ANALYSIS OF STOCK MARKET EXCHANGE WITH PARTICULAR INSTANCE FROM NIGERIA STOCK IN MATLAB AND R

VICTOR O. WAZIRI[1], PATIENCE WOYE ADAMA[2] & AUDU ISAH[3]
Department of Cyber Security Science,
School of Information and Communication Technology,
Federal University of Technology, Minna
[2]Department of Computer Science and Mathematics, FPI, Bida
[3]Department of Statistics, Federal University of Technology, Minna, Nigeria
E-mail: victor.waziru@futminna.edu.ng, Yapacie123@gmail.com, aisah@futminna.edu.ng
Phone No: +234-806-351-8931

Abstract
This paper presents an analysis of arbitrary dataset for Nigeria Stock Exchange Market in Matlab and R software. The simulated outputs from the dataset are applied to compute and predict the future trends of the Nigerian Stock Exchange Market. The analytical processes of the dataset are achieved in phases. First, the dataset is preprocessed, then learn and trained using Neural Network (NN) for Matlab and Multiple Linear Regression (MLR) for R. The simulation of the model is categorized into subsystems that are based on Artificial Neural Network Multilayer Perceptrons (ANN MLP). The ANN MLP that is applied in the simulation consists of three layers; one input layer, three Hidden Layers and the Output Layers. Their Mean Squared Errors and Coefficient of Determinants were used to compare and discuss results through graphical and simulated figures. The graphical representations and the mathematical models of Matlab and R were derived and compared. It is discovered that the ANN algorithm in Matlab gives more efficient and acceptable outputs forecasts than MLR in R.

Keywords: Nigeria Stock Exchange, Analysis, Neural Network, Multiple Linear Regression, Pre-processed

Introduction
Stock-exchange market is a public entity for the trading of company stock (shares) or equity and derivatives at an agreed price by the brokers. Stock exchange market prediction is a very important aspect of the economy that requires consideration due to the re-occurring challenges in the stock market. All the investors need to know when to make a buying or selling decision in the expected direction of the stock. Studies have also shown that predicting direction as compared to value can generate higher profit Kimoto, *et al.* (1990). The stock exchange markets have become an integral part of the global economy. Our personal and corporate financial lives and the economic health of a country are influenced by any fluctuation in the market.

The Nigerian Stock Exchange (NSE) which was established in 1960 has grown rapidly and would continue to evolve to meet the needs of its valued customers. For example, the Nigerian Stock Exchange (NSE) as of December 31, 2012 has about 198 listed companies with a total market capitalization of about ₦8.9 trillion ($57 billion) *(*www.nse.com.ng/…/PRESCO%20DEC%202012%20AUDITED.pdf*).* This makes NSE *champion* the acceleration of Africa's economic development and to become "the Gateway to African Markets". The prediction of the movement of stocks helps the investors to know when and where to invest in stocks, for analyzing price patterns and predicting stock prices and index changes. The investors and traders depend so much on the intelligent trading systems which help them in predicting prices with the knowledge of previous situations and conditions.

The research focused on developing a better approach to successfully predict stock market prices that could achieve higher profits with the aid of least complex stock market models. First, we pre-processed the data obtained from the Nigerian Stock Exchange (NSE) by normalizing it. The Methods to be adopted are Neural Network in Matlab and Multiple Linear Regression in R; its models would be given and discussed. Necessary diagrams and utilities shall also be stated and discussed appropriately.

The data from NSE daily returns was pre-processed using normal transformation of a normal curve $(0 < x < 1)$ using the MINITAB Version 13.0 software, and was also analyzed so that there are no leaking values (that is, redundant stock data owing to technical suspension of the stock by the Nigerian Stock Exchange) and the pre-processed data were used for the experiments.

The rest of the paper is arranged as follow: Section two reviews the related works, section three is on material and methods, section four presents software implementation of results, while section five presents the experimental results and section six is on conclusion and recommendation.

Related Works

We explored various literatures on the application of artificial intelligence systems and time series prediction models such as Artificial Neural Network, Support Vector Machine, Expert System, and Traditional Forecasting Method. Among them are these few that are related to our area of interest. Mackinlay and Lo, (1988) defined stock market as a complex, non-stationary, noisy, chaotic, nonlinear and dynamic system but it does not follow Random Walk Hypothesis (RWH). The financial market movement is caused by so many factors that include: Economic situations, Political situations, Traders' expectations and other unexpected events. Stock prices are very dynamic and are quick to changes because of the underlying nature some known parameters like Previous Day's Closing Price, and other unknown factors (like Election Results, Rumors, an outbreak of disease such as the recent Ebola disease in Africa. Haykin, (1994) described Neural Network as an adaptive machine or more specifically as a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects: Knowledge is acquired by the Network through a learning process, and interneuron connection strengths known as synaptic weights are used to store the knowledge. With the knowledge of Artificial Neural Networks techniques (ANN) many researchers have applied these techniques to various fields including stock market prediction. Zahedi, (1993) described Expert Systems and Artificial Neural Networks as the qualitative methods offer for business and economic systems than traditional quantitative tools in Statistics and Econometrics cannot quantify due to the complexity in translating the systems into precise mathematical function. Kartalopoulos, (1996), Vasant, and Roger (1996) and Ward, and Marge, (1995) all mentioned to varying degrees that Neural Networks have the capability to forecast financial markets. Bulter and Daniyal (2009) attempt to make accurate prediction of the movement of stock market with the aid of Evolutionary Artificial Neural Network (EANN) and it was constructed for Multi-Objective Optimization (MOO) which was trained to build up data and its effect on Market performance. It was observed that the main contribution was to show that an EANN trained to recognize direction and magnitude in the Stock Market and to recognize direction changes. ANN can be used to perform classification and regression tasks. It was also proved by (Clarence and Tan, 1997) as the one that best deals with uncertainty in finance. Primarily, it involves recognition of pattern in data and using these patterns to predict future events.

Hsieh (1993) states that many potential corporate finance applications including Financial Simulation, Predicting Investor's Behavior, Security and a host of others can be significantly improved with the adaptation to ANN technology. Khan and Gour (2013) used some technical indicators and BackPropagation Neural Network (BPNN) to predict the Stock price of the day. The stock rate prediction accuracy of the technical indicators is compared with that of the BPNN and the result show that the BackPropagation Neural Network is more accurate than the other techniques.

Jason (1998) examined and analyzed the use of Neural Networks as a forecasting tool. He tested the Neural Network's ability to predict future trends of Stock Market Indices. The accuracy was compared against a traditional forecasting method, multiple linear regression analysis. Finally, he calculated the probability of the model's forecast using conditional probabilities. His research determines the feasibility and practicality of using Neural Networks as a forecasting tool for the individual investor. Jason, (ibid) builds up his study on the work done by Gately (1996) in his book. The research validates the work of Gately, (1996), and described the development of a Neural Network that achieved a 93.3% probability of predicting a Market rise, and an 88.07% probability of predicting a Market drop in the Standard and Poor 500 (The Standard & Poor's 500, often abbreviated as the S&P 500, or just "the S&P", is an American stock market index based on the market capitalizations of 500 large companies having common Stock listed on the NYSE or NASDAQ. The S&P 500 index components and their weightings are determined by S&P Dow Jones Indices). He concluded that Neural Networks do have the capability to forecast financial Markets and, if properly trained, the individual investor could benefit from the use of this forecasting tool.

The ANN has been used by many other researchers to predict Stock Exchange of different Country's Markets. (Idowu, Osakwe, Kayode, and Adagunodo, 2012), used ANN for the prediction of Nigeria Stock Market. (Refenes, Zapranis, & Francis, 1995) applied ANN for the prediction of Tokyo Stock Exchange index.

Material and Methods

This section gives a general outlook of Neural Network (NN) as a Machine Learning tool

Neural Network

NN is a Connectionist model that is well suited for machine learning where connection weights are adjusted to improve the performance of a network. An ANN is a network of nodes connected with directed arcs each with a numerical weight, $w_{i,j}$, specifying the strength of the connection.

These weights indicate the influence of previous node, u_j, on the next node, u_i, where positive weights represent reinforcement; negative weights represent inhibition. Generally, the initial connection weights are randomly selected. During the training process a set of pattern examples is used, each example consisting of a pair with the input and corresponding target output. The patterns are presented to the network sequentially, in an iterative manner, the appropriate weight corrections being performed during the process to adapt the network to the desired behavior. This iteration continues until the connection weight values allow the network to perform the required mapping. Each presentation of the whole pattern set is named an epoch. The back propagation learning generally involves the following four steps:

Step 1: Initialization:

Initialize the weights and thresholds of the Network.

Step 2: Activation:

Activate the back-propagation Neural Network by applying inputs $x_1(p), x_2(p), \ldots\ldots x_n(p)$ and desired outputs $d_1(p), d_2(p), \ldots\ldots d_n(p)$

Step 3: Weight training:
Update the weights in the backpropagation Network by propagating backward the errors associated with output neurons.

Step 4: Iteration:
Increase iteration p by one, go back to Step 2 and repeat the process until the selected error criterion (usually mean squared error) is satisfied.

Fig. 3.1 depicts an abstractive concept of the Artificial Neural Network Multilayer Perceptron algorithm that would be applied in the dataset analysis for the prediction of the stock exchange Market.

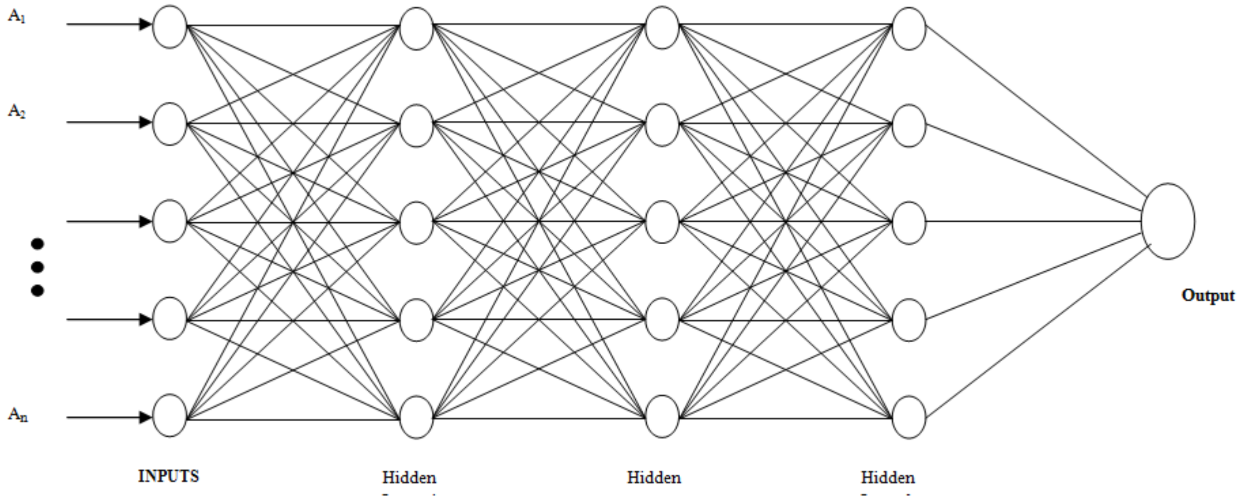

Fig. 3.1: The Multilayers depicting one Input layer, three hidden layers and one Output layer

Multilayer Perceptron
Fig. 3.1 expounds the structure of a Multilayer Perceptron. A Multi-Layer Perceptron (MLP) Network is composed of several layers containing nodes. The lowest layer is the input layer where external data is received. Generally, neural processing will not occur in the input layer. Therefore, the input layer is not treated as a layer in Neural Network processing units, but as merely input units Kecman, (2001). The highest layer is the output layer where the problem solution is obtained. In the case of predicting the currency market, the inputs will be the past observations of the Exchange Market Stocks and the output will be the future value of the Exchange Market Rate. Between the input and output layer, there can be one or more intermediate layers that are called the hidden layers.

The main advantage of MLP networks is their ease of use and approximation of any input/output map. The main disadvantage is that they train slowly and require lots of training data. The connections between individual neurons that make up the ANN allow the neurons to signal each other as information is processed. If the flow of information through the ANN is from the input layer to the output layer, the ANN is said to be *feed forward*. Each connection between the neurons is assigned a certain weight. The weight equals zero when there is no connection between two particular neurons. In addition, a bias neuron may be connected to each neuron in the hidden and output layers which has a value of positive one. These weights are what determine the output of the ANN.

To explain how the weights, determine the output, consider the following. Let the MLP inputs be represented by x_i with $i = (1,2,\dots,l)$, and representing the number of inputs. In addition, let the hidden nodes be represented by h_j with $j = (1,2,\dots.m)$, and m representing the number of hidden nodes. Finally, let the actual value and the MLP output be represented by $y_k$and$\hat{y}_k$, respectively, with $k = (1,2,\dots,p)$. The input vector X and the series of weight vectors w_j is then defined as x_i.

The output of each processing unit for the forward pass will be defined as follows:

$$S_i = \sum_{j=0}^{n} w_{ij}\eta_j \qquad (3.1)$$

where S_i is the total input at layer i, w_{ij} is the weight of the input from node j in layer i, *and*η_j *is the activation function which is defined as:*

$$\eta_j = f(S_i),\ \ where\ \ f(input) = \frac{1}{(1+e^{-input})} \qquad (3.2)$$

Training the Arbitrary Dataset and Normalization Output

Due to lack of time and space, it is not possible to present the raw arbitrary Stock Market Exchange dataset. The computing process for the normalization of the acquired raw dataset result has no computational space as an algorithm; but Table 3.1 depicts the dataset normalized outputs which shall be applied in experimental section.

Table 3.1: Normalized dataset of our

Training Parameters	NN1 Train	1st Retrain	2nd Retrain	3rd Retrain	4th retrain	5th retrain	6th retrain	7th retrain
Iterations	13	9	15	8	12	14	15	13
Training MSE	0.00618566	0.00735707	0.00583144	0.00787549	0.00662532	0.00551117	0.00657987	0.00553799
Validation MSE	0.00944252	0.00698658	0.00793371	0.00603267	0.00744463	0.00833006	0.00862000	0.00816243
Testing MSE	0.00788424	0.00681670	0.00954940	0.00520126	0.01047210	0.00934746	0.00962382	0.00838818
Regression	0.48749	0.47795	0.53002	0.50716	0.50484	0.5807	0.49665	0.59163
Duration	1sec	3secs	7secs	3secs	4secs	0sec	4secs	41secs
Mu	0.00100	0.00100	0.00001	0.00100	0.00010	0.00010	0.00100	0.00100

The dataset was pre-processed using normal transformation of a normal curve $(0 < x < 1)$ using the MINITAB Version 13.0 software tool. The experiment was carried out for Neural Network and the topologies based on the efficient computing of equations (3.3) - (3.5). The Training Function used was Levenberg-Marquardt Backpropagation Algorithm. The learning rate per layer used is 0.001 while the training tolerance is of maximum epoch size of 1000; the Levenberg-Marquardt Backpropagation layering equations are:

$x_i - s - s^2$; (3.3)

$x_i - s - s^2 - s^3$; (3.4)

$x_i - s - s^2 - s^3 - s^4$ (3.5)

The variable x_i represents the ith input into MLP, the s_i's in equations (3.3)-(3.4) are as defined in equation (3.1) are the output processing units of the MLP topology.

Software Implementation of Results

In this section, presentations of basic experimental details are given using two Machine Learning tools; Matlab and R software.

Implementation of the MLP in Matlab

The Neural Network Model (NN1) with the Topology 3-7-1 was implemented and the Training Function Levenberg-Marquardt Backpropagation was executed as the simulation model. The results from its First training and two other retraining are given in Table 4.1 below. The NARX Neural Network view and the NARX Neural Network Closed Loop view are presented in Fig 4.1 and Fig 4.2, respectively. Table 4.1 and Table 4.3 simultaneously presented the performance results from the best NN1 Model.

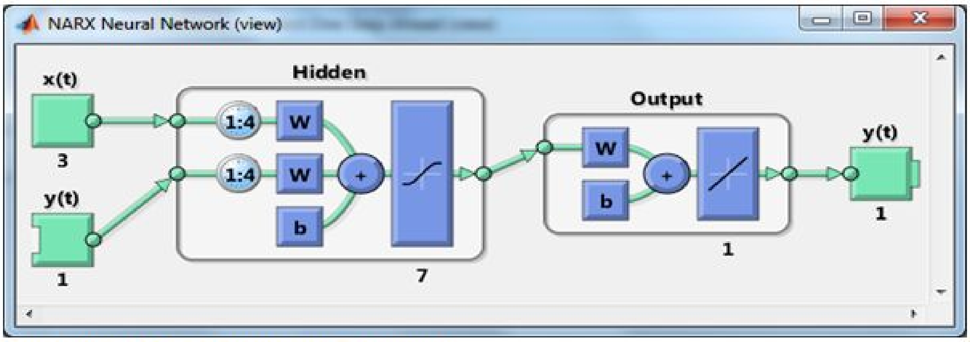

Fig4.1: A NARX MLP with 3-7-1 Topology

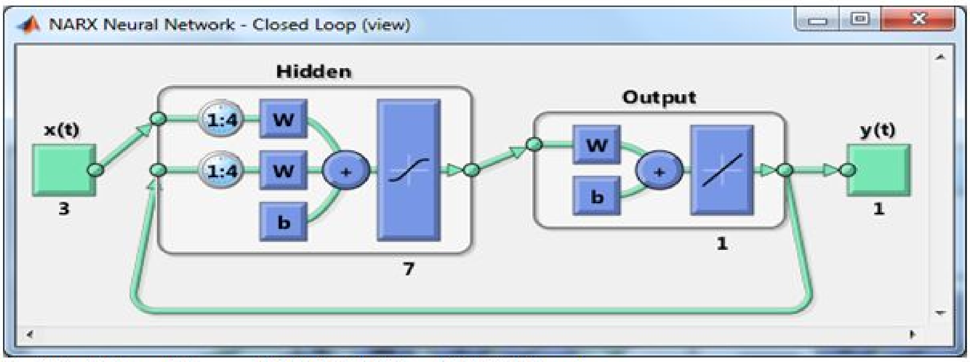

Fig4.2: A Closed Loop of NARX MLP with 3-7-1 Topology

Table 4.1: Performance Result of Model NN1

Topology	Types	Performance
3-7-1	NARX	0.0065
3-7-1	NARX Closed Loop	0.0094

Table 4.2: Performance Result of Model NN1

Table 4.2: Result of the best convergence training of NN1 Model

Data set	Target	Mean Squared Error	Regression
Training	608	0.00735707	0.589887
Validation	131	0.00698658	0.322966
Testing	131	0.00681670	0.326430

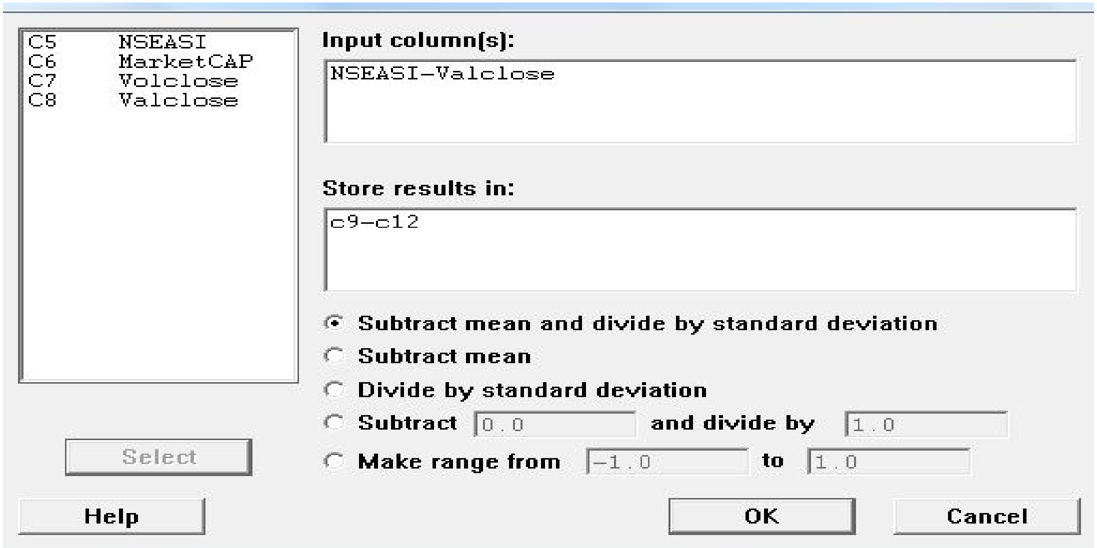

Fig. 4.3: Application of Matlab with stock inputs classified as c_i, $i = 5, \ldots, 8$

Matlab offers a unique simulation and prototype environment as demonstrated in fig 4.3 for analyzing dataset. It has a concise and descriptive dialogue spaces that allows one to model complex and dynamic systems like the Stock Market with small sections of easy-to-flow-code. MATLAB offers an array of tools for simulation and modeling techniques. Applying specific tool boxes, such as Optimization, Control System, and Neural Network allows one to quickly build simulations and models for applications across a range of disciplines with limited coding from scratch. To visualize the simulation results as they calculate or for post-processing, built-in animation functions with graphics allow one to view model behaviors for analysis, testing and debugging, and presentation purposes. All these features and more make MATLAB an indispensable tool for use in this work. A Nonlinear Autoregressive with External (Exogenous) input (NARX) was used to predict series $y(t)$ given the past values of $y(t)$ and another series$x(t)$.

This gives the equation:

$$y(t) = f(x(t-1) \ \ldots \ x(t-1)y(t-1)y(t-d)) \tag{4.1}$$

Equation (4.1) is an essential ingredient for making computational forecast (prediction) when a given dataset has been pre-processed.

Application of R Software

The R project for statistical computing (or in Short R) is a powerful data analysis tool. It is a programming language, a computational and a graphical environment for data analysis. It has the advantage to run on other operating systems like Mac, Windows and Unix. R has its root from S language developed by Chambers (1998) at Bell Laboratories.

Multiple Linear Regression (MLR)

This is a technique used for predicting an unknown value of a variable from a known value of two- or- more variables. The multiple regression equation is of the form:

$$y_i = \beta_o + \beta_1 x_{i1} + \cdots + \beta_q x_{iq} + \varepsilon_i \quad (4.2)$$

where β_o is the intercept and $\beta_1, \beta_2, ---, \beta_q$ are regression coefficients. $y_i{}'s$ are the attributes and the $x_{i1}{}'s$ are the individual predictors, ε_i *are error* terms and $i = 1,2,---,q$. The basic assumptions are that: the relationship between the attributes and referrers are linear, the referrers are not related among themselves, normality and homoscedasticity.

Experimental Results

This section gives both the tables and visualization of our computational experiments based in Matlab and R simulated outputs

Implementation of the NN with one hidden layer

The Neural Network Model (NN1) with the Topology 3 – 7 - 1 was implemented and the Training Function Levenberg-Marquardt Backpropagation was used. The results from its first training and six other retraining were done using the NARX Neural Network. The regression outputs are presented in table 5.1.

Table 5.1: Result from the iterations of NN1 model training

Training parameters for NN1	Iterations	Regression
First Train	13	0.48749
Second train	9	0.47795
Third train	15	0.53002
Fourth train	8	0.50716
Fifth train	12	0.50484
Sixth train	14	0.58070
Seventh train	15	0.59163

Table 5.1 shows the effect of the training of the stock dataset and the regression coefficients over iterative training epochs. It shows that the regression coefficients change alternatively as the iterative counts is modified. Pick out the fourth train and the second train effects, in the fourth execution of the simulation; the iteration count is 8 epochs with regression 0.50716 while the second train gives a count of 9 number of iteration with regression 0.47795. This interpretation depicts the fact that the training, the observed iterations and regression coefficients are independent. This means that the Reponses and the Predictors are not the same in each training interval. In a nutshell, the responses and the predictors' relation may turn out to be correct, while others can be way off the mark.

Table 5.2: Result of the best convergence training of NN Model

Data Set	Target	Mean Squared Error	Regression
Training	608	0.00735707	0.589887
Validation	131	0.00698658	0.322966
Testing	131	0.00681670	0.326430

The data set was divided into the Training set of 70%, the Validation of 15% and the Testing set of 15%. The Testing set performed best in the Mean Squared Error and the Training had the best Regression.

Interpretation

Table 5.2 shows that the more the training, the better the model gives realistic prediction. The more the test data, the more accurate the error estimates. As it is generally known, the error rate percentage gives the incorrectly classified instances of the dataset. For a cogent interpretation of Table 5.2, however, more Data Mining construct formulations such as the Occam's razor and the practical issues in cross-validation models would have to come into analysis to make adequate and comprehensive interpretation. This task is beyond the horizon of the current paper space and time.

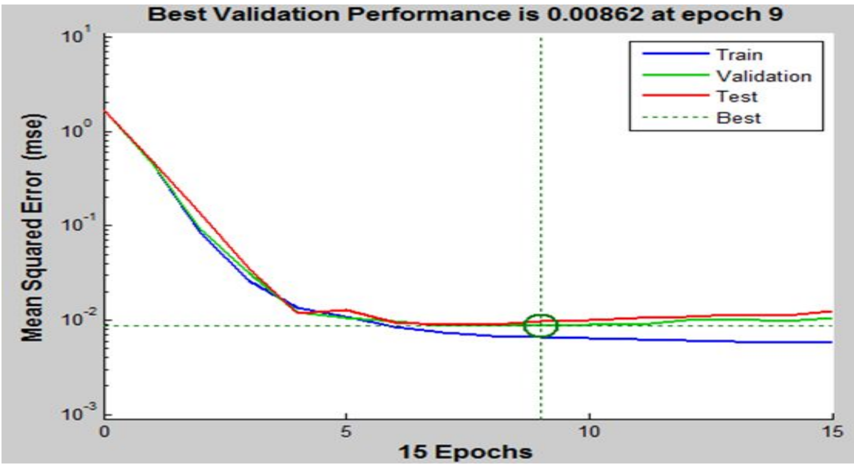

Figure 5.1: The Matlab Plot of Table 5.2

Figure 5.1 is the plot of the seventh train in Table 5.1 that has its Best Validation Performance as 0.00862 at epoch 9, and the Test and the Validation curves converged to the Best line. The training curve is close to the best line. Hence, it was considered to be the best plot and the train that had the best convergence.

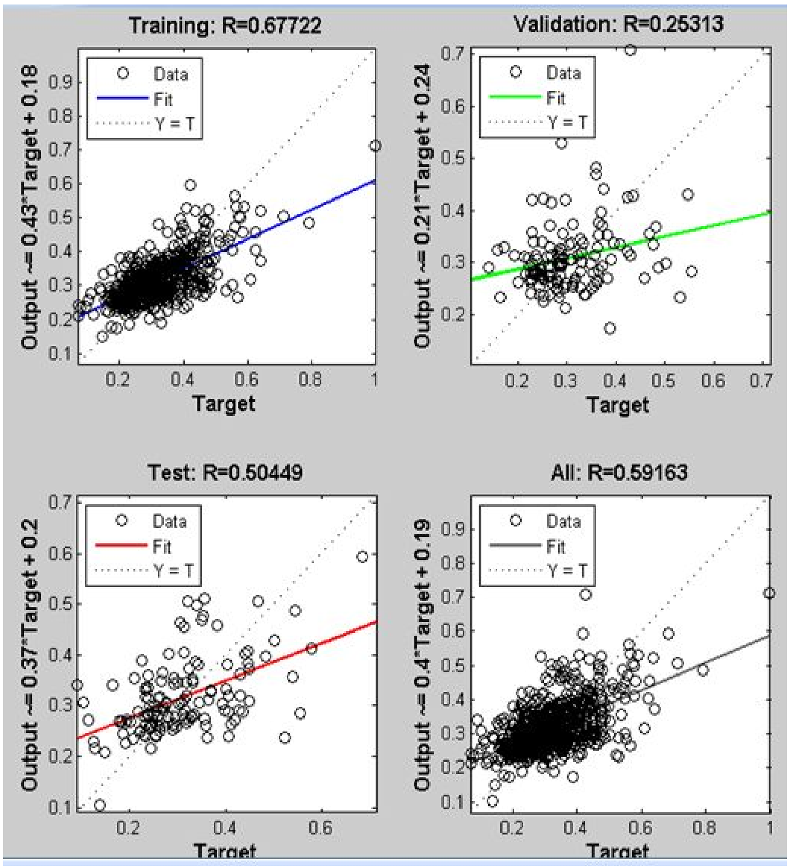

Figure 5.2: Best training Graphs showing the Training, Validation, Test and All Regression Value

The best train in figure 5.2 also gave the best training graphs. It`s All Regression value had the highest of 0.59163 with its coefficient of determination to be 0.3500. Note that the values given are acquired from the inherent Matlab Program Codes for the Training, Validation and all regressions. Thus, there is no need for the models to be outlined here as this work is implemented as a Machine Learning software, a tool; in this case, Matlab.

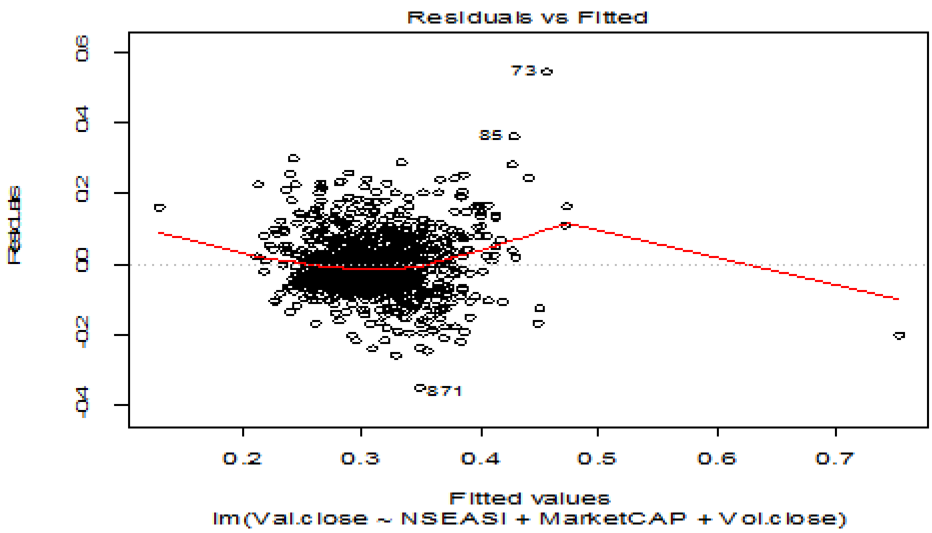

Figure 5.3: Plot of Table 5.1 Input Dataset

Figure 5.3 depicts the Training effects on Dataset Graphical Inputs of Table 5.1. The Plots in Fig 5.3 shows that at the seventh train it converged and had the best regression plot.

Implementation of the NN with two Hidden Layers

The Neural Network Model (NN2) with the Topology 3-7-5-1 was implemented and the Training Function Levenberg-Marquardt Backpropagation was applied.

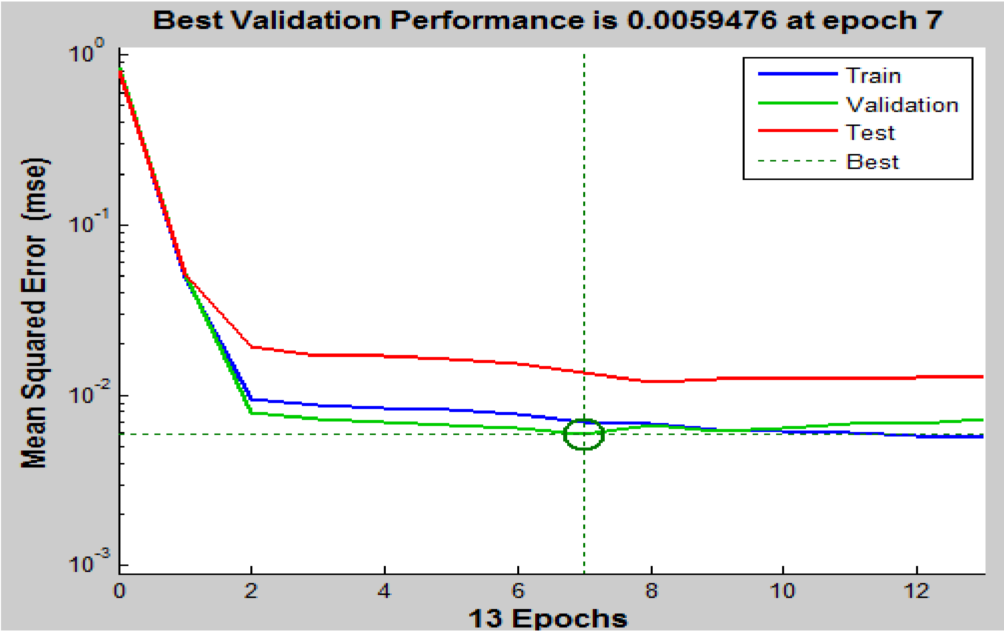

Figure 5.4: Plot from the Train of NN with two Hidden Layers

The plot of the train of NN in figure 5.4 shows that the test curve and the validation curve converged to the best line and had similar trend but the train curve did not converge to the Best line. The selection of the optimal number of units in the hidden layer is very important, because it impacts the model performance. Hence, the implication of this convergence to the NSE implementation is that at this probabilistic stage, the stock exchange would make predictive positive forecast on the Stock Market Exchange.

Table.5.3: Ranking results of the MSE and the R^2 of all the three NN models

Models	MSE	R^2	MSE Rank	R^2 Rank
NN1	0.0065	0.3500	1	1
NN2	0.0069	0.2826	2	2
NN3	0.0070	0.2735	3	3

From Table 5.3, the results were compared using their Mean Squared Error (MSE) and the coefficient of determination (R^2) to see which approach performed best. The result of the NN with one hidden layer was ranked first in both the MSE and the R^2. From Table 5.3, it implies that the training test which converges to the best line has its regression above 0.5 which gives better prediction equation.

Hence, the mathematical model for the ANN is:

$$A \cong 0.4 * T + 0.19 \quad (5.1)$$

The letter A is the actual stock price to be predicted while T is used as the training data (for today's stock price) which will be used for making prediction.

Multiple Linear Regression Analysis

Multiple linear regression analysis was performed on the normalized data using the R version 3.1.1. The output of this process for the final model is presented in figures 5.5 -5.7 below. The plot in figure 5.5 shows the residual errors plotted versus their fitted values. The residuals should be randomly distributed around the horizontal line representing a residual error of zero. For each variable, there did not seem to be major differences in the variability of the residual for different values of the independent variable. Therefore, the Homoscedasticity assumption is valid.

The plot in figure 5.5 suggests that the residual errors are normally distributed. The points on the plot seemed to deviate from a straight line in a random manner. This indicates normality which is also a necessary assumption. The plot in figure 5. 5 suggests that the residual errors are normally distributed. The points on the plot seemed to deviate from a straight line in a random manner. This indicates normality.

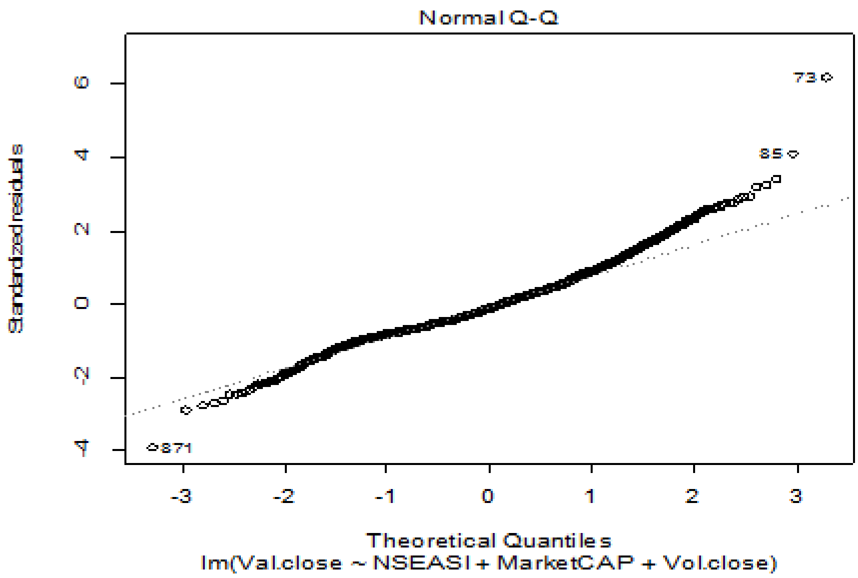

Figure 5.6: The Impact of Normal Distribution

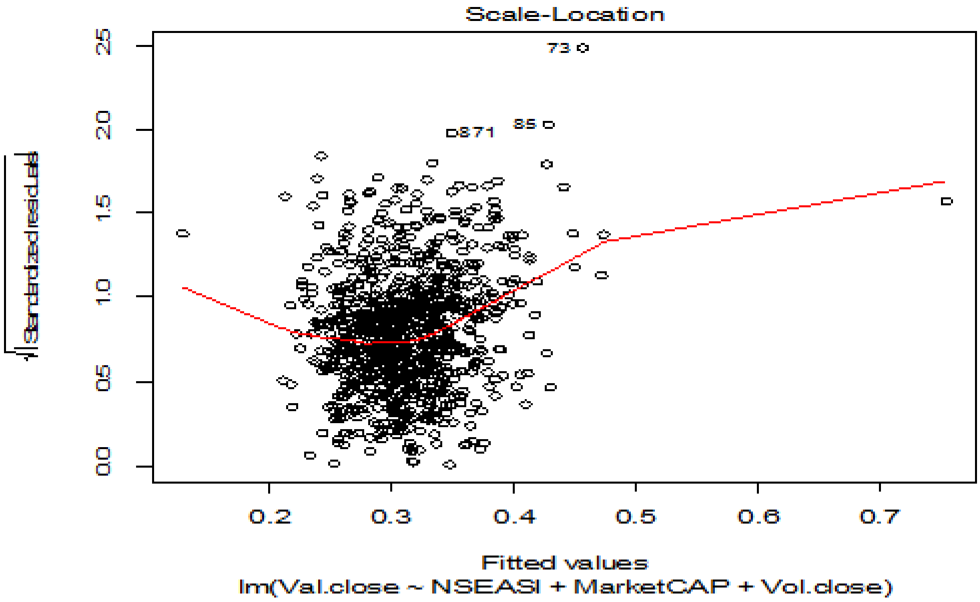

Figure 5.6: Regression Plot Scale versus Location

The scale-location plot in figure 5.6 shows the square root of the standardized residuals (sort of a square root of relative error) as a function of the fitted values. There was no pattern in the residual plots for each independent variable. Therefore, the linearity assumption is valid.

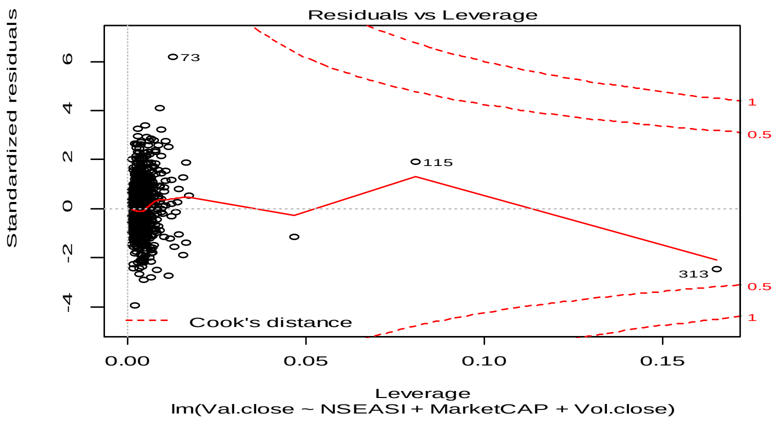

Figure 5.7: Regression Result Plot Residuals versus Leverage

Finally, Figure 5.7 shows each point`s leverage, which is a measure of its importance in determining the regression result. This shows no pattern and indicated the absence of Autocorrelation. Hence, the independence of errors assumption is valid. From the plots in figures 5.5 – 5.7 above, all the four assumptions of regression analysis were verified. Therefore, a linear regression model is appropriate. Hence, the Regression model equation is

$$Val@close = 0.0634 + 0.0523V@close + 0.240NSEASI - 0.024marketCap \qquad (5.2)$$

This can also be written as:

$$Y = 0.0634 + 0.0523X_1 + 0.240X_2 - 0.024X_3 \qquad (5.3)$$

Table 5.4 gives the outcomes of MLR, the Multiple Regression value was less than 0.5 which made the coefficient of determination (R^2) and the adjusted (R^2) very low and the MSE was very high Table 5.5: Ranking results of the MSE and the R^2 of Type A, Type B and MLR Models.

Table 5.4: The Outcome of the MLR

	Result
Multiple R	0.44
R^2	0.194
Adjusted R^2	0.192
MSE	0.08865
Observations	870

Table 5.5: MLR statistical outcomes.

Models	MSE	R^2	MSE Rank	R^2 Rank
NN1	0.0065	0.3500	1	1
NN2	0.0069	0.2826	2	2
NN3	0.0070	0.2735	3	3
MLR	0.0887	0.1940	4	4

Table 5.5 is the general comparison of the three subsystems of NN and the MLR result. The NN1 was ranked 1st in both MSE and the R^2 which came out to be the best, the NN2 was 2nd and NN3 was 3rd while the MLR was ranked 4th in both the MSE and R^2.

Conclusion

The NN was trained and retrained to see which one performed better. We discovered during the training process that the train, validation and test plot was shown on the same graph and at some points of the retrain they diverged from the best line but also converged at some points. We considered the retrain that gives better convergence and the higher regression value as the best.

The experiment carried out to predict the NSE future values was considered to be best in Model NN1, followed by NN2 and then NN3 in both MSE and R^2. This agrees with work done by Refenes Zapranis and Francis (1995), Gately (1996), Idowu, Osakwe, Kayode and Adagunodo (2012), and Khan and Gour (2013). Mathematical Model was developed from the best NN model as seen in equation (4.1). The study tested the ANN with different Hidden layers and different Neurons per layers, computing their MSE and R^2 keeping in mind the problem of over fitting the model. The NN1 was manually chosen as the best possible Network. This model ranks best in both the MSE and the Coefficient of Determination R^2.

Recommendation

It is recommended that more data mining construct formulations be used for the analysis of Stock Market data. Other practical issues in cross validation could also be explored.

References

Bulter, M. & Daniyal, A. (2009). *Multi-objective optimization with an evolutionary artificial neural network for financial forecasting.* Faculty of Computer Science, Dalhousie University, 6050 University Ave., Halifax, NS, Canada B3H 1W5.

Chambers, M. J. (1998). *Programming with data.* New York: Springer.

Clarence, N. & Tan, W. (1997). *An artificial neural networks primer with financial applications examples in financial distress predictions and foreign exchange hybrid trading system.* School of Information Technology, Bond University, Gold Coast, QLD 4229, Australia.

Egeli, B., Ozturan, M. & Badur, B. (2003). *Stock market prediction using artificial neural network.* http://www.hicbusiness.org/biz2003proceedings/birgul%20egeli.pdf

Gately, E. J. (1996). *Neural networks for financial forecasting.* New York: John Wiley & Sons.

Harris, M., Karper, E., Stacks, G., Hoffman, D., DeNiro, R. & Cruz, P. (2001). Writing labs and the Hollywood connection. *Journal of Film Writing,* 44 (3), 213 - 245.

Haykin, S. (1994). *Neural networks: A comprehensive foundation.* New York: Macmillian College Publishing Company.

Hsieh, C. (1993). Some potential applications of artificial neural systems in financial management. *Journal of Systems Management,* 44 (4), 12 - 24.

Idowu, A. P., Osakwe, C., Kayode, A. A. & Adagunodo, R. E. (2012). Prediction of stock market in Nigeria using artificial neural network. *I. J. Intelligent Systems and Applications,* 11, 68-74. Available at http://www.mecs-press.org/

Kaastra, I. & Boyd, M. (1996). Designing neural network for forecasting financial and economic time series. *Neurocomputing*, 10, 215 – 236.

Kartalopoulos, S. V. (1996). *Understanding neural networks and fuzzy logic: Basic concepts and applications.* New York: IEEE Press.

Khan, U. A. & Gour, B. (2013). Stock market trends prediction using neural network based hybrid, model. *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR),* 3 (1), 11 – 18.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. *Proceedings of the International Joint Conference on Neural Networks*, *IEEE, San Diego, CA.* pp. 1–6.

Jason, E. K. (1998). *Forecasting financial markets using neural networks: An analysis of methods and accuracy.* Naval Postgraduate School Monterey, California.

Mackinlay, C. A. & Lo, W. A. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *Review of Financial Studies,* 1, 41 – 66.

Refenes, P. A., Zapranis, D. A. & Francis, G. (1995). Modeling stock returns in the framework of APT: A comparative study with regression models. *Neural Networks in the Capital Markets*, 101 – 125.

Vasant, D., & Roger, S. (1996). *Intelligent decision support methods: The science of knowledge work.* New Jersey: Prentice Hall Business Publishing, Upper Saddle River,

Ward, S. & Marge, S. (1995). *The neural network financial wizards, technical analysis of stocks & commodities.* Reprinted, Technical Analysis Inc., Seattle, Washington

Zahedi, F. (1993). *Intelligent systems for business: Expert systems with neural networks.* Belmont, USA: Wadsworth,10 - 11.

www.nse.com.ng/.../PRESCO%20DEC%202012%20AUDITED.pdf