Stock price prediction using Artificial Neural Network for stocks listed in NSE of India

G. G. Rajput[1] and Bhagwat H. Kaulwar[2]*
[1]Department of Computer Science, Akkamahadevi Women's University, Vijayapura, India
[2]Deparment of Computer Science, Solapur University, Solapur, India
ggrajput@yahoo.co.in
bhagwat_kaulwar@yahoo.com

Abstract—The study in this paper focuses on designing a model for predicting the closing price of a particular stock using Artificial Neural Networks (ANN). ANN is suitable for classification or prediction as it is popular way to identify unknown and hidden patterns in data. A Multilayer Feed-Forward NARX network is designed to predict closing price of a share listed in NSE, in particular for National Thermal Power Corporation Ltd. (NTPC). The daily trading data of NTPC was used along with moving averages as technical parameters. Experiments are carried out for the NTPC data between 2017 and 2018 and the performance of the NN in predicting stock price has been presented.

Keywords—*Artificial Neural Network (ANN), Nonlinear Autoregressive with External Input (NARX), stock price prediction, Data Mining, Normalized Mean Square Error (NMSE)*

I. INTRODUCTION

A stock market is a public market for the trading of a company stock and derivatives at an agreed price. Stock market brings investors together to buy and sell the shares. Share market prices sets prices according to supply and demand. Stocks that are in demand will increase their price, whereas stocks that are being heavily sold will decrease their price. Stock market data are highly time-variant and are normally in a nonlinear pattern, which makes predicting the future price of a stock highly challenging for investors and traders [1]. In their quest to forecast the markets, they assume that future occurrences are based at least in part on present and past events and data. However, financial time series are among the 'noisiest' and most difficult signals to forecast [2].

Predicting stock or index prices in capital market, by using traditional forecasting methods like least square regression, are difficult. This is because capital markets are of chaotic nature. On the other hand, several AI based methodologies have been proposed for better market predictions [3]. Many researchers have proposed models using various fundamental, technical and analytical techniques to give a more or less exact prediction. They have used different prediction models like ANNs, Support Vector Machine (SVM), Autoregressive integrated moving average (ARIMA), etc. The use of Artificial Neural Network (ANN) is more accurate technique as compared to other nonlinear techniques as stock market returns are noisy, uncertain, chaotic and nonlinear in nature. The novelty of the ANN lies in their ability to discover nonlinear relationship in the input data set without a priori assumption of the knowledge of relation between the input and the output. The study in the area of Neural Network (NN) has proved that NN can be used to predict INDEX or stock price. The daily transaction data can be used to train the NN. The trained NN then can be used to predict future value of share or Index. A neural network trained to a particular input data set corresponding to a particular environment; can be easily retrained to a new environment to predict at the same level of environment.

In this paper we propose a prediction model based on Artificial Neural Network (ANN). After studying the various features of a Multilayer Feed-Forward NARX network, an optimal model is proposed for the purpose of forecasting.

II. LITERATURE SURVEY

Because irregular fluctuations occur in stock market, it is very difficult to model its behaviour. Stock market can be considered as non-linear deterministic system [4]. But ANN has ability to discover nonlinear relationship in input data set without a priori assumption of the knowledge of relation between input and output (Hagen et. al. 1996). Many researchers have proposed different types of models for prediction of an Index or stock. The neural network model is one of the popular models for classification and prediction. Some researchers have worked on combined prediction model based on PCA along with NN or Support Vector Machine.

Manna Majumder et. al. (2007) used different sets of input parameters to train Multi Layer Neural Network. They used past historical data of S&P CNX Nifty 50 – a index of 50 stock companies listed in Indian Stock Market. They used data from 2000 to 2005 to train the NN and predicted future values of S&P CNX Nifty 50 index between 2006 and 2009. They worked on six different structures of model for different inputs like input variables, hidden neurons, transfer function and training algorithm. In total they worked on 24 networks with different set of input parameters. They observed that three layer feed-forward back propagation neural network with 10 input variables, 5 hidden neurons, symmetric sigmoid transfer function in hidden layer, linear transfer function in output layer was optimal network structure [5].

Mayankkumar B Patel et. al. (2014) used Artificial Neural Network techniques to predict the stock price of companies listed under LIX15 Index of National Stock Exchange (NSE). The companies involved were Axis Bank, State Bank of India, Tata Motors, Tata Steel, Hindalco etc. The data of companies listed in LIX15 index of NSE for 36 months, during Jan 2011 to Jan 2014, was used to train Multilayer Perceptron (MLP). The MLP with 6 input neurons and 1 output neuron were designed with multiple hidden layers. The experiment achieved Median Normalized Error of 0.05995, Median Correct Direction % of 51.06 and Median Standard Deviation of 6.39825 [6].

Luis Gonzaga Baca Ruiz et. al. (2016) used the data provided by an energy consumption monitoring system in a compound of faculties and research centre at the University of Granada and proposed a methodology to predict future energy consumption using nonlinear autoregressive (NAR) and the nonlinear autoregressive neural network with exogenous inputs (NARX). The results revealed that both NAR and NARX neural networks are suitable for predicting energy consumption. The experiment observed that exogenous data may help to improve the accuracy of prediction [7].

III. ARTIFICIAL NEURAL NETWORK

From the literature review it is clear that, ANN can be constructed to predict a future value of an Index or particular share. During the study, a special class of ANN called Nonlinear Autoregressive with External Input (NARX) model was considered. NARX networks are recurrent neural networks and are well suited for modelling nonlinear systems and specially time series.

A. Network Architecture

The neural networks field was originally inspired by psychologists and neurobiologists who wanted to develop and test computational equivalence of neurons. A neural network is a set of connected input/output units. This is one of the reasons that neural networks are also called as connectionist learning. These connections have associated weights. Initially these weights are assigned random values. Theses weights then, during learning, are adjusted in such a way that the network predicts accurate class label for new input tuples. Following Fig. 1 shows multilayer feed-forward neural network.

As shown in Fig. 1, a Multilayer Feed-Forward Neural Network has one input layer, one or more hidden layers and one output layer. The inputs are provided to network through neurons in input layer. After passing through input layer, these inputs are weighted and fed to first hidden layer. The weighted outputs of first hidden layer are then given to second hidden layer if any and so on. The weighted outputs of last hidden layer are then given to output layer. The output layer finally gives the network prediction for given input tuples.

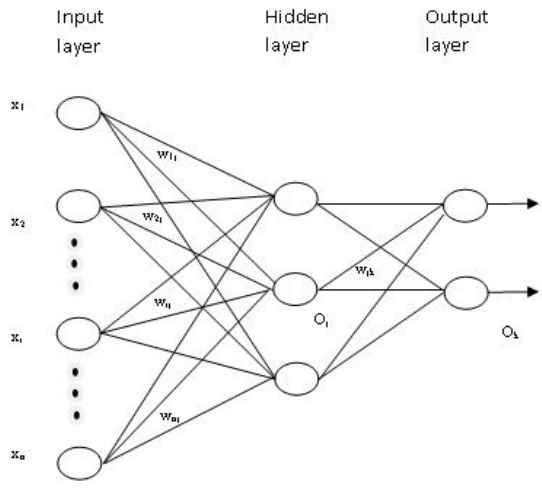

Fig. 1 : Multilayer feed-forward neural network

The back-propagation algorithm is used to train such neural networks. Back-propagation learns by iteratively processing the data set of training tuples. For each training tuples, the weights are modified so that the predicted values by network are more close to actual target values. The algorithm is named as back-propagation because these modifications are made in backward direction from output layer to hidden layer [8]. The Fig. 2 shows steps of back-propagation algorithm.

Step 1. Initialize all weights and biases in *network*.

Step 2. **while** terminating condition is not satisfied, perform steps 3 through 18.

Step 3. **for** each training tuple ***X*** in *D*, perform steps 4 through 18.

Step 4. Propagate the input forward

for each input layer unit *j*, perform steps 5 to 18

Step 5. $O_j = I_j$ // output of an input unit is its actual input value

Step 6. **for** each hidden or output layer unit *j*, perform steps 7 through 8.

Step 7. Compute the net input of unit *j* with respect to the previous layer *i*

$$I_j = \sum_I (w_{ij} O_i + \theta_j)$$

Step 8. Compute the output of each unit *j*

$$O_j = \frac{1}{1 + e^{-I_j}}$$

Step 9. Back-propagate the errors

for each unit *j* in the output layer , perform step 10.

Step 10. Compute the error

$$Err_j = O_j(1 - O_j)(T_j - O_j)$$

Step 11. **for** each unit *j* in the hidden layers, from the last to the first hidden layer, repeat step 12.

Step 12. Compute the error with respect to the next higher layer *k*

$$Err_j = O_j(1 - O_j)\sum_k (Err_k w_{jk})$$

Step 13. **for** each weight w_{ij} in network, perform steps 14 through 15.

Step 14. Increment the weights

$$\nabla w_{ij} = (l) Err_j O_i$$

Step 15. Update weights

$$w_{ij} = w_{ij} + \nabla w_{ij}$$

Step 16. For each bias θ_j in network, perform steps 17 and 18

Step 17. Increment the biases

$$\nabla \theta_j = (l) Err_j$$

Step 18. Update the biases

$$\theta_j = \theta_j + \nabla \theta_j$$

Fig. 2 : Back-propagation Algorithm

The back-propagation algorithm will help to get the neural network with optimised weights and biases. The obtained network then can be used for prediction.

IV. DATA AND METHODOLOGY

Many research studies have observed that technical parameters are more useful to predict share price in short term than macroeconomic indicators. The daily trading data along with moving averages as technical parameters were used to train the neural network. The company under observation was NTPC Ltd., A Maharatna Company, and India's largest power utility.

NTPC is India's largest energy conglomerate with roots planted way back in 1975 to accelerate power development in India. Since then it has established itself as the dominant power major with presence in the entire value chain of the power generation business. From fossil fuels it has forayed into generating electricity via hydro, nuclear and renewable energy sources. This foray will play a major role in lowering its carbon footprint by reducing green house gas emissions. To strengthen its core business, the corporation has diversified into the fields of consultancy, power trading, training of power professionals, rural electrification, ash utilization and coal mining as well.

NTPC became a Maharatna company in May 2010, one of the only four companies to be awarded this status. NTPC was ranked 512th in the '2018, Forbes Global 2000' ranking of the World's biggest companies [9].

In this study, we used daily transaction data like previous day close price of share, open price of share on trading day, high price of share during trading day, low price of share during trading day, turnover of share during trading day etc. The moving averages were used as technical parameters. The five technical parameters like 10 days, 20 days, 50 days, 100 days and 200 days moving averages were used.

A. Data used for Data Mining

The NTPC data of year 2017 was used to train and data of year 2018 was used to test the ANN. The daily trading data of NTPC between 2017 and 2018 is taken from National Stock Exchange website [10].

The daily transaction data of NTPC includes close price on previous day, day open price, day high price, day low price, day average price, total traded quantity, total turnover and closing price on trading day. From the closing prices of the day, various moving averages like 10 days, 20 days, 50 days, 100 days and 200 days were calculated. Thus there are 13 attributes, i.e. 8 attributes and 5 moving averages. The closing price of the share on trading day is used as output of the network. Out of 13 attributes, one attributes is used as networks target and remaining 12 attributes were used as an input to network. As attribute total traded quantity and attribute total turnover were highly correlated to each others, the former attribute was ignored. Finally 11 attributes were considered for network input and attribute closing price was used for network output.

B. Network Training

Initially a study was performed in order to fix the combinations of various parameters for ANN. Through that initial study it was observed that NARX network with 12 inputs variables, 1 hidden layer with 10 neurons, 1 output neuron, feedback delay of 2, Gradient descent adaptive back-propagation (GDA) training algorithm, symmetric sigmoid transfer function in hidden layer and pure linear transfer function in output layer is optimal network. During another study, it was observed that network trained with one year data gave better predictions as compared to network trained with more than one year.

Procedure for training the data and predicting the share value is presented below.

Step 1 : load input series in InputData
Step 2 : Convert data in InputData to standard neural network cell array by using *tonndata* function and store returned data in inputSeries.
Step 3 : load target series in InputTargets
Step 4 : Convert data in inputTargets to standard neural network cell array by using *tonndata* function and store returned data in targetSeries.
Step 5 : Set inputDelay to 1:2
Step 6 : Set feedbackDelays to 1:2;
Step 7 : Set hiddenLayerSize to 10;
Step 8 : Create a Nonlinear Autoregressive Network with External Input by using *narxnet* function with inputDelays, feedbackDelays and hiddenLayerSize and store return network in net.
Step 9 : set transfer function of hidden layer to tansig by using net.layers{1}.transferFcn = 'tansig';
Step 10 : set transfer function in output layer to purelin by using

```
                net.layers{2}.transferFcn = 'purelin';
Step 11 : Prepare the input data for network training and
          simulation by using function PREPARETS function.
                [inputs,inputStates,layerStates,targets] =
                    preparets(net,inputSeries,{},targetSeries);
Step 12 : Partition input data in three groups 70 % for training,
          15 % for validation, and remaining 15 % for testing.
                net.divideFcn = 'dividerand';
                net.divideMode = 'value';
                net.divideParam.trainRatio = 70/100;
                net.divideParam.valRatio = 15/100;
                net.divideParam.testRatio = 15/100;
Step 13 : Set a training function to traingda.
                net.trainFcn = 'traingda';
Step 14 : Train the network by using train function.
                [net,tr] = train(net, inputs, targets, inputStates,
                    layerStates);
Step 15 : Test the performance of network
                outputs = net(inputs,inputStates,layerStates);
                errors = gsubtract(targets,outputs);
                performance =
                perform(net,targets,outputs);
Step 16 : Recalculate Training, Validation and Test
          Performance.
                trainTargets = gmultiply(targets,tr.trainMask);
                valTargets = gmultiply(targets,tr.valMask);
                testTargets = gmultiply(targets,tr.testMask);
                trainPerformance = perform(net, trainTargets,
                    outputs);
                valPerformance = perform(net, valTargets,
                    outputs);
                testPerformance = perform(net, testTargets,
                    outputs);
Step 17 : If training, testing and validation performance are
          not acceptable then restart the process otherwise
          continue to step 18.
Step 18 : Convert the trained open loop network to close loop
          network
                netc = closeloop(net);
Step 19 : prepare data for use by close loop network and test
          the network performance.
                [xc,xic,aic,tc] = preparets(netc, inputSeries,
                    {}, targetSeries);
                yc = netc(xc,xic,aic);
                closedLoopPerformance = perform(netc,tc,yc);
Step 20 : Remove the delay in network for early prediction i.e.
          one step ahead prediction.
                    nets = removedelay(net);
Step 21 : Prepare the data for use by one step ahead network
          and test the performance.
                [xs,xis,ais,ts] = preparets(nets, inputSeries, {},
                    targetSeries);
                ys = nets(xs,xis,ais);
                earlyPredictPerformance = perform(nets,ts,ys);
Step 22 : If performance is proper, return trained network
          otherwise restart the process.
```

Fig. 3 : pseudo-code for predicting one step ahead value

In the algorithm, one step ahead means tomorrow's value based on today's data. Thus we predict the share price well before the event happens. This helps the investors to take their call, one day before.

C. Performance Measurement

To measure the prediction accuracy of the model, the predicted values were compared with actual outputs of sample data. Normalized Mean Square Error (NMSE) is used to evaluate prediction of accuracy of the model. Following formula in Eq. 1 is used to calculate NMSE.

$$NMSE = \frac{\sum_{t=1}^{N_1}(P_t - O_t)^2}{\sum_{t=1}^{N_1}(P_t - \bar{P}_t)^2} \dots\dots\dots\dots\dots\dots\dots\dots (1)$$

where P_t represents actual value of the pre-processed data series i.e. closing price of share, O_t represents observed value or the predicted value i.e. predicted closing price of share for the same day and $\overline{P}_t$ is the mean of the actual value.

In order to calculate the error percentage, actual closing price and predicted closing price were compared. The formula to calculate error percentage is as shown in Eq. 2.

$$Error\ \% = \frac{|P_t - O_t|}{P_t} * 100 \quad \dots\dots\dots\dots\dots (2)$$

Where P_t represents actual closing price and O_t represents observed or predicted closing price of share. From formula it is clear that, a network with less error % should be considered as best network. The value of 0 for error % indicates that there are no errors in actual and predicted values which indicate perfect prediction.

V. EXPERIMENTAL RESULTS

After predicting share prices of NTPC for year 2018, performance of model was calculated by using **Eq. (1) and (2)**. Table 1 gives performance details of model.

Table 1 : Performance measure details of ANN structures used in study

Sr. No.	Parameter	Performance measures of ANN
1	Input variables & neurons	11
2	Neurons in hidden layer	10
3	Epochs	211
4	NMSE	0.04
5	Average Error %	0.90 %

From **Table 1**, it is observed that model gave NMSE of 0.04 which is in the acceptable range. The Average Error % was observed at 0.90% which is also a quite below the accepted range.

The Fig. 4 shows plot of actual closing price and network output during training phase i.e. close values and predicted values of NTPC in year 2017. From Plot in Fig. 4, it is clear that the actual close value and predicted close values by network are in line with each others.

The Fig. 5 shows the plot between actual close price of NTPC and close price predicted by network for entire year 2018. Plot in Fig. 5 is providing clear idea about prediction accuracy of ANN model. Plot concludes that the predicted values by ANN model are in line with that of actual closing values of NTPC.

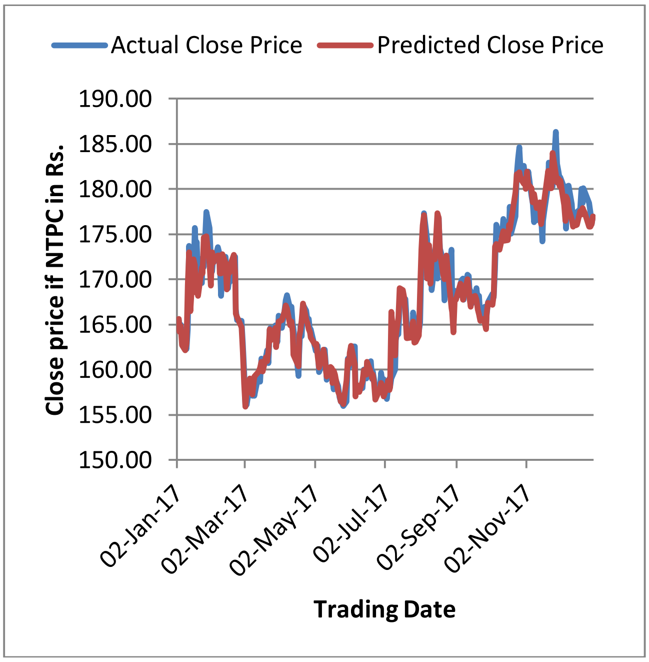

Fig. 4 : Plot of closing price and network output during training phase.

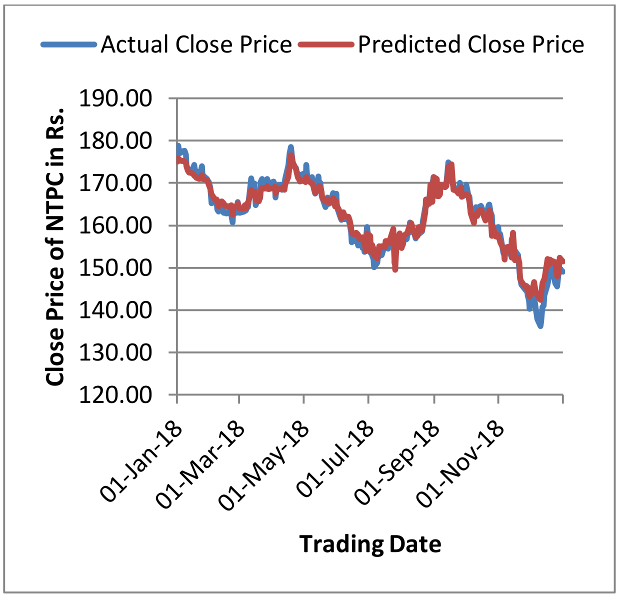

Fig. 5 : Plot of actual and predicted closing prices of NTPC by ANN model for year 2018 data.

The Fig. 6 shows the plot of actual closing price of NTPC and network predicted prices for year 2018 after ignoring last month data i.e. Dec 2018 data.

From plot in Fig. 6, it is clear that the network performance is improved if few last records are ignored. This is due to the need of retraining the network after particular time span. This will allow network to learn new trends in data.

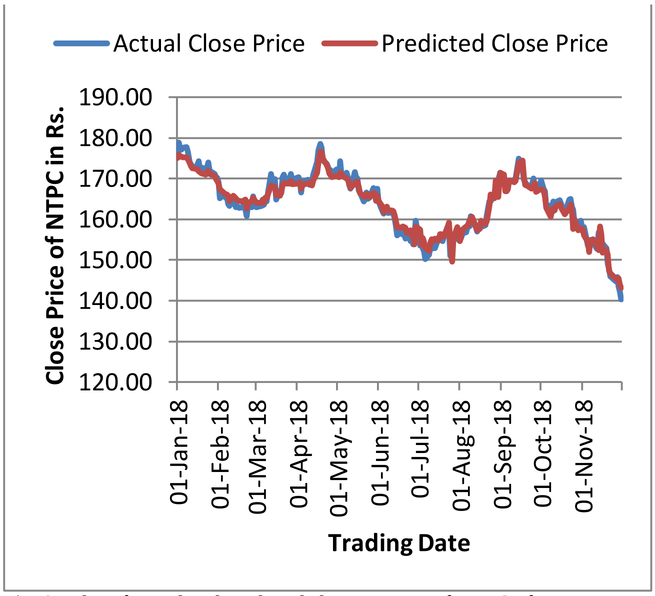

Fig. 6 : Plot of actual and predicted closing prices of NTPC after ignoring Dec 2018 data.

VI. CONCLUSION

In this paper ANN based NARX model has been proposed for prediction of closing price of NTPC. During this study, prediction model gave NMSE of 0.04 and average error % of 0.90%. Thus the performance of ANN based model is encouraging. Unlike other AI applications where model accuracy goes on increasing with data size, the prediction model for particular share gave better results for one year data. This is because in technical analysis, trends are given more importance. Due to current events and trends, the share price is affected in short term trends and eventually share price will continue with its long term trend. But for predicting next day price of a share short term trend play important role. Also in a short span, share prices are affected by seasonal factors like monsoon, RBI policies etc. So instead of using more data, the study used only one year data to train the network.

The study also observed that the network performance was improved if the last few records are ignored. The network observed NMSE of 0.04 and average error % of 0.90 for entire year 2018. But if Dec 2018 is ignored, NMSE was same at 0.04 but average error % was improved to 0.76. This indicates that the ANN based networks need to retrained after specific time so that they can learn new trends in data. The retraining process will improve prediction accuracy.

REFERENCES

[1] Chi-Jie Lu, “Integrating independent component analysis-based denoising scheme with neural network for stock price prediction”, *Expert Systems with Applications* 37(10), 7056-7064

[2] Yaser S. Abu-Mostafa, Amir F. Atiya, “Introduction to Financial Forecasting”, *Applied Intelligence* 6, 205-213, 1996

[3] Chen, W-H., Shih, J-Y. and Wu, S. (2006), “Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets”, *Int. J. Electronic Finance*, Vol. 1, No. 1, pp.49–67

[4] Abhyankar, A. Copeland, L.S., & Wong, W., “Uncovering nonlinear structure in real-time stock-market indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100”, *Journal of Business & Economic Statistics*, 15, 1–14., 1997

[5] Manna Majumder and MD Anwar Hussian, “Forecasting of Indian Stock Market Index using Artificial Neural Network”, https://nseindia.com/content/research/FinalPaper206.pdf, 2007

[6] Mayankkumar B Patel, Sunil R Yalamalle, “Stock Price Prediction Using Artificial Neural Network”, *International Journal of Innovative Research in Science, Engineering and Technology*, June 2014, Vol. 3, Issue 6, pp. 13755-13762, ISSN: 2319-8753

[7] Ruiz LGB, Cuellar MP, Calvo-Flores MD, Jimenez MDCP. “An Application of Non-Linear Autoregressive Neural Networks to Predict Energy Consumption in Public Buildings”, *Energies*. 2016; 9(9):684., doi:10.3390/en9090684

[8] Jiawei Han, Micheline Kamber, & Jian Pei, “Data Mining Concepts and Techniques”, 3rd edition, Morgan Kaufmann Publishers, imprint of Elsevier, pp. 398-406, ISSN : 978-93-80931-91-3

[9] https://www.ntpc.co.in/en/about-us/ntpc-overview

[10] https://www.nseindia.com/products/content/equities/eq_security.htm