TRANSLATED NIGERIA STOCK MARKET PRICES USING ARTIFICIAL NEURAL NETWORK FOR EFFECTIVE PREDICTION

Akinwale Adio T, Arogundade O.T. and Adekoya Adebayo F.
Department of Computer Science, University of Agriculture, Abeokuta, Nigeria
E-mail: atakinwale@yahoo.com, arogundadeo@acm.org, adekoyaaf@unaab.edu.ng

ABSTRACT

This paper used error back propagation algorithm and regression analysis to analyze and predict untranslated and translated Nigeria Stock Market Price (NSMP). Nigeria stock market prices were collected for the periods of seven hundred and twenty days and grouped into untranslated and translated train, validation and test data. A zero mean unit variance transformation was used to normalize the input variables in order to allow the same range which makes them to differ by order of magnitude. A *5-j-1* network topology was adopted because of five input variables in which variable *j* was determined by the number of hidden neurons during network selection. The untranslated and translated data served as input into the error back propagation algorithm and regression model which were written in Java Programming Language. The results of both untranslated and translated statements were analyzed and compared. The performance of translated NSMP using regression analysis or error back propagation was more superior than untranslated NSMP. The results also showed that percentage prediction accuracy of error back propagation model on untranslated NSMP ranged for 11.3% while 2.7% was for translated NSMP. The 2.7% percent accuracy as against 11.3% indicates the relative stability of translated NSMP prediction as against untranslated NSMP. The mean relative percentage error was very low in all hidden topologies of error back propagation of translated NSMP than untranslated NSMP. This indicates that translated NSMP prediction approach was superior to untranslated NSMP predicition.

Keywords: *Neural network, error back propagation, regression analysis, machine, learning, stock predictions, translated stock price*

1. INTRODUCTION:

Prediction of stocks is generally believed to be a very difficult task if a nation economy is being effected by inflation and fluctuation of exchange rate. The use of untranslated raw data for stock market prediction has been widely established using different methods such as fuzzy logic, regression analysis, artificial neural network, rough set, genetic algorithms, etc. The result always favours one of the methods of prediction without putting into consideration some economic factors such as inflation, fluctuation of exchange rate, global economic recession, etc.

Effective stock prediction is usually an investor's delight as investors wish to be sure whether the stock they are putting their money on, is able to pay back its liabilities, has enough working capital and is generally in good state of financial health. It is opined that a potential investor's need would be met when the information that could enable him make good and wise investment decision as regards stocks to purchase or invest in is easily accessible, available, understandable and reasonably rational [26]. With favourable investment climate in Nigeria since the advent of democratic government in 1999, investors are making great inroad into the country through foreign direct investment (*FDI*) and active participation in stocks trading in the Nigerian capital market via the Nigerian Stock Exchange. The Nigerian Stock Exchange market capitalization as at August 2007 stood at ₦8.3 trillion ($69.17billion) [27].

Many studies on stock market prediction using ANNs or statistical methods were performed on untranslated raw data. These data might have been affected by inflation or fluctuation of exchange rates especially in developed countries such as Nigeria. This study therefore, investigates the effect of daily stock price in current exchange of untranslated and translated Nigerian Stock Market Price (NSMP) using both error back propagation and statistical regression model. Each of the stock price listed daily on the Nigerian Stock Exchange were converted from their Nigerian Naira value (untranslated NSMP statement) to United State of America Dollar value (translated NSMP statement). This was done for seven hundred and twenty days using the prevailing daily currency exchange (Naira – USD) rate. This in effect will allow foreign investors to use either translated or untranslated report to effectively predict Nigerian stock market prices and thereafter take wise investment decisions. The result obtained thus fulfills a core objective of the investors' information component of the Investment Climate Program (*ICP*) being initiated by the Nigerian government in conjunction with World Bank [27].

2. LITERATURE REVIEWS

Since the early 90's when the first practically usable Artificial Neural Network (ANN) types emerged, it had been rapidly used in different fields of sciences. The areas in which ANNs have been successful implemented are speech, handwritten character, finger print and electrical recognition.[1]. Other areas are prediction of bank failure, stock market performance and pattern recognition [2]. The areas of application include system identification and control (vehicle control, process control), game-playing and decision making (chess, racing), medical diagnosis, financial application and data mining. A large number of studies have been reported in the literature with reference to the use of ANN in modeling stock prices in advanced countries. E. Birgul et al used ANN and Moving Average to predict Turkey Stock Market [3]. S.I Wu and H. Zheng developed recurrent neural networks to forecast the daily closing prices of these stock indexes [4]. K. Kim and W. Lee also used ANN with optimal feature transformation to predict stock market trends. In their study, a genetic algorithm (GA) was incorporated to improve the learning and generalizability of ANNs for stock market prediction. The results obtained by a feature transformation method using the GA when compared to other two feature transformation ANN methods without GA showed that the performance of the improved ANN-GA model was better [5]. Cirna, et al introduced a genetic programming technique (called Multi-Expression programming) for the prediction of two stock indices and compared the performance with an artificial neural network trained using Levenberg-Marquardt algorithm, support vector machine and Takagi-Sugeno neuro-fuzzy model [6]. The use of ANN in stock prediction comes in because they can learn to detect complex pattern in data. As ANNs are essentially non-linear statistical models, their accuracy and predictive capabilities can be both analytically and empirically tested. The performance of ANNs has been extensively compared to that of various statistical methods within the areas of prediction [7]. In particular, a fair amount of literature has been generated on the use of ANNs in time series forecasting. Lapedes and Farber concluded that basic neural network substantially outperform conventional statistical methods [8]. J. M. Hutchinson used time series analysis to measure stock market performance [9]. It was noted that researchers on stock price prediction used untranslated data.

Regression Analysis

This method represents the dependent variable, y_i, as a linear function of one independent variable, x_i, subject to a random error, u_i

$$y_i = \beta_o + \beta x_i + u_i$$

The error terms u_i is assumed to be uncorrelated with itself across observations

$$(\in (u_i u_j) = 0, i \neq j)$$

The purpose of prediction is to determine regression coefficients $\hat{\beta}_0$ *and* $\hat{\beta}_1$ as follows:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x$$

We define the predicted error with each pair of stock prices as

$$\hat{u}_i = y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$$

The basic technique for determining the coefficients $\hat{\beta}_0 + \hat{\beta}_1$ is ordinary least square. Values for $\hat{\beta}_0$ *and* $\hat{\beta}_1$ are chosen so as to minimize the sum of square residuals (SSR) as follows:

$$SSR = \sum_{i=1}^{n} (\hat{u})_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

To find the partial derivatives of SSR, we have the following:

$$\frac{\partial SSR}{\partial \hat{\beta}_0} = -2\sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \qquad \text{------(1)}$$

$$\frac{\partial SSR}{\partial \hat{\beta}_1} = -2\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \quad \text{--------------------(2)}$$

Equation 1 implies that

$$\sum_{i=1}^{n} y_i - n\hat{\beta}_0 - \hat{\beta}_1 \sum_{i=1}^{n} x_i = 0$$

$$\Rightarrow \hat{\beta}_0 = \bar{y} - \hat{\beta}\bar{x} \quad \text{--------(3)}$$

While equation 2 implies that

$$\sum_{i=1}^{n} x_i y_i - \hat{\beta}_0 \sum_{i=1}^{n} x_i - \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 = 0 \quad \text{------------(4)}$$

we can substitute for $\hat{\beta}_0$ in equation 4 and 3

This yields

$$\sum_{i=1}^{n} x_i y_i - (\bar{y} - \hat{\beta}_1 \bar{x})\sum_{i=1}^{n} x_i - \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 = 0$$

$$\Rightarrow \sum_{i=1}^{n} x_i y_i - \bar{y}\sum_{i=1}^{n} x_i - \hat{\beta}_1 (\sum_{i=1}^{n} x_i^2 - \bar{x}\sum_{i=1}^{n} x_i) = 0$$

$$\Rightarrow \hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \bar{y}\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2 - \bar{x}\sum_{i=1}^{n} x_i} \quad \text{---------------------(5)}$$

Equation 3 and 4 can be used to generate the regression coefficients. We use equation 5 to find $\hat{\beta}_1$ and equation 3 to find $\hat{\beta}_0$

There is no guarantee that $\hat{\beta}_1$ and $\hat{\beta}_0$ correspond exactly with the unknown parameters β_0 *and* β_1

The degree of freedom is equal to n - 2. The predicted standard error of the regression $(\hat{\sigma}_{\in}) = \sqrt{\frac{SSR}{n-2}}$ or root mean square error (RMSE)

The predicted equation is

$$R^2 = 1 - \frac{SSR}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \equiv 1 - \frac{SSR}{TSS}$$

where TSS is the total sum of square and $0 \le R^2 \le 1$

The adjusted

$$R^2 = \bar{R}^2 = 1 - \frac{\frac{SSR}{n-k-1}}{\frac{TSS}{n-1}} = 1 - \frac{n-1}{n-k-1}(1 - R^2)$$

where k + 1 represents the number of parameters being predicted.

The regression analysis is extended as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n$$

so as to form neural network analysis. In this case x_i serves as inputs to a node, β_i is taken as weight and β_0 as activation function.

Artificial Neural Network Model

An Artificial Neural Network is a computational model based on biological neural network. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. They are self training that imitate the work of human brain. They also have the ability of error tolerance whereby a neural network trained for a particular problem can recall correct results even if the problem to be solved is not exactly the same as the already learned one [10].

What has attracted most interest in ANN is the possibility of learning. For example, given a specific task to be solved and a class of function F, learning means using a set of observations in order to find

f* ε F → R --- (i)

such that for the optimal solution f*

$C(f^*) \le C(f) \forall f \in F$ ------------------------- (ii)

The cost function (C) is an important concept in learning as it is a measure of how far away; an optimal solution is to the problem at hand. Learning algorithm searches through the solution space in order to find a function that has the smallest possible cost. For an application where the solution is dependent on some data, the cost must necessarily be a function of the observations; otherwise there is nothing to be modeled in relation to the data set. For example, consider the problem of finding the model (f) which minimizes

$C = E[(f(x) - y)^2]$ -------------------------- (iii)

for pairs *(x, y)* drawn from some distribution D. In practical situations there would be only *N* samples from D and which would minimize

$$\hat{C} = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 \text{ ------------------------ (iv)}$$

Thus, the cost is minimized over a sample of the data rather than the true data distribution.

Architecture of this model

The architecture of this model consists of five variable, three intermediate variables and the output. Figure 1 illustrates a schematic diagram of an 5-3-1 topology. The five undependable variables are: x_1 = number of share sold, x_2 = share price, x_3 = number of deals, x_4 = volume of share and x_5 = price earning ratio. The hidden layer of H_1, H_2 and H_3 are intermediate variables which interact by means of weight matrices $W^{(1)}$ and $W^{(2)}$ with adjustable weights. The values of the hidden units are obtained from the formulas

$$H_f = f(\sum\nolimits_k W_{jk}^{(1)} X_k)$$

$$Y_i = f(\sum\nolimits_j W_{ij}^{(2)} H_j)$$

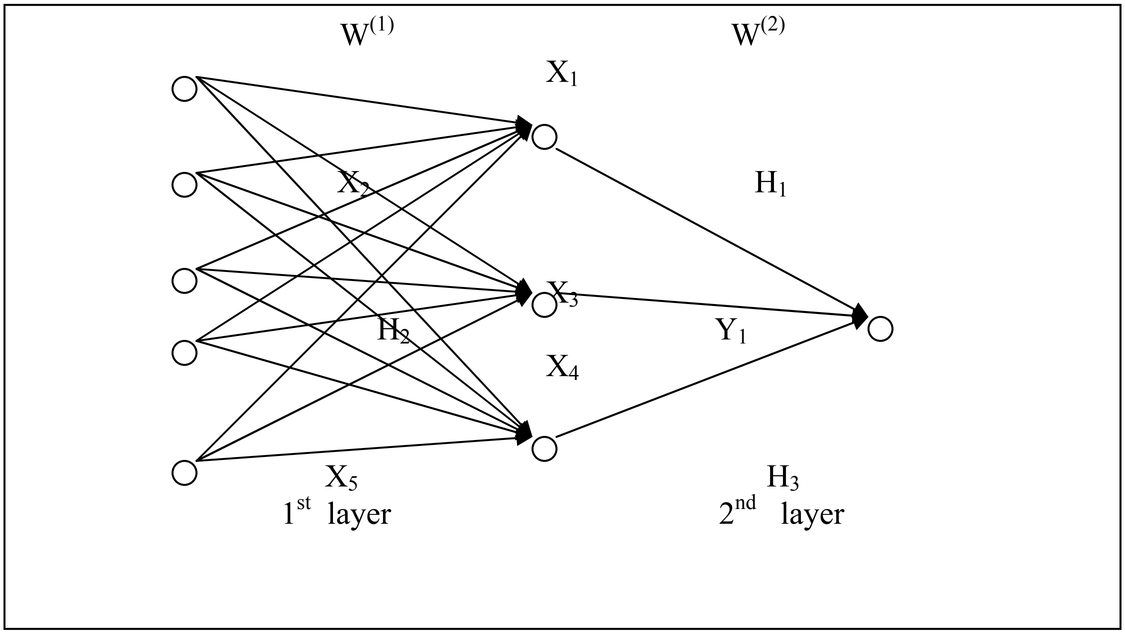

Figure 1: Architecture of ANN's NSMP Prediction

The first weight matrix of $(W_{jk}^{(1)})$ was multiplied by the five independent input variables and then applied an activation function f to each component of the result. The second weight matrix of $(W_{ij}^{(2)})$ was multiplied to the vector H = (H_1, H_2 and H_3) of the hidden unit values and then applied activation function f to each component of the result and obtain output result Y. The activation function f is typically of sigmoid form and may be a logistic function or hyperbolic tangent

$$\text{logistic function} = \text{l(u)} = \frac{1}{1+e^{-u}}$$

$$\text{hyperbolic function} = \text{h(u)} = \frac{e^{u} - e^{-u}}{e^{u} + e^{-u}}$$

The accuracy of the estimated output is improved by an interactive learning process in which the outputs for various input vectors are compared with targets and an average error term E is computed

$$E = \frac{\sum_{i=1}^{n}(y^{(i)} - t^{(i)})^2}{n}$$

The error back propagation algorithm

The following steps represent the procedure involved in the error back propagation algorithm.

Step 1: there is a need to set all weights and threshold levels of the network to random number uniformly distributed inside a small range *[-0.5 to 0.5]*.

Step 2: the activation of back propagation of neural network is effected by applying

$X_1(p), X_2(p), \ldots, X_n(p)$ and desired outputs $y_{d1}, y_{d2}, \ldots, y_{dn}$ ------------ (xi)

Then, the actual outputs y_j *(p)* of the neurons in the hidden layer is computed as

$$y_j(p) = \tanh\left[\sum_{j=1}^{m} x_{jk}(p) * w_{jk}(p) - \theta_k\right]$$

or/and (l(u))---------------- (xii)

where **m** is the number of inputs to neuron **k** in the output layer.

Step 3: the weight training is updated in the back propagation plus the errors associated with the output neurons.

(a) calculate the error gradient for the neurons in the output layer

$$\partial_k(p) = [1 - (y_k(p))^2] * e_k(p)$$ ----------- (xiii)

where $e_k(p) = y_{d,k}(p) - y_k(p)$ ---(xiv)

using the weight corrections $\Delta w_{jk}(p) = y_j(p) * \delta_k(p)$ ------------ (xv)

and the updated weights w_{jk} *(p+1)* at the output neurons

$$w_{jk}(p+1) = w_{jk}(p) + w_{jk}(p)$$ ------------- (xvi)

(b) calculate the error gradient for the neurons in the hidden layer

$$\delta_j(p) = [1 - (y_j(p))^2] * \sum_{k=1}^{m} \delta_k(p) * w_{jk}(p)$$ ---------------- (xvii)

where **m** is the number of neurons in the output layer.

using the weight corrections $\Delta w_{ij}(p) = \alpha * x_i(p) * \delta_j(p)$ --------------- (xviii)

and the updated weight w_{ij} *(p+1)* at the hidden neurons

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$ --------------------------- (xix)

Step 4: there is a need to increase iteration **p** by one, go back to *Step 2* and repeat the process until the selected error criterion is satisfied.

3. IMPLEMENTATION PROCESS

The error back propagation algorithm described previously and regression model were coded in Java Programming Language. Saenz and Pingitore said that the entire process of designing a network, train it, optimizing its performance entail considerable trails and errors. It is not possible to say which amount of neurons in a hidden layer is the best for the problem at hand, trail and errors will give this solution [11]. Therefore, in order to find the most appropriate topology, the topologies between 5-j-1 and 5-n-1 were used where j = 4, 5, 6,…, 13. The activation function f of sigmoid logistic function and hyperbolic tangent were taken for j as shown in table 1.

Hidden Layer for j	10-4-1	10-5-1	10-6-1	10-7-1	10-8-1	10-9-1	10-10-1	10-11-1	10-12-1	10-13-1
1st layer	l(u)	l(u)	l(u)	l(u)	l(u)	l(u)	l(u)	l(u)	l(u)	l(u)
2nd layer	h(u)	h(u)	h(u)	h(u)	h(u)	h(u)	h(u)	h(u)	h(u)	h(u)
3rd layer	l(u)	h(u)	l(u)	h(u)	l(u)	h(u)	l(u)	h(u)	l(u)	h(u)
4th layer	h(u)	l(u)	h(u)	l(u)	h(u)	l(u)	h(u)	l(u)	h(u)	l(u)

Table 1: The use of activation function for different hidden layer

The experiment was carried out to know the analytical basis of hidden layers. The topology behaviour of 5-j-1 where j is either l(u) or h(u) showed no significant different in result in first or second layer. The topology behaviour of 5-j-1 where j is either l(u), h(u) or h(u), l(u) show significant difference in third and fourth layer and both converge at topology 5-10-1 where the training got quite close to one. This indicates that the activation function should not be taken to be the same for all the topologies.

4. DATA COLLECTION

The dataset used for this paper were obtained from Nigerian stock market website www.nigerianstockmarket.com for the period of seven hundred and thirty. These data points were divided into train data, validation data and test data. These data were translated using current exchange rate; thus, we have untranslated and translated train, validation and test data. A zero-mean unit-variance transformation is applied to each data sets of number of shares sold, share price, number of deals, volume of shares and price earning ratio. The following model was used

$$f_{nsp} = f(nrs, sp, nd, vs, per)$$ -------------------- (xxiii)

where: nsp = nigeria stock price, nrs = number of shares sold, *sp* = share price,
nd = number of deals, *vs* = volume of shares and *per* = price earning ratio

The mean μ_i and the variance (δ_i^2) are calculated as follows:

$$\mu_j = \frac{1}{p}\sum_{i=1}^{p} x_{ij}$$ ----------- -------------------------- (xx)

$$\delta_j^2 = \frac{1}{p-1}\sum_{i=1}^{p}(x_{ij}-\mu_j)^2$$ -------------------- (xxi)

Thus,

$$x_{ij}^* = \frac{x_{ij}-\mu_j}{\delta_j}$$ -------------------------------- (xxii)

where x_{ij}^* is the normalized value of x_{ij} and
p is the number of patterns in the trained data set.

5. RESULT ANALYSIS

The untranslated and translated train data served as input into both back propagation and regression methods. These input variables were trained up to 10 neural networks by varying the number of neurons in the hidden layers with the best generalization performance on the untranslated and translated test and validation data. The learning constant and momentum of *0.01* and *0.2* were used respectively. The output values of error back propagation of untranslated Nigerian Stock Statement with the values of sum square error (SSE), root mean square error (RMSE), mean relative percentage error (MRPE) and percentage accuracy are shown in table 2 while the parameters for translated statement are shown in table 3. The output values of regression model of untranslated and translated Nigerian Stock Statement with values of sum square error, root mean square error, and mean relative percentage error and percentage accuracy are also depicted in table 4 and 5. The coefficients of SSE, RMSE, MRPE and percentage accuracy are the measure of the accuracy of prediction of trained error back propagation and regression model. For example, higher SSE values and percentage accuracy indicate better prediction and lower MRPE values also measure accuracy of prediction [12, 13]. Equation (xxiv) was used to calculate the relative error and the results were averaged and factored in percentage.

$$f_{nsp=\frac{(f_{nsp})_{actual}-(f_{nsp})_{predicted}}{(f_{nsp})_{actual}}}$$ -------------- (xxiv)

Hidden Layer	SSE	RMSE	MRPE	%Accuracy
5-4-1	82.1	2.513	3.926	82.1
5-5-1	80.1	2.513	3.911	83.5
5-6-1	79.8	2.511	3.891	84.3
5-7-1	79.7	2.491	3.901	90.8
5-8-1	77.3	2.419	3.829	83.2
5-9-1	78.1	2.391	3.719	91.3
5-10-1	77.9	2.401	3.609	93.4
5-11-1	78.7	2.389	3.826	83.7
5-12-1	79.7	2.411	3.714	83.2
5-13-1	79.8	2.429	3.725	89.4

Table 2: Result of Error back propagation using untranslated NSMP

Comparison of Untranslated and Translated NSMP Prediction

Looking at the percentage accuracy of both untranslated and translated column result of regression model in table 4 and 5, % accuracy of translated method was above 90% while 80% above was untranslated method. The 90% percentage accuracy indicates the performance of translated NSMP prediction over untranslated method. Looking at the tables 2 and 3, the percentage accuracy of untranslated NSMP of *10* hidden layers (5-4-1 to 5-13-1) varies from 82.1 to 93.4 with topology *5-10-1* has the highest percentage accuracy. The percentage accuracy of translated NSMP of *10* hidden layers varies from 96.7 to 99.44 of which topology *5-10-1* has the highest percentage accuracy. Topology 5-10-1 performed the best and thus could be in position for ANN's prediction. Untranslated method has the difference of 11.3% while translated has the difference of 2.74%. This 2.74% difference of translated method as against 11.3% difference of untranslated method indicates the relative stability of translated NSMP prediction.

Hidden Layer	SSE	RMSE	MRPE	%Accuracy
5-4-1	0.002	0.0136	1.826	98.2
5-5-1	0.002	0.0134	1.824	98.8
5-6-1	0.002	0.0132	1.823	96.7
5-7-1	0.002	0.0133	1.729	97.9
5-8-1	0.002	0.0129	1.517	98.6
5-9-1	0.002	0.0128	1.626	99.3
5-10-1	0.002	0.0127	1.416	99.4
5-11-1	0.002	0.0130	1.629	98.9
5-12-1	0.002	0.0132	1.623	97.5
5-13-1	0.002	0.0132	1.672	97.6

Table 3: Result of Error back propagation using translated NSMP

Days	SSE	RMSE	MRPE	%accuracy
30	23.108	1.333	6.375	86.3
90	27.212	1.631	6.852	84.7

Table 4: Result of regression model using untranslated NSMP

Days	SSE	RMSE	MRPE	%accuracy
30	0.001	0.0103	4.241	95.8
90	0.001	0.0104	4.470	95.9

Table 5: Result of Regression model using translated NSMP

The mean relative percentage errors of regression analysis in both untranslated and translated NSMP are higher than mean relative percentage errors of error back propagation model in untranslated and translated NSMP. The mean relative percentage errors of untranslated Nigeria stock price are very high with the range from 3.609 to 3.926 using back propagation method. The mean relative percentage error is very low in all hidden topologies of translated Nigeria stock market price from the range of 1.416 to 1.826 with the topology 5-10-1 has the lowest value. The performance of error back propagation with low error percentage values in translated NSMP is more superior to regression model in translated NSMP as depicted in table 3 and 5 under the field of MRPE. Translated Nigeria stock market price approach has a higher overall percentage accuracy using regression or ANN's method for effective prediction.

6. CONCLUSION

The study was aimed at finding the best method for the prediction of Nigerian stock market price values during the fluctuation of Nigerian currency as against USA dollar. The prevailing current exchange rate for each day was used to convert the value of each Nigerian stock market price into USA dollar. The translated NSMP from untranslated values were performed and served as input variables to application of both regression and error propagation. Based on the findings of the study, translated NSMP prediction approach was more accurate than untranslated NSMP using either regression analysis or error back propagation algorithm. The translated NSMP as against untranslated gives a fuller picture of stock prediction. Prediction of NSMP using current exchange rate was more accurate and reliable. The incorporation of other prediction methods such as genetic algorithm, fuzzy logic, rough sets, etc is proposed as further studies for finding the best method of translated NSMP prediction.

REFERENCES

[1] T. J. Sejnowski , B. P.Yuhas , Goldstein M.H and Jenkins R.E., Combine visual and acoustic speech signals with a neural network improves intelligibility, *Advances in Neural Information Processing Systems*, (1990) vol. 2, 232-239.

[2] K.Y.Tam and M.Y. Kiang, Managerial Application of Neural Networks: The

case of Bank Failure Predictions, *Management Science*, (1992) vol. 38, 926-947.

[3] E. Birgul, O. Melten and B. Bertan, Stock Market Prediction Using Artificial Neural Networks. www.hicbusiness.org/biz2003/proceedings/Birgul%Egeli.pdf (2003).

[4] S.I. Wu and H. Zheng, Modeling Stock Market using Neural Networks. Proceedings of Software Engineering and Applications, ACTA Press, USA, 2003.

[5] K. Kim and W. Lee, Stock Market Prediction using ANN with Optimal Feature Transformation, Neural Computing & Applications, Springer. (2004)Vol. 13, No.3, 225-260 (6).

[6] G. Cirna, A. Ajith, Y.H. Sang and V. Ramos, Stock Market Prediction Using Multi Expression Programming. www.softcomputing.net/alea.pdf

[7] B.D. Ripley., Neural Network and related methods for classification: *Journal of the royal statistical society*, (1994) series B, vol. 50, 409-556.

[8] A. Lapedes and R. Farber., Nonlinear signal processing using neural network: prediction and system modeling, Los Alamos National Laboratory (1987)

[9] Hutchinson J.M., A radial Basis Function Approach to Financial Time Series Analysis, Ph.D dissertation, Massachusetts Institute of Technology, (1994), USA

[10] C.M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press, UK 1995.

[11] Saenz G. and Pingitore N.E., Surface rganic Geochemical Prospecting for Hydrocarbons: Multivate Analysis: *Journal of Geogchemical Exploitation,* (2000) vol. 34, 227-349

[12] A.R. Ilersic and R.A. Pluck, Statistics, 14th Edition, HFL Publishers Limited, 1979

[13] J. Fox, Applied Regression Analysis, Linear Models and Related Methods, Sage, 1997.

[14] B. Widrow and M.A Lehr, 30 years of Adaptive Neural Networks, Perceptron, Madaline and BackPropagation, *proc. IEEE*, Vol 78, No 9, 1415-1442.

[15] B.S. Adefowoju and A.O. Osofisan, Cocoa Forecasting Using Neural Network. *International Journal of Mathematical and Computer Science*, (2001) pp 117-136 Nigeria.

[16] M. L. Minsky and S.A. Papert, Perceptorn, Cambridge MIT Press, 1969.

[17] N. Meade and T. Isla, Prediction Intervals for Growth Curve Forecasts. *Journal of Forecasting*, (1995), pp 413-430.

[18] N.R. Draper and N.A. Smith, Applied Regression Analysis, New York, John Wiley and Sons, (1966).

[19] S. Haykin, Neural Network: A comprehensive Foundation. 2nd Edition, Prentice Hall, (1998).

[20] S. Kotsiantis, P. Kanellopoulos and P. Pintelas, Local Additive Regression of Decision Stumps. Lecture Notes in Artificial Intelligence. Springer Verlag, Vol. 3955, pp 148-157, (2006).

[21] S.I. Gallant, Perceptron Based Learning Algorithms, IEEE Transactions on Neural Networks, Vol. 1 No. 2 pp 179-191.

[22] Y. Freund and R.E. Schapire, Large Margin Classification using the Perceptron Algorithm, In Proceeding of the11th Annual Conference on Computational Learning Theory. ACM Press, (1998).

[23] M.R.K. Maqsood and A. Araham , Neural Networking Ensemble Method for Weather Forecasting, *Neural Computing*, (2004)13, 112-122.

[24] C. Grosen and A. Abraham, Stock Market Modeling using Genetic Programming Ensembles, Genetic System, *Studies in Computational Intelligence*, Springer Verlag, (2006) 131=146, Germany.

[25] W. X. Zhou and D. Sornette, Testing the stability of the US Stock Market Antibubble, Physica A: *Statistical and Theoretical Physics*, (2000), 348, 328-452

[26] www.nationsencyclopedia,com/Africa/Nigeria-FOREIGN-INVESTMENT.html

[27] www.econsult.worldbank.org/suite/public/populateViewPublicNotification.bg