STOCK PRICE PREDICTION USING LSTM,RNN AND CNN-SLIDING WINDOW MODEL

Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, Soman K.P
Centre for Computational Engineering and Networking (CEN),
Amrita School of Engineering,Coimbatore
Amrita Vishwa Vidyapeetham, Amrita University,India
Email:sreelekshmyselvin@gmail.com

Abstract—**Stock market or equity market have a profound impact in today's economy. A rise or fall in the share price has an important role in determining the investor's gain. The existing forecasting methods make use of both linear (AR,MA,ARIMA) and non-linear algorithms (ARCH,GARCH,Neural Networks),but they focus on predicting the stock index movement or price forecasting for a single company using the daily closing price. The proposed method is a model independent approach. Here we are not fitting the data to a specific model, rather we are identifying the latent dynamics existing in the data using deep learning architectures. In this work we use three different deep learning architectures for the price prediction of NSE listed companies and compares their performance. We are applying a sliding window approach for predicting future values on a short term basis.The performance of the models were quantified using percentage error.**
Index Terms—**Time series, Stock market, RNN,LSTM,CNN**

I. INTRODUCTION

Forecasting can be defined as the prediction of some future event or events by analyzing the historical data. It spans many areas including business and industry, economics, environmental science and finance. Forecasting problems can be classified as

- Short term forecasting (prediction for few seconds,minutes,days, weeks or months)
- Medium term forecasting (prediction for 1 to 2 years)
- Long term forecasting (prediction beyond 2years)

Many of the forecasting problems involve the analysis of time . A time series data can be defined as a chronological sequence of observations for a selected variable. In our case the variable is stock price.It can either be univariate or multivariate. Univariate data includes information about only one particular stock whereas multivariate data includes stock prices of more than one company for various instances of time. Analysis of time series data helps in identifying patterns, trends and periods or cycles existing in the data. In the case of stock market, an early knowledge of the bullish or bearish mode helps in investing money wisely.Also the analysis of patterns helps in identifying the best performing companies for a specified period. This makes time series analysis and forecasting an important area of research.

The existing methods for stock price forecasting can be classified as follows[1]

- Fundamental Analysis
- Technical Analysis
- Time Series Forecasting

Fundamental analysis is a type of investment analysis where the share value of a company is estimated by analyzing its sales,earnings,profits and other economic factors. This method is most suited for long term forecasting. Technical analysis uses the historical price of stocks for identifying the future price. Moving average is a commonly used algorithm for technical analysis. It can be considered as the unweighted mean of past n data points. This method is suitable for short term predictions. The third method is the analysis of time series data. It involves basically two classes of algorithms, they are

- Linear Models
- Non Linear Models

The different linear models are AR,ARMA,ARIMA and its variations [2] [3] [4]. These models uses some predefined equations to fit a mathematical model to a univariate time series. The main disadvantage of these models is that, they do not account for the latent dynamics existing in the data. Since they consider only univariate time series, the inter dependencies among the various stocks are not identified by these models.Also the model identified for one series will not fit for the other. Due to these reasons, it is not possible to identify the patterns or dynamics present in the data as a whole.

Non-linear models involve methods like ARCH,GARCH,[3] TAR,Deep learning algorithms [5].In [6] an analysis on the inter dependency between stock price and stock volume for 29 selected companies listed in NIFTY 50 has been done.
The poposed work focuses on the application of deep learning algorithms for stock price prediction [7] [8]. Deep neural networks can be considered as non-linear function approximators which are capable of mapping non-linear functions . Based on the type of application, various types of deep neural network architectures are used. These include multi layer perceptrons (MLP),Recursive Neural Networks (RNN),Long Short Term Memory(LSTM), CNN(Convolutional Neural Network) etc [9].They have been applied in various areas like image processing, natural language processing, time series analysis etc.

1643

Deep learning algorithms are capable of identifying hidden patterns and underlying dynamics in the data through a self learning process.

In the case of stock market, the data generated is enormous and is highly non-linear. To model such kind of dynamical data we need models that can analyze the hidden patterns and underlying dynamics. Deep learning algorithms are capable of identifying and exploiting the interactions and patterns existing in a data through a self learning process. Unlike other algorithms, deep learning models can effectively model these type of data and can give a good prediction by analyzing the interactions and hidden patterns within the data. In [5], we can see the application of various deep learning models for multivariate time series analysis. The first attempt to model a financial time series using a neural network model was introduced in [10]. This work made an attempt to model a neural network model for decoding the nonlinear regularities in asset price movements for IBM. However the scope of the work was limited, but it helped in establishing evidences against EMH [11].

Researches in the area of financial time series analysis using NN models used different input variables for predicting the stock return. In some works, data from a single time series were used as input [10], [8]. Certain works considered the inclusion of heterogeneous market information and macroeconomic variables. In [12], a combination of financial time series analysis and NLP have been introduced. In [13] and [7], deep learning architectures have been used for the modeling of multivariate financial time series. In [14], a NN model using technical analysis variables have been implemented for the prediction of Shanghai stock market. The work compared the performance of two learning algorithms and two weight initialization methods. The results sh6wn that efficiency of back propagation can be increased by conjugate gradient learning with multiple linear regression weight initialization.

In 1996, [15] used back propagation and RNN models for the prediction of stock index for five different stock markets. In [16], application of time delay, recurrent and probabilistic neural network models were introduced for daily stock prediction. In [17], application of machine learning algorithms like PSO and LS-SVM have been used for the prediction of S&P 500 stock market. Implementation of genetic algorithms along with neural network models were introduced in [18]. The work combined the application of both genetic algorithm and artificial neural network for forecasting. In this work weights for NN was obtained from the genetic algorithm.However, the prediction accuracy for this model was low. Application of wavelet transform for prediction was introduced in [19]. The work used wavelet transform to describe short term features in stock trends. With the introduction of LSTM [20], the analysis of time dependent data become more efficient. These type of networks have the capability of holding past information. They have been used in stock price prediction by [8], [7].

The proposed method focuses on predicting stock price for NSE (National Stock Exchange) listed companies.The approach we have adopted is a sliding window approach with data overlap. The data set contains minute wise data of NSE listed companies. Here we are trying to obtain a generalized model for the purpose of prediction which can use minute wise data as input. This kind of modeling have applications in algorithmic trading where high frequency trading occurs.

The paper is structured as follows section[II] explains the proposed methodology.Results and discussions can be found in Section[III] and Section[IV] includes the conclusion.

II. METHODOLOGY

The data set consists of minute wise stock price for 1721 NSE listed companies for the period of July 2014 to June 2015. It includes informations like day stamp, time stamp, transaction id, stock price and volume of stock sold in each minute. For this work we have selected two different sectors, IT sector and Pharma sector. Two companies from IT sector and one company from Pharma sector were taken for the study. These companies were identified by the help of NIFTY-IT index and NIFTY-Pharma index. The data for these three companies were extracted from the available data and was subjected to preprocessing to obtain the stock price.

The work is based on a sliding window approach for a short term future prediction. The window size was fixed to be 100 minutes with an overlap of 90 minute's information and prediction was made for 10 minutes in future. The best window length was identified by calculating the error for various window sizes. The train data consists of stock price of Infosys for the period July-01-2014 to October-14-2014 and test data consists of stock price for Infosys, TCS and CIPLA for the period of October-16-2014 to November-28-2014.
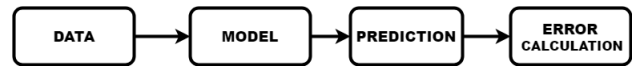

Fig. 1: Proposed Model Block Diagram

The data varies with in a range of 2000 to 4000 for Infosys and TCS and for Cipla it is 400 to 700. To unify the data range, it was subjected to normalization and was mapped to a range of 0 to 1. This normalized data was given to the network for training. All the models were trained for 1000 epochs by varying the layer size for fine tuning. If the loss (mean squared error) for the current epoch is less than the value obtained in previous epoch, the weight matrices for that epoch is stored. After the training process each of these models were tested and the model with least RMSE (Root Mean Squared Error) is taken as the final model for prediction.

We have used three different deep learning architectures, RNN, LSTM and CNN for this work. RNN is a class of neural network where connections between the computational units form a directed circle. Unlike feed forward networks, RNN can use their internal memory to process arbitrary sequence of inputs. Each of the computing unit in an RNN has a time varying real valued activation and modifiable weight. RNNs are created by applying the same set of weights recursively over a graph-like structure. Many of the RNNs use (1) to define the values of their hidden units.

$$h^t = f(h^{t-1}, x^t; \theta) \tag{1}$$

In the case of RNN, the learned model always has the same input size, because it is specified in terms of transition from one state to another. Also the architecture uses the same transition function with the same parameters at every time step. LSTM is a special kind of RNN, introduced in 1997 by Hochreiter and Schmidhuber [20] . In the case of LSTM architecture, the usual hidden layers are replaced with LSTM cells. The cells are composed of various gates that can control the input flow. An LSTM cell consists of input gate, cell state, forget gate, and output gate. It also consists of sigmoid layer, tanh layer and point wise multiplication operation.The various gates and their functions are as follows

- Input gate : Input gate consists of the input.
- Cell State : Runs through the entire network and has the ability to add or remove information with the help of gates.
- Forget gate layer: Decides the fraction of the information to be allowed.
- Output gate : It consists of the output generated by the LSTM.
- Sigmoid layer generates numbers between zero and one, describing how much of each component should be let through.
- Tanh layer generates a new vector, which will be added to the state.

The cell state is updated based on the outputs form the gates. Mathematically we can represent it using the following equations.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{3}$$

$$c_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(c_t) \tag{6}$$

where x_t: input vector, h_t: output vector, c_t: cell state vector, f_t: forget gate vector, i_t: input gate vector, o_t: output gate vector and W,b are the parameter matrix and vector. Convolutional neural networks or CNNs, are a specialized kind of neural network for processing data that has a known, grid-like topology. This include time-series data, which can be thought of as a 1D and image data, which can be thought of as a 2D grid of pixels.The network employs a mathematical operation called convolution and hence known as convolutional neural network. It is a specialized kind of linear operation. Convolutional networks use convolution instead of general matrix multiplication in at least one of their layers. The motivation behind using these three models is to identify whether there is any long term dependency existing in the given data. This can be identified from the performance of the models. RNN and LSTM architectures are capable of identifying long term dependencies and uses them for future prediction. However CNN architectures mainly focuses on the given input sequence and does not use any previous history or information during the learning process.The motivation behind testing the models with data from other companies is to check for interdependencies among the companies and to understand the market dynamics.

The train data was normalized. Test data was also subjected to the same normalization. After obtaining the predicted output, denormalization was applied and percentage error was calculated using the available true labels.The error percentage was calculated using (7)

$$ep = \frac{abs\left[X_{real}^i - X_{predicted}^i\right]}{X_{real}^i} \times 100 \tag{7}$$

where ep is the error percentage,X_{real}^i is the ith real value and $X_{predicted}^i$ is the ith predicted value. Error percentage gives the magnitude of error present in the output.

III. RESULTS AND DISCUSSION

The experiment was done for three different deep learning models. The maximum value of error percentage obtained for each model is given in Table[I]. From the table it is clear that CNN is giving more accurate results than the other two models. This is due to the reason that CNN does not depend on any previous information for prediction. It uses only the current window for prediction. This enables the model to understand the dynamical changes and patterns occurring in the current window. However in the case of RNN and LSTM, it uses information from previous lags to predict the future instances. Since stock market is a highly dynamical system, the patterns and dynamics existing with in the system will not always be the same. This cause learning problems to LSTM and RNN architecture and hence the models fails to capture the dynamical changes accurately.

TABLE I: ERROR PERCENTAGE

COMPANY	RNN	LSTM	CNN
Infosys	3.90	4.18	2.36
TCS	7.65	7.82	8.96
Cipla	3.83	3.94	3.63

For comparison we have used ARIMA, which is a linear model used for forecasting.The error percentage obtained for the three companies are as follows

TABLE II: ERROR PERCENATGE - ARIMA

COMPANY	Error Percentage
Infosys	31.91
TCS	21.16
Cipla	36.53

From Table[I] and Table[II] it is clear deep learning models are outperforming ARIMA.
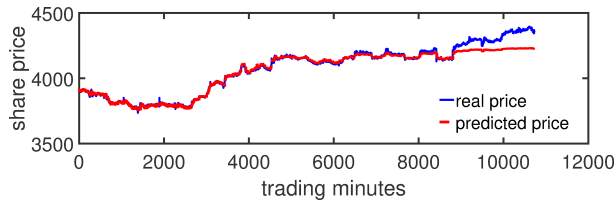
Fig. 2: Plot for Real value vs Predicted value for INFOSYS using RNN
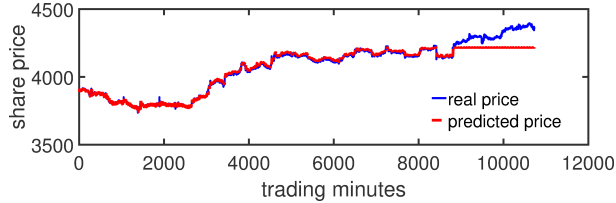

Fig. 3: Plot for Real value vs Predicted value for INFOSYS using LSTM
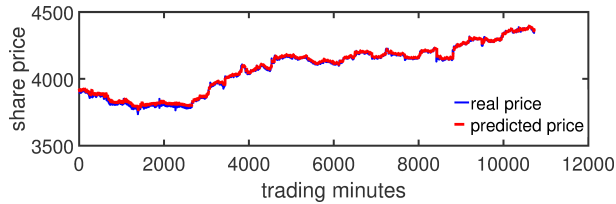

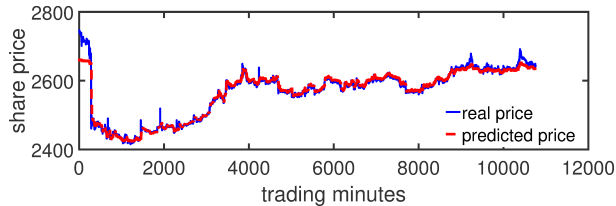Fig. 4: Plot for Real value vs Predicted value for INFOSYS using CNN

Fig. 5: Plot for Real value vs Predicted value for TCS using RNN

From Fig(2) and Fig(3) it can be observed that both RNN and LSTM fails to capture the trends and dynamics towards the end (between the time period 9000 to 11000) ie; there is a change in the behavior of the stock pattern for that time window when compared to the previous windows. In case of CNN it is evident from the Fig(4), that the network is capable of capturing the changes in the trend for the time period 9000 to 11000

In the case of TCS, Fig(5) and Fig(6), RNN and LSTM networks are not identifying the pattern in the beginning of the window (during the first 1000 minutes). There is a change in the trend followed by TCS during that period. This makes the predictions less accurate, whereas in Fig(7), we can see that CNN captures these changes more accurately
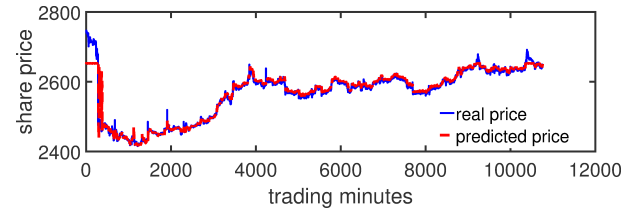

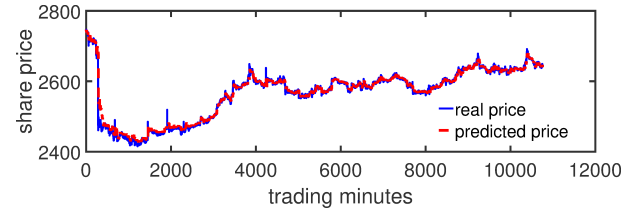Fig. 6: Plot for Real value vs Predicted value for TCS using LSTM

Fig. 7: Plot for Real value vs Predicted value for TCS using CNN

when compared to the other two networks.

In case of Cipla, Fig(8) and Fig (9) ,between the period of 2000 and 6000, it is clear that the predicted values of RNN and LSTM does not matches with the pattern of original data. This can be considered as a change in the behavior of the system. In case of Cipla, Fig(10), we can observe that CNN is capable of capturing the changes in the behavior of the stock price for the specified period.

It can be seen that CNN network is almost able to capture the trends and is giving accurate predictions compared to the other two models. CNN was able to analyze the change in trend for Infosys, TCS and Cipla. It should also be noticed that we trained the network with Infosys data for the period of July-01-
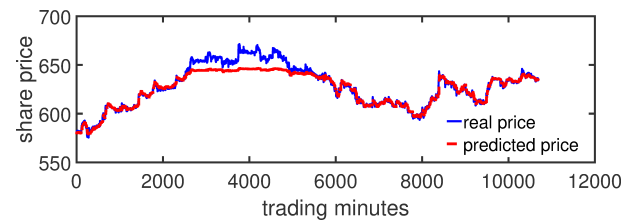

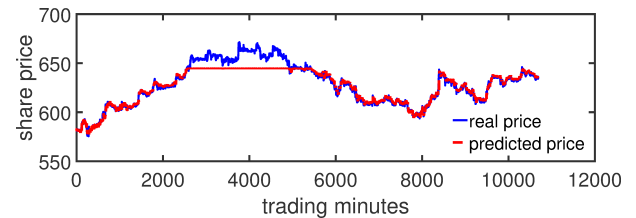Fig. 8: Plot for Real value vs Predicted value for CIPLA using RNN

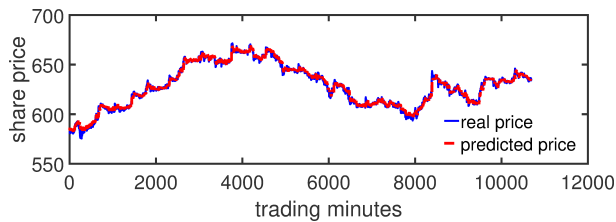Fig. 9: Plot for Real value vs Predicted value for CIPLA using LSTM

Fig. 10: Plot for Real value vs Predicted value for CIPLA using CNN

2014 to October-14-2014, even then the testing accuracy for Infosys is lower when compared to the other companies.This shows that whatever trend Infosys exhibits during the period of July to October 14 is not present in the test data as such(from October-16-2014 to November-28-2014) ie; there is a change in the dynamics.This accounts for the difference in the error percentage for Infosys when compared to other models. Also the model is capable of predicting stock price for companies other than Infosys. This shows that, the pattern or dynamics identified by the model is common to other companies also.

IV. CONCLUSION

We propose a deep learning based formalization for stock price prediction. It is seen that, deep neural network architectures are capable of capturing hidden dynamics and are able to make predictions. We trained the model using the data of Infosys and was able to predict stock price of Infosys, TCS and Cipla. This shows that, the proposed system is capable of identifying some inter relation with in the data. Also, it is evident from the results that, CNN architecture is capable of identifying the changes in trends. For the proposed methodology CNN is identified as the best model. It uses the information given at a particular instant for prediction. Even though the other two models are used in many other time dependent data analysis, it is not out performing the CNN architecture in this case. This is due to the sudden changes that occurs in stock markets. The changes occuring in the stock market may not always be in a regular pattern or may not always follow the same cycle. Based on the companies and the sectors, the existence of the trends and the period of their existence will differ. The analysis of these type of trends and cycles will give more profit for the investors.To analyze such information we must use networks like CNN as they rely on the current information.

REFERENCES

[1] A. V. Devadoss and T. A. A. Ligori, "Forecasting of stock prices using multi layer perceptron," *Int J Comput Algorithm*, vol. 2, pp. 440–449, 2013.
[2] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.
[3] V. K. Menon, N. C. Vasireddy, S. A. Jami, V. T. N. Pedamallu, V. Sureshkumar, and K. Soman, "Bulk price forecasting using spark over nse data set," in *International Conference on Data Mining and Big Data*. Springer, 2016, pp. 137–146.
[4] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
[5] G. Batres-Estrada, "Deep learning for multivariate financial time series," ser. Technical Report, Stockholm, May 2015.
[6] P. Abinaya, V. S. Kumar, P. Balasubramanian, and V. K. Menon, "Measuring stock price and trading volume causality among nifty50 stocks: The toda yamamoto method," in *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*. IEEE, 2016, pp. 1886–1890.
[7] J. Heaton, N. Polson, and J. Witte, "Deep learning in finance," *arXiv preprint arXiv:1602.06561*, 2016.
[8] H. Jia, "Investigation into the effectiveness of long short term memory networks for stock price prediction," *arXiv preprint arXiv:1603.07893*, 2016.
[9] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
[10] H. White, *Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns*, ser. Discussion paper- Department of Economics University of California San Diego. Department of Economics, University of California, 1988.
[11] B. G. Malkiel, "Efficient market hypothesis," *The New Palgrave: Finance. Norton, New York*, pp. 127–134, 1989.
[12] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction." in *IJCAI*, 2015, pp. 2327–2333.
[13] J. Roman and A. Jameel, "Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns," in *System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on,*, vol. 2. IEEE, 1996, pp. 454–460.
[14] M.-C. Chan, C.-C. Wong, and C.-C. Lam, "Financial time series forecasting by neural network using conjugate gradient learning algorithm and multiple linear regression weight initialization," in *Computing in Economics and Finance*, vol. 61, 2000.
[15] J. Roman and A. Jameel, "Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns," in *System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on,*, vol. 2. IEEE, 1996, pp. 454–460.
[16] E. W. Saad, D. V. Prokhorov, and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," *IEEE Transactions on neural networks*, vol. 9, no. 6, pp. 1456–1470, 1998.
[17] O. Hegazy, O. S. Soliman, and M. A. Salam, "A machine learning model for stock market prediction," *arXiv preprint arXiv:1402.7351*, 2014.
[18] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert systems with Applications*, vol. 19, no. 2, pp. 125–132, 2000.
[19] Y. Kishikawa and S. Tokinaga, "Prediction of stock trends by using the wavelet transform and the multi-stage fuzzy inference system optimized by the ga," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 83, no. 2, pp. 357–366, 2000.
[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.