Nigerian Journal of Mathematics and Applications
Volume **31**, (2021), 234 − 251. *Printed by Unilorin press*
©*Nig. J. Math. Appl.* *http://www.njmaman.com*

Analysis and Simulation of a Simplified Neural Network Model for Price Prediction in a Relation to the Inflationary Trend in Nigeria Markets

S. O. Momoh*, H. O. Edogbanya and Z. O. Usman

Abstract

This paper provides the concept of neural network model for price prediction is and what it can be used for as well as to solve a real data classification problem using the methods of Artificial Neural Network (ANN). However, in the application of ANN for predicting the financial market, the use of technical analysis variables for price prediction is predominant. In this study, it was shown that the Nigerian market is not efficient but chaotic. The simulation results and price forecast show that it is possible to consistently earn good return on investments on the Nigerian market using private information from an Artificial Neural Network indicator.

1. Introduction

In Nigerian market the prediction market involves a collection of people speculating on a variety of events, exchange, averages, election results, commodity price, quarterly sales results or even such things as gross move i.e. receipts. They are also known as information markets, or decision markets. A prediction market, contract trades between 0 and 100%. Despite significant advances in certain areas of mathematical finances, there is still no formal model that describes the mechanics of market [11] what has rather evolved are two distinct investment

Received: 03/09/2021, Accepted: 22/10/2021, Revised: 30/11/2021. * Corresponding author.
2015 *Mathematics Subject Classification.* 00A72 & 92B20.
Keywords and phrases. Neural Network, Model, Prediction, Sample, Market
Department Of Mathematics, Federal University Lokoja, Kogi State, Nigeria
E-mail of the corresponding author: momohsheidu23@gmail.com
ORCID of the corresponding author: 0000-0002-6664-5069

paradigms based on the efficiency theory and chaotic theory. Since the early work of Fama [4], it is common to discuss three forms of efficiency when examining the Efficiency Market Hypothesis (EMH). EMH States that:

(1) A market is weak form efficient, if it is not possible to consistently earn excess returns using past prices and returns.
(2) A market is 'semi – strong efficient' if it is not possible to consistently earn excess returns using public information including private information [4].

Chaos is a non-linear process which appears to be random, various theoretical tests has been developed to prove that a system which is chaotic has chaos in its time series. Chaos theory is an attempt to show that order does exist in apparent randomness. Chaos theory is a relatively new approach to modeling nonlinear dynamic systems and random process. By implying that Nigerian market is a chaotic and not simply random, chaos theory contradicts the market efficiency theory. Artificial Neural Network have the ability to capture both the deterministic, nonlinear random patterns in the Nigerian market time series data due largely to exploit in the learning capacity.
It is therefore, my goal in this study to exploit the learning capability of a feed forward neural network with back propagation for the prediction of daily prices of securities quoted in the Nigerian stock exchange market.
Mathematical projections to further show that the Nigerian market is not efficient but chaotic is done.
The Nigeria capital market was established in 1961 to provide and sustain the capital requirements of the economy. The mechanism for mobilizing long term funds for investments purposes is the stock Exchange. Between 1961 and 1997, the stock exchange operated a manual call over system with its inherent problems, which could be summarized as undue long transaction cycles, minimal transparency and therefore a general lack of confidence in the system [1]. Information and communication technology (ICT) transformation in the Nigerian Capital market began in 1997 with establishment of the automated Trading system (ACT). This is a system that enables dealers trade through a network of computers connected to a server using the queuing system.

1.1. **Classification and prediction.** Classification and predictions are two forms of data analysis that can be use to extract models describing important data classes to predict future data trends. Data classification is a two-step process: In the first step, a classifier is built describing a predetermined set of data classes or concepts. This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or learning from a training set. This first step of the classification process can also be viewed on the learning of a mapping or function, $y = f(x)$.In the second step, the model is used for classification. First, the predictive accuracy of the classifier is estimated. Data prediction is a

two-step process, similar to that of data classification. However, for prediction, we lose the terminology of class label attribute for which values are being predicted is continuous-valued (ordered) rather than categorical (discrete-valued and unordered). Prediction can also be viewed as a mapping or function, $y = f(x)$ where x is the input and output y is a continuous value.
There is also multilayer perception (MCP) as one of the most widely implemented neural network topologies. In terms of mapping abilities, the MCP is believed to be capable of approximating arbitrary functions [10] this has been important in the study of nonlinear dynamics, and other function mapping problems. Two important characteristic of the multilayer preceptor are its nonlinear processing elements (PE) which has a nonlinearity that must be smooth (the logistic function and the hyperbolic tangent are the most widely used) ; and their massive interconnectivity, i.e. any element of a given layer feeds all the elements of the next layer. For further work, see [3], [5], [7], [8], [9], [13] & [14].

2. Methodology

In this part, we look at the models formulation by considering the major concept, thus we explain First Feed Forward Artificial Neural Network (FFANN) and Artificial Neural Network (ANN).

2.1. **Feed Forward Artificial Neural Network (FFANN).** FFANN is a category of ANNS that collect historical data from input node and flow to a single direction out node where result is collected. Feed forward back propagation neural networks introduced hidden layer which made it possible to solve nonlinear problems. Feed forward networks is required to be trained with historical data which target value is known using any suitable algorithms, in so doing, the network acquires new knowledge base on pattern learn from historical data and be able to generalized on data it has never seen called “fesr set” [6].

2.2. **Artificial Neural Network (ANN).** Artificial Neural Network (ANN) is defined as an information processing paradigms inspired by the methods by which the mammalian brain process information [2]. These are assortments of mathematical models that initiate some of the observed phenomenon in a biological nervous system, most importantly adaptive biological learning.

2.2.1. *How does the neural network works.* In 1943, Warren McCullock and Walter Pitts proposed a very simple idea that is still the basis for most artificial neural network. The neuron computes the weighted sum of the input signal and compare the result with a threshold value, if the if net input is less than the threshold; the neuron output is -1. But if the net output greater than or equal to the threshold, the neuron becomes activated and his output attains a value +1. (McCullock and Pitts, 1943).

In other words, the neuron uses the following transfer or activation function (1):

$$X = \sum_{i=1}^{n} x_i w_i, \quad Y = \begin{matrix} +1 & if\ X \geq 0 \\ -1 & if\ X < 0 \end{matrix} \tag{1}$$

where X is the net weighted input to the neuron, x_i is the value of input i,w_i is the weighted input i,n is the number neuron inputs, Y is the output of the neuron. This type of activation function is called a sign function.

(1) Describe how many neurons are to be used and how the neurons are to be connected to form a network (choose the network architecture).
(2) Decide which learning algorithm & use (supervised, unsupervised or reinforcement).
(3) Finally train the neural network operators.

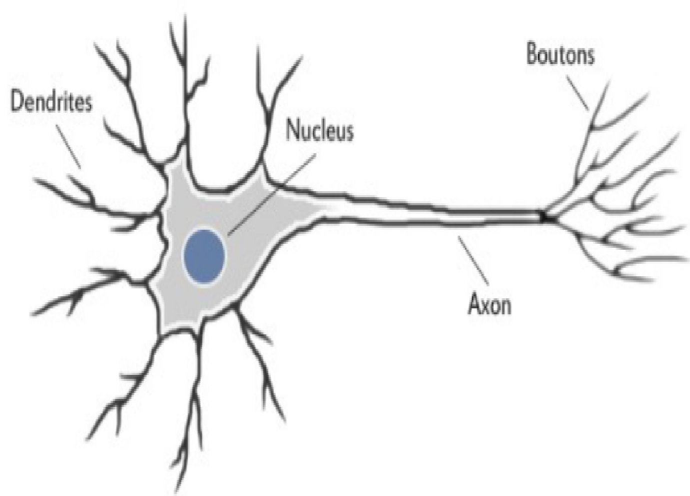

Figure 1: Neural Network

2.3. **The Neural Network Model.** Artificial neural network inspired by biological neural networks, learn from historical cases and make it possible to generate rules automatically. And thus avoid the tedious and expensive process of knowledge acquisition, validation and revision.
ANN is an information processing system that has been developed as generalization of mathematical models of human neural biology (figure1). ANN is composed of units connected by links (direct) and each link has a numerical weight associated with it. Weights are the basic means of long-term memory in ANNs. A Neural Network express the strength or in other words, the importance of each

neuron input. A Neural Network learns through repeated adjustment of these weights.

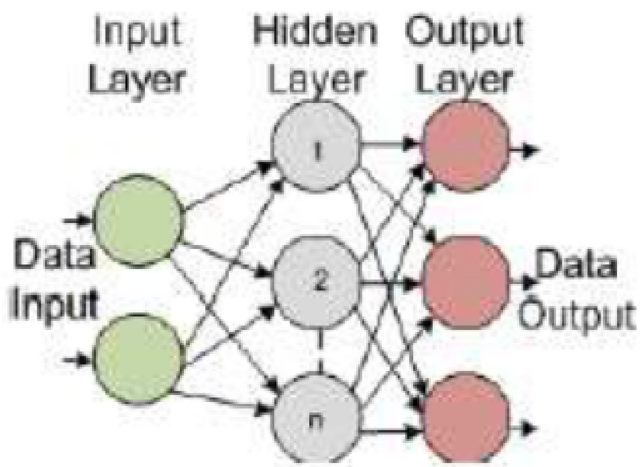

Figure 2: Model of ANN

2.4. **Feed Forward Neural Network with Back Propagation.** A typical ANN is made up of layers and the neurons in the networks are arranged along these layers. The neurons connected to the external environment from input and output layers. The weight are modified to bring the network input or output behavior into line with that of the environment. As shown in the figure above. As shown in (Figure 2), a bias is included with the purpose of setting the actual threshold of the activation function.

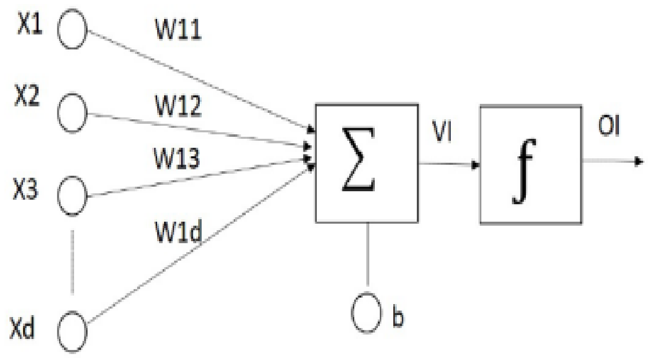

Figure 3: Mathematical Model of ANN

With the purpose of setting the actual threshold of the activation function (2), (3) & (4):

(2) $$V_i = \sum_{j=1}^{d} W_{ij} X_j$$

(3) $$oi = \gamma\,(V_i) = \ \gamma \sum_{j=1}^{d} W_{ij} X_j$$

(4) $$W = \begin{pmatrix} w_{11}w_{12}w_{13}\cdots w_{1d} \\ w_{21}w_{22}w_{23}\cdots w_{2d} \\ w_{31}w_{31}w_{33}\cdots w_{3d} \\ \vdots \\ w_{m1}w_{m2}w_{m3}\cdots w_{md} \end{pmatrix}$$

where γ is the activation function, Xj is the input neuron, Oi is the output of the hidden neuron i and W is the weight matrix. The NN learns by adjusting the weight matrix (Matlab Documentation). The learning function or the activation function that was used is a sigmoid function (5).

(5) $$f\,(x) = \frac{1}{1} + e^{-ax}$$

Because it was found from literature on related problem domain to be most widely used and perform better than other functions such as the unit step function, binary transfer function etc.
The Figure 3 depicts the basic model used, the input layer consist of P processing entities of Km$(m = 1, 2, 3, \cdots, p)$ and three output layer $y_{1,}$, $y_{2,}, y_3$ the output of the model could be presented in the functional form as:

$$y = g\left[\sum_{0}^{d} W_{my} g(\sum_{i=1}^{N} W_{im} X_i)\right] \tag{6}$$

2.4.1. *Training with the Backdrop trainer.* The back propagation algorithm was first developed in the 1970's, gaining more population in 1986 following a famous study conducted by Rumelhart, Hinton and Willliams. Since then, it has become one of the most important machine learning Algorithms for neural networks.
Training the Neural Network involves the algorithm for finding suitable weights and/or other network parameters. It can be viewed as a nonlinear optimization problem for finding a set of network parameters that minimizes the cost function for specific examples. This kind of parameters estimation is also called Training Algorithm. Just as one can operate a **flashlight** without knowing how the electronics inside it work.
The aim of this process is to give an intuitive, if not rigor understanding of how neural networks problem are solved.

2.4.2. *How to code a Neural Network.*

(1) Prepare your input/target.
(2) Decide your NN architecture.
(3) Do a feed-forward propagation.
(4) Build error.
(5) Perform a backward-propagation one Error.
(6) Update parameter.
(7) Loop again.

2.4.3. *Training times.* [10] present below a set of heuristics that will help decrease the training times and, in general, produce better performance:

(1) Normalizing training data.
(2) Using the tanh nonlinearity instead of the logistic function.
(3) Normalizing the desired signal to be just below the output nonlinearity rail voltages (when using the tanh, the desired signals of +/ 0.9 instead of +/1).
(4) Setting the step size higher towards the input (i.e. for a one hidden layer MLP, set the step size at 0.05 in the synapse between the input and hidden layer, and 0.01 in the synapse between the hidden and output layer). the

nets weights in the linear region of the nonlinearity (dividing the standard deviation of the random noise source by the fan-in of each PE).

(5) Using more sophisticated learning methods (quick prop or delta bar delta).

(6) Always having more training in the test set to be limited by the relation NW/e, where N is the number of training epochs, W the number of weights and e the performance error. The MLP should be trained until the mean square error is less than e/2.

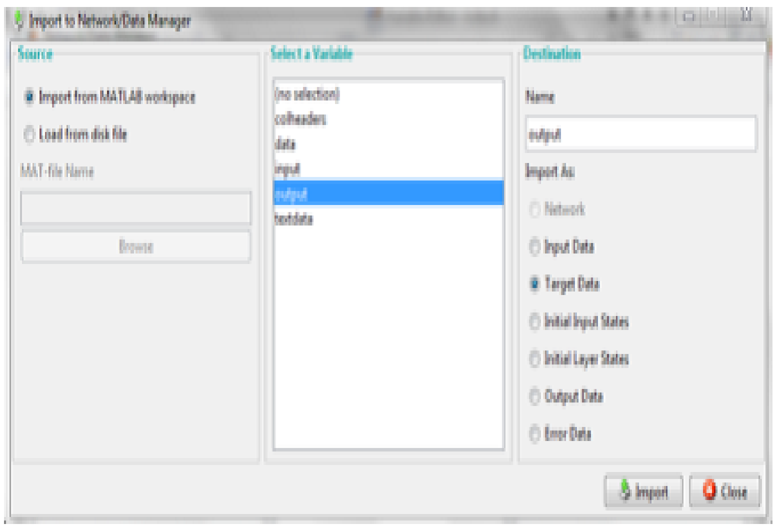

Figure 4: First stage of training(Input data)

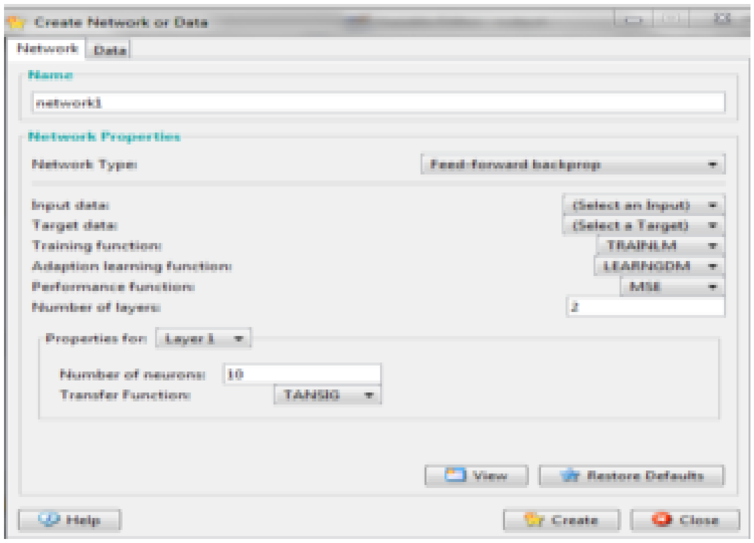

Figure 5: Second stage of training (Modeled data split create network)

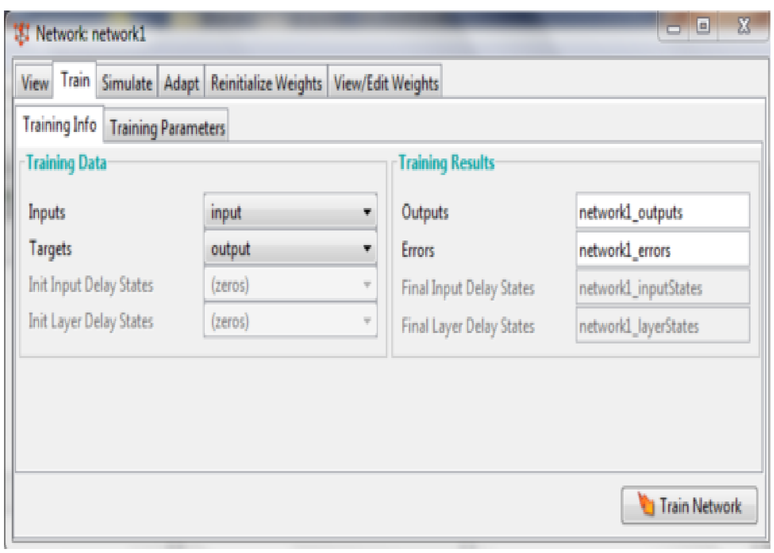

Figure 6: Third stage of training (Parameter and information)

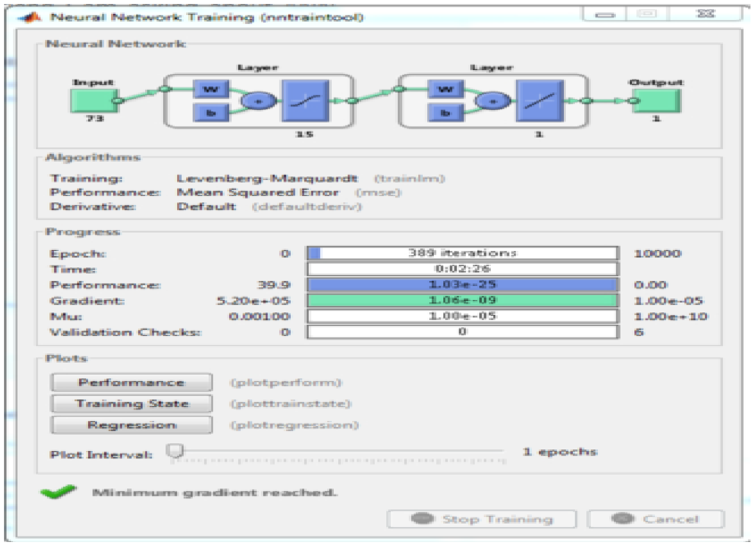

Figure 7: fourth stage of training(performance)

2.4.4. *Testing Phase.* Once the network has been trained after modeling the next step is to test for the input and output data to get the predicted data, and blow here is the example and the graph of the output and predicted data.

3. Main Result

Below here is the set of sample data gotten from the NIGERIAN STOCK EXCHANGE and the source is online following its website. from this data, I am predicting the closed prices and the modeled Data for the close price are shown in tables and graph of predictions below.

The training parameters were set as follows: learning rate=0.01, momentum term=0.9, and epoch size=1000, 2000, 5000. Finally, the network was tested with the data set to estimate its generalization ability. To determine the best performing model, simulation experiment was run on different ANN model configurations. Both training and testing data were carefully selected. However, the training was not done with test data. The model was trained with1000,2000 and 5000 epochs, respectively, while the mean squared error (MSE) for each training session of the different network structure was noted. The tables and graphs are the results of modeling prices of Nigeria market by using the training parameters for the ANN model.

3.1. **Sample of Daily Market Price.**

Date	Close	Volume	Open	High	Low
4/25/2018	160.66	3,176,260	168.95	169.6123	158.171
4/24/2018	160.66	3175980	169.19	169.6123	158.171
4/23/2018	167.51	1926393	169.92	170.23	166.62
4/20/2018	168.9	1860959	171.69	172.55	167.93
4/19/2018	172.99	1524237	173.29	174.36	170.885
4/18/2018	174.3	1658872	175.39	175.5399	171.21
4/17/2018	174.54	1700487	171	175.03	171
4/16/2018	169.45	1245694	168.51	170.72	167.27
4/13/2018	166.9	1342027	170.08	171.3	164.7889
4/12/2018	168.57	1554897	166.73	170.17	166.54
4/11/2018	166.5	2252560	165.08	168.24	164.47
4/10/2018	166.58	1774599	166.25	167.43	162.13
4/9/2018	162.89	1237068	164.56	166.345	162.65
4/6/2018	162.94	1177499	163.96	166.88	161.24
4/5/2018	165.77	1759255	166.78	168.16	164.621
4/4/2018	164.69	2058080	156.5	164.99	156
4/3/2018	161.77	1817910	166.6	167.97	160.07
4/2/2018	161.62	1900268	164.44	165.845	159.24
3/29/2018	165.45	2261467	161.6	167.46	160.27
3/28/2018	159.68	3078929	163	164.015	157.16
3/27/2018	162.82	2037557	173.21	173.38	161.36
3/26/2018	171.45	2042330	168.55	172.04	167.815
3/23/2018	164.77	1999689	168.98	170.63	164.52
3/22/2018	168.76	1633150	171.34	172.83	168
3/21/2018	173.19	1900401	171.99	176.195	171.5
3/20/2018	172.11	3001149	169.87	173.3702	169.23
3/19/2018	169.81	1974798	169.8	172.7799	167.42
3/16/2018	169.98	2442405	174.82	175	169.07
3/15/2018	173.83	1657275	175.96	176	170.91
3/14/2018	175.34	1402759	175.37	176.56	173.49
3/13/2018	173.96	1841753	175.99	175.99	172.17
3/12/2018	174.43	2119941	172.76	175.9	171.56

Date	Close	Volume	Open	High	Low
3/9/2018	172.76	2359876	174.34	174.99	172.46
3/8/2018	172.99	1841917	174.68	175.65	172.4
3/7/2018	173.06	1654484	169.45	174.19	169.38
3/6/2018	171.07	2734795	169.4	171.91	168.41
3/5/2018	168.3	2046948	164.76	169.1875	163.81
3/2/2018	166.67	2189762	158.65	167.1	157.525
3/1/2018	159.57	2001857	162.62	164.26	157.875
2/28/2018	161.01	1934671	163.27	165.68	160.865
2/27/2018	163.34	1995921	161.27	164.235	160.75
2/26/2018	160.57	1205066	160.8	162.64	159.66
2/23/2018	160.4	1035524	156.86	160.63	156.86
2/22/2018	156.05	1170255	157.78	159.31	155.693
2/21/2018	157.58	1924121	156.36	160.73	156.31
2/20/2018	156.76	1755310	152.9	158.13	152.03
2/16/2018	154.03	1904698	151.93	156.19	151.18
2/15/2018	152.54	1874684	148.95	152.79	147.8
2/14/2018	147.74	1852112	147.27	149	146.3
2/13/2018	147.55	1431527	146.04	148.32	145.57
2/12/2018	147.24	1614798	146.97	148.625	145.235
2/9/2018	145.16	2836220	142.14	146.765	137.6
2/8/2018	140.1	3737168	153	153.5	140.1
2/7/2018	153.17	2462816	148.62	154.03	148.07
2/6/2018	149.02	2963103	144.91	149.379	143.49
2/5/2018	147.39	3788712	147.29	157	145.43
2/2/2018	149.76	2826713	150.51	151.94	147.33
2/1/2018	152.81	4228906	151.76	156.545	151.33
1/31/2018	148.87	2872910	148.68	150.1	147.76
1/30/2018	146.59	2369521	146.07	147.44	145.63
1/29/2018	147.09	2414839	149.59	150.85	146.83
1/26/2018	149.74	1472513	147.16	149.82	146.38
1/25/2018	145.54	1220738	146.89	146.9	145.42
1/24/2018	145.44	1432496	144.42	146.67	144.37

3.2. **Sample of Daily Market Price Modeled Data.**

INPUT1	INPUT2	INPUT3	OUTPUT1	OUTPUT2
145.44	145.54	149.74	147.09	146.59
149.74	147.09	146.59	148.87	152.81
146.59	148.87	152.81	149.76	147.39
152.81	149.76	147.39	149.02	153.17
147.39	149.02	153.17	140.1	145.16
153.17	140.1	145.16	147.24	147.55
145.16	147.24	147.55	147.74	152.54
147.55	147.74	152.54	154.03	156.76
152.54	154.03	156.76	157.58	156.05
156.76	157.58	156.05	160.4	160.57
156.05	160.4	160.57	163.34	161.01
160.57	163.34	161.01	173.83	169.98
161.01	173.83	169.98	169.81	172.11
169.98	169.81	172.11	173.19	168.76
172.11	173.19	168.76	164.77	171.45
168.76	164.77	171.45	162.82	159.68
171.45	162.82	159.68	165.45	161.62
159.68	165.45	161.62	161.77	164.69
161.62	161.77	164.69	165.77	162.94
164.69	165.77	162.94	162.89	166.58
162.94	162.89	166.58	166.5	168.57
166.58	166.5	168.57	166.9	169.45
168.57	166.9	169.45	174.54	174.3
169.45	174.54	174.3	172.99	168.9
174.3	172.99	168.9	167.51	160.66

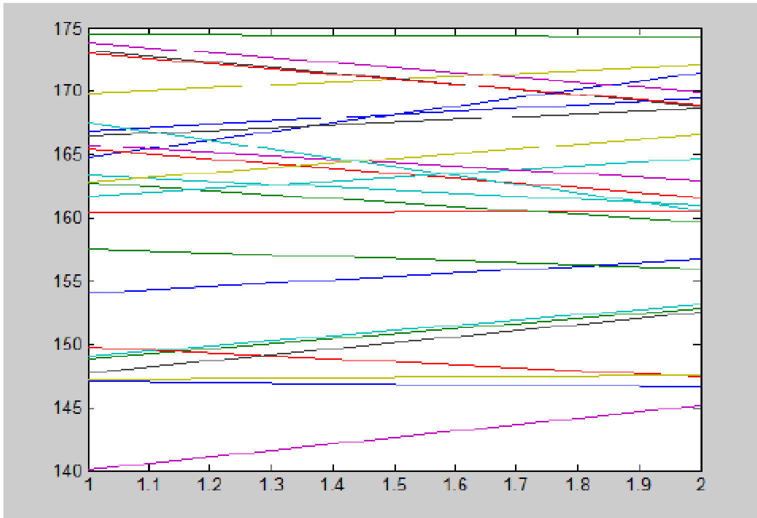

Figure 8: Graph of modelled output (MATLAB result from the model training algorithms)

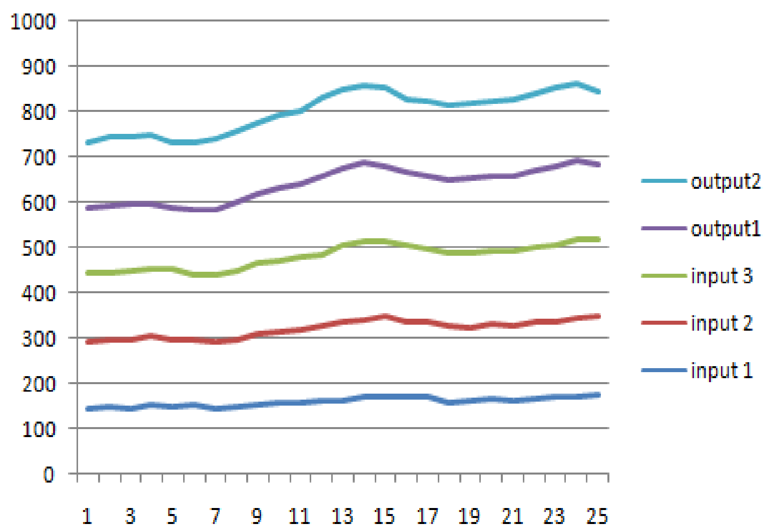

Figure 9: Graph of modeled data (Year against input/output)

3.3. **Sample of Daily Market Price. Table of Predicted Historical Data**

OUTPUT1	OUTPUT2
147.0690235	148.3514925
149.5607892	151.8618819
148.1178697	148.7250424
153.6024524	154.3039123
148.6022922	149.0483526
149.4708231	148.7422922
147.7690336	150.1459605
148.2672531	148.8362841
154.3398803	152.7398081
160.7415647	159.6256108
161.3587079	156.5505228
167.9661528	163.8407394
155.3738049	163.5771922
170.3438636	167.8533986
168.0247674	168.2058175
166.740702	167.3213272
177.1162806	180.0484551
163.9083573	165.4540878
163.8767523	159.8380313
169.9505414	172.3317053
160.8475236	162.4313727
165.1883537	167.6966965
168.1893587	168.7421109
170.9099959	166.9501045
168.3465484	161.2993598

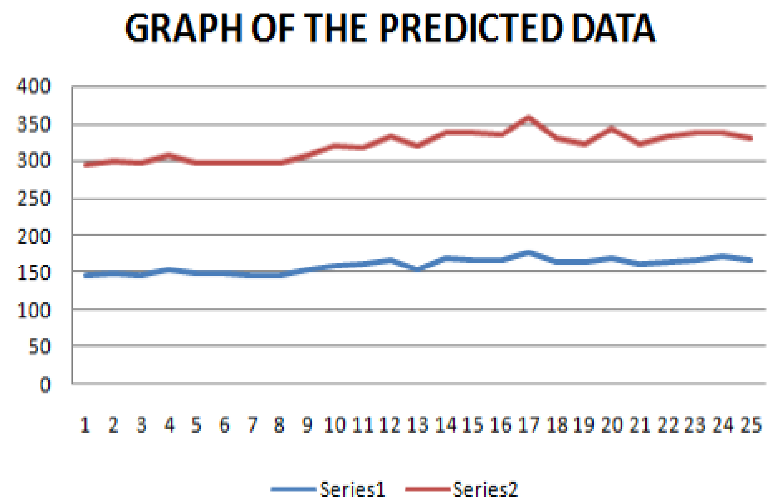

Figure 10: Graph of predicted data (years against input/output)

3.4. **Sample of Daily Market-Price. Table of Predicted Historical Data**

Output 1	Y1	Error	Output 2	Y2	Error
147.09	147.0690235	0.0209765	146.59	148.3514925	-1.7614925
148.87	149.5607892	-0.6907892	152.81	151.8618819	0.9481181
149.76	148.1178697	1.6421303	147.39	148.7250424	-1.3350424
149.02	153.6024524	-4.5824524	153.17	154.3039123	-1.1339123
140.1	148.6022922	-8.5022922	145.16	149.0483526	-3.8883526
147.24	149.4708231	-2.2308231	147.55	148.7422922	-1.1922922
147.74	147.7690336	-0.0290336	152.54	150.1459605	2.3940395
154.03	148.2672531	5.7627469	156.76	148.8362841	7.9237159
157.58	154.3398803	3.2401197	156.05	152.7398081	3.3101919
160.4	160.7415647	-0.3415647	160.57	159.6256108	0.9443892
163.34	161.3587079	1.9812921	161.01	156.5505228	4.459472
173.83	167.9661528	5.8638472	169.98	163.8407394	6.1392606
169.81	155.3738049	14.4361951	172.11	163.5771922	8.5328078
173.19	170.3438636	2.8461364	168.76	167.8533986	0.9066014
164.77	168.0247674	-3.2547674	171.45	168.2058175	3.2441825
162.82	166.740702	-3.920702	159.68	167.3213272	-7.6413272
165.45	177.1162806	-11.6662806	161.62	180.0484551	-18.4284551

Output1	Y1	Output2	Y2
161.77	163.9083573	164.69	165.4540878
165.77	163.8767523	162.94	159.8380313
162.89	169.9505414	166.58	172.3317053
166.5	160.8475236	168.57	162.4313727
166.9	165.1883537	169.45	167.6966965
174.54	168.1893587	174.3	168.7421109
172.99	170.9099959	168.9	166.9501045
167.51	168.3465484	160.66	161.2993598

In statistic, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity)measures the average of the squares of the errors, that is the average squared difference between the estimated value and what is estimated. The MSE is a measure of the quality of an estimator and it is always non negative and values closer to zero are better. The MSE access the quality of an estimator (that is a mathematical function mapping a sample of data to an estimate of a parameter of the population from which the data is sampled) or a predictor. The Mean Squared Error is given as:

$$MSE = \sum_{n=1}^{N} inf, n \in \mathbb{N} \tag{7}$$

Conclusion: Using Developed System: Matlab to predict the future market price values with Feed forward Neural Networks some analyses were done to know the performance of the Back-propagation Algorithm. Using the previous historical data of stock market which involves 3 inputs, of trying to predict market price for future of 2 days of January 2018 from Back-propagation algorithm, we are now able to compare the predicted values with the real values. Since the first simulation 3 involves 3 inputs, then the second simulation involves 2 inputs for the prediction of A Day ahead prediction. The input historical data is from January to February 2018. We conclude that: Artificial neural networks (ANN) are an appropriate technique for predicting prices and other market variables and also testing of ANN systems had tended to be based on predicting than comparing the prediction with the real data that is already known (testing data). The paper investigated the training data and transfer functions required to produce efficient neural network architecture for predicting Nigerian market price. The stock market dataset considered for this analysis was for the market prices movement. Once a neural network is exposed to a good training dataset, it can be used to predict complex Nigerian market price. In this analysis, the Neural Network was used to predict market prices independent of each other. This is why it is profitable to use neural networks to predict the behavior of financial instruments such as stocks. From our simulation and projection results it is recommended that: Sensible profit can be obtained in stock markets (especially [12]) trading with ANNs.

Acknowledgement: The authors are very grateful to Federal University Lokoja for the supports they received during the compilation of this work and to anonymous referees for their careful reading of the paper and helpful suggestions.

Competing interests: The authors declare that they have no competing interests.

Funding: We declare that no funding from external bodies was received for this research.

References

[1] Adebiyi M. A., Adenuga A. O., Abeng M. O. & Omanukwue P. N. (2009). Oil price shocks, exchange rate and stock market behaviour: empirical evidence from Nigeria. *African Econometric Society Conference Proceedings*. Sheraton Hotels and Towers, Abuja.

[2] Caiamiello (1999). Prediction with genetic-algorithm designed neural networks: an empirical comparison with modern statistical models. *Computational Intelligence*. **15** (3), 171-184.

[3] Elliott D. L. (1993). A better activation function for artificial neural net-works. *ISR technical report*. **TR9**, 3-8.

[4] Fama E. F. (1969). The adjustment of stock prices to new information. *International Economic Review*.

[5] Fatma K. & Bouri A. (2009). Fuzzy logic and investment strategy. *Global Economy and Finance Journal*.

[6] Hu X., Wang H., Xu L. & Li H. (2008). Predicting stock index increments by neural networks: The role of trading volume under different horizons. *Expert Systems with Applications*. **34**, 3043-3054.

[7] Kumar P. R. & Ravi V. (2007). Prediction in banks and firms via statistical and intelligent techniques- A review. *European Journal of Operational Research*. **180**, 128.

[8] Lippman R. (1987). An introduction to computing with neural nets. *IEEE Transactions ASSP Magazine*. **4**, 4-22.

[9] McCulloch W. S. & Pitts W. (1943). A logical calculus of the ideas immanent inner activity. *Bulletin of Mathematical Biophysics*. **5**, 115-137.

[10] Principe J. C., Euliano N. R. & Lefebvre W. C. (1999). *Neural and Adaptive Systems: Fundamentals through Simulations*. New York, USA: John Wiley & Sons.

[11] Ravichandran K. S. (2005). estimation of return on investment in share market through ANN. *Journal of Theoretical and Applied Information Technology (JATIT)*. India

[12] Nigerian Stock Market (2016). *Predicting the future survey of research papers and patents-think path-Appreciating innovation (n.d)*. Retrieved April, 2016, fromhttp://www.thinkpatori.com/2013/10/stock-market-prediction predicting.html.

[13] Simon H. (1998). *NN A comprehensive foundation, second edition*. Prentice. Hall 161-173.

[14] Zhang Y. and Wan X. (2007). Statistical fuzzy interval neural networks for currency exchange rate time series prediction. *Applied Soft Computing*. 7 pages.