Journal of the Nigerian Association of Mathematical Physics
Volume 53 (November *2019 Issue*), pp103 – 116

STOCK MARKET MOVEMENT DIRECTION WITH ENSEMBLE DEEP LEARNING NETWORK

D.O. Oyewola[a], [b]K. A. Al-Mustapha, [c]E.G. Dada and [d]O.A. Kennedy

[a]Department of Mathematics and Computer Science, Federal University Kashere, Gombe state, Nigeria
[b]Department of Science, Baze University, Abuja, Nigeria
[c]Department of Computer Engineering, Faculty of Engineering, University of Maiduguri, Borno State, Nigeria
[d]Department of Mathematics, National Open University, Abuja, Nigeria

Abstract

In this study the ability of Ensemble Deep Learning Network in forecasting the daily closing price of Nigerian Stock Exchange was investigated. An ensemble method consisting of several deep learning network which are feed forward neural network (FFN), recurrent neural network (RNN) and cascade forward network (CFN) were used to train the outputs of Levenberg-Marquardt (LM), Resilient Back Propagation (RBP) and Scaled Conjugate Gradient (SCG)for enhanced performance. An ensemble deep-supervised was proposed by combining deep learning and supervised learning to achieve an optimal result. The dataset used was obtained from historical closing stock prices with integrating five technical analysis as inputs. Daily stock exchange rates of first city monument bank and zenith bank extracted from Nigerian Stock Exchange from January 4^th^, 2016 to December 31^st^ ,2018 are used to develop a robust model. January 4^th^, 2016 to December 31^st^, 2017 are selected as a training dataset and January 4^th^, 2018 to December 31^st^, 2018 are used for testing the model prediction ability. Experimental result analysis shows that our proposed model has 99% accuracy higher than the three deep learning network used in this study.

Keywords: stock, ensemble learning, recurrent neural network, scaled conjugate, technical analysis

1. Introduction

Research on stock prices prediction has been the pattern for a long time [1]; experts, brokers and analysts accept that movements on stock prices are unsurprising predictable in light of statistical techniques and historical data. Distinguishing historical pattern of time series was the subject of many research works [2, 3]. In the last 20 years and on account of the advancement of technology and tracking systems, an enormous amount of historical information is accessible for analysis, therefore, machine learning procedures turned into the fundamental hub for networks. Motivated from human mind, artificial neural systems [4] (ANN) represent an association between various sources of inputs called neurons, every neuron is weighted where weights are found out from historical data, they can learn complex patterns and extract valuable data. Since mid 1990's [5], ANN was being alluring for new looks into prediction of time series. A paper by [6] displayed a measured neural system with simulation on TOPIX index stocks predicting, [7] utilized neural systems with data gain procedures to forecast future prices, [8] proposed an improved S&P500 forecasting model dependent on neural systems and bacterial chemotaxis optimization (IBCO), [9] utilized fuzzy neural systems to research the effect of public mood on securities exchanges. Stock exchange index prediction with a machine learning strategies was utilized by [10] , [11] and[12].

Support vector machines (SVM) have additionally been shown to effectively predicate stock price. A paper by [13] demonstrate that utilizing support vector machines for regression (SVR) could result in preferred financial forecasting over neural systems and [14] shown that SVR outperforms or is on a par with the MLP for a transient

Corresponding Author**:** Oyewola D.O., Email: davidakaprof01@yahoo.com, Tel: +2348063106663

prediction as mean squared error and risk premium. SVR is a valuable and ground-breaking machine learning strategy to recognize patterns in time series datasets and connected SVR to predict securities exchange prices just as well as trends as being suggested by [15]. SVM and a prescient variable selection technique within Technical Analysis (TA) indicators was used by [16]. While trying to build up an optimal market exchanges strategy, [17] utilized k-means to predict stock market volatility and SVR to foresee stock prices in the Indian securities exchange. The authors analyzed every day data to estimate stock prices for two days. In [9] the authors study, annual data from more than 5000 European organizations were utilized in classifiers for example logistic regression, neural networks, KNN and SVM. Classifiers are utilized to decide the direction of the individual company stocks in the following year. The authors contrasted the effects of those classifiers with ensemble approaches, for example, Random Forests (RF), Ada Boost and kernel factory. The supposed ensemble systems include different classifiers, ordinarily of a similar kind or algorithm, bringing about autonomous results yet with some classification technique for deciding a single final classification. Applying these strategies, determined predictive variables, considering companies accounting reports and budget reports. In view of the forecast of the direction that a specific stock would take during the year, the authors indicated how a productive methodology could be built. Additionally, trying to predict the direction of stock prices, [18] and [19] connected SVM-based systems to Technical Indicators. A paper by [20] proposed a technique that utilize computationally less difficult classifiers for intraday systems in the Foreign Exchange (FOREX) market. Decision tree algorithms and lazy models were connected to Technical Analysis variables for strategies on USDJPY, GBPUSD and EURUSD prices and [21] likewise utilized Technical Analysis variables as inputs in their model yet in an alternate way from different authors. Rather than utilizing the indicator values directly, as [20] and [16] utilized the pattern indication given by the indicators. A Technical Analysis indicator can identify the market pattern as bullish or bearish. In this manner, [21] model uses this data as a predictive variable in algorithms, for example, SVM, irregular random trees and neural networks to predict trend rather than price. [22] built up another scientific model to forecast price changes, estimated in seconds, of organizations listed on the BM&F Bovespa. in the meantime, [23] examined the profitability of a genetic learning algorithm for the FOREX stock exchange, connected to price changes estimated in minutes. SVR is up-to-the-minute prices in the Chinese market in [24] study.

2. Methodology

2.1 Feed Forward Neural Network (FFN)

The neural network Feed Forward [25] data from the inputs through the hidden layers to the outputs. This movement of data through the network is called forward propagation. Between layers in the Neural Network, the output of neurons in one layer, or the activations of these neurons, are associated with the input of neurons in the next layer, every connection are related with a weight [26]. These weights are the tuning knobs of the network. One can say that the weight is the strength of the connection between two neurons, or the amount of the activation from one neuron that is carried through to the next. This can be illustrated using Figure 1

Equation for activation function [27] of an i^{th} hidden neuron is given by:

$$h_i = f(u_i) = f(\sum_{k=0}^{K} w_{ki} x_k) \quad (1)$$

Where h_i is the i^{th} hidden neuron, $f(u_i)$ is the link function which gives non-linearity among input and hidden layer, w_{ki} is the weight in the ki^{th} entry in a ($K \times N$) weight matrix, x_k is the K input value.

$$y_j = f(u_j^1) = f(\sum_{i=1}^{N} w_{ij} h_i) \quad (2)$$

Where y_j is the j^{th} output value

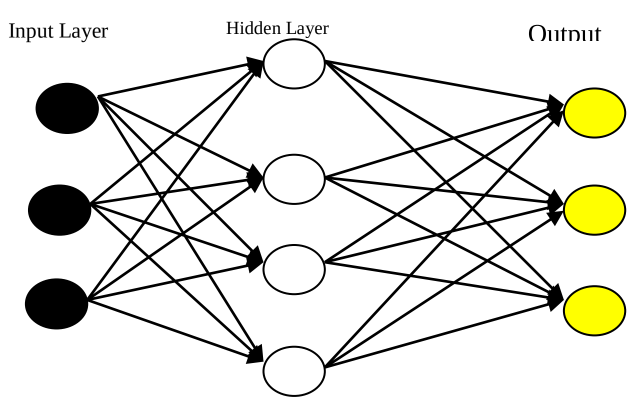

Fig. 1: Feed Forward Neural Network

2.2 Recurrent Neural Network (RNN)

A RNN[28] has an inner state that is fed back to the input, illustrated in Figure 2.

It utilizes the present information on the input and the prediction of the last input. The time steps of a recurrent neural network are often illustrated as in Figure 2. RNN can have one-to-many, many-to-one or many-to-many input and output layers. Input to hidden layer equation is given as:

$$h_t = g_n(W_{xh}X_t + W_{hh}h_{t-1} + b_h) \qquad (3)$$

Where h_t is the hidden layer at t^{th} instant, g_n is the function, W_{xh} is the input to hidden layer of weight matrix, X_t is the input at t^{th} instant, h_{t-1} is the hidden layer at $t-1$ instant, b_h is the bias or threshold value.

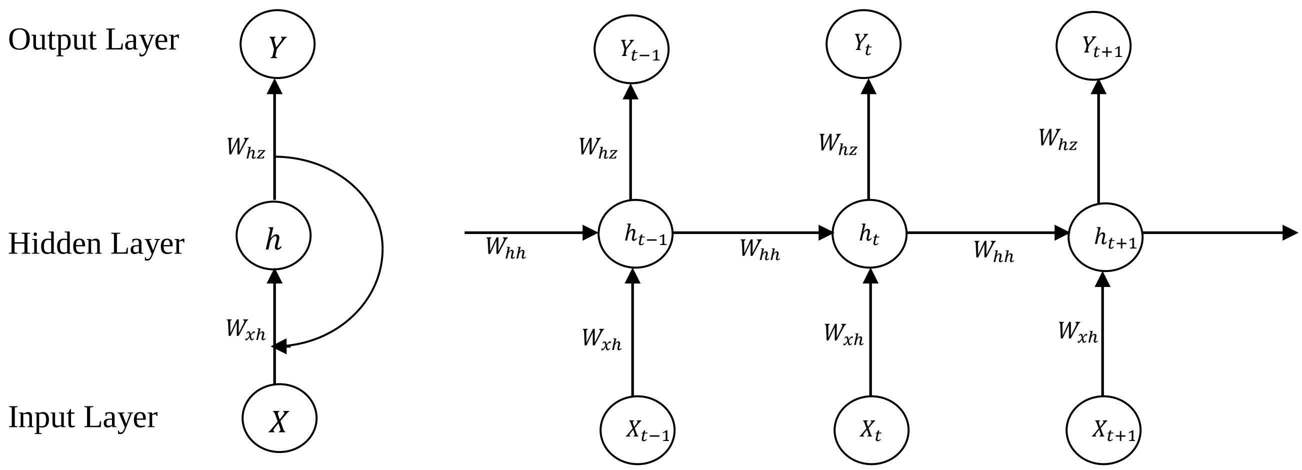

Fig. 2: Recurrent Neural Network

hidden to output layer equation is given as :

$$Z_t = g_n(W_{hz}h_t + b_z) \qquad (4)$$

Where as Z_t is the output vector, W_{hz} is the hidden to output layer weight matrix, b_z is the bias or threshold.

2.3 Cascade Forward Network (CFN)

Cascade forward network is also like feed-forward networks, but it incorporate a weight connection from the input to each layer and from each layer to the successive layers. Cascade forward network model is similar to feed forward neural network in utilizing the back propagation algorithm for weights updating. However, the primary significance of this network is that each layer of neurons are related to all previous layer of neurons [29]. Fig. 3.3 shows pictorial representation of Cascade Forward Network (CFN).

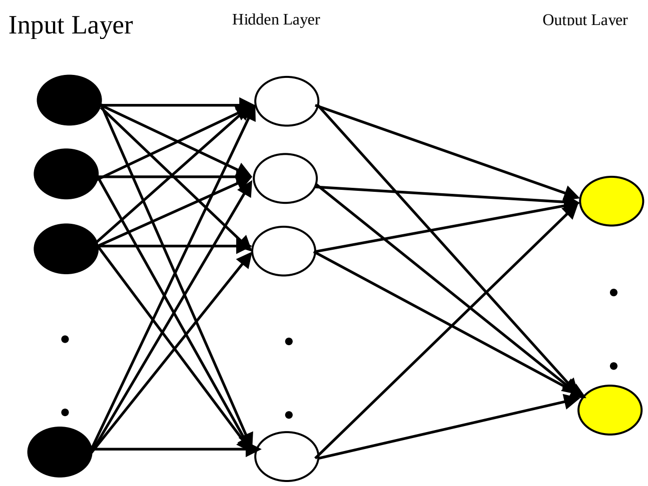

Fig. 3: Cascade Forward Network (CFN)

The mathematical equation is given as:

$$y=\sum_{i=1}^{n} f^{i}(w_i x_i)+f^{o}(\sum_{j=1}^{K} w_j^{o} f_j^{h}(\sum_{i=1}^{n} w_{ji}^{h} x_i)) \qquad (5)$$

Where f^{i} is the activation function from the input layer to the output layer, w_i^{i} is weight from the input layer to the output layer. In the event a bias is added to the input layer and the activation function of each neuron in the hidden layer is f^{h} then equation (5) becomes

$$y=\sum_{i=1}^{n} f^{i}(w_i x_i)+\ f^{o}\left(w^{b}+\sum_{j=1}^{K} w_j^{o} f^{h}\left(w_j^{b}+\sum_{i=1}^{n} w_{ji}^{h} x_i\right)\right) \qquad (6)$$

2.3.1 Training Algorithms for Neural Network

Three training algorithms were tested in this study to evaluate the three neural network performance. In this study, we attempt to implement three training algorithms such as Levenberg-Marquardt, Scaled Conjugate Gradient, and Resilient Back-propagation. The detailed explanation related to the three training algorithms, is described. The Training Algorithms is used for weighting adjustments of RNN, CFN and FFN model used in this research.

2.3.2 Levenberg Marquardt (LM)

The Levenberg-Marquardt algorithm joins the steepest descent technique with the Gauss-Newton technique and operates accurately in search for parameters both far from and close to the optimum one. In the previous case the algorithm of the linear model of steepest descent is utilized, and in the last one - squared convergence [30].The Levenberg-Marquardt algorithm is an iterative technique, in which the vector of unknown parameters is determined during step $K+1$ by the equation:

$$x_{k+1}=x_k^{T}-[J^{T}(x_k,t)J(x_k,t)+\mu_k I]^{-1}J^{T}(x_k,t)y(x_k,t) \qquad (7)$$

With the error:

$$I_2=\int_0^{T} y^2(x_k,t)dt \qquad (8)$$

Where:

$$y(x_k,t)=\int_0^{t} k(t-\tau)u(\tau)d\tau \qquad (9)$$

$$J(x_k,t)=\begin{bmatrix} \frac{\partial y(x_k,t_1)}{\partial x_1} & \frac{\partial y(x_k,t_1)}{\partial x_2} & \cdots & \frac{\partial y(x,t_1)}{\partial x_m} \\ \vdots & & \ddots & \vdots \\ \frac{\partial y(x_k,t_n)}{\partial x_1} & & \cdots & \frac{\partial y(x_k,t_n)}{\partial x_m} \end{bmatrix} \qquad (10)$$

Where

$k=1,2,\ldots p$; p is the number of iteration loops; $J_{n\times m}$ is the Jacobian matrix; $I_{m\times n}$ is the unit matrix; μ_k is the scalar and its value changes during iteration; $x=[x_1,x_2,\ldots,x_m]$ is the model parameters searched for.

In the event that the parameters of the vector are not optimum ones, and the value of error (10) is not at the minimum level. At this point:

$$J^{T}(x_k,t)J(x_k,t)\ll \mu_k I \qquad (11)$$

can be accepted and this leads to the steepest descent method, then we have:

$$x_{k+1}=x_k^{T}-\frac{1}{\mu_k}J^{T}(x_k,t)y(x_k,t) \qquad (12)$$

In the event the value of coefficient μ_k is small, it implies that the values of the parameters of vector x are close to the optimum solution. At this point:

$$J^{T}(x_k,t)J(x_k,t)\gg \mu_k I \qquad (13)$$

which means that the Levenberg-Marquardt algorithm is reduced to the Gauss-Newton method:

$$x_{k+1}=x_k^{T}-[J^{T}(x_k,t)J(x_k,t)]^{-1}J^{T}(x_k,t)y(x_k,t) \qquad (14)$$

Back propagation is utilized to calculate the Jacobian J with respect to the weight and bias variables x. Each variable is adjusted according to Levenberg-Marquardt,

2.3.3 Resilient Back propagation (RBP)

The RBP algorithm just refers to the direction of the gradient. It is a supervised learning strategy. It works in correspond to back propagation, except that the weight updates is done in a different manner. In back propagation the adjustment in weight is determined with partial derivative:

$$\Delta_{wij}(t)=\alpha\times x_i(t)\times\delta_j(t) \qquad (15)$$

where α is the learning rate, $x_i(t)$ represents the inputs propagating back to the i^{th} neuron at time step t, and δ is the corresponding error gradient. Resilient propagation, on the other hand, calculates an individual delta Δ_{ij}, for each connection, which determines the size of the weight update. The following learning rule is applied to calculate delta:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^{+} \times \Delta_{ij}^{(t-1)}, if \ \frac{\partial E^{(t-1)}}{\partial w_{ij}} \times \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^{-} \times \Delta_{ij}^{(t-1)}, if \ \frac{\partial E^{t-1}}{\partial w_{ij}} \times \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)}, else \end{cases} \quad (16)$$

Where $0 < \eta^{-} < 1 < \eta^{+}$

The update-value Δ_{ij}evolves during the learning process based on the sign of the error gradient of the past iteration,$\frac{\partial E^{(t-1)}}{\partial w_{ij}}$and the error gradient of the current iteration, $\frac{\partial E^{(t)}}{\partial w_{ij}}$. Each time the partial derivative (error gradient) of the corresponding weight w_{ij} changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-valueΔ_{ij}isdecreased by the factor η^{-} which is a constant usually with a value of 0.5. If the derivative retains its sign, the update value is slightly increased by the factorη^{+} so as to accelerate convergence in shallow areas.η^{+} is a constant usually with a value of 1.2. If the derivative is 0 then we do not change the update-value. When the update-value is determined for each weight, the

weight-update is then determined. There are two principles to follow to calculate the weight-update. The first principle is that if the present derivative and the past derivative retain their signs then Equation 3 is utilized to calculate the weight-update.

$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)}, \ if \ \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t-1)}, if \ \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, \ else \end{cases} \quad (17)$$

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$$

In the event that the present derivative is a positive value meaning the past value was also a positive value (increasing error), at that point the weight is decreased by the update value. If the present derivative is negative value meaning the past value was also a negative value (decreasing error) then the weight is increased by the update value. The second principle is that if the present derivative and the past derivative have changed

their signs, for example, there was a big step taken then chances are that a minimum was missed. To avoid such big jumps, the weights need to be reverted to the past state.

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)}, if \ \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \quad (18)$$

2.3.4 Scaled Conjugate Gradient (SCG)

The Scaled Conjugated Gradient (SCG) algorithm is a network training function that has the capacity to deal with large scale problems viably. It was created by [31] so as to lessen the computational time, by utilizing LM algorithm method of scaling the step size to avoid the line search per learning iteration [32]. It is a good at managing training pattern recognition networks and large network. The SCG algorithm is gotten from quadratic minimization of objective function E within N iterations [33]. It uses second order information from neural network, similar to LM algorithm. Since the gradient information calculation is inexpensive, only a modest memory is required. The vector sequence as in (19) and (20) is created with initial gradient $g_{initial} = \frac{\delta E}{\delta w}$ and direction vector $d_{initial} = -g_{initial}$, where $w = w_{initial}$, t is the initial time.$t + 1$ is the next iteration time, d is the conjugate direction and H is the Hessian matrix of the objective function E[34].

$$g(t+1) = g(t) + \lambda(t)Hd(t) \quad (19)$$

$$d(t+1) = -g(t+1) + \gamma(t)d(t) \quad (20)$$

$$\lambda(t) = \frac{g(t)^T g(t)}{d(t)^T H d(t)} \quad (21)$$

$$\gamma(t) = \frac{g(t+1)^T g(t+1)}{g(t)^T g(t)} \quad (22)$$

2.3.5 Random Forest

Extension of decision tree methods is random forest, and, very close to Neural Network. It attempts to learn the basis functions from the dataset. The random forest expands decision trees to reduce the variance of the estimates and utilizes a technique bagging (bootstrap aggregating) by training Tdifferent trees given as:

$$y(x) = \sum_{t=1}^{T} \frac{1}{T} y_t(x) \tag{23}$$

Where y_t is the t'th tree.

2.3.6 Technical Indicators

Researchers and investors have used different kinds of technical indicators to monitor the development of stock prices and in setting up trading rules for buy-sell-hold decisions. In this study, five well known technical indicators, for example, RSI, DPO, MACD, EMA, STOCH are selected as input to the model. The technical indicators are determined from historical prices as follows:

Relative Strength Index (RSI)

RSI is a standard momentum indicator which chooses if the stock is overbought or on the other hand oversold. A stock is said to be overbought when the demand freakishly pushes the price upwards. This condition is ordinarily interpreted as a sign that the stock is overvalued and the price is most likely going to go down. A stock is said to be oversold when the price goes down unequivocally to a level beneath its real value. It is a price following an oscillator that extends from 0 to 100, when RSI is above 70, it might show that the stock is overbought when RSI is beneath 30, it might show the stock is oversold.

The mathematical formula for calculating RSI is:

$$RSI = 100 - \frac{100}{1+(\sum_{i=0}^{n-1} \frac{Up_{t-i}}{n})/(\sum_{i=0}^{n-1} Dw_{t-i}/n)} \tag{24}$$

Where Up_t is the upward-price-change and Dw_t is the downward-price-change at time t

Detrended Price Oscillator (DPO)

The Detrended Price Oscillator shows the distinction between a previous price and a simple moving average. Instead of other price oscillators, DPO is not a momentum indicator. It is simply planned to identify cycles with its peaks and troughs. Cycles can be assessed by counting the periods between peaks or troughs.

Detrended price oscillator is calculated by mathematical equation:

$$DPO = C_t - SMA(\frac{n}{2} + 1) \tag{25}$$

Where C_t is the closing price at time t, SMA the simple moving average, $n = 21$

Moving Average Convergence Divergence (MACD)

MACD (Moving Average Convergence/Divergence) is a specialized indicator created by Gerald Appel in the late 1970. It is used to spot changes in the strength, direction, momentum, and duration of a trend in a stock's price. The MACD is a computation of the difference between two exponential moving averages (EMAs) of closing prices. This
distinction is charted after some time, close by a moving average of the contrast. Exponential moving averages include recent changes in a stock's price. By contrasting EMAs of different periods, the MACD line outlines changes in the pattern of a stock. At that point by differentiating that difference to an average, an expert can outline unobtrusive moves in the stocks pattern.

The equation for calculating MACD is:

$$MACD = EMA_{12}(C_t) - EMA_{26}(C_t) \tag{26}$$

$$Signal\ Line = EMA_9(MACD) \tag{27}$$

Where C_t is the closing price at time t, $EMA_n = n =$ 9, 12 and 26 day Exponential Moving Average. When the $MACD$goes below the $Signal\ Line$,it indicates a sell signal. When it goes above the $Signal\ Line$, it indicates a buy signal.

Exponential Moving Average (EMA)

Exponential Moving Average is used correspondingly as the Simple Moving Average, the only refinement being that EMA reacts quicker to recent price values than the simple moving average. EMA is additionally used for processing various different indicators, for example, Moving Average Convergence Divergence (MACD) and so forth.

Exponential moving average of n days is calculated as:

$$EMA_t = C_t \times \propto + EMA_{t-1} \times (1 - \propto) \tag{28}$$

Where EMA_t is the exponential moving average at time t, $\propto$ is the exponential smoothing
Factor given as $\propto = 2 \div (n + 1)$

Stochastic Oscillator (Stoch)

The stochastics oscillator is a momentum indicator that uses support and resistance levels. Dr. George Lane propelled this indicator in the 1950s. The term stochastic implies the the location of a current price in connection to its price extend over a period of time. This strategy endeavours to predict price movements by contrasting the closing price of a security to its price range.

The indicator is calculated as follows:

$$Stoch = \frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100 \qquad (29)$$

Where C_t closing price at time t, LL_t is the lowest low and HH_t highest high in the last t days, respectively.

Performance Evaluation

We assess prediction performance utilizing three measures: Mean Absolute Error (MAE),
Root Mean Square Error (RMSE) and Mean Absolute Scaled Error (MASE).

Mean Absolute Error (MAE)

Consider a set of target or actual returnsr_t^n and their predicted values$\hat{r}_t^n$
MAE is defined as follows:

$$\frac{1}{n}\sum_{n=1}^{n} |r_t^n - \hat{r}_t^n| \qquad (30)$$

Root Mean Square Error (RMSE)

RMSE is defined as:

$$\sqrt{\frac{1}{n}\sum_{n=1}^{n} (r_t^n - \hat{r}_t^n)^2} \qquad (31)$$

Mean Absolute Scaled Error (MASE)

MASE is defined as follows:

$$\frac{1}{n}\sum_{n=1}^{n} \left(\frac{|r_t^n - \hat{r}_t^n|}{\frac{1}{n-m}\sum_{n=m+1}^{n} |r_t^n - r_{t-m}^n|}\right) \qquad (32)$$

Where m is the seasonal period of return r_t^n

2.4 Proposed Ensemble Deep Learning Method (PDLM)

An ensemble learning technique is a machine learning procedure used to achieve better prediction performance by deliberately combining different learning algorithms [35]. There are two essential point of interest brought by ensemble methods.

The initial one is called statistical reason which is identified with absence of adequate data to appropriately represent the data distribution. Ensemble techniques can thus be able to decrease the risk of choosing the wrong model by aggregating all these models.

The second is computational reason. Many learning algorithms, for example, decision tree and neural network, work by performing some type of local search. These techniques can much of the time result in locally optimal solutions. Ensemble methods demonstrate their advantages in this situation by running numerous local search from diverse starting points.

In this research due to the nature of stock market we combine all the best outputs generated by FNN, RNN and CFN trained with LM, RBP and CFN. By analyzing the relationships between these outputs and target output values, we assign each output a corresponding weight value to compute the overall predicted output value. We chose an ensemble of deep learning algorithm composed of FNN, RNN and CFN trained using LM, RBP and CFN and a

Random Forest (RF) as the top layer. Random Forest is very similar to Neural Network and is a supervised learning. The inputs as the outputs of the FNN, RNN and CFN output as the final prediction. Fig. 3 shows the Schematic Diagram of the proposed Ensemble Deep Learning.

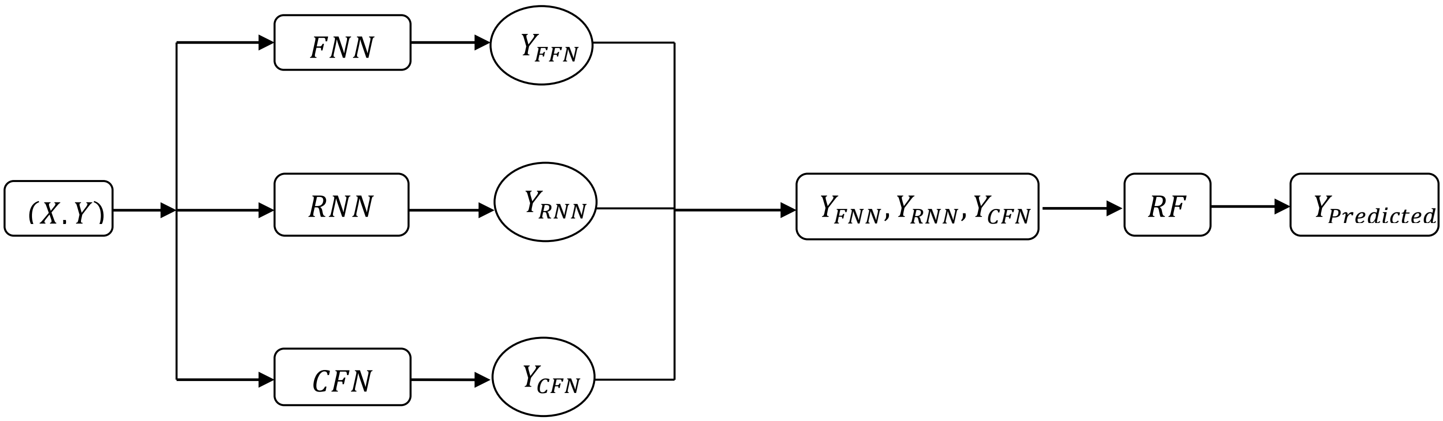

Fig. 3 Schematic Diagram of the proposed Ensemble Deep Learning.

3. Result and Discussion

Here we make an account on how the experiments were conducted and present the results that we obtained. Data from the Nigerian Stock Exchange (NSE) was used for this study and First City Monument Bank (FCMB) and Zenith Bank (ZENITH) was gathered for the period 1 January 2016 to 31 December 2018 for each of the stock. The training data is further
partitioned in a training set and a test set and consists of 708 examples. This gives an input
matrix of dimension 708×5 and a vector of 708 labels.
The training and test data was partitioned as follows:
1 January 2016 – 31 December 2017 is the data for training (66.7%)
1 January 2018- 31 December 2018 is the data for testing (33.2%) as shown

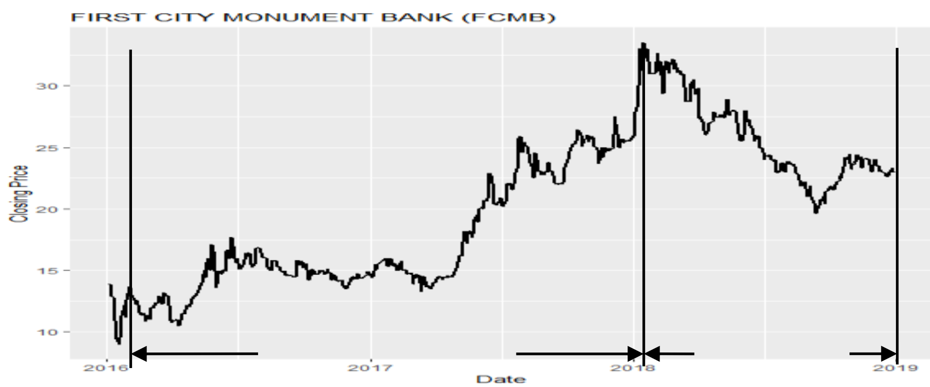

Fig. 4: Daily Closing Price of First City Monument Bank

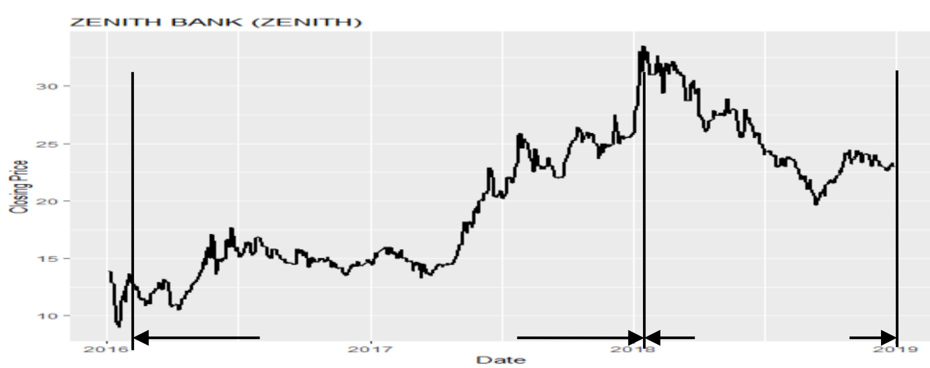

Fig. 5: Daily Closing Price of Zenith Bank

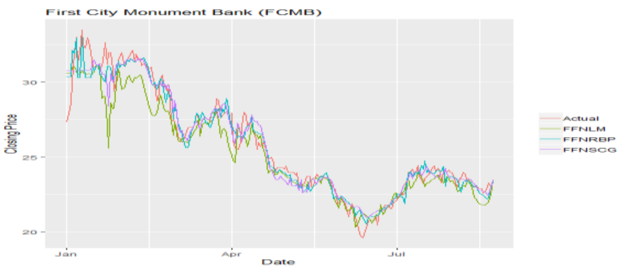

Fig. 6: Actual and Predicted First City Monument Bank Stock Closing Price

The Fig. 6 shows the Actual value of the closing price and the predicted value of the Neural Networks. FFNLM means Feed Forward Neural Network trained with Levenberg Marquardt, FFNRBP means Feed Forward Neural Network trained with Resilient Back Propagation and FFNSCG means Feed Forward Neural Network trained with Scaled Conjugate Gradient.

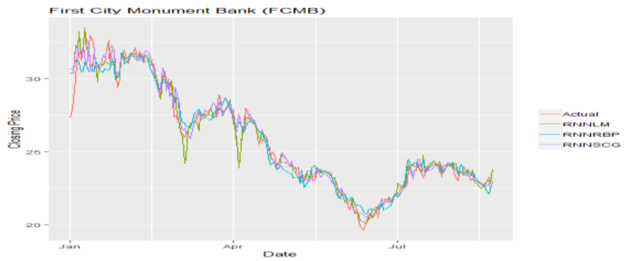

Fig. 7: Actual and Predicted First City Monument Bank Stock Closing Price

The Fig. 7 shows the Actual value of the closing price and the predicted value of the Neural Networks. RNNLM means Recurrent Neural Network trained with Levenberg-Marquardt, RNNRBP means Recurrent Neural Network trained with Resilient Back Propagation and RNNSCG means Recurrent Neural Network trained with Scaled Conjugate Gradient.

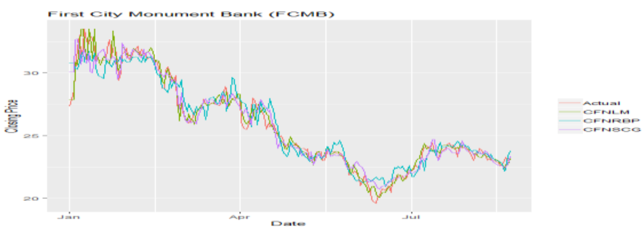

Fig. 8: Actual and Predicted First City Monument Bank Stock Closing Price

The Fig. 8 shows the Actual value of the closing price and the predicted value of the Neural Networks. CFNLM means Cascade Forward Neural Network trained with Levenberg Marquardt, CFNRBP means Cascade Forward Neural Network trained with Resilient Back Propagation and CFNSCG means Cascade Forward Neural Network trained with Scaled Conjugate Gradient.

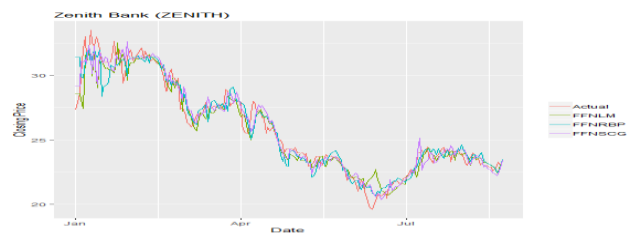

Fig. 9: Actual and Predicted Zenith Bank Stock Closing Price

The Fig. 9 shows the Actual value of the closing price and the predicted value of the Neural Networks. FFNLM means Feed Forward Neural Network trained with Levenberg Marquardt, FFNRBP means Feed Forward Neural Network trained with Resilient Back Propagation and FFNSCG means Feed Forward Neural Network trained with Scaled Conjugate Gradient.

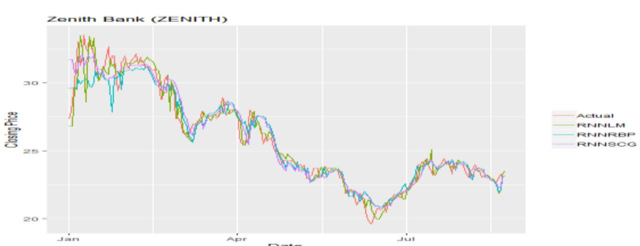

Fig. 10: Actual and Predicted Zenith Bank Stock Closing Price

The Fig. 10 shows the Actual value of the closing price and the predicted value of the Neural Networks. RNNLM means Recurrent Neural Network trained with Levenberg-Marquardt, RNNRBP means Recurrent Neural Network trained with Resilient Back Propagation and RNSCG means Recurrent Neural Network trained with Scaled Conjugate Gradient.

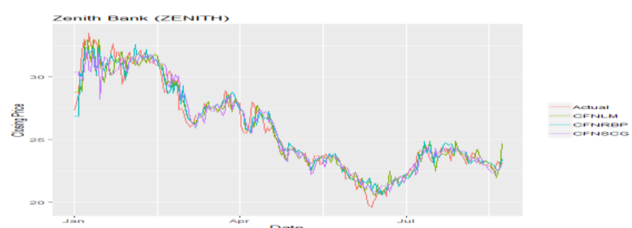

Fig. 11: Actual and Predicted Zenith Bank Stock Closing Price

The Fig. 11 shows the Actual value of the closing price and the predicted value of the Neural Networks. CFNLM means Cascade Forward Neural Network trained with Levenberg Marquardt, CFNRBP means Cascade Forward Neural Network trained with Resilient Back Propagation and CFNSCG means Cascade Forward Neural Network trained with Scaled Conjugate Gradient.

Table 1: Performance Evaluation of Neural Network

			Performance Metrics			
Stocks	DL	Training Algorithm	MAE	RMSE	MASE	R^2
	FFN	LM	0.8505	1.2030	0.4819	0.9227
		RBP*	0.5103	0.7353	0.6637	0.9576
		SCG	0.5891	0.8347	0.7661	0.9455
	RNN	LM*	0.3706	0.5853	0.4819	0.9728
FCMB		RBP	0.5362	0.7481	0.6974	0.9565
		SCG	0.4383	0.6435	0.5701	0.9676
	CFN	LM**	0.3939	0.5456	0.5122	0.9769
		RBP	0.6838	0.9179	0.8893	0.9350
		SCG	0.5107	0.7219	0.6642	0.9589
	FFN	LM	0.6177	0.9086	0.8033	0.9374
		RBP	0.6310	0.8919	0.8206	0.9371
		SCG*	0.5580	0.7915	0.7257	0.9505
	RNN	LM*	0.4899	0.7025	0.6371	0.9609
ZENITH		RBP	0.6511	0.9251	0.8467	0.9373
		SCG	0.5841	0.8556	0.7597	0.9424
	CFN	LM	0.5617	0.7668	0.7305	0.9535
		RBP*	0.4986	0.6921	0.6485	0.9625
		SCG	0.6350	0.8957	0.8258	0.9364

Table 1 shows three Neural Network we used in this research which are Feed Forward Neural Network (FFN), Recurrent Neural Network (RNN) and Cascade Forward Neural Network trained with Levenberg Marquardt (LM), Resilient Back Propagation (RBP) and Scaled Conjugate Gradient (SCG). The performance of each of Neural Network using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Scaled Error (MASE) and R^2 for different Neural Networks. The * indicated the best training algorithms selected in each neural network and ** indicated the overall best predicted value of all the Neural Network which means training Levenberg Marquardt with Cascade Forward Network can achieve best prediction result.

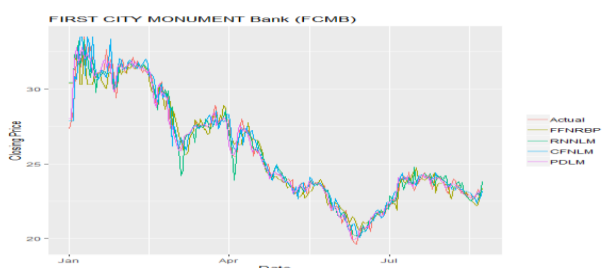

Fig. 12: Actual and Predicted First City Monument Bank Stock Closing Price

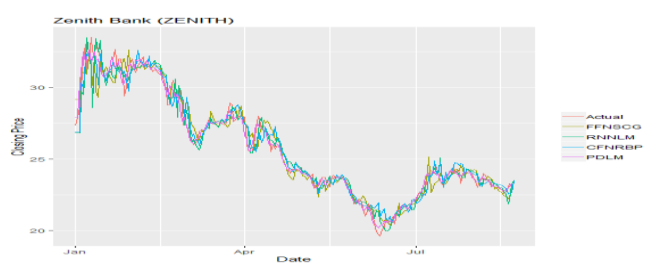

Fig. 13: Actual and Predicted Zenith Bank Stock Closing Price

The Fig. 12 shows the Actual value of the closing price and the predicted value of the Neural Networks. FFNRBP, RNNLM and CFNLM was selected since they perform best among the three training algorithms and PDLM is the proposed Deep Learning algorithms according to section 2.

The Fig. 13 shows the Actual value of the closing price and the predicted value of the Neural Networks. FFNSCG, RNNLM and CFNRBP was selected since they perform best among the three training algorithms and PDLM is the proposed Deep Learning algorithms as explained in section 2.5.

Table 2: Performance Evaluation of Proposed Neural Network

			Performance Metrics			
Stocks	DL	Training Algorithm	MAE	RMSE	MASE	R^2
FCMB	FNN	RBP	0.5103	0.7353	0.6637	0.9576
	RNN	LM	0.3706	0.5853	0.4819	0.9728
	CFN	LM	0.3939	0.5456	0.5122	0.9769
	PDLM	RF	0.1603**	0.2168	0.2084	0.9963
ZENITH	FFN	SCG	0.5580	0.7915	0.7257	0.9505
	RNN	LM	0.4899	0.7025	0.6371	0.9609
	CFN	RBP	0.4986	0.6921	0.6485	0.9625
	PDLM	RF	0.2201**	0.2983	0.2863	0.9931

Since this research is on predicting and beating the stock market we combine the * in table 4.1as the input and the actual values as the target values. Table 4.2 shows an improvement on the proposed algorithms (PDLM) stacked with random forest to perform better results. In both stocks used in this research the R^2 shows a better fit of 99%. The results indicated that with good input we can have best prediction results and better fit.

4. Conclusion

This study has proposed a novel system for developing efficient stock trading strategies, which may provide attractive benefits for investors. The model has integrated technical analysis with machine learning techniques for efficient generation of stock trading decisions. In this study the stock trading decision generation was developed using three deep learning which includes Feed Forward Neural Network (FFN), Recurrent Neural Network (RNN) and Cascade Forward Network (CFN) with training algorithms such as Levenberg-Marquardt, Resilient Back Propagation (RBP) and Scaled Conjugate Gradient (SCG). Table 1 shows that the three Neural Network best fit the test data and from the performance metrics we see that they produced best prediction output results. The computational efficient artificial neural network (PDLM) with random forest learning approach is proposed for generating the stock trading decisions.

From the experimental result analysis it is clearly apparent that the proposed model provides
superior profit percentage compared to the three Neural Network such as Feed Forward Neural Network (FFN), Recurrent Neural Network (RNN) and Cascade Forward Network (CFN) as shown in Table 2.The experimental result shows that our proposed model has 99% accuracy which is higher than the single Neural Network forecasting model.

Hence instead of taking trading decision based on particular technical indicators, it is more
profitable to take trading decision using combination of technical indicators with computational intelligence tools. Furthermore the work can be extended by validating the proposed model over more real world datasets. More work will be done on structure optimization of the model by using efficient optimization algorithms such as differential evolution, particle swarm optimization, genetic algorithms and soon. More technical analysis will also be explored in future.

References

[1] Granger, Clive Wiliam John, and Oskar Morgenstern (1970). Predictability of stock market prices. Heath Lexington Books, 1970.

[2] Marszałek, A., and T. Burczyński(2014). Modeling and forecasting financial time series with ordered fuzzy candlesticks. Information sciences 273 144-155.

[3] Mills, Terence C., and Raphael N. Markellos(2008). The econometric modelling of financial time series. Cambridge University Press, 2008.

[4] Schalkoff, Robert J (1997). Artificial neural networks. Vol. 1. New York: McGraw-Hill, 1997.

[5] Arévalo, Andrés, Jaime Niño, German Hernández, and Javier Sandoval (2016). High-frequency trading strategy based on deep neural networks." In International conference on intelligent computing, pp. 424-436. Springer, Cham, 2016.

[6] Kimoto, Takashi, Kazuo Asakawa, Morio Yoda, and Masakazu Takeoka (1990). Stock market prediction system with modular neural networks." In Neural Networks, 1990., 1990 IJCNN International Joint Conference on, pp. 1-6. IEEE, 1990

[7] Enke, David, and Suraphan Thawornwong(2005). The use of data mining and neural networks for forecasting stock market returns. Expert Systems with applications 29, no. 4 (2005): 927-940

[8] Zhang, Yudong, and Lenan Wu (2009). "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network." Expert systems with applications 36, no. 5 (2009): 8849-8854.

[9] Bollen, Johan, Huina Mao, and Xiaojun Zeng (2011). Twitter mood predicts the stock market." Journal of computational science 2, no. 1 (2011): 1-8.

[10] Wang, Jie, and Jun Wang (2015). "Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks." Neurocomputing 156 (2015): 68-78.

[11] Moghaddam A. H., Moghaddam M. H., and Esfandyari M. (2016). "Stock market index prediction using artificial neural network." Journal ofEconomics, Finance and Administrative Science 21 (41):89-93.

[12] Chen, An-Sing, Mark T. Leung, and Hazem Daouk (2003). "Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index." Computers & Operations Research 30, no. 6 (2003): 901-923.

[13] Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In Neural networks, 2000. ijcnn 2000, proceedings of the ieee-inns-enns international joint conference (Vol. 6, pp. 348-353).

[14] Ince, H., & Trafalis, T. B. (2008). Short term forecasting with support vector machines and application to stock price prediction. International Journal of General Systems , 37 (6), 677- 687.

[15] Meesad, P., & Rasel, R. I. (2013). Predicting stock market price using support vector regression. In Informatics, electronics & vision (iciev), 2013 international conference on (pp. 1-6).

[16] Zbikowski K(2015). Using Volume Weighted Support Vector Machines with walk forward testing and feature selection for the purpose of creating stock trading strategy.Expert Syst Appl. 2015;42(4):1797-1805.

[17] Choudhury S, Ghosh S, Bhattacharya A, Fernandes KJ, Tiwari MK (2014). A real time clustering and SVM based price-volatility prediction for optimal trading strategy. Neurocomputing. 2014;131(1):419-426.

[18] Kumar D, Meghwani SS, Thakur M. Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets.J Comput Sci. 2016;17(1):1-13.

[19] Kim K(2003). Financial time series forecasting using support vector machines. Neuro computing. 2003;55(1-2):307-319.

[20] Gerlein EA, McGinnity M, Belatreche A, Coleman S (2016). Evaluating machine learning classification for financial trading: an empirical approach. Expert Syst Appl. 2016; 54 (1): 193 -207.

[21] Patel J, Shah S, Thakkar P, Kotecha K (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. Expert Syst Appl. 2015a;42(1):259-268.

[22] Arau´jo R, Oliveira AL, Meira S.(2015) A hybrid model for high-frequency stock market forecasting. Expert Syst Appl. 2015;42(8):4081-4096.

[23] Manahov V, Hudson R, Gebka B (2014). Does high frequency trading affect technical analysis and market efficiency? And if so, how? Journal of International financial markets. Inst Money. 2014;28(1):131-157.

[24] Budhani N., Jha C. K., and Budhani S. K. (2014). "Prediction of stock market using artificial neural network. " In Soft Computing Techniques for Engineering and Technology (ICSCTET): 1-8.

[25] Sibi P., Jones S. A. and Siddarth P. (2013). "Analysis of different activation functions using back propagation neural networks." Journal of Theoretical and Applied Information Technology 47 (3): - 1264 1268.

[26] Jia H. (2016). Investigation into the effectiveness of long short term memory networks for stock price prediction.arXiv preprint arXiv :1603.07893.

[27] H. Demuth, M. Beale and M.Hagan (2009). "Neural Network Toolbox User‟ s Guide". The MathWorks, Inc., Natrick, USA. 2009.

[28] Levenberg, K. (1944) A Method for the Solution of Certain Non-Linear Problems in Least Squares. The Quarterly of Applied Mathematics. 2: 164-168.

[29] S. Chatterjee, A. Dash, and S. Bandopadhyay (2014), "Ensemble support vector machine algorithm for reliability estimation of a mining machine,"Quality and Reliability Engineering International, 2014.

[30] Aderibigbe, J. O. (2004). An overview of the Nigeria financial System, Bullion, the publication of the Central Bank of Nigeria, January-December, 28(1): 5-12.
[31] Tran, C., Abraham, A., and Jain, L. (2004). Decision Support Systems Using Hybrid Neurocomputing. Neurocomputing.61: 85-97.
[32] K. Levenberg (1944), " A method for the solution of certain problems in least squares, Quart. Appl. Math., 1944, V ol. 2, pp. 164–168.
[33] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik (1997), Support vector regression machines,Advances in neural information processing systems, vol. 9, pp. 155–161, 1997.
[34] Wei, J. T., Zhang, Z., Barnhill, S. D., Madyastha, K. R., Zhang, H., and Oesterling, J. E. (1988). Understanding Artificial Neural Networks and Exploring Their Potential Applications for The Practicing Urologist. Elsevier Science. 52: 161-172.
[35] S. Chatterjee, A. Dash, and S. Bandopadhyay (2014), "Ensemble support vector machine algorithm for reliability estimation of a mining machine,"Quality and Reliability Engineering International, 2014.
[36] George S Atsalakis and Kimon P Valavanis (2009). 'Surveying stock market forecasting techniques–Part II: Soft computing methods'. In: Expert Systems with Applications 36.3(2009), pp. 5932–5941.
[37] Martin Riedmiller, H. B. 1992. RPROP - A Fast Adaptive Learning Algorithm. Proc. of ISCIS (VII). Universitat.
[38] Aguwamba, S. M. (2005). The role of Nigerian stock exchange in the provision of capital for industrial development, The Nigerian Banker, Journal of the Chartered Institute of Bankers of Nigeria, April-June, 33-38.
[39] Andrew W Lo and A Craig MacKinlay (1988). 'Stock market prices do not follow random walks: Evidence from a simple specification test'. In: Review of financial studies 1.1 (1988), pp. 41–66.
[40] Brownlees C, Gallo G. Financial econometric analysis at ultra-high frequency: data handling concerns. Comput Stat Data Anal.2006;51(4):2232-2245.
[41] C. Cortes and V. Vapnik (1995), "Support-vector networks," Machine learning,vol. 20, no. 3, pp. 273–297, 1995.
[42] D. Marquardt (1963), An algorithm for least-squares estimation of nonlinear parameters, SIAM J. Appl. Math., 1963, V ol. 11, pp. 431–441.
[43] David Oyewola, Emmanuel Gbenga Dada, Ezekiel Olaoluwa Omole and K. A. Al-Mustapha(2019), Predicting Nigerian Stock Returns using Technical Analysis and Machine Learning, European Journal of Electrical and Computer Engineering Vol. 3 (2), 1-8.
[44] Emekekwue, P. (2005). Corporate financial management, Kinshasa, Democratic Republic of Congo, Africa, Bureau of Educational Sciences.
[45] Eugene F Fama (1995). 'Random walks in stock market prices'. In: Financial analysts journal 51.1 (1995), pp. 75–80.
[46] Eunsuk Chong, Chulwoo Han, FrankC.Park (2017). Deep learning networks for stock Market Analysis and prediction: Methodology,data representations, and case studies, Expert Systems with Applications, 83, 187-205.
[47] Johnson, N.F., P. Jefferies and P.M. Hui (2003), Financial Market Complexity, Oxford U. Press, N.Y., 2003.
[48] J. C. Sousa, H. M. Jorge, and L. P. Neves (2014), "Short-term load forecasting based on support vector regression and load profiling," International Journal of Energy Research, vol. 38, no. 3, pp. 350–362, 2014
[49] Luckyson Khaidem, Snehanshu Saha and Sudeepa Roy Dey (2016). Predicting the Direction of stock market prices using random forest. Applied Mathematical Finance 1-20.
[50] Moghaddam, Amin Hedayati, Moein Hedayati Moghaddam, and Morteza Esfandyari (2016). Stock market index prediction using artificial neural network." Journal of Economics, Finance and Administrative Science 21, no. 41 (2016): 89-93.
[51] Martin. Fodslette. Moller. (1993) A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Neural Network. 6: 525-533.
[52] Rajashree Dash and Pradipta Kishore Dash (2016).A hybrid stock trading framework integrating technical analysis with machine learning techniques. The Journal of Finance and Data Science 2 (2016) 42-57.

[53] Olagunde, A.O, D. O Elumilade & T .O Asolu (2006). Stock market capitalization and interest rate in Nigeria: A time series analysis, International Research Journal of Finance and Economics, 4:154-166.
[54] Qu H, Zhang Y(2016). A new kernel of support vector regression for forecasting high-frequency stock returns. Math Probl Eng. 2016;2016(1):1-9.
[55] T. G. Dietterich (2000), "Ensemble methods in machine learning," in Multiple classifier systems. Springer, 2000, pp. 1–15.
[56] V. Vapnik (1998), Statistical learning theory, New York: Wiley, pp. 736, 1998.