Precision forecasting for intraday and daily stock prices with neural networks and fuzzy models.

[1]Shilpa Amit Verma, [2]G.T. Thampi, [3]Madhuri Rao
[1]Associate Professor, [2]Professor and Principal, [3]Professor and HOD (IT)
[1]Computer Engineering Department,
[1]Thadomal Shahani Engineering College, Bandra, Mumbai, India

Abstract: Precision financial market forecasting plays an important role in financial prediction and investing. There are various global, local and psychological factors that affect financial market forecasting making it a non-trivial, complex problem. Numerous soft computing techniques based on application of Artificial Neural Network (ANN), Fuzzy Systems (FS) and hybrid models, employing a combination of varied models have been applied by various researchers for forecasting stock prices, with modest results. In the present work, the time-series data, from ICICI Bank Ltd, obtained from the National Stock Exchange (NSE), India is considered as a reference data to demonstrate precise prediction on application of varied ANN and fuzzy algorithms. A thorough and detailed comparison of these algorithms under the ANN & Fuzzy domains, on widely varying time scales - daily price forecasting on one-year data and minute to minute forecasting on intraday data, has been attempted. Forecasting accuracies on the optimized methods has resulted in better than 99% accuracy.

***IndexTerms* - Stock Market, Forecasting, Efficacy Parameters, ANN, ANFIS.**

I. INTRODUCTION

Artificial Intelligence methods are being increasingly applied in various fields of science and engineering. In the last few decades, artificial intelligence (AI) methods have especially found favour in various forecasting problems. Rather than just applying the conventional backpropagation ANN methods, researchers are now studying other optimization methods in the AI domain to select the most optimized method for the problem in hand. In the present work we have undertaken an in-depth study for the stock market forecasting using ANN, and hybrid of ANN and fuzzy methods. Share price of a stock depends on several factors like the financial performance of the respective company, confidence level of the shareholders in brand management and its demand/supply at any given time. Forecasting of share price movements in stock exchanges has historically been a challenging problem mainly due to their nonlinear, volatile and heteroscedastic nature in time, more so in developing economies than in developed ones. This has been a subject of intense research in recent past, though there is some progress in forecasting accuracies since the days of efficient market hypothesis and random walk hypothesis which postulated against accurate forecast of share prices. Researchers, who did not believe in the above hypotheses, however continued with their efforts of prediction employing both classical statistical methods and modern techniques like the AI methods obtaining increasingly better forecasts. AI techniques such as artificial neural networks (ANN), fuzzy systems (FS) and their hybrids such as the adaptive neuro fuzzy inference systems (ANFIS) are being increasingly used to make more accurate forecasts. The reason apparently being their superiority in capturing non-linearity, volatility and heteroscedasticity of the price data as compared to earlier methods. Hybrid methods such as ANFIS combine advantages both of ANN, the ability to adjust structurally to the given data and of FS, the ability to handle uncertainty of the data well. Higher degree of accuracy attained in forecasting is critically important in the selection of the methods. In the present study both the approaches of ANN and ANFIS have been applied to the time series data from ICICI Bank listed in NSE. The choice of the stock data being rather arbitrary, the main consideration was to choose a highly liquid and volatile stock so that capability of the ANN & FS methods employed can be put to test rather rigorously. Though there is significant literature on ANN applications on financial data, however, for the first time, we report highly accurate and precise predictions for an Indian stock.

AI techniques have been applied in numerous fields. A few of those are mentioned here. E. A. Mlybari et al. employed support vector machines (SVM) to predict daily tidal levels along the Jeddah Coast, Saudi Arabia [1]. Hossein Mombeini et al. employed an ANN model to forecast gold prices [2]. V.K. Dhar et al. have done an inter-comparison of several ANN algorithms on benchmark and function approximation problems [3]. In another AI application time series forecasting of daily network traffic, Haviluddin et al. used Radial Basis Function Neural Network (RBFNN) achieving acceptable accuracy [4]. Hyunjung Shin et al. employed a semi-supervised learning algorithm (SSL) to forecast direction of movement in crude oil prices [5]. In yet another work an ANN model was developed by Deepak Singhal et al. to predict one-day ahead energy market-clearing prices [6].

II. PARAMETERS FOR EVALUATING EFFICACY

We denote actual stock prices and the forecasted stock prices by y_d and y respectively. The error in the forecasted price is then defined by "$e = (y_d – y)$". Let n denote the total number of input-output data sets constructed from the stock prices, and let "y_{mean}" denote the mean of all the actual stock prices i.e. the desired output values. The parameters listed below are considered in the present work to investigate efficacy of the employed AI methods in stock prices forecasting.

1) Mean Error (ME) $= \frac{1}{n}\sum_{t=1}^{n} e_t$

2) Mean Absolute Deviation (MAD) $= \frac{1}{n}\sum_{t=1}^{n} |e_t|$

3) Mean Square Error (MSE) $= \frac{1}{n}\sum_{t=1}^{n} (e_t)^2$

4) Percent Forecast Error (% FE) $= \left(\frac{e}{y_d}\right) * 100$

5) Mean Percent Forecast Error (MPE) $= \frac{1}{n}\sum_{t=1}^{n} \%FE_t$

6) Mean Absolute Percent Forecast Error (MAPE) $= \frac{1}{n}\sum_{t=1}^{n} |\%FE_t|$

7) Coefficient of determination $R^2 = 1 - \frac{\sum(y_d - y)^2}{\sum(y_d - y_{mean})^2}$

8) Mean Percent Accuracy (MPA) $= 100 - \text{MAPE}$

III. REVIEW OF FORECASTING EFFICIENCIES ACHIEVED BY RESEARCHERS

Stock price forecast accuracies reported in relevant literature employing AI methods are briefly reviewed here for a comparison with our study. Kyoung-jae Kim et al. (2000) used Genetic Algorithm (GA) approach for feature selection in ANN for forecasting the direction of movement of the Tokyo stock exchange prices index (TOPIX) and achieved accuracies ranging between 52% and 62% [7]. Kyoung-jae Kim (2003) used a support vector machine (SVM) algorithm to forecast direction of daily price change in the Korean Stock market. The predictions of SVM were found to be better (58%) than the other two methods based on backpropagation (55%) and case-based reasoning (CBR) (52%) [8]. Kyoung-jae Kim (2006) predicted direction of movement of the South Korean market index (KOSPI) using ANNs where instance selection was done using a GA achieving accuracies ranging between 59% to 65% [9]. Ebrahim Abbasi et al. (2008) designed an ANFIS model to investigate current trend of stock prices of "IRAN KHODRO Corporation" at Tehran Stock Exchange. MAPE as low as 0.9 could be achieved [10]. George S. Atsalakis et al. employed an ANFIS expert system to predict changes in trends of the stock prices of the National Bank of Greece and the General Electric achieving an accuracy of about 68% [11]. Melek Acar Boyacioglu et al. (2010) successfully forecasted the monthly return of the Istanbul Stock Exchange (ISE) National 100 Index with an accuracy rate of 98.3% [12]. Esmaeil Hadavandi et al. (2010) combined a genetic fuzzy systems (GFS) and ANN to predict direction of share price movement of the stocks of IBM, Dell, British airline and Ryan airline achieving (MAPE) ranging from 0.63 to 1.9 [13]. Yakup Kara et al. predicted (2010) direction of Istanbul Stock market index (ISE). ANN model gave an accuracy of approximately 76% as compared to the SVM Model which gave an accuracy of 72% [14]. Jonathan L. Ticknor (2013) proposed a Bayesian regularized ANN to predict the one day future closing prices of the stocks of Microsoft and Goldman Sachs achieving MAPE ranging from 1.06 to 1.3 [15]. Amin Hedayati Moghaddam et al. (2016) investigated the ability of their back propagation network (BPNN), using various training and transfer functions, in forecasting the daily NASDAQ stock exchange index. The performance of ANNs was evaluated using the determination coefficient (R^2). Values of R^2 as high as 0.9622 were obtained [16]. Aparna Nayak et al. (2016) attempted to predict stock market trends. Supervised machine learning algorithms were used to build the models using historical prices combined with sentiments from social media obtaining up to 70% of accuracy [17]. Amin Hedayati Moghaddam et al. (2016), examined ability of several feed forward ANNs forecasting daily trends of NASDAQ stock exchange rates. Values as high as 0.96 could be attained for the determination coefficient (R2) for some models [18]. Siham Abdulmalik et al. forecasted Russian Trading System Index (RTSI) using fuzzy inference rules, achieving a confidence level of about 90% in the condition of incomplete initial data [19]. In the work done by Devadoss, closing share prices of Tata Consultancy Services Ltd, Wipro Ltd, Dr. Reddy’s Laboratories Ltd and Sun Pharmaceutical Ltd. listed under the Bombay Stock Exchange (BSE) were satisfactorily predicted using standard Backpropagation algorithm of ANN achieving MAPE ranging between 1.1% and 11.7% [20]. Yunus Yetis et al. (2014) employed a generalized feed forward based ANN model to predict NASDAQ's stock values obtaining an error 2% [21]. Mingyue Qiu et al., used BPN combined with genetic algorithm GA and simulated annealing (SA) to predict the returns of the Neikki 225 index. MSE as low as 0.0043 was attained [22]. Sachin Kamley et al. reviewed performance of various machine learning techniques giving share market forecasting accuracies ranging between 88 to 97 % during a period of 15 years [23].

IV. DAILY AND INTRADAY PRICE VARIATIONS OF STOCK PRICE

As mentioned earlier, highly liquid and volatile stock, ICICI Bank, of NSE was chosen [24] for the present study. Two time frames were considered for the same: a) daily opening prices in a trading year (from 1st January 2015 to 31st December 2015) a total of 249 prices on 249 days (events) and b) minute to minute prices in a trading day (from 9.15 am to 3.30 pm on 15th September 2015) a total of 376 prices on 376 minutes (events). The two price variations are shown in Figure 1 and 2 below.

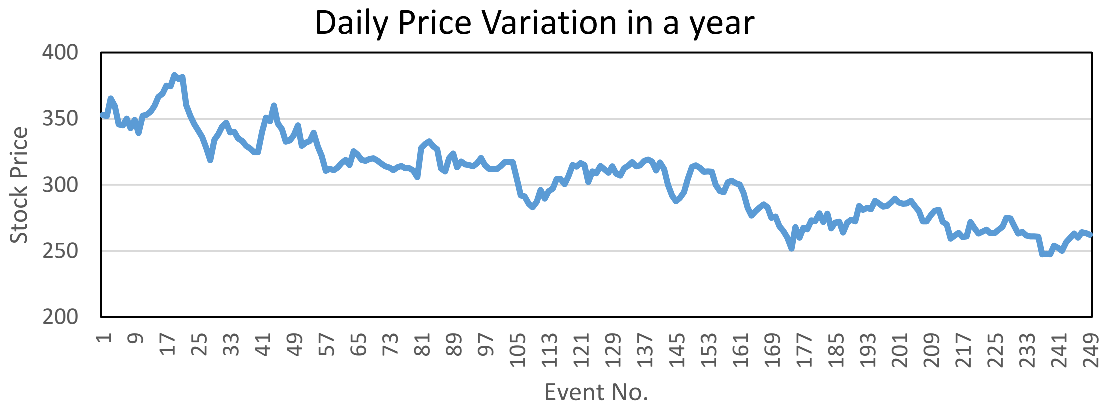

Figure 1. Plot of daily open price of ICICI Bank stock for a period of one year (Jan to Dec 2015)

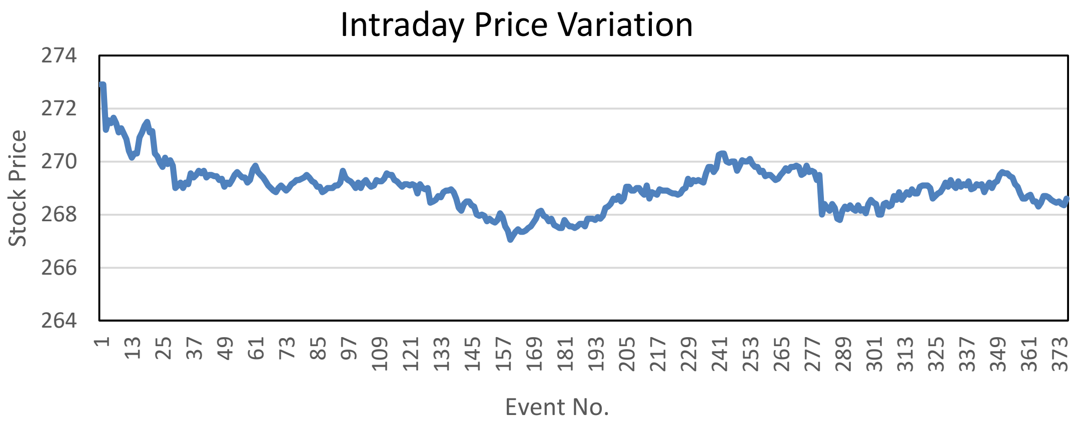

Figure 2. Plot of minute to minute share prices of ICICI Bank in a trading day (15th September 2015)

V. FORECAST OF STOCK PRICES

The forecasting efficacy of some prominent AI techniques, within the ANN domain and ANFIS technique within Fuzzy System domain, for data obtained from NSE is presented. A very brief introduction to the techniques is provided here. The reader is referred to the references provided for details on these methods.

5.1 ANN Technique

The backpropagation (BP) algorithm that was introduced by Rumelhart is the well-known method for training multilayer feed-forward artificial neural networks [25]. The Multi-layer perceptron (MLP) networks trained by back propagation (BP) algorithm have been a popular choice in ANN applications in finance [26]. Figure 3 below illustrates the basic multilevel perceptron (MLP) network without using biases.

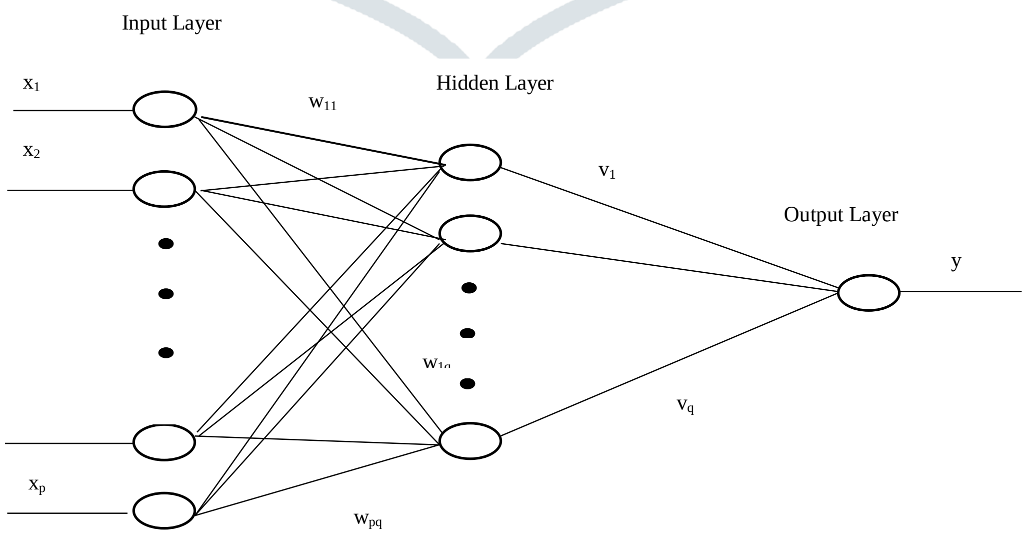

Figure 3. ANN/MLP network

The MLP consists of three types of layers. The first layer is the input layer and corresponds to the input variables of the problem with one node for each input variable. For ‘p’ input variables, ‘p’ nodes are required in the input layer. The second layer is the hidden layer, containing ‘q’ nodes, and it helps to capture non-linear relationships among variables. The third layer is the output layer. The number of nodes required in this layer is equal to the number of outputs for each set of ‘p’ inputs. In the present problem there is only one output and therefore only one node is required in the output layer. The relationship between the output y and the input vector x is given by:

$$y = \sum_{j=1}^{q} v_j \cdot f\left(\sum_{i=1}^{p} w_{i,j} \cdot x_i\right) \quad (1)$$

where $w_{i,j}$ (i=0,1,2,…,p; j=1,2,…,q) and v_j (j=0,1,2,…,q) are the connection weights. The nonlinear activation function ‘f’ enables the network to learn nonlinear features of the input-output interdependence. The most widely used activation functions for the output layer are the sigmoid and hyperbolic functions. In this paper, the hyperbolic tangent transfer function is employed and is defined as:

$$\tanh(r) = \frac{e^r - e^{-r}}{e^r + e^{-r}} \quad (2)$$

The MLP is trained using the BP algorithm and the weights are optimized. The objective function to be minimized is the sum of the squares of the differences between the desirable output (y_d) and the predicted output (y) and is given by:

$$E = 0.5 \sum_{t=1}^{n} (y_d - y)_t^{\;2} = 0.5 \sum_{t=1}^{n} e_t^{\;2} \quad (3)$$

The training of the network is performed by BP algorithm trained with the steepest descent algorithm given in Eq. (4), m denoting the iteration index

$$\Delta w_m = -\alpha_m . g_m \quad (4)$$

The BP algorithm suffers from slow convergence and traps in local minima of error function. Several modifications have been introduced by eminent researchers for improving the local minima problem. These are: gradient descent with momentum, gradient descent with adaptive learning rate, gradient descent with both momentum and adaptive learning rate and resilient backpropagation [27]. Some other methods are based on variations of numerical optimization technique itself. These variations formed bases of Conjugate-Gradient, Quasi-Newton, and Levenberg-Marquardt algorithms. Conjugate-Gradient algorithms perform searches along conjugate directions requiring calculation of second derivatives. Nevertheless its convergence is generally faster than that of the steepest descent direction method [28]. Quasi-Netwon method may converge still faster than conjugate gradient methods since it avoids calculation of second derivatives. It approximates the Hessian matrix and updates it at each iteration. The Levenberg-Marquardt method interpolates between the Gauss-Newton and the gradient descent methods. It also avoids calculation of the Hessian Matrix by replacing it by the Jacobian needing lesser computation [29]. Some details of these methods can be found in the work done by V.K. Dhar et al. [30] and Lahmiri [31]. AI methods used in the present work are briefly discussed below in Table 1.

Table 1. Description of ANN Methods used

ANN Method	Formulation	Explanation
Gradient Descent Method with momentum (GDM)	$\Delta w_m = -\alpha_m . g_m + p\Delta w_{m-1}$ where, Δw_m is a vector of weight changes g_m is the current gradient α_m is the learning rate p is the momentum parameter	The weights and biases are updated in the negative gradient direction of the performance function.
Resilient Back Propagation (RP)	$\Delta w_m = -sign\left(\frac{\Delta E_m}{\Delta w_m}\right)\Delta_m$ where, E is the error vector	The sign of the partial derivative determines the direction of updating the weight.
Scaled Conjugate Gradient Method (SCG)	$\Delta w_m = \alpha_m . p_m$ $p_m = -g_m + \beta_m p_{m-1}$ $\beta_m = \frac{g_m^T\, y_m}{d_{m-1}^T\, y_m}$ where,	α is the optimal current search direction found out by search line method. β is the next search direction found in such a manner

	d_m is the conjugate gradient search direction	that it is conjugate to the previous search directions.
Levenberg- Marquadt (LM)	$\Delta w_m = -H'_m . g_m$ $H' = J'J$ $g = J'e$ where, H is the Hessian matrix(second derivatives) J is the Jacobian matrix(first derivatives) e is vector of network errors	It is an iterative method for locating minimum of a function. It may be considered as an ideal combination of steepest descent and Gauss Newton methods.

5.2 ANFIS Technique

Artificial neural networks (ANN) are trained to learn and recognize patterns by adapting the connecting weights called learning. Fuzzy inference systems (FIS) are designed to act somewhat akin to human inference. Behavior of fuzzy systems is designed by simple fuzzy rules stated as: IF premise THEN consequent. A fuzzy inference system consists of three main parts. In the first part, crisp inputs (e.g. stock prices in the present study) are described by specifying a few number and type of membership functions (MF). This procedure is defined as fuzzification. In the second part the above fuzzified system frames the rule base depending on the input strength. Finally in the third part defuzzification is performed to convert the fuzzy output back to crisp numerical values. Integration of these two systems, ANN and FIS, forms the so called neuro fuzzy system. Out of several types of such systems, the most commonly employed and popular method is the ANFIS, wherein applications of fuzzy rule based systems are implemented within the framework of neural networks.

Structure of an ANFIS model consists of various nodes and directional links connecting them. Some of the nodes are adaptive (square node in the figure) and assigned with membership functions having parameter sets affecting their outputs. Other nodes (circle nodes in the figure) are not bound by any parameters and perform simple operations like taking product of the inputs or normalizing them. Each input is assigned a chosen type of a membership function and also the number of such functions. Various MFs used in this study are Triangular MF (Trimf), Generalised Bell MF (Bellmf), Gaussian MF (Gaussmf) and Difference of Sigmoidal MF (Dsigmf). For example an input (stock price) may be represented by three Bellmfs, each representing the price to some extent. The input space of the given data is thus partitioned according to type and number of membership functions selected for the inputs. The total number of fuzzy rules n_r, one for each partition, thus equals the total number of the equations of the type Eq. (5) and is given by $(n_{mf})^{ni}$, where n_i is the number of inputs and n_{mf} is the number of membership functions associated with each input. In order to determine a suitable set of parameters associated with the rules adaptive network concepts are utilized. Figure 4 below shows the architecture of a 2 input-1 output, 5 layered ANFIS as originally proposed by Jyh-Shing Roger Jang [32].

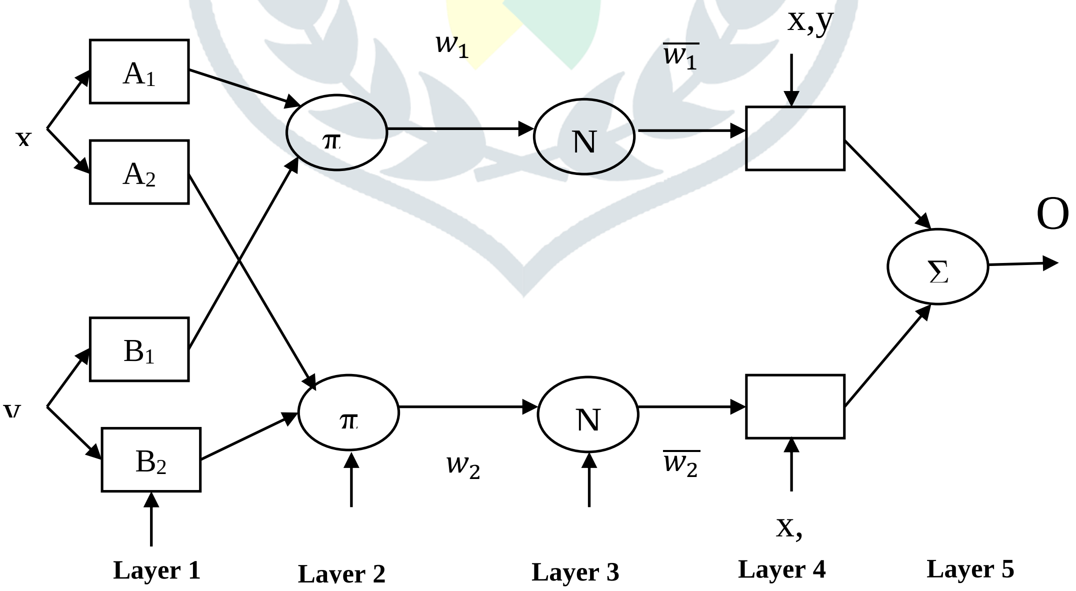

Figure 4. Architecture of a 5-layer ANFIS model.

Of the many types of ANFIS the one used here in the present work is a first-order Sugeno fuzzy model having the following form:

$$\text{If input x is A and input y is B then output } O = px + qy + r \tag{5}$$

Here A and B are fuzzy sets giving fuzzy equivalents of the crisp inputs x and y. O is the crisp output. Layer wise description of its working principle follows.

Layer - 1:
Every node in this layer is shown by a square node. The associated membership function O_i^1 (the node output) is defined as:

$$O_i^1\ (x) = f(x) \tag{6}$$

Where x is the input to the node i of this layer and f(x) is a membership function. In the present work several types of membership functions, as given in Table 2, have been employed.

Layer - 2:
This layers consists of circle nodes with the function of finding and passing the product W_i, the firing strength of a rule, of the incoming signals O_i^1 to the next layer.

$$W_i = O_i^1\ (x) * O_i^1\ (y) \tag{7}$$

Layer - 3:
The nodes in this layer normalize the outputs of all the nodes of the earlier layer as follows:

$$\overline{W}_i = W_i / \textstyle\sum_i W_i \tag{8}$$

The resultant output $\overline{W}_i$ produced by this layer is termed as the normalized firing strength.

Layer - 4:
This layer consists of square nodes with a node function

$$O_i^4 = \overline{W}_i * f_i = \overline{W}_i * (p_i x + q_i y + r_i) \tag{9}$$

where p_i, q_i, r_i are the node parameters referred to as consequent parameters.

Layer-V:
This layer has only one summation node computing the net output.

$$O_i^5 = \textstyle\sum_i \overline{W}_i * f_i \tag{10}$$

The training of ANFIS is quite similar to that of ANN. The error is computed as the difference between O_i^5 and the desired output. In the forward pass, least squares estimate method is used to identify consequent parameters. The error signals, which are the derivatives of the squared error with respect to each node output, propagate backward from the output layer to the input layer updating the premise parameters by the gradient descent algorithm in the backward pass.

Table 2: Types of Membership functions in an ANFIS

Membership Function	Expression	Explanation
Gaussmf	$e^{\frac{-(x-c)^2}{2\sigma^2}}$	Parameters c and σ define the location and width of the function
Bellmf	$\frac{1}{1+\ ((x-c_i)/a_i)^{2*b_i}}$	Parameters a and c define the location and width of the function and b is a positive number
Trimf	$\max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$	The parameters a and c locate the feet of the triangle and b locates the peak.
Dsigmf	$\left\{\frac{1}{1+\ e^{-a_1(x-c_1)}} - \frac{1}{1+\ e^{-a_2(x-c_2)}}\right\}$	Difference between two sigmoidal membership function

VI. RESEARCH METHODOLOGY

In the present work historical normalized share prices of the stock are used to forecast next day's or next minute's stock prices. Forecasting is done in two time horizons described below.

6.1 Daily Forecasting:

In this time horizon opening prices of each trading day of full one trading year, from 1st Jan 2015 to 31st Dec 2015, a total of 249 numbers were downloaded from a NSE site. Time series price patterns are formed from this data in different ways or strategies. Under a '5in_1out' strategy a price pattern is formed by taking the first 5 days opening prices as 5 inputs and the 6th day opening price as the corresponding 1 output. The next pattern is made by shifting the price window by 1 day till all the downloaded 249 price data gets exhausted. Out of total such patterns formed, a set of about 70% (chosen randomly to avoid any bias) is used for training the network and the remaining 30% for the testing. Similarly training and testing sets of '7in_1out' and '10in_1out' strategies are formed.

6.2 Intraday Forecasting:

For this forecasting trading prices at each minute in one full trading day (from 9.15 a.m. to 3.30 p.m.) of 15th Sep 2015, a total of 376 numbers, were downloaded from Kerala Bank site [33].Patterns of minute to minute prices are formed and divided between training and testing sets in the same manner as above.

Forecasting efficacy of both the approaches, ANN and ANFIS, is aimed to be examined using the above mentioned price patterns. Four types of algorithms of ANN namely that of Gradient Descent with momentum, Resilient Propagation, Scaled Conjugate Gradient and the Levenberg-Marquadt were chosen in the ANN approach. In the ANFIS four types of fuzzy membership functions (MF) namely Trimf, Bellmf, Gaussmf and Dsigmf were used. Each of the inputs was fuzzified by using 2 or 3 MFs. The program MATLAB_R_2013a version was employed for this work [34].

VII. RESULTS AND DISCUSSION

Results obtained in both the time frames with the ANN and ANFIS approaches are presented below.

7.1 Daily Forecasting

The trading year considered in the present study consisted of 249 trading days as mentioned above. The results using various strategies with ANN and ANFIS algorithms are discussed below.

7.1.1 ANN approach

Once the database i.e. all the sets of patterns is ready along the lines described above, GDM is employed on the training and testing sets for all the strategies, i.e. '5in_1out', '7in_1out' and '10in_1out' to choose the one giving the least MSE. The strategy '5in_1out' turns out to be optimum. All the other three ANN algorithms namely, RP, SCG and LM are then employed for the so chosen strategy for various number of hidden neurons i.e. 3, 5 and 7 to eventually find out the best possible forecast for the stock under consideration in the day-to-day time horizon. The results are summarized below in Table 3.

Table 3. Daily Forecasting Performance of various configurations of ANN using GDM, RP, SCG and LM

Training Algorithm	No. of neurons in hidden layer	Training		Testing	
		MSE	%Acc	MSE	%Acc
GDM	3	0.022908	82.93	0.022736	83.86
	5	0.00143	96.21	0.00196	95.08
	10	0.00215	95.68	0.00222	95.28
RP	3	0.000212	98.58	0.000286	98.21
	5	0.000323	98.22	0.000412	97.94
	10	0.000245	98.44	0.000312	98.14
SCG	***3***	***0.000193***	***98.64***	***0.000275***	***98.27***
	5	0.000180	98.52	0.000298	98.18
	10	0.000186	98.51	0.000276	98.21
LM	3	0.000161	98.76	0.00034	98.18
	5	0.00014	98.92	0.000313	98.21

	10	0.000089	99.09	0.006288	96.18

As can be seen from the table, the best performance for one day ahead forecasting is obtained when the SCG algorithm is chosen for the training and 3 neurons are taken in the hidden layer. This gives an accuracy of 98.27% in the testing phase. Figure 5 compares the desired and the forecasted output at various test points (event numbers).

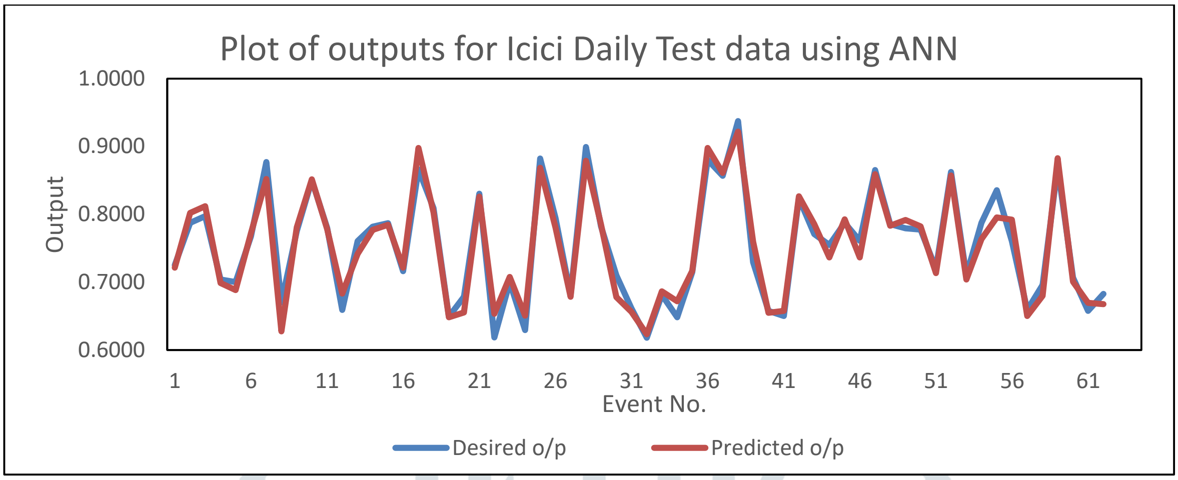

Figure 5. Plot of desired and forecasted outputs for ICICI daily test data for SCG with 3 neurons

7.1.2 ANFIS approach

Both daily and intraday forecasting has also been attempted using the ANFIS approach. Various structures of ANFIS were deployed with different type and number of membership functions. The same sets of patterns as those for the ANN approach were used. Many structures of ANFIS have been tested with different numbers (2 and 3) and types of MFs (Trimf, Bell, Gauss and DSig). The results are summarized in Table 4. Best performance is obtained when 3 Bell MFs represent the 5 inputs, a constant or linear MF is used at the output and a BP learning method is employed. This gives an accuracy of 98.16% in the testing phase. Figure 6 plots the desired and forecasted outputs at each test point.

Table 4. Forecasting Performance of various configurations of ANFIS using 3 membership functions for each input for ICICI daily data.

Input mf type	**Output mf type**	**Optimization method**	**Training**		**Testing**	
			Mse	**% acc**	**mse**	**% acc**
Trimf*	Constant	Hybrid	9.8E-05	99.3	4.2E-03	95.09
	Constant	Backprop	5.6E-03	96.81	7.9E-03	94.52
	Linear	Hybrid	5.9E-05	99.6	6.7E-03	94.23
	Linear	Backprop	5.6E-03	97.38	7.7E-04	97.12
Bellmf	Constant	Hybrid	6.9E-04	99.5	3.0E-03	96
	Constant	***Backprop***	***2.6E-04***	***98.36***	***2.9E-04***	***98.16***
	Linear	Hybrid	4.4E-05	99.6	5.9E-03	94.8
	Linear	***Backprop***	***2.6E-04***	***98.36***	***2.9E-04***	***98.16***
Gaussmf	Constant	Hybrid	8.6E-05	99.4	3.9E-03	95.12
	Constant	Backprop	9.9E-04	97.2	1.3E-03	97.8
	Linear	Hybrid	5.2E-05	99.6	6.1E-03	94.8
	Linear	Backprop	4.7E-04	97.75	6.2E-04	97.27
Dsigmf	Constant	Hybrid	1.0E-04	98.3	1.5E-02	93
	Constant	Backprop	1.2E-03	97.2	1.5E-03	97.4

	Linear	Hybrid	3.5E-05	99.7	4.9E-03	95
	Linear	Backprop	4.0E-04	98.1	5.0E-04	97.8

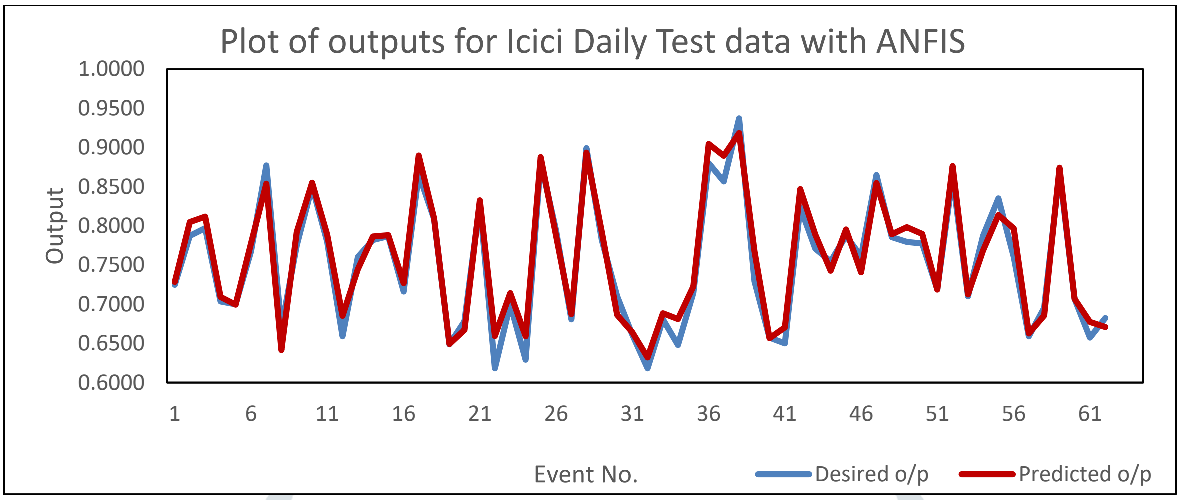

Figure 6. Plot of desired and forecasted outputs obtained by ANFIS for ICICI daily test data

7.2 Intraday Forecasting

The trading day considered in the present study consisted of 376 trading minutes as mentioned earlier. The results, using various strategies with ANN and ANFIS algorithms, are discussed below.

7.2.1 ANN Approach

The results of intraday forecasting obtained by employing all the four ANN algorithms for the '5in_1out' strategy are summarized below in Table 5.

Table 5. Intraday Forecasting Performance of various configurations of ANN using GDM, RP, SCG and LM.

Training Algorithm	**No. of neurons in hidden layer**	**Training** MSE	**%Acc**	**Testing** MSE	**%Acc**
GDM	3	1.11E-02	86.98	1.08E-02	87.37
	5	2.42E-02	89.12	2.12E-02	88.38
	7	3.64E-02	80.2	3.94E-02	79.1
RP	3	6.46E-05	99.2	7.85E-05	99.21
	5	7.35E-05	99.12	7.4E-05	99.06
	7	2.22E-03	95.2	1.85E-03	95.22
SCG	3	1.52E-06	99.85	2.039E-06	99.84
	5	1.35E-05	99.75	6.59E-06	99.76
	7	4.48E-05	99.38	8.10E-05	99.26
LM	***3***	***2.48E-07***	***99.95***	***2.58E-07***	***99.94***
	5	2.11E-07	99.95	2.5E-07	99.94
	7	2.35E-07	99.95	2.35E-07	99.94

As can be seen from Table 5, LM gives the most accurate (99.94%) forecast for all the architectures. The best performing architecture can however be considered the one corresponding to 3 hidden neurons as it gives the most compact network. Figure 7 compares the desired and the forecasted output at various test points.

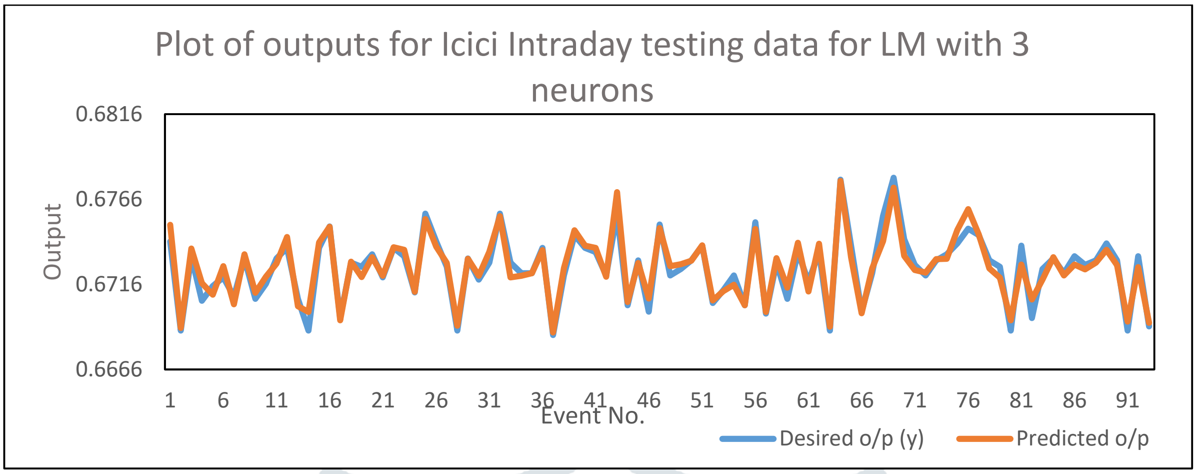

Figure 7. Plot of desired and forecasted outputs for ICICI intraday test data for LM with 3 neurons

7.2.2 ANFIS Approach

Intraday forecasting has also been attempted using the ANFIS methodology. In an approach similar to the one adopted for daily forecasting using ANFIS, various structures of ANFIS have been applied by varying the number of membership functions used to represent each of the 5 inputs and their types. These are tabulated in Table 6 below.

Table 6. Forecasting Performance of various configurations of ANFIS using 2 membership functions for each input for ICICI intraday data

Input mf type	**Output mf type**	**Optimization method**	**Training**		**Testing**	
			MSE	**% acc**	**MSE**	**% acc**
Trimf	Constant	Hybrid	2.1E-07	99.95	3.2E-07	99.94
	Constant	Backprop	2.8E-01	24.5	2.6E-01	24.15
	Linear	Hybrid	2.0E-07	99.95	3.6E-07	99.94
	Linear	Backprop	2.9E-04	97.47	2.9E-04	97.46
Bellmf	***Constant***	***Hybrid***	***2.3E-07***	***99.96***	***2.9E-07***	***99.94***
	Constant	Backprop	7.3E-05	98.76	7.3E-05	98.78
	Linear	***Hybrid***	***2.0E-07***	***99.96***	***2.6E-07***	***99.94***
	Linear	Backprop	2.3E-04	97.75	2.3E-04	97.75
Gaussmf	Constant	Hybrid	2.1E-07	99.95	2.9E-07	99.94
	Constant	Backprop	6.8E-05	98.81	6.5E-05	98.8
	Linear	Hybrid	2.1E-07	99.95	2.9E-07	99.94
	Linear	Backprop	2.1E-04	97.8	2.1E-04	97.5
Dsigm	Constant	Hybrid	2.7E-07	99.95	3.4E-07	99.94
	Constant	Backprop	8.6E-05	98.82	8.4E-07	98.83
	Linear	Hybrid	1.9E-07	99.95	5.8E-07	99.94
	Linear	Backprop	2.6E-04	97.67	2.6E-04	97.66

As can be seen in the table, best results are obtained when 2 Bell MFs represent the 5 inputs, a linear or constant MF is used at the output and a hybrid learning method is employed. Figure 8 plots these ANFIS results with the linear MF at the output.

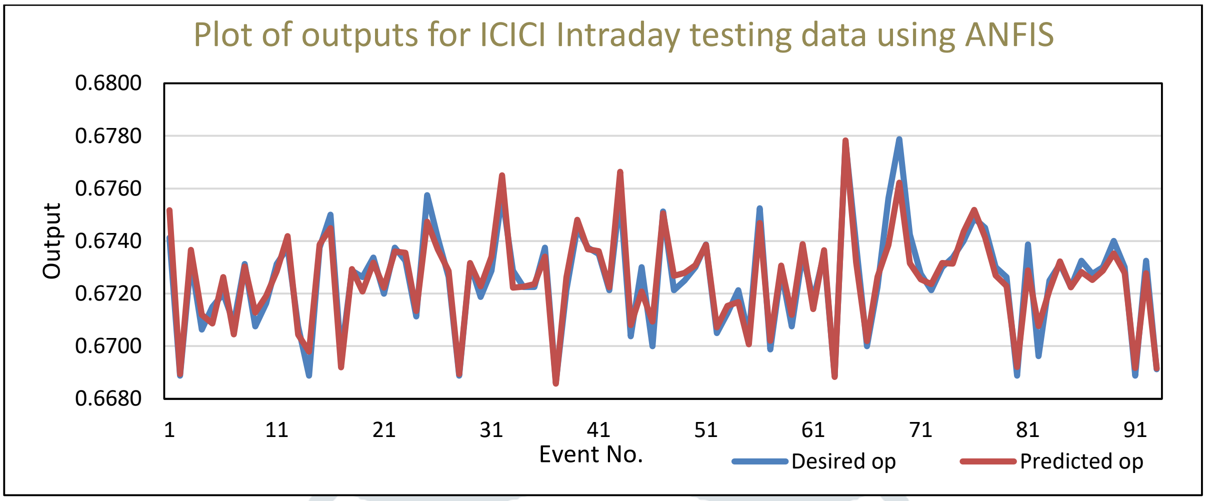

Figure 8. Plot of desired and forecasted outputs for ICICI intraday test data obtained by ANFIS

VIII. EFFICACY PARAMETERS

Various efficacy parameters given in section 2 have been calculated for all the approaches discussed above and summarized in the Table 7 given below.

Table 7: Efficacy parameters used in forecasting.

	ME	MAD	MSE	MPE	MAPE	MPA	R^2
ANN, Daily	1.26E-03	1.28E-02	2.75E-04	1.3E-01	1.73E00	98.27E00	9.56E-01
ANFIS, Daily	-5.98E-03	1.36E-02	2.93E-04	8.4E-01	1.84E00	98.16E00	9.53E-01
ANN, Intraday	-4.30E-05	3.90E-04	2.58E-07	1.0E-02	6.00E-02	99.94E00	9.31E-01
ANFIS, Intraday	1.20E-05	3.93E-04	2.64E-07	1.64E-03	5.84E-02	99.94E00	9.29E-01

It can be observed from the above table of efficacy parameters that both for daily and intraday forecasts ANN and ANFIS are about equally efficient though ANN has an edge over ANFIS in daily forecasts and ANFIS is slightly better than ANN in intraday forecasts.

IX. CONCLUSION

In the present work, application of ANN and ANFIS techniques, has been presented in order to make highly accurate forecast of a typical NSE stock viz. ICICI BANK. Forecasting has been done from stock price time series in two different time domains: day-to-day (daily) and minute-to-minute (intraday). Time series of daily prices over one trading year was used for the daily forecast while prices at each minute were used for intraday forecast. Both the AI techniques gave highly accurate forecasts; mean percentage accuracy (MPA) often exceeding 98% for daily forecasts and 99% for intraday forecasts. ANN showed an edge over ANFIS in daily forecasts while for intraday forecasts it was the other way round. Apart from MPA other efficacy parameters like mean absolute deviation (MAD), mean square error (MSE) and coefficient of determination (R^2) etc. also testify very high degree of efficacy of the AI methods presented here. The results obtained underline two very significant conclusions: 1) time lagged variables of stock price time series are capable of making highly accurate forecasts, 2) there is considerable variation in forecasting efficacy of the AI techniques considered here. One has to make a judicious choice and considerable testing of the techniques before attempting forecasting of stocks rather than picking them of the shelf. It is planned in future to employ AI techniques to forecast other stocks, commodities and currency markets in general.

REFERENCES

[1] E. A. Mlybari, M. S. Elbisy, A. H. Alshahri, O. M. Albarakati, "The Use Support Vector Machine and Back Propagation Neural Network for Prediction of Daily Tidal Levels along the Jeddah Coast, Saudi Arabia," *International Journal of Mathematical, Computational, Physical and Quantum Engineering*, vol/issue: 8 (1), pp: 13-18, 2014.
[2] Hossein Mombeini and Abdolreza Yazdani-Chamzini, "Modeling gold price via artificial Neural Network," *Journal of Economics, Business and Management*, vol/issue 3 (7), pp. 699-703, 2015.
[3] V.K. Dhar, A.K.Tickoo, R.Koul, B.P.Dubey, "Comparative performance of some popular artificial neural network algorithms on benchmark and function approximation problems," *Volume 74 Issue 2 February pp 307-324, 2010*
[4] Haviluddin and Imam Tahyudin, "Time Series Prediction Using Radial Basis Function Neural Network," *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue 5 (4), pp. 765~771, 2015.
[5] Hyunjung Shin, Tianya Hou, Kanghee Park, Chan-Kyoon Park, Sunghee Choi, " Prediction of movement direction in crude oil prices based on semi-supervised learning, " *Decision Support Systems*, vol. 55, pp. 348-358, 2013.
[6] Deepak Singhal, K.S. Swarup, "Electricity Price forecasting using Artificial Neural Networks," *Electrical Power and Energy Systems*, vol. 33, pp. 550-555, 2011.
[7] Kyoung-jae Kim, Ingoo Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Systems with Applications*, vol. 19, pp. 125-132, 2000.
[8] Kyoung-jae Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, pp. 307–319, 2003.
[9] Kyoung-jae Kim, "Artificial neural networks with evolutionary instance selection for financial forecasting," *Expert Systems with Applications*, vol. 30, pp. 519–526, 2006.
[10] Ebrahim Abbasi, Amir Abouec, "Sock Price forecast by using Neuro-Fuzzy Inference System," *World Academy of Science, Engineering and Technology*, vol. 46, pp. 320-323, 2008.
[11] George S. Atsalakis, Kimon P. Valavanis, "Forecasting stock market short-term trends using a neuro-fuzzy based methodology," *Expert Systems with Applications*, vol. 36, pp. 10696–10707, 2009.
[12] Melek Acar Boyacioglu, Derya 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, pp. 7908–7912, 2010.
[13] Esmaeil Hadavandi, Hassan Shavandi, Arash Ghanbari, "Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting," *Knowledge-Based Systems*, vol. 23, pp. 800–808, 2010.
[14] Yakup Kara, Melek Acar Boyacioglu, Ömer Kaan Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange," *Expert Systems with Applications*, vol. 38, pp. 5311–5319, 2011.
[15]Jonathan L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, pp. 5501–5506, 2013.
[16] Amin Hedayati Moghaddam, Moein Hedayati Moghaddam, Morteza Esfandyari, "Stock market index prediction using artificial neural network," *Journal of Economics, Finance and Administrative Science*, vol. 21, pp. 89–93, 2016.
[17] Aparna Nayak, M. M. Manohara Pai, Radhika M. Pai, "Prediction Models for Indian Stock Market," *Procedia Computer Science*, vol. 89, pp. 441 – 449, 2016.
[18] Amin Hedayati Moghaddam, Moein Hedayati Moghaddam, Morteza Esfandyari, "Stock market index prediction using artificial neural network," *Journal of Economics, Finance and Administrative Science*, vol. 21, pp. 89–93, 2016.
[19] Siham Abdulmalik, Mohammed Almasani, Valery Ivanovich Finaev, Wadeea Ahmed Abdo Qaid, Alexander Vladimirovich Tychinsky, "The Decision-making Model for the Stock Market under Uncertainty", *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue 7 (5), pp. 2782-2790, 2017.
[20] A. Victor Devadoss, T. Antony Alphonnse Ligori, "Stock Prediction Using Artificial Neural Networks," *International Journal of Data Mining Techniques and Applications*, vol. 2, pp. 283-291, 2013.
[21] Yunus Yetis, Halid Kaplan, and Mo Jamshidi, "*Stock Market Prediction by Using Artificial Neural Network*," In Proc. World Automation Congress, IEEE Computer Society, U.S., 2014, pp. 718-722.
[22] Mingyue Qiu, Yusong et al., "Application of ANN for prediction of stock market returns: Case of Japanese stock market," *Chaos Solitions & Fractals*, vol. 85, pp. 1-7, 2016.
[23] Sachin Kamley, Shailesh Jaloree, R. S. Thakur, "Performance Forecasting of Share Market using Machine Learning Techniques: A Review", *International Journal of Electrical and Computer Engineering (IJECE)*, vol/issue 6(6), pp. 3196-3204, 2016.
[24] www.nseindia.com
[25] Rumelhart D.E., Hinton G.E., Williams R.J., "Learning Representations by Back–Propagating Errors," *Nature*, vol. 323, pp. 533-536, 1986.
[26] Atsalakis G.S., Kimon P., Valavanis K.P., "Surveying Stock Market Forecasting Techniques –Part II: Soft Computing Methods,'' *Expert Systems with Applications*, vol. 36, pp. 5932-5941, 2009.
[27] M. Reidmiller, "Advanced supervised learning in multilayer perceptrons from backpropagation to adaptive learning algorithms," *Computer Standards Interfaces*, vol 16, pp. 265-278, 1994.
[28] M.F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol 6, pp. 525-533, 1993.
[29] D.W. Marquardt, "An algorithm for least square estimation of non-linear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol/issue 11(2), pp. 431-441, 1963.
[30] V. K. Dhar, A. K. Tickoo, R. Koul and B.P. Dubey, "Comparative performance of some popular artificial neural network algorithms on benchmark and function approximation problems," *Pramana*, vol/issue 74 (2), pp. 307-324, 2010.
[31] Salim Lahmiri, "A Comparative Study of Backpropagation Algorithms in Financial Prediction", *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, vol/issue 1(4), 2011.
[32] Jyh-Shing Roger Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on Systems, Man and Cybernetics*, vol/issue 23(3), pp. 665-685, 1993.
[33] www.keralabanking.com
[34] https://in.mathworks.com/