

## A COMPARATIVE ANALYSIS OF VECTOR AUTOREGRESSIVE MODEL AND NEURAL NETWORKS

Ugwuowo, Ifeanyi Fidelis<sup>1</sup>, Asogwa Oluchukwu<sup>2</sup> Eze, Chinonso Michael<sup>3\*</sup>

<sup>1</sup>Department of Statistics, University of Nigeria Nsukka Tel: +2347035089574 E-mail: [fidelis.ugwuowo@unn.edu.ng](mailto:fidelis.ugwuowo@unn.edu.ng)

<sup>2</sup>Department of Statistics, Federal University, Ndufu Alike-Ikwo. Tel: +2348036785840 E-mail: [qackasoo@yahoo.com](mailto:qackasoo@yahoo.com)

<sup>3</sup>Department of Statistics, University of Nigeria Nsukka

**\*Corresponding Author:-**

E-mail: [chinonso.eze@unn.edu.ng](mailto:chinonso.eze@unn.edu.ng) Tel: +2348069464222

---

### Abstract:-

*In this work, vector autoregression and neural network approach to multivariate time series analysis is presented. A vector autoregressive model and multilayer perceptron network with back-propagation, gradient descent algorithm have been designed to model the monthly average exchange rates of three international currencies with respect to naira. The series span over the period of January, 2012 to August, 2017. The original series were preprocessed to smoothen the distribution and facilitate fast convergence in the network algorithm. In training the network to learn the combined series of the exchange rates, a remarkable achievement was made. Adding to the beauty of the network model is the fact that the number of units of the input layer was predetermined through the VAR model. Using some model performance measures (RMSE, MBE and  $R^2$ ), it was recorded that the neural network approach performs better than the VAR model as it yielded minimum error of prediction.*

**Keywords:-**Back-propagation algorithm, Multi-layer perceptron, neural network, Vector autoregressive

## 1. INTRODUCTION

The need for appropriate decision making against the future occurrences of a system has always been the major reason behind time series modeling. As decision makers look for appropriate decision tool (model), they invariably try to come up with a predictor which has a minimum associated cost function. In an effort to get the best model, the analyst always tries to explore every relevant information space so as to get a forecast about the future event which will have relatively minimum error; Lutkepohl (2006). In currency arbitrage and macroeconomic policy formulation, the knowledge of the future exchange rate of foreign currencies, using an appropriate statistical model is of great essence to effective planning. The knowledge of the law underlying these exchange rates can be grasped by establishing the mathematical relationship that exist between the past, present and future exchange rates of the selected currencies. Daniya (2015), in studying the depreciation speed of Kazakhstan local currency (tenge) against major world currencies: USD, EURO and Singapore dollar (SGD) fitted an appropriate ARIMA model credited to Box-Jenkins (1976) to each of the series. Although his study gave individual model that explains the generating process of each of the series, the structural relationship inherent in the series was not established.

In the study of macroeconomic observations, the level of interdependence among contemporaneous variables is always so significant that, it becomes very advantageous to carry the related variables all along so as to extract the structural relationship inherent in them. The multivariate time series analysis (also known as VARMA model) has the ability of extracting the dynamic relationship obtainable in multiple equal-spaced-time interrelated variables (Sims, 1980). Vector autoregressive moving average (VARMA) models are special statistical tools that study the behavior of multiple time dependent variables and forecast future values depending on the history of variations in the data, some other related variables and shocks.

A multiple time series  $z_t = (z_{1t}, z_{2t}, \dots, z_{kt})'$ ;  $t = 1, 2, 3, \dots, T$  is a vector containing sequences of observations measured over equal-spaced time, in discrete or continuous time units. A better understanding is often obtained by studying these many related variables together than by just studying one variable as that will help to ascertain the cause and effect relationship among the variables of interest (Granger, 1969).

An important concept in the family of VARMA models is a vector autoregressive (VAR) model which captures the correlation among a set of variables and analyzes a certain aspect of the relationship between the variables of interest. In forecasting new values for each variable, better predictions are made available if variations in the other variables are also taken into account.

Chaotic noise, nonstationarity and non-linearity are the main features of economic and financial time series; they should be taken into account before fitting an appropriate model to any series; Kondratenko and Kuperin (2003). Most statistical models for learning and predicting time series are based only on assumption of linear recurrence; Kanad et al. (1990). Because of this assumption, many shortfalls exist in analysis of such series with chaotic and noisy elements using linear models. Thus, it becomes necessary to use the technique which has the ability of cushioning the effects of non-linearity in the analysis of real-life temporal data. We resorted to the use of neural network model in this research work since it has the ability of modeling appropriately both linear and non-linear series.

Neural network is an exploratory data analysis technique which has wide applications in different areas of life. The complex nature of some series suggests the use of neural networks which have been confirmed to be capable of modeling complex non-linear relationship among variables without prior assumption of the nature of the relationship; Sana et al. (2003). Artificial neural networks are built out of a highly interconnected set of units, where each unit takes a number of real valued inputs (possibly the output of previous units) and produces a single real valued output which may become input to subsequent units. Eric (2002) tendered an empirical report that confirmed the applicability of simple neural network model to the prediction of foreign exchange rates. In his work, simple average weekly indicators (including political factors) were fed to a three-layer perceptron to capture the pattern of the exchange rates of five currencies against US dollar. The results of his study showed that neural network model is robust for forecasting of exchange rates. Despite his findings, he recommended a further investigation into the network methodology in order to improve its performance. It has always been a great challenge in neural network methodology to determine before hand, the optimum number of units in the input and hidden layers that will yield minimized MSEs in the out-of-sample forecasts. Most of the works done in the literature of neural networks are based on heuristics (trial and error method) in determining the number of units in the network layers. As pointed out by Kanad et al. (1990) the major shortfalls of neural network modeling are determination of the optimum number of units (layers) in the network and establishment of a viable theoretical framework or statistical characteristics that underlie the phenomenon. However, Part of the interests in this research work is to pre-determine the number of units (lagged variables) of the input layer needed by the network to appropriately learn the series. This is to be done by fitting a preliminary VAR model to the multiple variables (exchange rates) and using the number of regressors in the fitted model as the number of units in the input layer of the network. An investigation will further be made to know whether the neural network model has better prediction ability than the VAR model.

**2. VECTOR AUTOREGRESSIVE MODEL:** Let  $Z_t$  be the value of the variable of interest at time  $t$ . For any lead time  $l$  ( $l > 0$ ), the value of the variable at time  $t + l$  made at the end of time  $t$  is a function of present and past observations of  $Z_t$  and it is given as

$$Z_{t+l} = f(z_t, z_{t-1}, z_{t-2}, z_{t-3}, \dots).$$

If the relationship is linear (which is always the assumption in classical problems), the forecast function given as

$$Z_{t+l} = c + c_0 z_t + c_1 z_{t-1} + c_2 z_{t-2} + \dots \quad (1)$$

However, in multiple time series analyses, the value of one variable often times, is not only related to itself lagged in time but, also to the present and(or) past observations of other variables. Unlike the univariate autoregressive models that

explain a variable solely by its own history and deterministic component, VAR models in addition, involve “outside regressors”. Introducing the information about the outside regressors into the forecast model of a particular variable will yield a more viable forecast. Let the  $k$  –variables under study be denoted as  $z_{1t}, z_{2t}, \dots, z_{kt}$ . The forecast function for say, the first variable, for lead time  $l$ , made at the end of time  $t$  is given as  $Z_{1t+l} = f(z_{1t}, z_{2t}, \dots, z_{kt}, z_{1t-1}, z_{2t-1}, \dots, z_{kt-1}, z_{1t-2}, \dots)$ . To generate the model, we consider the following  $k$ -equations having  $k$ -variables  $z_1, z_2, \dots, z_k$ , each of which regressed on itself lagged and lagged values of the other variables.

$$z_{1t} = c_{10} + a_{11,1}z_{1(t-1)} + a_{12,1}z_{2(t-1)} + \dots + a_{1k,1}z_{k(t-1)} + \dots + a_{11,p}z_{1(t-p)} + a_{12,p}z_{2(t-p)} + \dots + a_{1k,p}z_{k(t-p)} + e_{1t} \quad (2)$$

$$z_{2t} = c_{20} + a_{21,1}z_{1(t-1)} + a_{22,1}z_{2(t-1)} + \dots + a_{2k,1}z_{k(t-1)} + \dots + a_{21,p}z_{1(t-p)} + a_{22,p}z_{2(t-p)} + \dots + a_{2k,p}z_{k(t-p)} + e_{2t} \quad (3)$$

$$z_{kt} = c_{k0} + a_{k1,1}z_{1(t-1)} + a_{k2,1}z_{2(t-1)} + \dots + a_{kk,1}z_{k(t-1)} + \dots + a_{k1,p}z_{1(t-p)} + a_{k2,p}z_{2(t-p)} + \dots + a_{kk,p}z_{k(t-p)} + e_{kt} \quad (4)$$

In general matrix notation with  $k$  – variables and  $p$  – lags,

$$Z_t = C + B_1 Z_{t-1} + B_2 Z_{t-2} + \dots + B_p Z_{t-p} + u_t \quad (5)$$

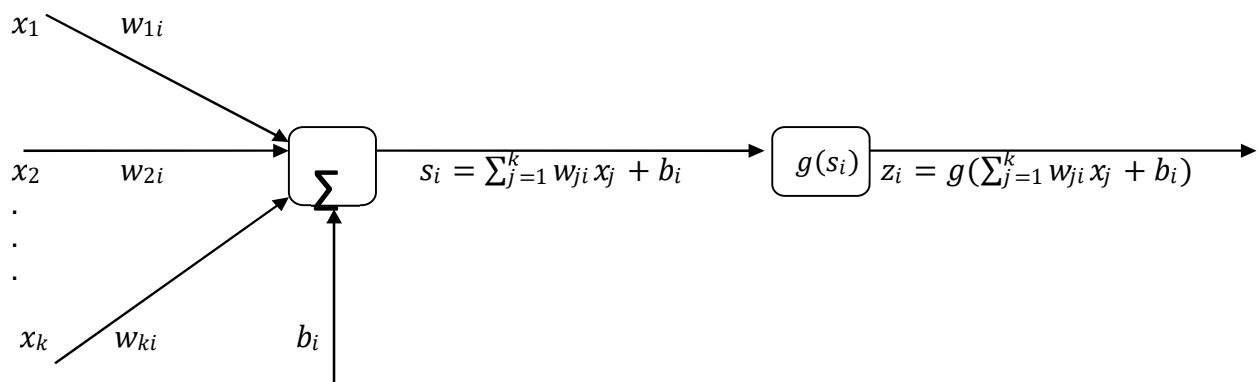
Where  $Z_t, C$  and  $u_t$  are  $K \times 1$  column vectors and  $B_1, B_2, \dots, B_p$  are each,  $K \times K$  matrices of coefficients. The vector  $u_t$  is a  $K$  – element vector of error components that satisfy  $E(u_t u_t') =$

$D$  and  $E(u_t u_s) = 0$  for  $t \neq s$  (no serial correlation), where  $D$  is a time invariant positive definite covariance matrix with the diagonal elements defined as  $E(u_t^2) = \sigma^2$

**3. NEURAL NETWORK MODEL:** This is a modeling mechanism that learns a given series (mostly chaotic) with an intention of understanding the general behavior of the series. Neural network system is made up of processing units that have the tendencies for storing experiences gotten while learning a series and making it available for the prediction of future values. The approach belongs to the family of data-driven statistical tools as against model based tools since there is no distributional assumption behind the use of the approach. It has the ability to model linear and nonlinear systems without the need to make assumptions implicitly as in most traditional statistical approaches. Yoa et al (2012) posited that classical time series analysis does not perform satisfactorily on economic series because of the nonlinear and noisy features of the series. Hence, the idea of applying nonlinear models, like neural networks to problem with chaotic series becomes imperative.

**3.1. Multilayer perceptron (MLP):** This is a network made up of more than one layer with each layer having a definite number of units. The arcs that connect to the units in the network have associated weights known as the network weights or synapses. Given an input output series  $(x_j, z_j); j= 1, 2, \dots, T$ , interest is to find the best MLP network that fits the series appropriately. The topology of a multilayer perceptron in Fig (1) shows how the inputs are fed to the network, processed (refined) and presented as the network output.

**Fig 1: The topology of a MLP.**



$s_i = \sum_{j=1}^k w_{ji} x_j + b_i$  is the weighted sum of the network input plus the bias associated with each unit in the hidden layer. The hidden layer upon activation gives  $z_i = g(\sum_{j=1}^k w_{ji} x_j + b_i)$ . Thus, the network model becomes output

$$y_t = \sum_j^n w_{jo} g(\sum_j^k w_{ji} x_j + b_i) + b + e_t \quad (6)$$

where  $w_{ji}$  is the weight of the  $j^{\text{th}}$  input in the input layer associated with the hidden layer,  $w_{jo}$  is the weight of the  $j^{\text{th}}$  input in the hidden layer associated with the output,  $b_i$  is the bias associated with the  $i^{\text{th}}$  unit of the hidden layer,  $b$  is the bias associated with the output layer and  $e_t$  is the network forecast error.

### 3.2. THE NETWORK LAYERS

- a) **First layer:** This layer is also known as the input layer. It consists of n-units, where n is a finite number of variables or lagged variables which each observation depends on. Using too few inputs can result in inadequate modeling whereas too many inputs can inordinately complicate the network. Also, too many inputs would imply slower training which will give rise to delayed convergence and may in fact worsen the prediction ability of the network model. This therefore, prompts the necessity to systematically determine the input units beforehand. In this initial stage of the network system, the n-tuples of inputs  $x_i$  fed to the network are weighted with the network weights  $w_{ji}$  (synapses) and summed up together with the constant bias term  $b_i$ . The weighted sum of the inputs ( $s_i$ ) is systematically fed to the second (hidden) layer for proper transformation using a specific activation function.
- b) **Second layer:** This is also known as the hidden layer. It can be termed the network refinery owing to the fact that it is the layer where the weighted sums of the inputs are refined to suite the network target. This transforms the result of the previous layer into a linear or nonlinear function (.) depending on the activation function used and the type of output the network is expected to produce. The complexity of the network increases with the addition of more hidden layers.
- c) **Third layer:** This is also known as the output layer. This layer receives refined observations from the hidden layer as inputs, processes them using a specific output activation function and presents the result as the network estimated value.

#### 4. DATA ANALYSIS AND RESULT

The data used in this research work are the monthly average exchange rate of U.S dollar, British Pound and Euro with respect to naira. The data were sourced from the depository of Central Bank of Nigeria. The data set contains a time series of 68 lengths for each of the currencies starting from January, 2012 to July, 2018. The original series are not stationary, so, we went on to difference the series ones so as to make them stationary.

##### 4.1. Model order selection

Consider a k-dimensional time series  $z_1, z_2, \dots, z_T$  with  $z_t = (z_{1t}, z_{2t}, \dots, z_{kt})^T$ , generated by a vector autoregressive process of order p given as

$$z_t = c + B_1 z_{t-1} + \dots + B_p z_{t-p} + e_t \quad (7)$$

The order p is unknown but taken to be an upper bound of the process order. However, the approximate MSE matrix of the 1-step predictor is directly proportional to the order p. In other words, it changes as the order p changes. Choosing p to be excessively large will reduce the forecast precision of the corresponding estimated VAR (p) model (Lutkepohl, 1991). To identify the optimum order of the VAR model (7), we consider the likelihood ratio test method. The likelihood ratio statistics compares the likelihood function values under the null and alternative hypothesis. In this method, the model with smaller number of lags p is treated as the restricted version of a higher dimensional model. The interest is to test, under the null hypothesis that the restricted model is better than the higher dimensional (unrestricted) model. Thus, the differences in the likelihood ought to be very small so as to facilitate the acceptance of the null hypothesis.

Given that the error component  $e_t$  is a white noise, that is  $e_t \sim (0, D)$ , where  $D$  is the covariance matrix of the error component. Let the likelihood function for the restricted version be given as  $(\omega^r, D^r)$  and for the unrestricted version be given as  $(\omega^{ur}, D^{ur})$ , where  $\omega^r$  and  $\omega^{ur}$  are  $(k^2 p + 1) \times 1$  dimension of  $vec(B_i)$  for the restricted and unrestricted version of the model respectively.  $v(B_i)$  is the stacked columns of the coefficient matrix  $B_i$ . Then, the likelihood ratio

$$LR = \frac{L(\omega^{ur}, D^{ur})}{L(\omega^r, D^r)} \quad (8)$$

The log of LR gives

$$LR^* = \ln L(\omega^{ur}, D^{ur}) - \ln L(\omega^r, D^r) \quad (9)$$

Differentiating  $LR^*$  with respect to  $\omega^r$  and  $\omega^{ur}$  reduces the test statistic to

$LR = (\ln|D^r| - \ln|D^{ur}|) - Xv^2$ , where  $v$  is the number of tolerated restrictions. However, a likelihood ratio test is biased against the null hypothesis in small samples. Therefore, it is appropriate to use a corrected likelihood ratio test given as  $LR_c = (T - kp)(\ln|D^r| - \ln|D^{ur}|)$ , where  $kp$  is the number of estimated parameters in each equation of the unrestricted model. The algorithm for lag length selection using the corrected likelihood ratio test is as follows:

- a) Choose an upper bound for the lag order  $p_u$
- b) Test a restricted VAR ( $p_u - 1$ ) against an unrestricted VAR ( $p_u$ ) using likelihood ratio test. If the null hypothesis is not rejected,
- c) Test a further restricted VAR ( $p_u - 2$ ) against VAR ( $p_u - 1$ ) using likelihood ratio test. Continue in this order until the null hypothesis is rejected.

With this algorithm, each of the hypotheses is tested conditional on the fact that the entire previous null hypothesis is true. The sequential test has its significant level as an increasing function of an integer  $i$  defined as  $\alpha(i) = 1 - (1 - \alpha_0)^i$ , where  $\alpha_0$  is the initial significant level and  $i = 1, 2, \dots, n^{th}$  stage.

The rejection of VAR ( $p_u - i$ ) implies that all the VAR ( $p_u - i'$ ),  $\forall i' > i$  have been rejected. The likelihood function algorithm discussed above is based on level of significance (which is at researcher's volition), therefore, there is a high tendency of the iterative test converging at the wrong order. Since the VAR model construction is aimed at optimum prediction of the variables under study, our interest is on obtaining the good model for that purpose; that is a model that minimizes the forecast (out-of-sample) error. Define the  $l$ -step ahead forecast error based on time information to be  $Z_{t+l} - \hat{Z}_t(l)$  and define the  $l$ -step ahead mean square error (MSE) of the forecast model to be

$D_z(l) = E[z_{t+l} - \hat{z}_t(l)][z_{t+l} - \hat{z}_t(l)]'$ . For  $l = 1$ , the approximated value of  $D_z(1) = D \left[ \frac{T+kp+1}{T} \right]$  (see Lutkepohl, 1991, pg 88),  $D$  is the covariance matrix of the error component. Having approximated (1), we consider the metrics of the following information criteria: Akaike information criteria (AIC), Schwarz criterion (SWC), Hannan and Quinn criterion (HQC) and final prediction error (FPE). See Table. 1 for the results of the information criteria considered.

**Table 1: The information criteria result for lag length selection.**

Lags	AIC	SWC	HQC	FPE
1	16.671134	17.093684	16.836081	4372.02
2	16.843033	17.582495	17.131689	5319.231
3	16.773977	17.830352	17.186343	5516.128
4	17.017169	18.390456	17.553245	8704.201
5	17.031157	18.721357	17.690942	11288.67
6	17.193977	19.201089	17.977472	15286.28
7	17.188750	19.512775	18.095956	18383.19
8	17.353866	19.994803	18.382781	28731.87

From the result, it is apparent that all the information criteria considered take a minimum value at lag, 1. Thus VAR (1) model is identified to be the most appropriate for the series. That is

$$z_t = c + Bz_{t-1} + u_t \quad (10)$$

#### 4.2. Estimation of the model parameters

Let the general model given in (5) be written as  $z_t = Ax_t + u_t$

$$z_t = (z_{1t}, z_{2t}, \dots, z_{kt}) \quad A = (c, B_1, B_2, \dots, B_p) \text{ is a } k \times (kp + 1) \text{ Matrix.}$$

$x_t = (1, z_{t-1}, z_{t-2}, \dots, z_{t-p})$  is a  $(kp + 1) \times 1$  matrix.

$u_t$  is a normally distributed  $(k \times 1)$  vector of the error components.

Then  $z_t$  is multivariate normally distributed;  $z_t \sim (Ax_t, D)$  with the density function

$$f(z_t) = \left( \frac{1}{2\pi} \right)^{k/2} |D|^{-1/2} \exp \left\{ -\frac{1}{2} (z_t - Ax_t)' D^{-1} (z_t - Ax_t) \right\}$$

The likelihood function is  $L(A, D) = \prod_{t=1}^T f(z_t)$ . Taking the log of the likelihood function, gives the sum of the log of all the density functions. That is

$$\sum_{t=1}^T \text{Log}L(A, D) = -T \frac{k}{2} \log 2\pi - \frac{T}{2} \log |D| - \frac{1}{2} \sum_{t=1}^T \{ (z_t - Ax_t)' D^{-1} (z_t - Ax_t) \}$$

Differentiating the above equation with respect to the parameter A and equating to zero, results to the maximum likelihood estimator of the VAR coefficients

$$\hat{A} = \left[ \sum_{t=1}^T z_t x_t' \right] \left[ \sum_{t=1}^T x_t x_t' \right]^{-1}$$

Thus, the parameters of the VAR (1) model identified earlier are for the constant and slop respectively.

$$\begin{bmatrix} 0.5095 \\ 0.6187 \\ -0.0069 \end{bmatrix} \text{ and } \begin{bmatrix} 0.3248 & -0.0336 & -0.1003 \\ 0.8078 & -0.0528 & -0.0874 \\ 0.1647 & 0.1357 & -0.1132 \end{bmatrix}$$

The robustness of the fitted model was checked by testing for the presence of serial correlation among the residuals of the three variables considered. The p-value of the test (0.9011) is relatively high, thereby leading to the acceptance of the fitted model.

#### 4.3. Granger causality

Granger (1969) launched a concept of causality which reveals the structural relationship (causality) among multivariate variables. The term causality suggests cause and effects relationship between two sets of variables (Lutkepohl, 1991. Pg. 41). This concept extracts the intrinsic interdependency that exists among the variables. The idea is that a cause cannot come after effect. Thus, if a variable say x affects a variable say y, x should help in improving the predictions of y. If x causes y and y causes x, the system is called a feedback system. Here, we test for causality under the null hypothesis that variable  $i$  do not granger cause variable  $j \forall i \neq j$ .

From Table. 2 at 0.05 level of significance there is dual causality effect (feedback system) among the variables except between USD and Euro where euro only granger cause USD (the reverse does not hold).

**Table 2: Granger causality test result.**

Variable Granger Variable		F-	P-value
cause		ratio	
Usd	Pound	4.7812	0.03244
Usd	~ Euro	3.0734	0.08437
Pound	~ Usd	31.88	0.00000
Pound	~ Euro	6.8637	0.01097
Euro	~ Usd	4.9107	0.03025
Euro	~ Pound	4.3262	0.04154

**4.4. Design and neural network modeling of the series**

*Back-propagation algorithm:* David et al, (1986) in their paper entitled “*Learning representations by back-propagating errors*”, elaborated the usefulness of back-propagation in learning a network. Their work, describes several neural networks where back-propagation works faster than earlier approaches to learning, making it possible to use neural nets to solve problems which previously could not be solved. Back-propagation algorithm is a mathematical algorithm that acts on the weights of the network in order to improve the performance. It does not only improve the performance of the network, but also gives a detailed insight into how changing the weights and biases alters the general behavior of the network. The general procedures for designing a neural network model are:

- a) Data collection: The data are the monthly average exchange rate of three ‘most instrumental’ currencies (US dollar, British pound and Euro) against Naira, in currency arbitrage. The data were procured from Central Bank of Nigeria online depository.
- b) Data preprocessing: The series were preprocessed through log-transform and normalization. This helps in bringing the series closer to the range of the transfer function used in the training and prevents overshooting the deviation between the network output and the target variable. The log-transformed input vector is normalized into [-1, 1] before being fed into the first layer and the network outputs are de-normalized through reverse function to the log-value. Preprocessing (transforming) the series smoothes its distribution and facilitates convergence in the algorithm.
- c) Building the network: As stated earlier, a MLP network with gradient descent backpropagation algorithm was used to train the network. A three layer network was used with the input layer having 3-units, hidden layer having 3-units and a unit output layer. The choice of the units of the input layer as three is based on the knowledge gotten from the VAR model fitted earlier. From the VAR model, it was learnt that the model has an optimum order (1) with cross causality effects among the exchange rates. Therefore, better performance will be obtained using (for each currency) information from the three currencies instead of treating each of them in isolation. Thus, we used the order (p=3) to be the number of units in the input layer. There was a systematic ordering of the series such that  $z_{1t}$ (dollars) values were available before  $z_{2t}$  (pound) values, and  $z_{2t}$  values were available before  $z_{3t}$  (euro) values and in the same order, all these series were available before  $z_{it+1}$  ( $\forall i = 1, 2, 3$ ). In this work, we considered a sigmoid activation function for the hidden layer, defined as  $g(s_i) = \frac{1}{1+e^{-s_i}}$  with a learning rate of 0.01 and a purely linear function  $[(x) = x]$  for the output layer. The sigmoid function has the characteristics of being continuous, differentiable at all points and monotonically increasing. It accepts inputs defined on a real line  $[-\infty, \infty]$  and produces outputs that lie between 0 and 1. This prompts the re-scaling (log transform) of the original data.
- d) Training: The network was trained using 60% of the combined series, while 20% each was used for validation (confirmation of the truthfulness of the network model) and testing respectively. The training was done with Matlab (R2013a). During the training, the weights are adjusted at each epoch (iteration) in order to make the predicted output close to the network target. The input vectors are iteratively (in line with weight adjustment) presented to the network until the network performance is optimized. The Training stops when any of these conditions is met:
  - i. The maximum number of epochs (iteration) is reached. Here, the maximum number of epochs is set at hundred (100).
  - ii. Performance has been minimized to the goal (usually set at zero).
  - iii. The maximum amount of time has been exceeded. In this training, the maximum time is set at infinity.

The synaptic weights and biases associated with the optimized network are presented below:

For the input layer, the weight and bias matrices are

$$\begin{bmatrix} 1.2887 & 2.5078 & -2.8209 \\ 0.9420 & 0.0257 & -0.0798 \\ -2.1107 & -0.8994 & 2.7290 \end{bmatrix}$$

and

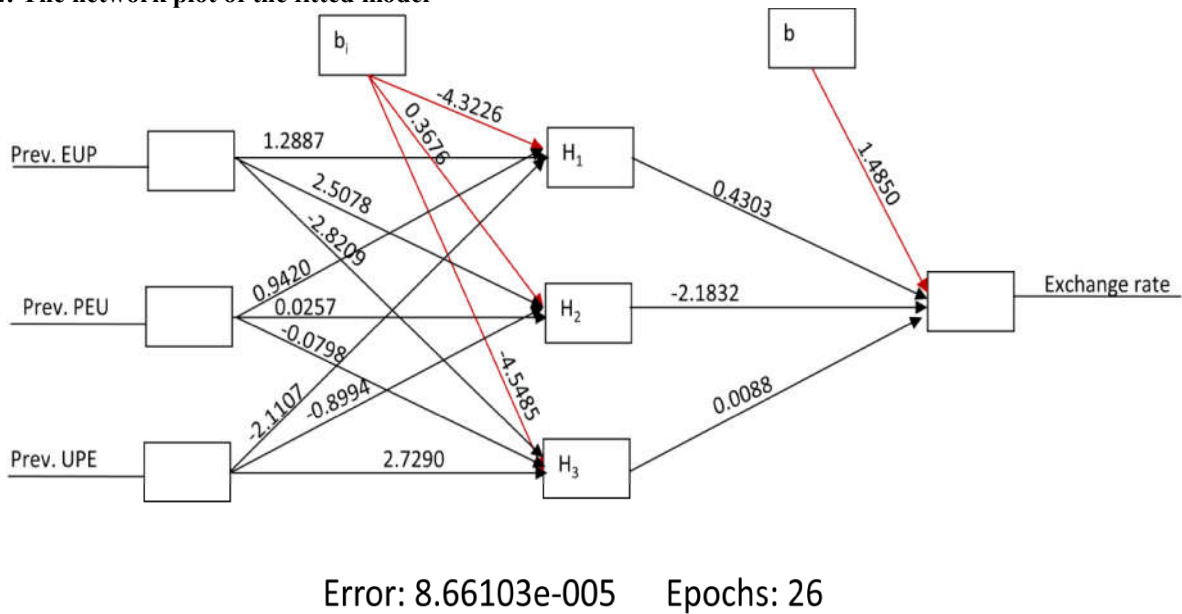
$$\begin{bmatrix} -4.3226 \\ 0.3676 \\ -4.5485 \end{bmatrix}$$

respectively.

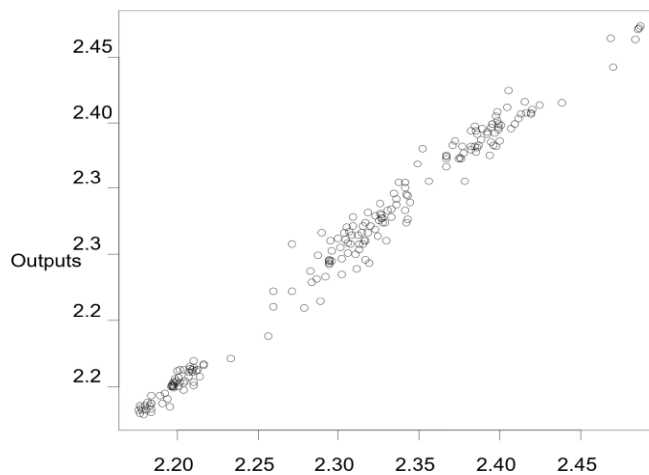
For the hidden layer, the weight and bias matrices are [0.4303 -2.1832 0.0088] and [1.4850] respectively. The generated network with the associated synapses is presented in Fig. 2. A scatter plot of the target against the network output is presented in Fig. 3. From the graph, the observation points for both the targets and network outputs (dots) are closely

clustered. Inscribing a linear regression line from the origin into the graph, divides the dots almost into two equal parts. This is evidence that shows that the prediction model is appropriately fitted.

**Fig 2: The network plot of the fitted model**



**Fig 3: A scatter plot of the target and the network output**



**Targets**

□ Testing the performance of the model: at this stage, the performance error of the network at each epoch is investigated. Since the goal of the network is set at zero, the training algorithm is continued until the network training performance error converges to a value very close to zero. Here, the convergence occurred at the 26<sup>th</sup> epoch. That is the epoch at which the network training performance error, network validation error and network testing error are each taking minimum value.

**Table. 3 shows the performance error of the network.**

Epoch showing progress	Network Training performance error	Network validation error	Network testing error	Gradients associated with the network training
0	0.0198182	0.0267	0.0131	0.0497621
25	8.69318e-005	0.0001	0.0001	1.92123e-005
26	8.66103e-005	0.0001	0.0001	1.8268e-005

**Table 4: The performance measures of the two models**

Performance measures	VAR model	Network model
Root mean square error	4.7501	0.0099
Mean bias error	0.00000913	0.0002
Co-efficient of determination	0.0831	0.9274

## 5. Discussion and Conclusion

As pointed out earlier, the determination of the number of units in the network layers has been based on trial and error method. In this work, we have been able to determine the optimal number of units in the network layers through VAR modeling. The performance of the network model is examined using the ‘test set’ of the data. Here, approximately 20% (the last 40 points) of the ordered series were used to test the performance of the model. In order to quantitatively evaluate the general performance of the two developed models (VAR and Neural networks models), statistical measures (root mean square error (RMSE), mean bias error (MBE) and coefficient of determination ( $R^2$ )) were calculated. The results are presented in Table. 4. The coefficient of determination of 0.0831 for the VAR model, indicates that 8.31% of the variability in the predicted values are caused by the lagged variables. This value, being very small indicates that the independent variables (previous values of the series) do not predict the series appropriately. On the other hand, the coefficient of determination of the network model shows that 92.74% of the variability in the predicted values is caused by the inputs. This is an indication that the neural network model performs very well based on the input. Also, the RMSE of the network model being smaller than that of the VAR model shows that the network model is better for prediction than the VAR model. However, an essential contention to the claim of improved performance using the neural network approach in comparison to the Box and Jenkins approach is that neural networks are more complex and have many more parameters (weights and biases). Thus, they are often taken to be non-parsimonious. This situation occurs when the complexity of the network is increased by the addition of more hidden layers with the intention of improving the network performance. However, the interest of analysts who use neural network modeling technique should always be to strike a balance between parsimony and network improvement.

## References

- [1]. Daniya, T. (2015). Forecasting exchange rates using time series analysis: the sample of the currency of Kazakhstan. Cornell university press.
- [2]. David, E. R., Geoffrey, E. H. & Ronal, J. W. (1986). Learning representations by back-propagating errors. Nature Publishing Group.
- [3]. Eric, S. (2002). Forecasting foreign exchange rates with neural networks. Computer science institute, University of Neuchatel.
- [4]. George, E.P, Gwilym, M.J & Gregory, C.R. (2008). Time series analysis forecasting and control. John Wiley & Sons, Inc., Hoboken, New Jersey.
- [5]. Granger, C.W.J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, vol. 37, 424-438.
- [6]. Kanad, C., Kishan, M., Sanjay, R. & Chilukuri, K.M. (1990). Forecasting the behavior of multivariate time series using neural network. School of computer and information science, Syracuse University, Syracuse.
- [7]. Kondratenko, V. & kuperin, Yu. (2003). Using recurrent neural network to forecasting of forex. Cornell University Press.
- [8]. Lutkepohl, H. (1991). Introduction to multiple time series analysis. Springer-Verlag Berlin Heidelberg GmbH.
- [9]. Lutkepohl, H. (2006). New introduction to multiple time series analysis. Springer, Berlin.
- [10]. Sana, B., Nejib, S. & Mahmoud, G. (2003). The box-jenkins analysis and neural networks: prediction and time series modeling. *Journal of applied mathematical modeling*. Vol. 27, 805-815
- [11]. Sims, C.A. (1980). Macroeconomics and reality. *Econometrica*, vol. 48, 1-48
- [12]. Yoa, Q., Zongwu, C. & Jianqing, F. (2012). Functional-coefficient regression models for non-linear time series. *Journal of the America statistical association*.