonality of the data around that quarter of those very years has increase considerably.

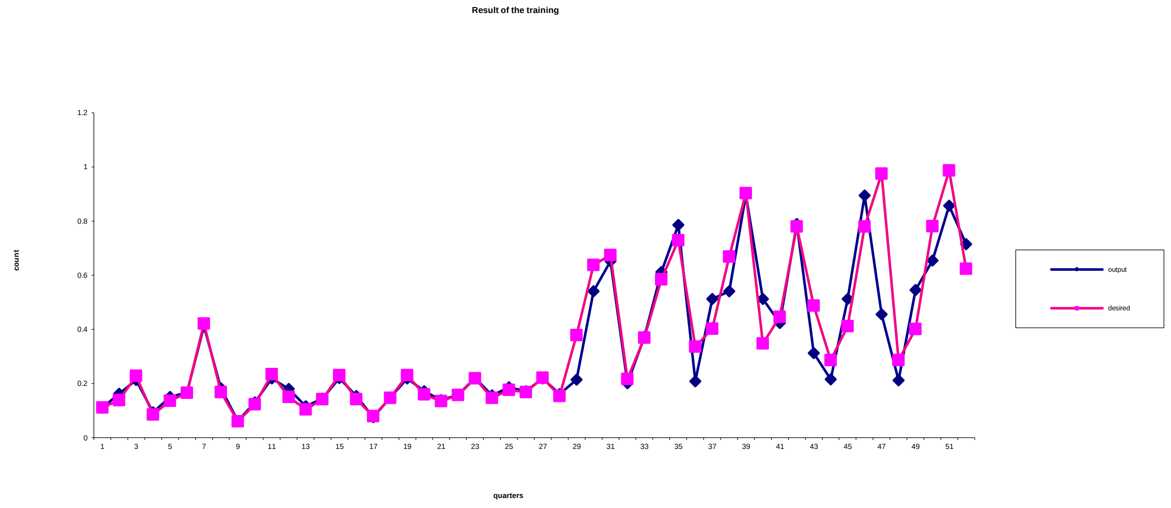

Figure 3: Final Result of the Training

4.0 CONCLUSION

This paper has shown how to train an artificial neural network composition and capabilities in the field of organizational research in addition to or as an alternative to multiple linear regressions. Because of their dynamic composition, neural networks are efficient at analyzing problems where the data are incomplete or fuzzy, and accurate predictions are sought more heavily than explanations. Neural network research has grown remarkably in the last fifteen years. Though most of this growth has been in the areas of Mathematical and Computer Sciences.

Hence the computing world has a lot to gain from neural networks; their ability to learn by example makes them very flexible and powerful. Furthermore there is no need to derive an algorithm in order to perform a specific task; there is no need to understand the internal mechanisms of that task. They are also very well suited for real time systems because of their fast response and computational times, which are due to their parallel architecture. Neural networks also contribute a lot in area like forecasting where it is found to be more suited using quarterly data.

A coefficient of determination ($R^2 = 0.9946$) shows that the training of the data with the first set was successful, whereas this ($R^2 = 0.5759$) shows that the second set of data is low in relation to the first set because of the high cases of malaria after the years following the data set use in the first training set.With $R^2 = 0.9665$ of the forecasting using the entire data means that we can validate the training of the neural networks and use it in a forecasting of the future.

It has been noted that malaria cases are highly concentrated during the raining season and the season after it. The cases of malaria is on the increase every year due to resistance of anopheles to malaria treatment drugs.

REFERENCE

Andrade, B. B., Reis-Filho, A., Barros, A. M., Souza-Neto, S. M., Nogueira, L. L., Fukutani, K. F. and Barral-Netto, M. (2010): Towards a precise test for malaria diagnosis in the Brazilian Amazon: comparison among field microscopy, a rapid diagnostic test, nested PCR, and a computational expert system based on artificial neural networks. *Malaria journal*, *9*(1): 117.

Chaudhuri, T. D., and Ghosh, I. (2016): Artificial Neural Network and Time Series Modeling Based Approach to Forecasting the Exchange Rate in a Multivariate Framework. *Journal of Insurance and Financial Management, 1(5): 92-123.*

Duliba, K. A. (1991):Contrasting Neural Networks Nets with Regression in Predicting Performance in the Transportation Industry.*IEEE*,(2)163-170.

Garson, G. D. (1998):*Neural Networks: An Introductory Guide for Social Scientists.* Thousand Oaks, CA: Sage.

Moreno, J. J. M., Pol, A. P. and Gracia, P. M. (2011): Artificial Neural Networks applied to forecasting time series. *Psicothema*, *23*(2): 322-329.

Niaki, S. T. A., and Hoseinzade, S. (2013): Forecasting S&P 500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International*, *9*(1): 1.

Pandit, P. and Anand, A. (2016): Artificial Neural Networks for Detection of Malaria in RBCs. *arXiv preprint arXiv:1608.06627*.

Stern, H. S. (1991): Neural Networks in applied statistics. *American Statistical Association;*(3):150-154.