IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. I (Sep – Oct. 2014), PP 112-122
www.iosrjournals.org

Machine Learning Application for Stock Market Prices Prediction.

C. Ugwu and OnwuachuUzochukwu C
Department of Computer Sciences, University of Port Harcourt, Choba, Nigeria.

Abstract: *The development of a vibrant application for analyzing and predicting stock market prices is a basic tool aimed at increasing the rate of investors' interest in stock markets. This paper explains the development and implementation of a stock price prediction application using machine learning algorithm and object oriented approach of software system development. The algorithm was used in training a set of market data collected for the period of one thousand, two hundred and three days. The implementation was done with Java programming language and Neuroph software. From the experiments conducted and observed results from the application indicates a better predictive accuracy and high minimum error. When tested with Nestle Nigerian Plc. recorded a mean squared error (MSE) and regression (R) values of 61876e-6 and 0.99975 respectively, Guinness Nigerian Plc. recorded a mean squared error (MSE) and regression (R) values of 9.95839e-7 and 0.99853 respectively and Total Nigerian Plc. recorded a mean squared error (MSE) and regression (R) values of 8.03493e-6 and 0.992193 respectively.*
Keywords: *Machine learning, Supervised Learning, Regression value, Mean Square Error.*

I. Introduction

Introducing an intelligent computer application into stock prices prediction will reduce skepticism and conservative options for many investments professionals and market participants. It will guide the investors in overcoming the uncertainty and imprecision in predicting the stock prices at most times. The characteristic that all stock markets have in common is the uncertainty, imprecision which is related with their short and long-term future state. Prediction in stock market has been a hot research area for many years [6]. If any system which can consistently predict the trends of the dynamic stock market be developed, it would make the owner and the users of the system wealthy. The purpose of

prediction is to reduce uncertainty associated to investment decision making. [11]. This feature is undesirable for the investor but it is also unavoidable whenever the stock market is selected as the investment area of interest. Stock market follows a random walk, which implies that the best prediction you can have about tomorrow's value is today's value. Indisputably, forecasting stock indices is very difficult because of the market volatility that needs accurate forecast model. The stock market indices are highly fluctuating and it affects the investor's belief. Determining more effective ways of stock market index prediction is important for stock market investors in order to make more informed and accurate investment decisions. These data might have been affected by inflation or fluctuation of exchange rates especially in developed countries such as Nigeria [2].Back propagation neural network is commonly used for price prediction [14]. This paper demonstrates the development of a neural network application for analyzing and predicting of Stock Market Prices considering the technical and fundamental factors as our input against one factor used by many papers reviewed. It is expected that the accuracy of the application will be high compared to similar application.

II. Literature Review

Tsanga (2007), Developed and implemented a model of NN5 for Hong Kong stock price forecasting'. The system was tested with data from Hong Kong stock, the system achieved an overall hit rate of over 70%. [8]

Alhassan (2011), Designed a neural network application used to demonstrate the application of the WNN in the forecasting of stock prices in the market which was designed and implemented in Visual FoxPro 6.0. The proposed network was tested with stock data obtained from the Nigeria Stock Exchange. This system was compared with Single Exponential Smoothing (SES) model. The WNN error value was found to be 0.39 while that of SES was 9.78, based on these values, forecasting with the WNN was observed to be accurate and closer to the real data than those using the SES model.[15]

An-Sing et al (2003) in their study Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index., they attempt to model and predict the direction of return on market index of the Taiwan Stock Exchange, one of the fastest growing financial exchanges in developing Asian countries. The probabilistic neural network (PNN) was used to forecast the direction of index return after it was trained by historical data. Statistical performance of the PNN forecasts are measured and compared with that of the generalized methods of moments (GMM) with Kalman filter. Moreover, the forecasts are applied to various index trading strategies, of which the performances are compared with those generated by the buy-and-hold strategy as well as the investment strategies guided by forecasts estimated by the random walk model and the parametric GMM models. Empirical results shows that the PNN-based investment strategies obtain higher returns than other investment strategies examined in their study. [16]

Peter et al (2012) in their work, Prediction of Stock Market in Nigeria Using Artificial Neural Network, the prediction models were developed using Artificial Neural Network. The result of the prediction of Nigerian Stock Exchange (NSE) market index value of selected banks using Artificial Neural Network was presented. The multi-layer feed forward neural network was used, so that each output unit is told what its desired response to input signals ought to be. This work has confirmed the fact that artificial neural network can be used to predict future stock prices. [17]

Yi-Hsien, in his study integrated new hybrid asymmetric volatility approach into artificial neural networks option-pricing model to improve forecasting ability of derivative securities price. Owing to combines the new hybrid asymmetric volatility method can be reduced the stochastic and nonlinearity of the error term sequence and captured the asymmetric volatility simultaneously. Hence, in the ANNS option-pricing model, the results demonstrate that Grey-GJR– GARCH volatility provides higher predictability than other volatility approaches. [12]

Pei-Chan et al. in their study, an integrated system, CBDWNN by combining dynamic time windows, case based reasoning (CBR), and neural network for stock trading prediction is developed and it includes three different stages, beginning with screening out potential stocks and the important influential factors; and using back propagation network (BPN) to predict the buy/sell points (wave peak and wave trough) of stock price and adopting case based dynamic window (CBDW) to further improve the forecasting results from BPN. The empirical results show that the CBDW can assist the BPN to reduce the false alarm of buying or selling decisions. [7]

Sheng-Hsun et al. (2008), in their study employs two-stage architecture for better stock price prediction. Specifically, the self-organizing map (SOM) was first used to decompose the whole input space into regions where data points with similar statistical distributions are grouped together, so as to contain and capture the non-stationary property of financial series. After decomposing heterogeneous data points into several homogenous regions, support vector regression (SVR) is applied to forecast financial indices. The proposed technique was empirically tested using stock price series from seven major financial markets. The results show that the performance of stock price prediction can be significantly enhanced by using the two-stage architecture in comparison with a single SVR model. [11]

Zhang in their paper proposed an improved bacterial chemo taxis optimization (IBCO), which is then integrated into the back propagation (BP) artificial neural network to develop an efficient forecasting model for prediction of various stock indices. Experiments show its better performance than other methods in learning ability and generalization. [13]

Akinwale examined the use of error back propagation and regression analysis to predict the untranslated and translated Nigeria Stock Market Price (NSMP). The author was used 5 -j -1 network topology to adopt the five input variables. The number of hidden neurons determined the variables during the network selection. Both the untranslated and translated statements were analyzed and compared. The Performance of translated NSMP using regression analysis or error propagation was more superior to untranslated NSMP. The result was showed on untranslated NSMP ranged for 11.3% while 2.7% for NSMP. [2]

Haven examined other scholars work; this research looks at the development of a Neural Network application for Stock Price Prediction. In this research the important steps taking to arrive at the desired application is shown using an object oriented methodology.

III. Materials and Methods

A . Component Testing

Components testing (sometimes called unit testing) is the process of testing individual components in the system. UML Class diagram will be used to explain the following individual component.

The layer is linked to both the parentNetwork and the Neurons also the neuron is linked to the parentLayer while the NeuralNetwork is linked to the Layer. The UML Class diagram for the Layer is shown in figure 1.

Layer
Attributes Private long serialVersionUID = 2L
Operations Public Layer( ) Public Layer(intneuronsNum, NeuronPropertiesneuronProperties)

Public void setParentNetwork(NeuralNetwork parent) Public NeuralNetworkgetParentNetwork( ) Public iterator<Neuron>getNeuronsIterator( ) Public Neuron[0..*] getNeurons( ) Public void addNeuron(Neuron neuron) Public void addNeuron(intidx, Neuron neuron) Public void setNeuron(intidx, Neuron neuron) Public void removeNeuron(Neuron neuron) Public void removeNeuronAt(intidx) Public Neuron getNeuronAt(intidx) Public intindexOf(Neuron neuron) Public intgetNeuronCount ( ) Public void calculate( ) Public void reset( ) Public void randomizeWeights( )

Fig 1.UML Class Diagram for the Layer.

The NeuralNetwork is linked to the LearningRule, the Layer, outputNeurons and the inputNeuron while the Layer and the LearningRuleis linked to the NeuralNetwork.The UML Class diagram for the Network is shown in figure 2.

NeuralNetwok
Attributes
Private long serialVersionUID = 3L Private thread learningThread
Operations
Public NeuralNetwork Public void addLayer(Layer layer) Public void addLayer(intidx, Layer layer) Public void removeLayer(Layer layer) Public void removeLayer(intidx) Public Iterator<Layer>getLayersIterator( ) Public Layer[0..*] getLayers( ) Public Layer getLayerAt(intidx) Public intindexOf(Layer layer) Public intgetLayersCount( ) Public void setInput(Double inputVector[0..*]) Public void setinput( ) Public Double[0..*] getOutput( ) Public Double[0..*] getOutputAsArray( ) Public void calculate( ) Public void reset( ) Public void run( ) Public void learn(TrainingSettrainingSetToLearn) Public void learnNewThread(TrainingSettrainingSetToLearn) Public void learnNewThread(TrainingSettrainingSetToLearn, LearningRulelearningRule) Public void learnInSameThread(TrainingSettrainingSetToLearn) Public void learnSameThread(TrainingSettrainingSetToLearn, LearningRulelearningRule) Public void stopLearning( ) Public void randomizeWeights( ) Public NeuralNetworkTypegetNetworkType( ) Public void setNetworkType(NeuralNetworkType type) Public Neuron[0..*] getInputNeurons( ) Public void setInputNeurons(Neuron inputNeurons[0..*]) Public Neuron[0..*] getoutputNeurons( ) Public void setoutputNeurons(Neuron outputNeurons[0..*]) Public LearningRulegetLearningRule( ) Public void setLearningRule(LearningRulelearningRul) Public void notifychange( ) Public void createConnection(Neuron fromNeuron, Neuron toNeuron, double weightVal) Public String toString( )

Public void save(String filePath)
Public NeuralNetwork load(String filePath)
Public void addPlugin(PluginBase plugin)
Public PluginBasegetPlugin(String pluginName)
Public void removePlugin(String pluginName)

Fig 2.UML Class Diagram for the NeuralNetwork.

The Connection is linked to both the Neuron and the weight while the Neurons is linked to both the inputConnection and the outputConnection. The UML Class diagram for the Connection is shown in figure 3.

Connection
Attributes
Private long serialVersionUID = 1L
Operations
Public Connection(Neuron connectTo)
Public Connection(Neuron connectTo, Weight weight)
Public Connection(Neuron connectTo, double weightVal)
Public Connection(Neuron from, Neuron connectTo)
Public Weight getWeight( )
Public Neuron getConnectedNeron( )
Public double getinput( )
Public double getWeightedInput( )

Fig 3. UML Class Diagram for the Connect.

The connection links the weight. The UML Class diagram for the Weight is shown in figure 4.

Weight
Attributes
Private long serialVersionUID = 1L
Private double value
Private double previousValue
Operations
Public Weight( )
Public Weight(double value)
Public void inc(double amount)
Public void dec(double amount)
Public void setValue(double value)
Public double getValue( )
Public void setPreviousValue(double previousValue)
Public double getPreviousValue( )
Public String toString( )
Public void randomize( )

Fig 4.UML Class Diagram for the Weight.

The LearningRule is links the NeuralNetwork and the LearningRule links the NeuralNeural. The UML Class diagram for the LearnigRule is shown in figure 5.

LearningRule
Attributes
Private long serialVersionUID = 1L
Private Boolean stopLearning = false
Operations
Public LearningRule( )
Public LearningRule(NeuralNetwork network)
Public void setTrainingSet(TrainingSettrainingSet)
Public TrainingSetgetTrainingSet( )
Public NeuralNetworkgetNeuralNetwork( )
Public void setNeuralNetwork(NeuralNetworkneuralNetwork)
Public void run( )
Public void stopLearning( )
Public booleanisStopped( )
Protected void notifyChange( )
Public void learn(TrainingSettrainingSet)

Fig 5.UML Class Diagram for the LearningRule.

The InputFunction links the SummingFunction. The UML Class diagram for the SummingFunction is in figure 6.

SummingFunction
Attributes
Operation
Public Double[0…*] getOutput(Connection inputs[0…*])
Public String toString( )

Fig 6.UML Class Diagram for the SummingFunction.

The inputFunction is linked to both the SummingFunction and WeightFunction. The UML Class diagram for the InputFunction is shown in figure 7.

InputFunction
Attributes
Private long serialVersionUID = 2L
Operation
Public InputFunction ( )
Public InputFunction(WeightsFunctionweightsFunction, SummingFunctionsummingFunction)
Public doublegetOutput(Connection inputConnections[0…*])
Public SummingFunctiongetSummingFunction( )
Public WeightsFunctiongetWeightsFunction( )

Fig 7.UML Class Diagram for the InputFunction.

The Neuron links the Transfer Function. The UML class diagram for the InputFunction is shown below in figure 8.

TransferFunction
Attributes
Private long serialVersionUID = 1L
Operations
Public double getOutput(double net)
Public double getDerivative(double net)
Public String toString( )

Fig 8. UML Class diagram for theTransferFunction

The Neuron is linked to the InputConnection, the OutputConnection, the Layer, the InputFunction and the transferFunction. The Connection and the layer are linked to the neuron while theoutputNeuron and the InputNeuron are linked to the NeuralNetork. The UML Class diagram for the Neuron is shown in figure 9.

Neuron
Attributes
Private long serialVersionUID = 3L
Protected double netInput = 0
Protected double output = 0
Protected double error = 0
Operation
Public Neuron ( )
PublicNeuron(InputFunctioninputFunction, TransferFunctiontransferFunction)
Public void calculate)
Public void reset( )
Public void setinput(double input)
Public double getNetInput( )
Public double getOutput( )
Public booleanhasInputConnections( )
Public Iterator<Connection>getInputsIterator( )
Public void addInputConnection(Connectio connection)
Public void addInputConnection(Neuron fromNeuron, double weightVal)
Protected void addOutputConnection(Connection connection)
Public Connection[0…*] getInputConnections( )
Public Connection[0…*] getOutputConnections( )
Public void removeInputConnectionFrom(Neuron fromNeuron)
Public Connection getConnectionFrom(NeuronfromNeuron)
Public void setInputFunction(

InputFunctioninputFunction)
Public void setTransferFunction(
inputFunctioninputFunction)
Public inputFunctiongetInputFunction( )
Public TransferFunctiongetTransferFunction( )
Public void setParentLayer(Layer parent)
Public Layer getParentLayer( )
Public Weight[0…*] getWeightsVector( )
Public double getError( )
Public void setError(double error)
Public void setOutput(double output)
Public void randomizeInputWeights( )

Fig 9. UML Class Diagram for the Neuron

The Connection links the Weight. The UML Class diagram for the WeightFunction is shown in figure 10.

WeightsFunction
Attributes
Operation Public Double[0…*] getOutput(Connection inputs[0…*]) Public String toString( )

Fig 10. UML Class Diagram for the WeightFunction

B The Case Diagram of the Proposed System

The case diagram of the proposed system is shown in fig 11. The proposed system allows the user to provide the training data and the threshold constant. Training Neural network requires initialization of input weights and the first input weight will be adjusted until the optimal prediction is achieved.

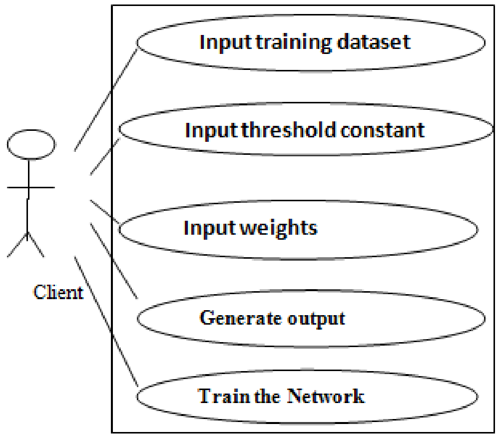

Fig 11.The Case Diagram of the Proposed System.

C. Activity Diagram and Sequence Diagram of the Proposed System.

An activity diagram is essentially a flowchart, showing flow of controls from one activity to another. Unlike a traditional flowchart, it can model the dynamic functional view of a system. An activity diagram represents an operation on some classes in the system that results to changes in the state of the system.

From the diagram shown in fig 12, the client is expected to provide the input dataset and the required output. The required output is used in back propagation, for the system uses it to compare its predicted value from time to time in other to get the optimal prediction. The client select a threshold constant and looking at the input value the neural network initializes the weight for the input layer and the hidden layer. Then the calculation for the activation function of both the input and the output layer is done and the system calculates the error. If the is large then the system adjust the weight and backpropagate it to the input layer for further calculation to be done. The system outputs its prediction whenever the error is small.

Fig 13, shows the sequence diagram of the proposed system. The sequence diagram number the actions starting from the data input to the optimal prediction. There is an arrow direction to show the sequence of flow for the action taking to arrive at the optimal prediction.

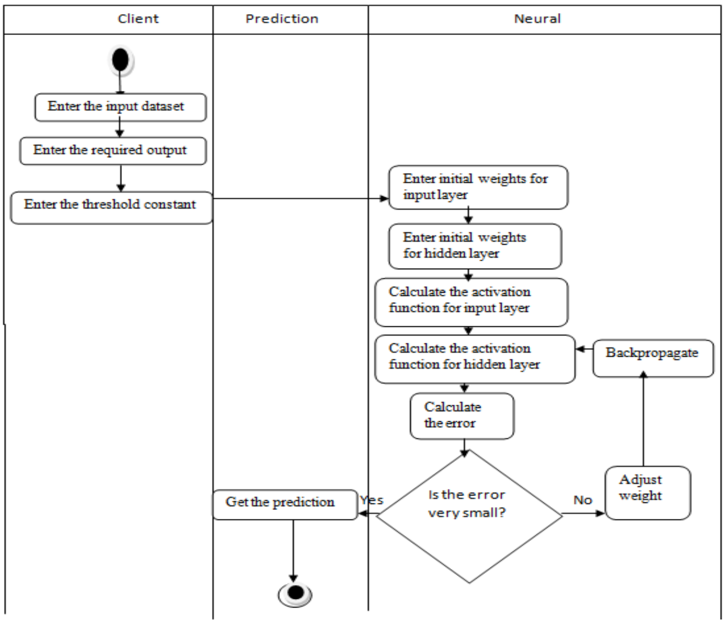

Fig 12: Activity Diagram of the Proposed System

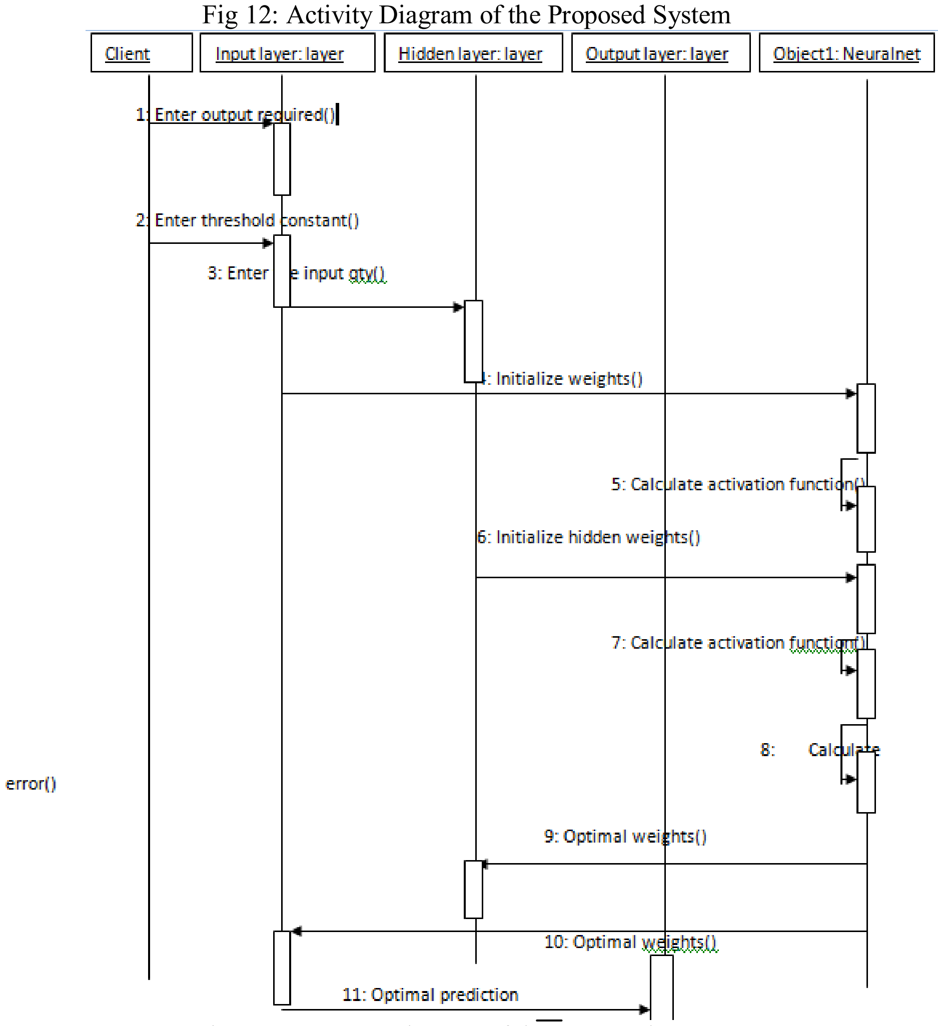

Fig 13: Sequence Diagram of the Proposed System

Fig 14.shows the class diagram for the proposed system. The layer is connected to both the parentNetwork and the Neurons also the neuron connects to the parentLayer while the NeuralNetwork connects to the Layer. The NeuralNetwork is connected to the LearningRule, the Layer, outputNeurons and the inputNeuron while the Layer and the LearningRule are connected to the NeuralNetwork. The Connection connects the Neuron and the Weight while the Neurons connect the inputConnection and the outputConnection. The Connection connects the Weight. The LearningRuleconnects the NeuralNectwork and the LearningRule connects the NeuralNeural. The InputFunction connects SummingFunction. The inputFunction is connected to the SummingFunction and WeightFunction. The Neuron Connects the Transfer Function. The Neuron connects the InputConnection, the OutputConnection, the Layer, the InputFunction and the transferFunction. The Connection and the layer are connected to the neuron while the outputNeuron and the InputNeuron are connected to the NeuralNetork. The Connection connects the Weight.

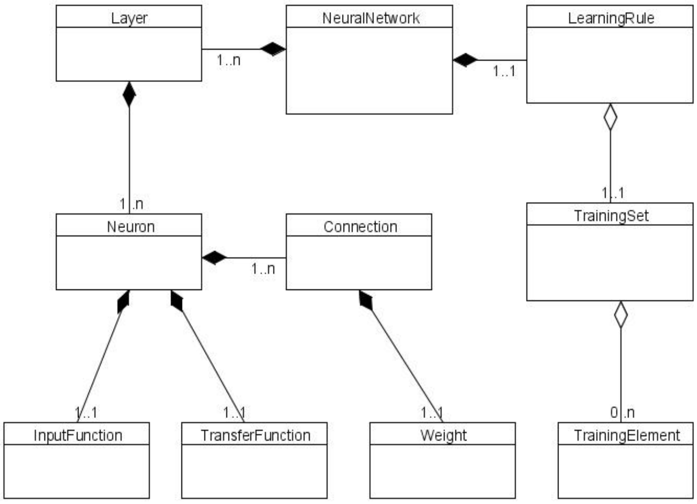

Fig 14: the architectural design of the system

V. Experiments And Results

The developed application was tested using data from the daily index values of the Nigerian stock exchange (NSE) for three selected companies. The selected companies include Total Nigeria Plc., Nestle Nigeria Plc. and Guinness Nigeria Plc. The data were collected from January 2008 to march 2013. The source of data used is www.cashcraft.com which provides daily stock market data reports of these companies. The system requirement includes the system input variables which comprises of both the technical and the fundamental input variables. [1] The technical and the fundamental input variables are listed as

Oi-1 the opening price of day i-1
Oi-2 the opening price of day i-2
Hi-1 the daily high price of day i-1
Hi-2 the daily high price of day i-2
Li-1 the daily low price of day i-1
Li-2 the daily low price of day i-2
Ci-1 the closing price of day i-1
Ci-1 the closing price of day i-2
Vi-1 the trading volume of day i-1
Vi-2 the trading volume of day i-2
Gi-1 thegross domestic product of year i-1
Gi-2 thegross domestic product of year i-2
Ii-1 the inflation rate of year i-1
Ii-2 the inflation rate of year i-2

We have a total of 14 input variables, these inputs were normalized which is an appropriate stage in training the data obtained using neural networks applications that was developed. The input data is normalized into the range of [0, 1] or [-1, 1] according to the activation function of the neurons. [5]. In this paper the value of the stock market is normalized into the range of [0, 1] using a sigmoid function and the neural networks are trained and tested using the back propagation algorithm.

Normally graphs are used to aid our understanding; in figure 15, different Actual values are compared with the predicted values using a line graphs. The graph demonstrates the direction of Nestle Nigerian Plc. Prediction. From the graph one can see that the predicted value follows the actual value.

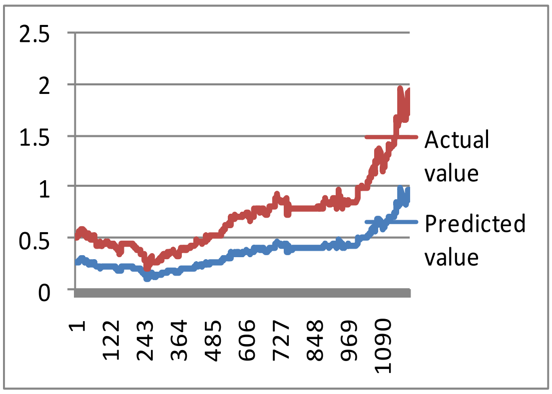

Fig 15.The graphic representation of the Actual vs. the predicted value in Nestle Nigerian Plc.

In fig 16 the different Actual values are compared with the predicted values using a line graphs. The graph demonstrates the direction of Guinness Nigerian Plc. Prediction. From the graph one can see that the predicted value follows the actual value.

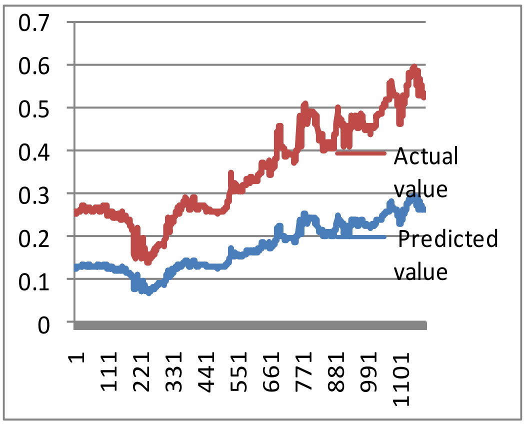

Fig 16. The Graphic Representation of Actual Vs. Predicted Value in Guinness Nigerian Plc. Prediction

In fig 17, different Actual values are compared with the predicted values using a line graphs. The graph demonstrates the direction of Nestle Nigerian Plc. Prediction. From the graph one can see that the predicted value follows the actual value.

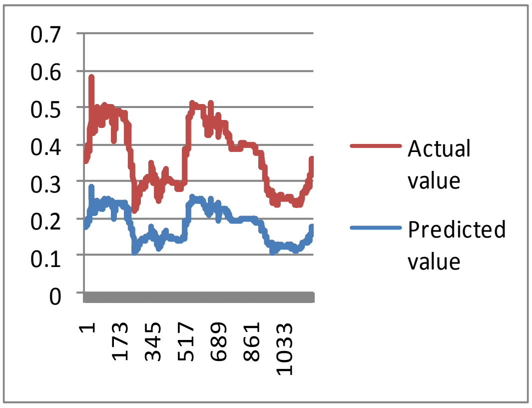

Fig 17. The Graphic Representation Of Actual Vs. Predicted Value in Total Nigerian Plc. Prediction.

In the table 1, the neural network structure of different hidden neurons was compared using their respective Actual and Predicted value. The result shows that the neural network structure of 14 - 18 – 1 gives the optimal Predicted value and it remains the best network model for predicting Nestle Nigerian Plc. Stock.

Table 1. Result From Different Neural Network Structure In Nestle Nigerian Plc. Prediction.

Date	Neural network structure	Actual value	Predicted value
10/04/2013	14 - 21 – 1	972	971.69
10/04/2013	14 - 20 – 1	972	972.2
10/04/2013	14 - 19 – 1	972	971.6
10/04/2013	14 - 18 – 1	972	972
10/04/2013	14 - 17 – 1	972	970.9
10/04/2013	14 – 16 – 1	972	972.4

In the table 2, the neural network structure of different hidden neurons was compared using their respective Actual and Predicted value. The result shows that the neural network structure of 14 - 17 – 1 gives the optimal Predicted value and it remains the best network model for predicting Guinness Nigerian Plc. Stock.

Table 2. Result From Different Neural Network Structure In Guinness Nigerian Plc. Prediction.

Date	Neural network structure	Actual value	Predicted value
10/04/2013	14 - 21 – 1	180	179.78
10/04/2013	14 - 20 – 1	180	179.1
10/04/2013	14 - 19 – 1	180	180.5
10/04/2013	14 - 18 – 1	180	180.1
10/04/2013	14 - 17 – 1	180	179.5
10/04/2013	14 – 16 – 1	180	177.8

In the table 3, the neural network structure of different hidden neurons was compared using their respective Actual and Predicted value. The result shows that the neural network structure of 14 - 18 – 1 gives the optimal Predicted value and it remains the best network model for predicting Total Nigerian Plc. Stock.

Table 3. Result From Different Network Structure In Total Nigeria Plc, Prection

Date	Neural network structure	Actual value	Predicted value
10/04/2013	14 - 21 – 1	264	263.66
10/04/2013	14 - 20 – 1	264	263.8
10/04/2013	14 - 19 – 1	264	263.8
10/04/2013	14 - 18 – 1	264	263.9
10/04/2013	14 - 17 – 1	264	264
10/04/2013	14 – 16 – 1	264	263.8

V. Discussion of Results

Table 1 shows the predicted values with different neural network predictive model. The table contains the sample data, the actual value, and values of different predictive models. The results from the different predictive models where compared with the actual value and it was observed that 14-18-1 predictive model gave the best prediction for Nestle Nigeria Plc. The result of 14-18-1 predictive model and the actual value are in bold for easy comparison and identification. The graph in figure15, demonstrate the closeness of the predicted value against the actual value, and it were seen from the graph that the prediction was done with minimum amount of error.

Table 4.5 shows the predicted values with different neural network predictive model. The table contains the sample data, the actual value, and values of different predictive model. The results from the different predictive models where compared with the actual value and it was observed that 14-18-1 predictive model gave the best prediction for Total Nigeria Plc. The result of 14-18-1 predictive model and the actual value are in bold for easy comparison and identification. The graph in figure 16, demonstrate the closeness of the predicted value against the actual value, and it were seen from the graph that the prediction was done with minimum amount of error.

Table 4.2 shows the predicted values with different neural network predictive model. The table contains the sample data, the actual value, and values of different predictive model. The results from the different predictive models where compared with the actual value and it was observed that 14-17-1 predictive model gave the best prediction for Guinness Nigeria Plc. The result of 14-17-1 predictive model and the actual value are in bold for easy comparison and identification. The graph in figure 17, demonstrate the closeness of the predicted value against the actual value, and it were seen from the graph that the prediction was done with minimum amount of error.

VI. Conclusion

The developed application does its predication of stock market prices with minimum amount of error. It was also observed from the experimental results that optimal prediction can be achieved by varying the number of the hidden neuron. The initialization scheme may be improved by estimating weights between input nodes and hidden nodes, instead of random initialization. Enrichment of more relevant inputs such as fundamental data and technical data from derivative markets may improve the predictability of the network. Applying Neural Network back propagation learning algorithm in training data for stock prediction has been shown in this paper to be an efficient tool in developing applications for stock prediction.

References

[1] A. AdebiyiAyodele, K. Ayo Charles., O. Adebiyi Marion., and O. Otokiti Sunday, "Stock Price Prediction using Neural Network with Hybridized Market Indicators," Journal of Emerging Trends in Computing and Information Sciences, Vol3 No 1, Pp 1-9, 2012.

[2] T. AkinwaleAdio, O.T. Arogundade and F Adekoya Adebayo. "Translated Nigeria Stock Market Price Using Artificial Neural Network for Effective Prediction," Journal of Theoretical and Applied Information TechnologyVol1 No 1. Pp 36-43. 2009

[3] N. C. Ashioba, E.O. Nwachukwu, and O. Owolobi. "Finding the Optimal Solution for a Transportation Problem Using Neural Network. Microwave," International Journal of Science and Technology, Vol. 3 No 1, Pp 36-40, 2012.

[4] Kyoung-Jae Kimand and Ingoo Han, " Genetic Algorithms Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index," Institute of Science and Technology, " Vol19, .Pp 125-132, 2000

[5] Mahdi PakdamanNaeini, HomaBaradaranHashemiand HamidrezaTaremian, ".Stock Market Value Prediction Using Neural Networks," International Conference on Computer Information Systems and Industrial Management Applications (CISIM), Pp132 – 136 2010.

[6] NeelimabBudhani,C. K. Jha, and K Sandeep.Budhani,. "Application of Neural Network In Analysis of Stock Market Prediction," International Journal of Computer Science And Engineering Technology (IJCSET), Vol 3No 4,, Pp61-68. Vol. 3 No. 4 , April 2012.

[7] Pei-Chann Chang and Chen-Hao Liu,. "A Neural Network With A Case Based Dynamic Window for Stock Trading Prediction; Expert Systems With Applications " Vol 36, No3,Pp 6889-6898, 2008

[8] M. Tsanga, Paul Kwoka, S.O. Choya, Reggie Kwanb, S.C. Nga, Jacky Maka, Jonathan Tsangc, Kai Koongd, Tak-Lam Wong (2007). Design and implementation of NN5 for Hong Kong stock price forecasting, Engineering Applications of Artificial Intelligence 453–461.

[9] Reza GharroieAhangar, MahmoodYahyaZadehfar and Hassan Pournaghshband. "The Comparison of Methods Artificial Neural Network With Linear Regression Using Specific Variance for Predicting Stock Price In Tehran Stock Exchange," International Journal of Science and Information Security, Vol 7 No 2. Pp 38-46. .2010.

[10] SalimLahmiri , "Neural Networks and Investor Sentiment Measures for Stock Market Trend Prediction," Journal of Theoretical and Applied Information Technology. ISSN 1992-8645, Vol 27 No 1, 2011.

[11] Sheng-Hsun Hsu and JJ Po-An Hsieh, "A Two-Stage Architecture For Stock Price For Recasting By Integrating Self-Organizing Map and Support Vector Regression, Expert Systems with Applications," Vol36, No 4, Pp 7947-7951. 2008

[12] Yi-Hsien Wang, "Nonlinear Neural Network Forecasting Model for Stock Index X Option Price: Hybrid GJR–GARCH Approach," Expert Systems with ApplicationsVol36.,No1, Pp564-570. 2007.

[13] Zhang Yudong and Wu Lenan,. "Stock Market Prediction of S&P 500 Via Combination of Improved BCO Approach and BP Neural Network, Expert Systems With Applications," Vol 36, No5, Pp8849-8854. 2008

[14] B. Manjul, S.S.V.N. Sarma, R. LakshmanNaik and G. Shruthi, "Stock Prediction using Neural Network," International Journal Of Advanced Engineering Sciences and Technologies (IJAEST) Vol No. 10, No. 1, Pp 013 – 018, 2011

[15] Alhassan1 J. K. and Sanjay Misra (2011),Using a weightless neural network to forecast stock prices: A case study of Nigerian stock exchange, Scientific Research and Essays Vol. 6 (14), pp. 2934-2940, Available online at http://www.academicjournals.org/SRE

[16] An-Sing Chen, Mark T. Leung and HazemDaouk (2003), Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index,Computers & Operations Research 901–923

[17] Peter Adebayo Idowu, Chris Osakwe, AderonkeAnthoniaKayode, Emmanuel RotimiAdagunodo (2012), Prediction of Stock Market in Nigeria Using Artificial Neural Network, I.J. Intelligent Systems and Applications, 11, 68-74.