A Fuzzy-Neural Intelligent Trading Model for Stock Price Prediction

Uduak A. Umoh[1] and Udoinyang G. Inyang[2]

[1]Department of Computer Science, Faculty of Science, University of Uyo,
Uyo, Akwa Ibom State, 52001, Nigeria

[2]Department of Computer Science, Faculty of Science, University of Uyo,
Uyo, Akwa Ibom State, 52001, Nigeria

Abstract

In this paper, Fuzzy logic and Neural Network approaches for predicting financial stock price are investigated. A study of a knowledge based system for stock price prediction is carried out. We explore Trapezoidal membership function method and Sugeno-type fuzzy inference engine to optimize the estimated result. Our model utilizes the performance of artificial neural networks trained using back propagation and supervised learning methods respectively. The system is developed based on the selection of stock data history obtained from Nigerian Stock Exchange in Nigeria, which are studied and used in training the system. A computer simulation is designed to assist the experimental decision for the best control action. The system is developed using MySQL, NetBeans, Java, and MatLab. The experimental result shows that the model has such properties as fast convergence, high precision and strong function approximation ability. It has shown to perform well in the context of various trading strategies involving stocks.

Keywords: ***Fuzzy Logic, Neural Network, Stock Price prediction, Fuzzy-Neural System***

1. Introduction

In the world today, every economy is strongly influenced by the operation of stock markets; in addition, stock market, as an available means for investment, is of special importance for both the investor(s) and the beneficiaries of return on investment (ROI). Stock market prediction is important and of great interest because successful prediction of stock prices may promise attractive benefits. The most important part of this business is to obtain more profits through estimating future stock prices and market direction.

With the continual development of social economy, high-speed increase takes place in the emerging capital markets in the developing countries. Today, stock investment has become an important means of individual finance. Apparently, it is significant for investors to estimate the stock price and select the trading chances accurately in advance, which will bring high return to stockholders. In the past long-term trading process, classical approaches are mainly based on stochastic models by using the time series techniques such as autoregressive moving average and multiple regression models. Also, many technical analysis methods for stock market such as *K*-line figure and moving average etc. were put forward. These methods are based on the statistical data generally. However, the accuracy of state forecasting may be vulnerable to qualitative factors from macro-economical, political and psychological effects. Given the drawbacks imposed by statistical methods, soft computing techniques [20], such as artificial neural networks (ANNs), Fuzzy Logic and genetic algorithms, were introduced alternatives. The main advantage of these techniques over the statistical ones is their ability to explore the tolerance of systems to data uncertainty, imprecision and partial truth. Thus, it is difficult to achieve accurate stock market direction and prediction through traditional analysis tools. More so, there is usually inconsistency in the analysis conclusions of various persons using even the same tool, which demonstrates they are not suited to be used by common investors without professional knowledge and experience [7][19].

It is believed that past information can be modeled into a system that explains the current behavior and predict the future state. Classical approaches are mainly based on stochastic models by using the time series techniques such as autoregressive moving average and multiple regression models; however, the accuracy of state forecasting may be vulnerable to qualitative factors from macro-economical and political effects.

Intelligent systems techniques are widely applied to stock market problems. They offer useful tools in forecasting noisy environments like stock markets, capturing their non-linear behavior effectively [2]. Soft computing paradigms, such as neural networks and fuzzy systems have extensive applications, for the purpose of prediction in the field of finance. Lately, intelligent systems have been successfully applied to solve the problems of predicting financial time series, including financial stock market prediction. [10], investigate Neuro Fuzzy approaches for predicting financial time series and shown to perform well in the context of various trading strategies involving stocks. The horizon of prediction is typically a few days and trading strategies are examined using historical data. The results reveal that the Neuro Fuzzy techniques are able to generate forecasts with significant predictive ability. [15], develop a self-organized, five-layer neuro-fuzzy model to model the dynamics of stock market by using technical indicators. The model effectiveness in prediction and forecasting is validated by a set of data containing four indicators: the stochastic oscillator (%K and %D), volume adjusted moving average (VAMA) and ease of movement (EMV) from TAIEX (Taiwan Stock Exchange Capitalization Weighted Stock Index). Results show that the model is effective in prediction and accurate in forecasting.

[17], develop a fuzzy-neural network model, using AI technologies and applies the model for effective control of profitability in paper recycling to improve production accuracy, reliability, robustness and to maximize profit generated by an industry, despite varying cost of production upon which ultimately profit, in an industry depend. [4], investigate the predictability of stock market return with Adaptive Network-Based Fuzzy Inference System (ANFIS). The objective of this study is to determine whether an ANFIS algorithm is capable of accurately predicting stock market return. The experimental results reveal that the model successfully forecasts the monthly return of Irish Stock Exchange (ISE) National 100 Index with an accuracy rate of 98.3%. [8], apply fuzzy neural networks in multi-ahead forecast of stock price. The prediction is done by the two linear and nonlinear models for one ahead and multi ahead in stock price by using exogenous variable of stock market cash index, and the results show the preference of non-linear neural-Fuzzy model to classic linear model and verify the capabilities of Fuzzy-neural networks in this prediction.

[6], design a stock trading system using Artificial Nero Fuzzy Inference Systems (ANFIS) and technical analysis approach. The proposed model uses 14 last days information and predicts 14 next day's variables' values of Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence - Divergence (MACD), Relative Strength (RSI) , stochastic oscillator (SO). Results show that the mean of all generated networks percentage of correct predictions (96/55%) is more than random cases (50 %). [11], propose a weather prediction model in this article based on neural network and fuzzy inference system (NFIS-WPM), and then apply it to predict daily fuzzy precipitation given meteorological premises for testing. Result gives more accurate predictive precipitation values than by using traditional artificial neural networks that have relatively low predictive accuracy.

[14], investigate equity market data prediction is developed based on neural network and neuro fuzzy model by using the past equity market data. [18], propose a Sugeno-type fuzzy inference system for stock price prediction using technical indicators as its input values. The study explores Sugeno-type fuzzy inference engine to optimize the estimated result. The degree of participation of each input parameter is evaluated with trapezoidal membership function. Result indicates that the system provides vital support to stock traders, researchers and other financial experts in making decisions as regards stock trading.

[12], introduce an intelligent decision-making model, based on the application of Fuzzy Logic and Neurofuzzy systems (NFs) technology. The decision-making model is used to capture the knowledge in technical indicators for making decisions such as buy, hold and sell. Experimental results have shown higher profits than the Neural Network (NN) and “Buy & Hold” models for each stock index. [9], survey neuro fuzzy systems (NFS) development using classification and literature review of articles for the last decade (2002–2012) to explore how various NFS methodologies have been developed during this period. The review study indicates mainly three types of future development directions for NFS methodologies, domains and article types: (1) NFS methodologies are tending to be developed toward expertise orientation. (2) It is suggested that different social science methodologies could be implemented using NFS as another kind of expert methodology. (3) The ability to continually change and learning capability is the driving power of NFS methodologies and will be the key for future intelligent applications.

[10], investigates Neuro Fuzzy approaches for predicting financial time series. Methods are designed to predict 10-15 days of stock returns in advance. The methodologies are tested with actual financial data and show to perform well in the context of various trading strategies involving stocks. Results show considerable promise as a decision making and planning tool. [5] develop a neural network-driven fuzzy reasoning system. The stock price forecast is proposed on the basis of the trends of stock market. The stock market is a very complicated non-linear dynamic system, it has both the high income and high risk properties. So the forecast of stock market trend has been

always paid attention to by stockholders and investment organizations. Forecasting stock prices and their trends are important factors in achieving significant gains in financial markets. [16], apply fuzzy logic models as one of the best techniques for the effective control of profitability in paper recycling production to ensure profit maximization despite the varying cost of production upon which profits depend. To reinforce the proposed approach, the model is applied to a case study performed on the paper recycling industry in Nigeria. [13], presents the design and performance evaluation of a hybrid decision tree- rough set based system for predicting the next day's" trend in the Bombay Stock Exchange (BSESENSEX). Technical indicators were used in the present study to extract features from the historical SENSEX data. C4.5 decision tree is then used to select the relevant features and a rough set based system is then used to induce rules from the extracted features. Performance of proposed system is compared to that of an artificial neural network based trend prediction system and a naive bayes based trend predictor. Results show that the proposed system outperforms both the neural network based system and the naive bayes based trend prediction system.

[1], investigate the current trend of stock price of the Iran Khodro Corporation at Tehran Stock Exchange by utilizing an Adaptive Neuro-Fuzzy Inference System. The findings of the research demonstrate that the trend of stock price can be forecast with a low level of error. [3], develop a neuro-fuzzy adaptive control system to forecast the next day's stock price trends of the Athens Stock Exchange (ASE) and the New York Stock Exchange (NYSE) index. The experimental results reveal that the proposed system performs very well in trading simulations, returning results superior to the buy and hold strategy. It also demonstrates solid and superior performance in terms of percentage of prediction accuracy of stock market trend.

In this paper, we propose a fuzzy-neural network model for stock price and stock market direction prediction. Stock market prediction is important and of great interest because successful prediction of stock prices may promise attractive benefits. A Fuzzy system can uniformly approximate any real continuous function on a compact domain to any degree of accuracy. An ANN model essentially mimics the learning capability of the human brain. This paper also investigates Fuzzy-Neuro approaches for predicting financial time series. These approaches are found to perform well in the context of various trading strategies involving stocks. In order to achieve the objective of this work, a study of a knowledge based system for stock price prediction is carried out. An exploration of Sugeno-type fuzzy inference engine to optimize the estimated result is also performed. The degree of participation of each input parameter is evaluated using trapezoidal membership function. A survey of the performance of artificial neural networks trained with back-propagation and supervised learning methods respectively are carried out. The development of this system is based on the selection of stock data history obtained from Nigerian Stock Exchange in Nigeria, which are studied and used for training the system. A computer simulation is designed to assist the experimental decision for the best control action. The system is developed using MySQL, NetBeans, Java and MatLab. The decision-making model is used to capture the knowledge in technical indicators for making decisions such as buy, hold and sell. The experimental result shows that the fuzzy neural network has such properties as fast convergence, high precision and strong function approximation ability. Furthermore, it also demonstrates that the fuzzy neural network system is good at predicting stock price and performs well in the context of various trading strategies involving stocks because of its non-strict requirement for input variables and not needing plenty of sample data, which ensure that it is suitable for actual forecasting and outperforms the classic prediction methods.

Section 2 presents the objective while in Section 3 the research methodology is presented. Section 4 presents the model experiment while in Section 5 results of findings are discussed. Finally in Section 6, some recommendations are made and conclusion is drawn.

2. Research Objective

The objective of this research is to investigate fuzzy-neural network methodologies in stock market prediction, develop a model based on fuzzy-neural network. Apply the developed model for effective prediction of stock price to improve financial trading, reliability, robustness and high ROI.

3. Research Methodology

Fuzzy systems lack self tuning capability but it enables computing with words and provides inference mechanism under cognitive uncertainty. It requires rule sets by human experts and it is good at explaining how their decisions are reached. Neural networks on the other hand, are good at recognizing patterns and they have the capability to learn, adapt and generalize but they are not good at explaining how they arrive at their decisions.
In this research work, fuzzy logic and neural networks system are fused together in order to create an intelligent hybrid system that overcomes the limitations of the two techniques mentioned above by complementing each other's weakness thereby reducing the development time and cost while improving the performance of the system.

Architecture of fuzzy-neural network for effective stock price prediction is shown in Figure 1. It is basically a five-layer fuzzy rule-based neural network.

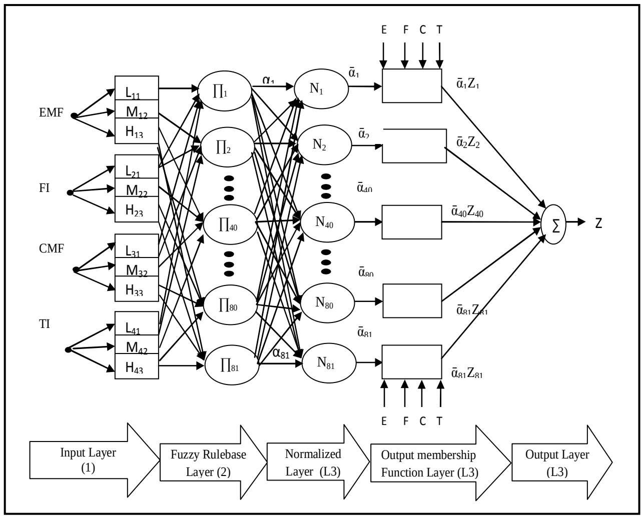

Fig. 1: Architecture of fuzzy-neural network for stock price prediction

Layer 1: The nodes in this layer serve as the fuzzy input linguistic variables which transmit input values to the next layer directly.

Layer 2: The nodes in this layer are the input membership functions which perform fuzzification of the input linguistic variables. They contain 14 neurons corresponding to the 14 fuzzy sets of the four linguistic variables. Three fuzzy sets of each input linguistic variables are *{Low, Medium, High}*. The membership functions are normal distribution with a range between 0 and 1. Trapezoidal membership functions are used in this work to describe the variables in order to suppress noise in the inputs. It is defined as:

$$\mu_A (X_i : a_1, a_2, a_3, a_4) = [0, \frac{Xi-a1}{a2-a1}, 1, \frac{a4-Xi}{a4-a3}, 0] \qquad (1)$$

Where i = 1, 2, …, 81. A can be the term *Low, Medium, High*. X_i is the normalized input, $\{a_1, a_2, a_3, a_4\}$ are the trapezoidal membership function parameters, with a_1 and a_4 defining the trapezoidal end points while a_2 and a_3 defines the trapezoidal peak locations. The degree to which the given input satisfies the linguistic label associated to this layer determines the output of this layer.

Layer 3: Nodes in this layer consist of the fuzzy rule base; there are 81 neurons in this layer. The nodes in this layer perform the root sum square operation to determine the firing strength of the associated rule. The degrees of truths of the rules are determined for each rule by evaluating the nonzero minimum values using AND operator as shown in (2). They assign the firing strength to the individual rule output (3) and finally determine the overall system output using the root sum square.

$$Net_a{}^{\wedge}3 = \lambda_a = \min (TI_a (w_0), EMV_a (x_0), FI_a (y_0), CMF_a (z_0)) \qquad (2)$$

$$C_a \quad (w)_a \quad =_a \qquad (3)$$

$$a \quad {}^{\wedge}3 \quad a \quad (w) = m_{a1} \quad a1 \quad (w)) \qquad (4)$$

$$C_{am} \quad (w) = max_{a1} \quad {}^{\wedge}3_{a2} \quad {}^{\wedge}3_{a3} \quad {}^{\wedge}3, \ldots_{am} \qquad (5)$$

Where, a = 1, 2,…, n and = 1, 2,...,m

Layer 4: Nodes in this layer represent the output membership functions

$$Net_j^4 \quad b \quad (v \qquad (6)$$

and $f(net_j^4) = net_j^4 \quad j \quad (v$ (7)

Where, j = 1, 2, …, m and ϒ is the link weight.

The initial connection weight values are generated randomly in this project. These weights can be assigned directly by an expert or can be determined outside the network from historical data and then incorporated into the network.

Layer 5: The single node in this layer evaluates the overall system output. This is accomplished by summing up the weights in layer 4. This is expressed in equation 8:

$$Netj{}^{\wedge}5 \; C0 \, (w) = \sum C_i'(w \qquad (8)$$

The crisp output is evaluated by averaging equation 8 with the sum of the connection weight values as shown in equation 9.

$$f \; (netj{}^{\wedge}5 \; C0(w)) = \frac{\sum_{i-1}^{n} C_i'(w)\Upsilon_i}{\sum_{i=i}^{n} \Upsilon_i} \; k \qquad (9)$$

3.1 Procedure in Fuzzy Neural Systems

The basic idea of the learning procedure rule is to find a way of optimizing the overall performance of the system. This is achieved by updating the links' weights that connect the nodes and minimizing the system's mean squared error between the expected outputs and actual outputs. For this design, the supervised learning method is used with the objective of minimizing the error function, E_k as presented in equation 10 by means of a learning algorithm.

$$k \quad = ½[y^k - O \qquad (10)$$

Where, y^k is the desired (expected) output and O^k is the actual (computed) output, k is the input/output pattern. We adapt and modify our learning algorithm based on (Umoh et al, 2011) to suit the requirements of this study as follows:

(i) Initialize weight. Set all weights to some random values between 0 and 1.

(ii) Select the first training vector pair from the training pair vectors and call this vector pair (x, y), where x represents the input vector and y the output vector.

(iii) Use the input vector x, as the output from the input layer of processing elements.

(iv) Complete the appropriate learning or activation function of the hidden layer denoted by f (net)$_h$ and for the output layer to each unit of the subsequent layer denoted by f (net)^o. Sigmoidal function is used as the activation function and it is defined by:

$$f(net)^0 = 1/(1 + e^{-} \quad (11)$$

where f(net)^o is the output of neuron and

$$net =_{ij\ i} \quad (12)$$

w_{ij} is the weight link that connects node i to node j and x_i is the node value.

(v) Repeat step (iii) for each layer in the network.

(vi) Compute the error signal, denoted by δ^k_i called delta, produced by the i-th output neuron as:

$$_i \quad = (y^k - 0^k)f(ne \quad (13)$$

Where is the computed output, is the actual output and

$$f(net)\, 0^k = 0^k(1 - \quad (14)$$

(vi) Compute the error δ^h_i for all hidden layer units (if any) as:

$$\xi \qquad f(n \qquad w \quad (15)$$

If E_{max} > 0.001 are chosen, further iterations are needed, thus steps (v) and (vi) are repeated.

(vii) Update the connection weight values to the output and hidden layers as:

$$_{ij} \quad (n+1)\ _{ij} \quad (n) +_{i\ \ j} \quad (16)$$

$$_{ij} \quad (n+1)\ _{ij} \quad (n) +_{i} \quad f(n\epsilon$$

(17)

Where n is the step, and η is the learning rate.

(viii) Repeat steps (ii) through (vii) for all vector pairs in the training set and call it training instance.

(ix) Repeat steps (i) through (viii) for as many epochs (iterations) as possible to reduce the sum squared error of equation 15.

3.2 Learning rate and Momentum

The learning procedure requires that the change in weight is proportional to $\partial E^p/\partial w$. The constant of proportionality is the learning rate, η. The smaller the learning parameter η, the smaller the change to the weight is, but then, the learning takes longer time. We choose a learning rate that is as large as possible without leading to quick oscillation. One way to avoid quick oscillation at large η is to make the change in weight dependent on the past weight by adding a momentum term. When no momentum term is used, it takes a long time before the minimum is reached with a low learning rate, whereas for high learning rate the minimum is never reached because of oscillation. When adding the momentum term, the minimum is reached faster. The momentum factor can speed up the training and stabilizes convergence, thus reducing the risk of getting stuck in a local minimum. The momentum term can be implemented in two ways:

(i) Add the past weight change to the current weight change, Δw_{jk}

(ii) Use the accumulated weight change over the previous cycle, $\sum\Delta w_{jk}$

The later is adopted for this research because it adds a term that is significant for all patterns and thus contributes to global error reduction. We choose a momentum constant, ω, which is small in relation to the value of $\eta\delta^o_x$. In training our fuzzy-neural network, the output weights vector components are updated using equation 18.

$$Wkj(n+1) := wkj(n) + \eta\, \delta^0 k\, xj + \omega\sum\Delta wj \quad (18)$$

Where n is the training cycle (epochs).

$w_{kj}(n)$ is the weight between the output and hidden neurons at epoch (n), $w_{kj}(n+1)$ is the weight between the output and hidden neurons at epoch (n+1), η is the Learning rate factor, δ^o_k is the output error signal, called the local gradient. x_j is the input to neuron j, ω is the momentum constant. $\sum\Delta w_{jk}$ is the cumulative weights change of the output neuron.

4. Model Experiment

The study adopts back propagation method in weights adjustment. The weights of layer 5 are tuned using back propagation algorithm and the error between the desired output (y^k) and computed output (O^k) are determined. This is illustrated as follows: Suppose a 2-layered network having 4 inputs nodes representing the firing levels of rules for Medium (x_{30}), Low (x_{31}), Medium (x_{32}) and High (x_{33}) and one output node in layer(5) is designed as shown in Figure 3,

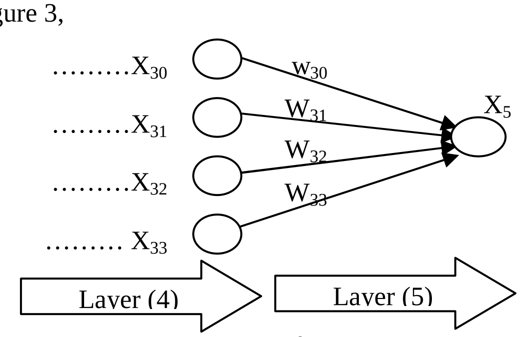

Fig. 3: 2-Layered FNN Architecture for stock price prediction

The weights of the nodes and the link weights are evaluated and presented in Tables 1 and 2 respectively.

Table 1: Weights of the nodes

W_{30}	W_{31}	W_{32}	W_{33}
0.8	0.2	0.5	0.7

Table 2: Initial links Weights

W_{30}	W_{31}	W_{32}	W_{33}
0.8	0.2	0.5	0.7

Table 3: Results of Back Propagation on the Training Data

n	W_{30}	W_{31}	W_{32}	W_{33}	$\sum w_{jk} x_j$	(y^k)	(O^k)	(δ^o_k)
1	0.2	0.4	0.6	0.8	1.100	0.9	0.7503	0.0263
2	0.21	0.42	0.63	0.84	1.155	0.9	0.7604	0.0239
3	0.24	0.45	0.67	0.86	1.219	0.9	0.7719	0.0213
4	0.25	0.47	0.66	0.82	1.198	0.9	0.7682	0.0221
5	0.29	0.49	0.6	0.8	1.190	0.9	0.7667	0.0193

From equation 12, net = $\sum w_{jk} x_j$, where w_{jk} for j = 1, 2, 3, 4, k = 1, represents the link weight between node j and node k, and x_j, j = 1,2, 3, 4 represents nodes weights respectively. "Net" is calculated and the function of these weights, f(net) is evaluated using equation 11 and 12 respectively. The output of the network with initial values of node and link weights = 0.7503 and this is the computed (actual) output (O^k). This result is used to find the error, δ^o_k as shown in equation 15. If we assume our expected output, y^k to be 0.9, the weight is updated and the new weight is computed and obtained using equation 16.

We choose a learning factor, η = 0.9, x_j is the node j weight, n is the step for the output layer and ω = 0.0381(momentum term). For example, using the values in Table 1 and 2, we net_k = 0.8 x 0.2 + 0.2 x 0.2+ 0.6 x 0.5 + 0.7 x 0.8 = 1.1. $O^k = f(net)_k = (1 + e^{-1.1})^{-1} = 1/(1 + e^{-1.1})$ = 0.7503, which is the computed output. Assuming the desired output y^k = 0.9. The error, δ^o_k is then computed thus:

$$\delta^o k = (y^k - O^k) O^k (1 - O^k) =$$
$$0.1497)(0.7503)(0.2497) = 0.0277.$$

The change in weight, Δw_{kj} is defined as: $_{jk}$ $= \eta$ ἰ$_{j}$,
for j = 1 to 4 (19)

That is, for j = 1, 0.9 x 0.0277x 0.8 = 0.0199
j = 2, 0.9 x 0.0277x 0.2 = 0.005
j = 3, 0.9 x 0.0277x 0.5 = 0.0125
j = 4, 0.9 x 0.0277x 0.7 = 0.0175

The new weights w_{kj} are obtained using equation 20 and the momentum constant, ω is assumed to be 0.0381.

$$Wkj(n+1) := wkj(n) + \eta\, \delta^0 k\, xj + \omega \Sigma \Delta wj \quad (20)$$

W_{30} (n+1) = 0.2 + 0.0199 + (0.0381)(0.0199) = 0.2207
W_{31} (n+1) = 0.4 + 0.0050 + (0.0381)(0.0050) = 0.4052
W_{32} (n+1) + 0.6 + 0.0125 + (0.0381)(0.0125) = 0.6130
W_{33} (n+1) + 0.8 + 0.0175 + (0.0381)(0.0175) = 0.8182

Net weight is computed using the new weights and the output as:
net = 0.2207 x 0.8 + 0.4052 x 0.2 + 0.6130 x 0.5 + 0.8182 x 0.7 = 1.1368

$$f(net)^0 = 1/(1 + e^{-1.1368}) = 0.$$

The error,

$$\delta^o k = (y^k - O^k) O^k (1 - O^k) =$$
$$(0.1429)(0.7571)(0.2429) = 0.0263$$

The same process is manually cycled in 5 epochs with new calculated weights showing how the results tend to converge to solution. The sample results of the back propagation on the training data are presented in Table 3.

The learning process is iterated until an acceptable minimum error (δ^o_k) between actual output y^k and predicted output O^k is achieved.

5. Result And Discussion

Our Fuzzy-Neural network system was developed using MatLab Version 7.5.0, MySQL Database Management System 6.0, NetBeans 6.5.1, and Java 1.6. An application example of FNN using the developed system is presented in our paper. Here we predict financial time series, and the result is shown to perform well in the context of various trading strategies involving stocks. In this application, four input parameters, which are thought to have an influence on stock price of the industry (*z*), are selected. They include, Ease of Movement *(EMV)*, Force Index *(FI)*, Chaikin Money Flow *(CMF)* and Trend Index *(TI)*. These parameters constitute the fuzzy logic input variables used to generate the fuzzy logic model. Membership functions are constructed for each input variable. A total of 108 sample data sets are obtained through fuzzy inference system developed in [18].

The outputs of fuzzy inference engine form the input to the neural networks. The 108 sample data sets obtained in [18] are used in network training. The initial connection weight values are generated randomly. These weights can also be determined either by the experts or outside the network from historical data and then incorporated into the network. The connection weights are subsequently tuned by back propagation algorithm. After the fuzzy-neural network is trained, test data are fed to the trained network to predict stock price movement. Figure 3 shows the graph of the Fuzzy-Neural Networks prediction results for low stock price. This result indicates low stock price movement and hence buy of stock is recommended to the traders. Figure 3 shows the graph of the Fuzzy-Neural Networks prediction results for no change in stock price, indicating no change in stock price movement and hence stock investors are recommended to buy stocks. Figure 3 shows the graph of the Fuzzy-Neural Networks prediction results for high price. This result indicates high stock price movement and hence traders are recommended to sell the available stocks.

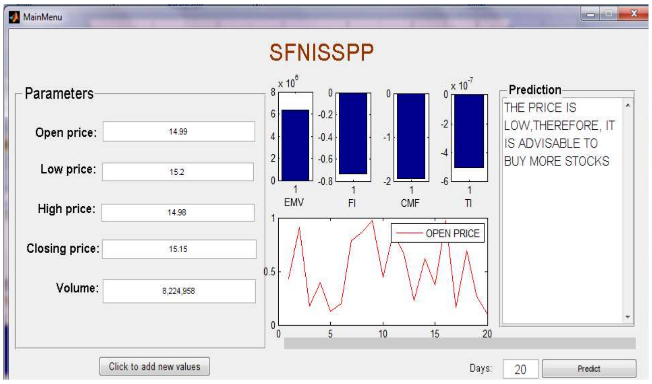

Fig. 4: The graph of the Fuzzy-Neural Networks prediction results for low Price

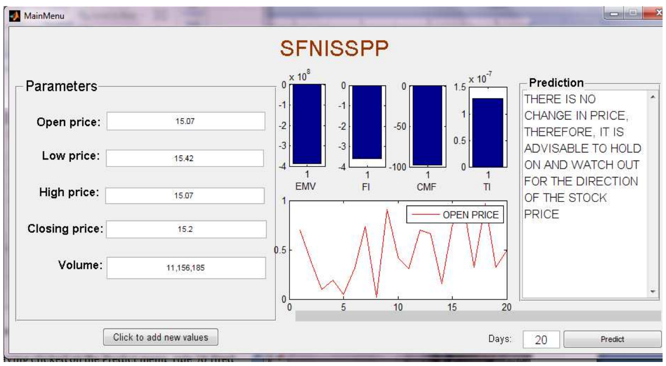

Fig. 5: The graph of the Fuzzy-Neural Networks prediction results for No Change in Price

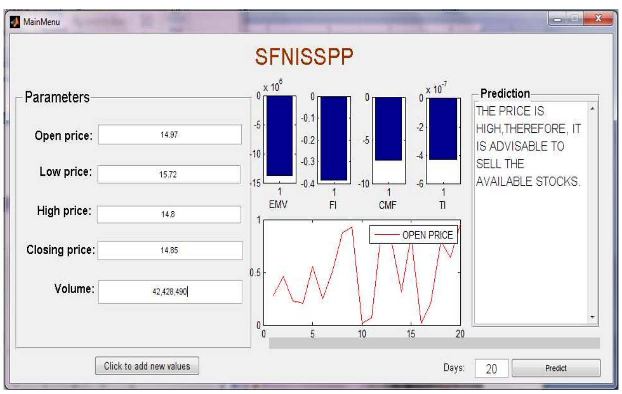

Fig. 6: The graph of the Fuzzy-Neural Networks prediction results for High Price

The Sum of the Squared Error (SSE) for the 5 cycles is evaluated by the formula: $\mathrm{SSE} = \sum_{k=1}^{n}($ ^{2} $, k = 1, 2, 3$, that is, SSE = $(0.0263+ 0.0239+ 0.0213+ 0.0221+ 0.0193)^2$ = 0.1129. Base on the fact that 0.1129 happens to be the single point or centroid of this cluster, the computed value now is = 0.9 – 0.1129 = 0.7871 (Positive). This result presents a 78% certainty that there is a rise (in the prices of stocks thereby recommending traders to sell their stocks.

The error measures based on Mean Square Error (MSE), defined (20).

$$MSE = \frac{\sum_{i=1}^{n}(y_i - o_i)^2}{N} \qquad (20)$$

Where y = desired output, O = network computed output; N = number of dataset and Correlation Coefficient (R), defined in (21).

$$R = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2 \sum (Y - \bar{Y})^2}} \qquad (21)$$

Where, Y = desired Y, ˆX = estimated Y, and Y X, = the means of Y. Its limits are 0≤ R ≥1. Y = Profitability level, is performed with 108 datasets. MSE is computed to measure the average of the squares of the difference between fuzzy neural results and the desired output with Mean Square Error value (MSE) of 0.001, and the difference between the fuzzy logic result and the desired output with MSE value of 0.022. Correlation coefficient (R) is computed to measure the degree of fit between Fuzzy-Neural Networks estimated values and desired values of output target with R value of 0.90; this indicated a perfect linear relationship.

6. Conclusion

The study develops a system that will foretell stock market price movements by utilizing adaptive fuzzy-neural inference system because it is capable of resolving conflicts by collaboration, propagation and aggregation and can mimic human decision making process or reasoning with its ability to work from approximate reasoning and ultimately find a precise solution. Neural network has ability to learn from the historical patterns of uncertain, incomplete, imprecise, vague and inconsistent data. Since prediction of stock market returns is an important issue in finance and the ultimate aim of any stock trading is to make profit, taking the low level of errors in the long and short – term modeling into account, it could be concluded that our model is capable of forecasting stock price movement. The most significant outcome is that the stock price movement is non-linear model and forecasting stock error estimation of the stock price with non-linear methods could decrease the stock price. It is therefore evident that the adaptive fuzzy-neural model developed is an effective tool for stock trade profit maximization based on its reasonability.

7. References

[1] E. Abbasi and A. Abouec,"Stock price forecast by using neuro-fuzzy inference system", Proceedings of World Academy of Science, Engineering and Technology. 2008, Vol.36, pp. 320–323.

[2] G. Atsalakis and K. Valavanis, "Surveying stock market forecasting techniques – part II: Soft computing methods". Expert Systems with Applications, Vol. 36, No. 3, 2009a, pp. 5932–5941.

[3] G. S. Atsalakis and K. P. Valavanis, "Forecasting stock market short-term trends using a neuro-fuzzy based methodology". Expert Systems with Applications, Vol. 36, No. 3, 2009b, pp. 10696 – 10707.

[4] M. Boyacioglu and D. Avci, "An Adaptive Network-Based Fuzzy Inference System (ANFIS) for the prediction of stock market return: The case of the Istanbul Stock Exchange". Expert Systems with Applications, Vol. 37, 2010, pp. 7908 – 7912.

[5] A. Devadoss and A. Ligori, "Adoption of Neural Network in Forecasting the Trends of Stock Market". International Journal of Computing Algorithm, Vol. 2, 2013, pp. 387-392.

[6] F. Faghani, M. Abzari, S. Fathi, and S. A. H. Monajemi, "Designing a Stock Trading System Using Artificial Nero Fuzzy Inference Systems and Technical Analysis Approach". International Journal of Academic Research in Accounting, Finance and Management Sciences, Vol. 4, No. 1, 2014, pp. 76–84.

[7] R. Gencay, "The predictability of security returns with simple technical trading rules". Journal of Empirical Finance, 5, 1998, pp 47–359.

[8] G. Jandaghi, R. Tehrani, D. Hosseinpour, R. Gholipour, and S. Shadkam, "Application of Fuzzy-neural networks in multi-ahead forecast of stock price Gholamreza". African Journal of Business Management Vol. 4, No. 6 2010, pp 903-914.

[9] S. Kara, S. Dasb and P. Ghoshb. "Applications of neuro fuzzy systems: A brief review and future outline". Applied Soft Computing. Vol. 15, 2014, pp 243–259.

[10] P. Kumar, K. Prashanth, T. Nirmala, S. Basavaraj, "Neuro Fuzzy based Techniques for Predicting Stock Trends". IJCSI International Journal of Computer Science Issues, Vol. 9, No. 4, 2012, pp 385-392.

[11] J. Lu, S. Xue, X. Zhang, S. Zhang, S. and Lu, W., "Neural Fuzzy Inference System-Based Weather Prediction Model and Its Precipitation Predicting Experiment". Atmosphere, Vol 5, 2014, pp 788-805.

[12] M. L. Kasemsan and M. Radeerom, "Intelligence Trading System for Thai Stock Index Based on Fuzzy Logic and Neurofuzzy System". Proceedings of the World Congress on Engineering and Computer Science, 2011, Vol 1.

[13] B. Nair, V. Mohandas and N. Sakthivel, "A Decision Tree- Rough Set Hybrid System for Stock Market Trend Prediction". International Journal of Computer Applications, (0975 – 8887) Vol.6 No. 9, 2010, pp 1-6.

[14] Seref, B and Barrisci, N. "Neural Network and Neuro Fuzzy Model for Forecasting Equity Market Data". ISITES2014 Karabuk – TURKEY, 2014 pp 669-679.

[15] C. Su, C. Chen and S. Yang, "A Self-Organized Neuro-Fuzzy System for Stock Market Dynamics Modeling and Forecasting". International Journal of Education and Information Technologies, Vol. 4 No. 3, 2010, pp 174-185.

[16] U. Umoh, E. Nwachukwu, and O. Okure, "Fuzzy Inference System for Control of Profitability in Paper recycling plant: Mamdani's Design approach". Pacific Journal of Science and Technology, Vol. 10 No. 2, 2010, pp 391-407.

[17] U. Umoh, E. Nwachukwu and O. Okure, "Fuzzy-Neural Network Model for Effective Control of Profitability in a Paper Recycling Plant". American Journal of Scientific and Industrial Research (AJSIR). Vol. 2, No. 4, 2011, pp 552-558.

[18] U. Umoh and A. Udosen, "Sugeno-Type Fuzzy Inference Model for Stock Price Prediction". International Journal of Computer Applications (IJCA), Vol. 103, No. 3, 2014, pp 1-12.

[19] Z. Yudong and W. Lenan, "Stock market prediction of S&P 500 via combinationof improved BCO approach and BP neural network". Expert Systems with Applications, Vol. 36 No. 5, 2009, pp 8849–8854.

[20] A. Zomaya, J. Anderson, D. Fogel and G. Milburn, "Nonconventional Computing Paradigms in the New Millennium: A Roundtable". Computing in Science and Engineering, 2001,pp 82-99.

Umoh U. A. had received her Doctor of Philosophy (PhD) degree in Soft Computing from University of Port Harcourt, Rivers State, Nigeria in the year 2012, Master's degree in Database Management System from University of Port Harcourt, Rivers State, Nigeria in the year 2006 and Bachelor's degree from University of Uyo, Akwa Ibom State, Nigeria in 2007. She is currently working as a Lecturer I, in the University of Uyo in the Department of Computer Science. She has published several articles in her areas in reputable national and international journals and has written some quality books in Computer discipline. Her area of interest include; Fuzzy System, Neural network, Hybrid models, Database System, etc. She is a member of Nigerian Computer Society (NCS), Member, Computer Professional Registration Council of Nigeria (CPN), etc.

U. G. Inyang is a Lecturer in the Department of Computer Science University of Uyo. He holds a Bachelor of Science (B.Sc.) degree in Computer Science from University of Uyo, Nigeria with Second class upper division. He also holds a Master of Technology and Doctor of Philosophy Degrees from Federal University of Technology, Akure, Nigeria in 2005 and 2012 respectively. He belongs to notable professional bodies including Computer Professional Registration Council of Nigeria (CPN) and has held some administrative positions in the University of Uyo. He has worked extensively with soft computing tools and Knowledge discovery/Datamining and has published several articles in these areas, in reputable national and international journals.