NSE Stock prediction:The Deep learning way

Ankit K. Barai[1] and Dr.Pooja Jain[2]

Department of computer science & Engineering
Indian Institute of Information Technology, Nagpur, India
[1]ankit.barai@cse.iiitn.ac.in
[2]poojaalld@yahoo.com

Abstract. Stock market forecasting plays a vital role in the decision making of financial firms and investors. This paper focuses and details a comparative study for stock price prediction of Indian industries with stock data from NSE (National Stock Exchange). A lot of research is concentrated for stock forecasting from last decades which got significance with emergence of deep learning. The deep learning techniques focused are LSTM (Long Short Term Memory), GRU (Grated Recurrent Unit) and RNN (Recurrent Neural Network). Stock data of automobile and financial industries are taken for analysis. This paper compares the results with ARIMA model, a statistical model for stock prediction as baseline. MAPE (Mean Average Percentage Error) is used as performance criterion. This work reveals how the investors can make use of deep learning techniques to revise their investment decisions and strategies to hone better returns over time. It helps the financial analysts and business communities to make informed decisions.

Keywords: LSTM, GRU, RNN, ARIMA, Deep Learning, stock market

1 Introduction

Stock market is a platform where buyers and sellers can exchange their financial assets in the form of shares in companies listed in the stock market. NSE is national stock exchange of India and BSE stands for Bombay stock exchange. Stock market characterizes for each trading company with parameters like open price, close price, lowest price of that day and the highest price of the day and volume of stock exchanges in terms of the number of transactions of that company's stocks on a particular day. Prediction in stock market is very important to make informed decisions which might result in loss or gains of millions of dollars. Traditionally, linear statistical models like ARIMA (Autoregressive integrated moving average) [1], ARMA, MA were used. The problem with such statistical models was that they might model a given company stock data well but may not model other companies data. During this decade there had been a number of researches on how the DNN (Deep Neural Network) for time series prediction like LSTM, RNN and GRU may model the stock data. Stock market prediction is a source of interest for financial companies as well as individuals. Closing price of stock market is one of the major parameters used to make decisions for trading over that stock on the next day. It make use of DNN for time series modeling like LSTM, RNN and GRU etc. for stocks from three different sectors namely SBI, NTPC and Infosys with all three DNN. It has been observed that a DNN which may fit given company stock in best might not be best for other stock data as visible in results. All the DNN like LSTM, RNN and GRU are related to each other with slight modifications and each of them has its own pros and cons as discussed in the background work.

2 Related Works

There are a number of research groups working on application of DNN for time series modeling especially stock market. [2] makes use of RNN to capture the non linearity of stock markets.[3] gives insight on how the Recurrent fuzzy networks could be used to approximate complex systems. Various Studies on the optimization of artificial neural networks with the use of genetic algorithms are carried out as in [4]. Neural networks finds application in Optimising the cycle time of manufacturing plants [5]. Traditional neural networks are not able to retain information over time, While RNN simulates human brain's power of inference based on previous findings. RNN contains loops which favors for passing information from one step of network to the next. Selection of Activation functions is important part of neural network architecture as studied in [6]. Deep Learning for stock prediction is studied on Chinese stock markets[7] ;it also shows how it outperforms the traditional back propagation neural network and radial basis neural network.Comparative study of Support Vector Machine and DL algorithms for stock prediction[8] shows DL algorithms to be better. Study on efficient approach to forecast Indian stock market price helps naive investors in making informed decisions. Stock market is affected by National and International events which could also be taken into consideration for event driven stock prediction[9] by extracting information from news text.

3 Background

3.1 Neuron

Neurons in AI are inspired by biological neuron which works like a mathematical function taking number of inputs with weights and corresponding output. The local induced field of j^{th} neuron v_j is given by Equation 1 and effective output $y_j$is as shown in equation 2 below:

$$v_j = \sum_{i=1}^{m} (x_i * w_i) + b \tag{1}$$

$$y_j = \phi(v_j) \tag{2}$$

3.2 Neural Network

Neural net can be mathematically defined as a differentiable function that maps one kind of variable to another kind of variable. A classification problem involves vectors to vectors mapping while regression problem involves vector to scalar mapping. NN contains interconnection between neurons, which can be visualized as layers of neurons where info is transferred from one layer to other layers. Three types of layers input, hidden and output layer. There is no connection in between neurons of same layer. MLP (Multi-Layer Perceptron) is a type of feed forward neural networks with at least three layers and utilizes back propagation for training. It has nonlinear activations that differentiate it from linear perceptron.

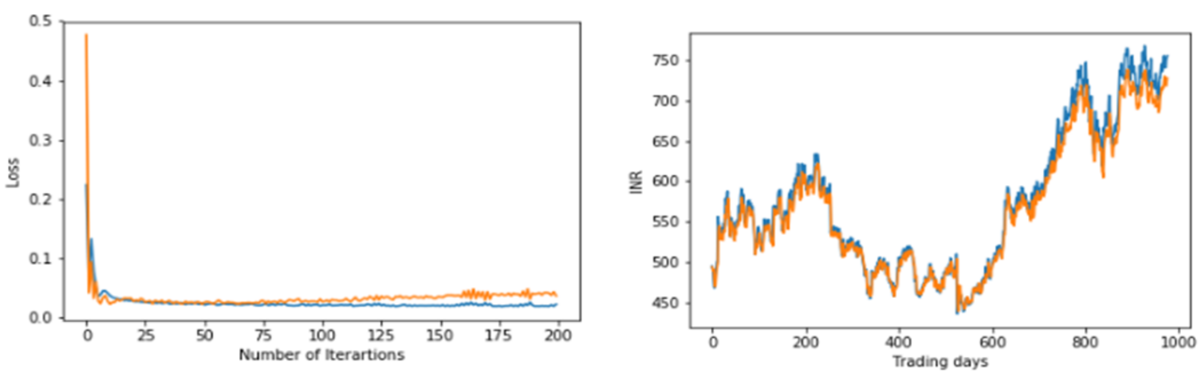


Fig. 1. Training loss curve and Prediction(Validation data) curve for INFY

3.3 Recurrent Neural Network

They are used for a variety of applications like: speech recognition [13], Language modeling [14], image captioning [15] and many more applications. The connection between nodes of RNN forms a directed graph exhibiting a temporal sequence. RNN uses their internal states to process onto a sequence of inputs. It takes two inputs as compared to one

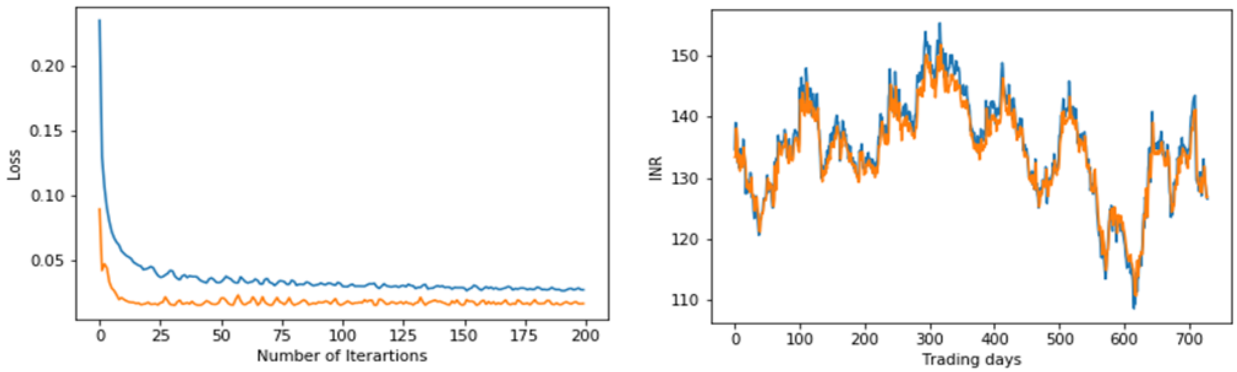


Fig. 2. Training loss curve and Prediction(Validation data) curve for NTPC

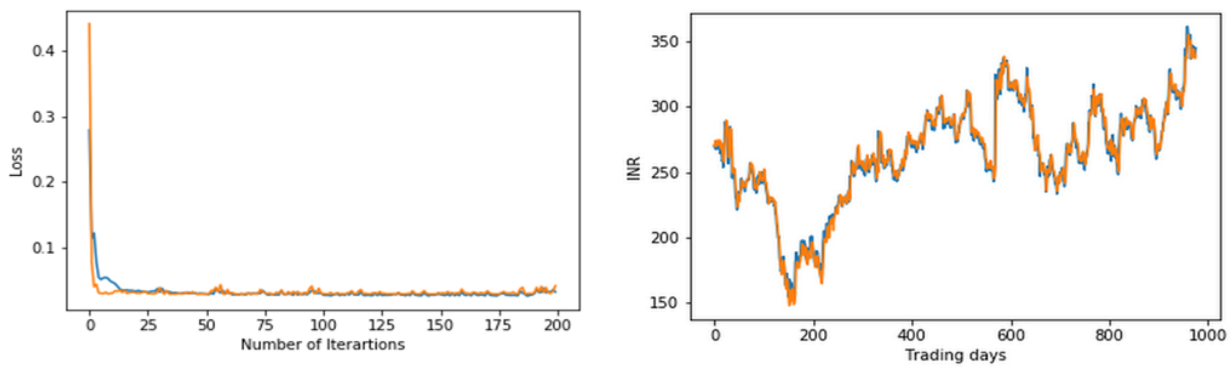


Fig. 3. Training loss curve and Prediction(Validation data) curve for SBI

input in case of MLP, one from past and other from present. RNN represents a general structure for two broad classes of networks where one is finite impulse and the other is infinite impulse. A finite impulse recurrent network can be unrolled to form a feed forward neural network while infinite impulse neural network can't be unrolled. Both of them have additional stored state which is directly controlled by neural network and can be replaced by a network or a graph. Such controlled states form a part of long short term memory and gated recurrent units.

3.4 Grated Recurrent Unit

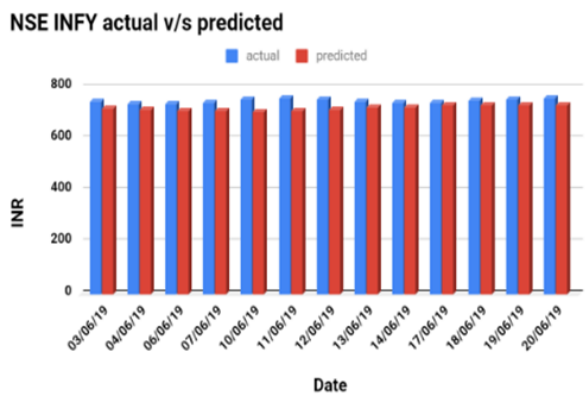
GRU is advanced RNN cell with memory unit whose architecture has a update gate which decides if previous output is to be passed to next cell or not and acts accordingly.

3.5 Long Short Term Memory

LSTM are specific RNN with capacity of learning long time data dependency.Simple RNN has a simple feedback loop in traditional NN but LSTM contains memory blocks. It has two additional gates, Forget gate and Output gate along with update gate as in GRU. Forget gate adds new set of mathematical operations with new weights. LSTM is more controllable for outputs compared to other two but with increased computational costs.

MAPE %			
Industry	GRU	RNN	LSTM
SBI	2.11	3.89	4.82
NTPC	5.41	1.66	2.94
Infosys	2.4	2.78	2.29

Table 1. MAPE for 10 days prediction with DNN model



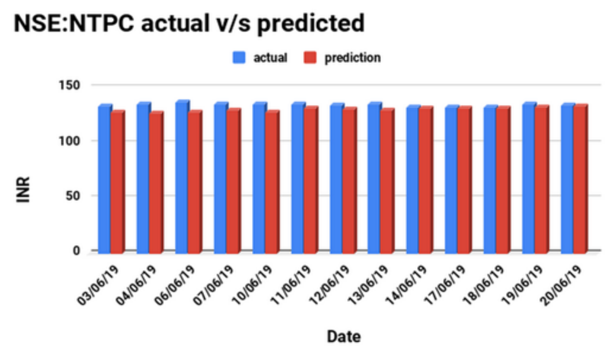


Fig. 4. Actual-predicted graph for INFY,NTPC (left to right)

4 Experiments

The historical dataset is taken from NSE stock data from the year 2000 to till date for three different sectors namely Banking, Mining and Information Technology. The companies from these sectors are SBI, NTPC and Infosys respectively. The models are trained with training data from over 19 years of stock data. The dataset comprises of various attributes namely date, low, high, open, close and volume. Closing price of stocks is extracted from the dataset to train the model since the decision about a stock is made based upon the closing price for that stock on previous day. The training data range is high and hence it is normalized before feeding it to DNN for training using Equation 3

$$y_{norm} = \frac{(y - y_{min})}{(y_{max} - y_{min})} \tag{3}$$

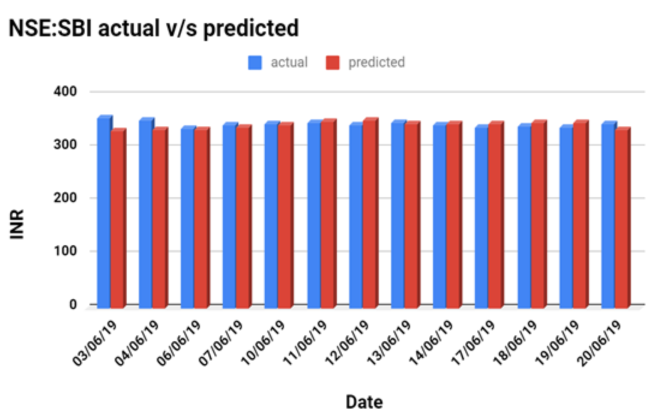


Fig. 5. Actual-predicted graph for SBI

The training data is trained over 200 epochs with varying window size($batchsize$) to tune the hyperparameters.

The error criterian used in MAPE($MeanAveragePercentageError$) as given in Equation4 where y is actual output and d is desired output.

$$MAPE = \sum\left(\frac{(y-d)}{d} * 100\right) \tag{4}$$

The algorithms used are 1 and 2. It is needed to fine tune the hyper parameters to get better predictions. Dropout layers are used to prevent over fitting of the model. The batch size and the number of epochs for training were optimized, after doing prediction on one of the industry from each of the sectors; prediction is done for some more industries from those sectors.

Algorithm 1: get_Model

```
Input: train_x.shape, n, dropout, modelType, predWindow
/* Where train_x-training data,                                        */
/* n-list of no. of Neurons in each layer,                             */
/* drop-Dropout ratio between layers,                                  */
/* modelType-lstm,gru,rnn, predWindow- Number of days to predict       */
Output: Neural network is returned
Result: DNN ready for stock prediction
// Sequential model
model = sequential() ;
input_shape= (train_x.shape[1], train_x.shape[2]);
// add Layers
model.add( modelType(n[0], input_shape, return_sequences=True));
// add dropout to prevent overfitting
model.add( Dropout(drop)) ;
model.add( modelType(n[1], input_shape, return_sequences=True));
model.add( Dropout(drop)) ;
model.add( Dense(predWindow)) ;
model.compile( loss='mape',optimiser='adam') ;
return model ;
```

Algorithm 2: Predict Stocks

```
Input: ep, bt, ntype, dataset
/* Where ep- Number of epochs, bt-batch size, ntype-DNN type,
   dataset-Dataset of historical stock prices                          */
Output: Prediction of stock prices
Result: Predicted stock prices helps in informed decision making and thus reduced loss
// get train and test data
train_x,train_y,test_x,test_y=splitData(datset) ;
// Normalise the data
train_x,train_y,test_x,test_y=Normalisation( train_x,train_y,test_x,test_y ) ;
// get the DNN for prediction
dnn=get_model( train_x.shape,n,dropout,modelType,predWindow) ;
// fit the DNN with training data
model_fit_output= dnn.fit( train_x,train_y,epochs=ep,batch_size=bt,validation_data=(test_x,test_y)) ;
// Predict with trained model
output=dnn.predict( test_x ) ;
return output ;
```

MAPE %	
Industry	ARIMA
SBI	3.99
NTPC	12.78
Infosys	7.16

Table 2. MAPE for 10 days prediction with ARIMA model

MAPE %		
Industry	DNN	ARIMA
BOI	1.37	2.57
SBI	2.11	3.99
UBI	8.08	12.99

Table 3. Banking Industry stock prediction

5 Results and Discussions

This work uses ARIMA model as its primary benchmark which is a statistical model used by business analyst extensively for stock prediction. MAPE is used for comparing the performances of DNN (Deep Neural Network) with the statistical ARIMA model. The DNNs were trained for historical stock prices from NSE for past 19 years from 2000 to 2019. It compared the prediction results for 10 future days from 03 June 2019 till 20 June 2019 with the actual data for all three DNN and the best fitting models for each of the stocks with minimal MAPE is selected as the results. For SBI, statistical model ARIMA gave MAPE of 3.99% which reduced to 0.89% with DNN, for Wipro error reduced from 6.36% to 0.77% with shift from statistical model to DNN while as for Infosys it reduced from 7.16% to LSTM was better for NTPC with 2.94% error and with 2.24% error for Infosys. The results of predictions for validation on test data for Infosys in figure 1;for NTPC in figure2; for SBI in 3 . Table 1 and table 2 compares the performance of statistical ARIMA with other Deep learning algorithms. Figure 4 and 5 compares the predictions with actual result for ten future days, it is evident that predicted prices are very close to actual stock prices and will help businesses to make informed decisions. A Study of Deep learning algorithms is also validated on Various Industry sectors as given in table3,4 and 5.

6 Conclusions and Future Scope

This paper makes use of three DNN for time series modeling of stock data from three different sectors namely banking, Mining and information technology. The companies from these sectors were SBI, NTPC and Infosys respectively. The prediction was made for ten days in future and was compared with actual results. DNN models were found to be better than statistical models like ARIMA and they model stock data well. LSTM and GRU are related to each other and either of them may give a better fit for a given stock depending upon tuning of hyper parameters. It is concluded that DNN better models stock data compared to linear statistical models like ARIMA and outperforms it in performance. This work does not take into account other factors affecting stock markets like political and social media trends. Sentiment analysis using social media analytics can also be integrated with stock prediction to account for these factors. This work would be further extended by integrating sentiment analysis with stock prediction.

MAPE %		
Industry	DNN	ARIMA
INFY (Infosys)	2.29	7.16
ROLTA	1.47	6.07
WIPRO	0.77	6.36

Table 4. Information technology Industry prediction

MAPE %		
Industry	DNN	ARIMA
NTPC	1.66	12.78
ONGC	2.11	4.45
RCF	1.9	3.58

Table 5. Mining Industry prediction

References

1. A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the arima model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 106–112, IEEE, 2014.
2. A. M. Rather, A. Agarwal, and V. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3234–3241, 2015.
3. A. Dass and S. Srivastava, "On comparing performance of conventional fuzzy system with recurrent fuzzy system," in *Soft Computing: Theories and Applications*, pp. 389–403, Springer, 2018.
4. H. Bhasin and N. Gupta, "Critical path problem for scheduling using genetic algorithm," in *Soft Computing: Theories and Applications*, pp. 15–24, Springer, 2018.
5. J. P. Giri, P. J. Giri, and R. Chadge, "Neural network-based prediction of productivity parameters," in *Soft Computing: Theories and Applications*, pp. 83–95, Springer, 2018.
6. B. Dhamayanthi, J. S. Vaiz, and M. Ramaswami, "A study of deep neural networks in stock trend prediction using different activation functions," in *International Conference on Recent Trends in Engineering, Computers, Information Technology and Applications (ICRTECITA-2017)*.
7. L. Chen, Z. Qiao, M. Wang, C. Wang, R. Du, and H. E. Stanley, "Which artificial intelligence algorithm better predicts the chinese stock market?," *IEEE Access*, vol. 6, pp. 48625–48633, 2018.
8. R. Y. Nivetha and C. Dhaya, "Developing a prediction model for stock analysis," in *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, pp. 1–3, IEEE, 2017.
9. X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.