Application of Deep Learning Techniques for Precise Stock Market Prediction

Saikat Mondal
School of Computing and Analytics
NSHM Knowledge Campus
Kolkata, India
email: saikatmondal1.18@nshm.edu.in

Abhishek Dutta
School of Computing and Analytics
NSHM Knowledge Campus
Kolkata, India
email: abhishekdutta.18@nshm.edu.in

Piyali Chatterjee
School of Computing and Analytics
NSHM Knowledge Campus
Kolkata, India
email: piyalichatterjee.18@nshm.edu.in

***Abstract*— the purview of stock price analysis largely depends on the ability to identify the movement of the stock prices and predict the hidden patterns and trends which the market follows. The sole idea is to gain profit from the investments that we make, therefore the more sure we will be with our predictions, safer will be the outlay. Predictions based on stock prices has been a constant field of research work in the past, however, obtaining the desired level of precision is still an engaging challenge. In this script, we are proposing a combined effort of using efficient machine learning techniques coupled with a deep learning technique (like LSTM) to use them to predict the stock prices with a high level of accuracy. We are considering the daily index values of three different companies namely HDFC Bank, Tata Consultancy Services, Cipla which are from different segments of the market – finance, IT and medical science. We are using their daily data of previous 6 years (2013-18) to prepare a training model and implement the results on the test data set to predict the closing values of these National Stock Exchange (NSE) listed companies from January 1 to December 31, 2019. For the prediction of the patterns of the price movements, we are using efficient classification techniques, and for the actual closing values, we are using various techniques of regression. Methods which have been implemented involve logistic regression, SVM (Support Vector Machines), ANN (Artificial Neural Network), Random Forest, ensemble learning techniques (Bagging, Boosting). We also use the deep learning technique of Long Short-Term Memory (LSTM) for the prediction of the closing prices of the stocks and then superimpose the accuracy measures by comparing the LSTM results with the other machine learning models.**

Keywords—stock price analysis, machine learning, deep learning, regression, classification, support vector machines, random forest, artificial neural networks, and long short-term memory.

I. INTRODUCTION

The concept of making money from stock market investments is very popular in the world at this moment. In India, the National Stock Exchange and the Bombay Stock Exchange are the two most involving names which provide us with the silhouette of this part of the country's economy. As we know, stock market investments are 'subjected to market risks' and hence this common question always arises whether there is a way to be sure about our gamble. Stock market investments are all about making profits from the money that we decide to bestow to the companies. However, the returns that follow don't always stalk our speculations. A large segment of prominent stalwarts believed that the stock prices are the most difficult entities to be predicted, which involves complications which are not interpret-able due to their regular fluctuations due to impact from every segment of the society starting from politics to agriculture. On the other side of this challenge, there have been works based on this topic where it was shown that with proper modelling of data, the stock process can be predicted with quite a remarkable level of accuracies. Analyzing stock prices has a major challenge in the form of detecting the patterns in which the data is distributed. If we can identify it using proper modelling of the data, then predicting the future values becomes much easier [1-6]. However, even with efficient machine learning techniques, it is difficult to force through certain threshold levels of accuracy. This is where deep learning steps in, which can provide results with much higher levels of precision [8-9]. Various techniques like moving average, moving average convergence divergence (MACD), relative strength index, Fibonacci fan etc. have been applied for the same quest.

In this paper, we are suggesting a combined effort involving multiple machine learning techniques and deep learning techniques for providing visual representations through tables for enabling us to get a clear outline which helps the regular public with amateur knowledge in this domain, to anticipate their future moves. We are using the data of three major companies enlisted in the National Stock Exchange (NSE) from three different domains. The fields that we have chosen are financial banking – HDFC Bank, information and technology sector – Tata Consultancy Services (TCS), medical sector _ Cipla. The selection was made to cover a wide purview of the economy with these three, forming the core of the market in their respective domains. We have used daily data of these companies for a period of six years (2013-2018) to prepare the training model. Based on the results which the model provides us, we have put the 2019 data for testing. The predicted values of 2019 or the test data are then being compared to the actual values which give the error in prediction. In the process, the accuracy of the model built is also affected. Subsequently, we can forage for improvements of the models by further tweaking and improving the approach of problem-solving. The machine learning techniques produced the results which were significantly boosted by the deep learning technique *LSTM*. Coagulation of these two approaches has produced quite striking results with accuracy rates high enough to be accepted as benchmarks. However, more complicated models of deep learning can notch it up higher, which we can pursue later for more interesting results.The paper organization is as follows:

Section II. Problem definition – an in-depth look at our problem at hand. Section III contains previous works on which we have based our idea. Section IV shows the methodology of our approach, accompanied by the results from our hands-on work based in R and Python. Section V contains Details of all the predictive models and their characteristics. Section VI provides the conclusion.

II. PROBLEM STATEMENT

The goal which we had set in long sight was to utilize the daily data of three major companies from three different sectors and predict their stock price movement for future endeavours. Application of machine learning techniques is used to model the data and identify the hidden patterns in which the data is behaving (regression and classification results). An efficient deep learning technique like Long Short-Term Memory is used, which significantly improves our predicting powers. The training set results are then pondered upon the test results of the 2019 daily closing values of stocks. This approach does not provide us with instantaneous results from a practical point of view, because deep learning methods require huge bulk of data from which the machine can learn more efficiently. Therefore, we have results in our hand which are more suited for long term analogy and investments. In this paper, we have monitored data over a large span of time, and then based our conclusions on a recent time slot (2019) to justify the approach and its advantages.

The behaviour of a stock market is very unpredictable due to various reasons, one which is its dependence on widespread factors. Multiple key factors which contribute to the cause are industrial growth, economic growth, agricultural turnovers, natural resources reserve, and livestock etc. Sectors like fast-moving consumer goods (FMCG), information technology (IT) fluctuate rapidly over a short period of time. On the other side, metal sectors and oil and natural gas sectors are comparatively slow bloomers. The amount of money which gets availed in the market is always a constant, hence, any spike or trough in the market is always complemented by another subsequent spike or trough from somewhere else. Thus the effective value of the market remains stagnant, while the individual prices go up and down. We propose to predict these movements of stock prices in this paper. We can predict the seasonality and trend in a data set and detect out the patterns but the third factor which is the randomness of data can never be accurately determined, where we can only try and improve our quest. For example – the demonetization of India in November 2016 was an unpredictable move by the government which severely affected the market and economy cannot be predicted beforehand and hence can be categorized as a random event.

III. RELATED WORK

The field of stock prediction involves using simple regression techniques on the data, which provides mediocre results in terms of accuracy as we know that stock prices fluctuations are non-linear in nature [7]. Second, in terms of approach come the traditional econometric methods of time series analysis like ARIMA A*utoregressive Integrated Moving Average (*ARIMA)[10,15] and A*utoregressive Moving Average* (ARMA)[12], *Granger causality test*, regressive models with distributed lag and *Quantile Regression* for forecasting of the values [16]. The third in chronology lies the modern machine learning [13,14,17] and deep learning techniques [19], natural language processing (NLP) [11], which represents the state of the art and produces impressive results.

However, the challenge which the traditional mathematics-based approaches ponder is their ability to accurately predict the forecasts because of their non-linear nature. We have tried to eliminate these adversaries and construct a versatile approach for this venture.

IV. METHODOLOGY

As the introduction of the document states that we are trying to predict the daily closing data for the \three NSE listed prominent companies namely HDFC Bank, TCS and Cipla. We have gathered their raw data from the NSEIndia website [18]. For the training set, we have data starting from 1st January 2013 up to 31st December 2018. For the test data set whose values we will predict using the models, we have considered the data spread from 1st January 2019 to 31st December 2019.

The raw data from the website was downloaded in Microsoft Excels comma-separated values format (CSV file). The files consisted of columns headed as 1) Date 2) Open values 3) Closed values 4) High values 5) Low values 6) Volume.

We have cleaned the data according to our necessities, especially by changing the open, close, high, low, volume values into a normalized form to simplify the job by reducing the amount of processing that is needed.

a) *day*: this variable is referred to as the day of the month which ranges from *[1-31]* corresponding to the number of a day of a single month.

b) *month:* this variable is referred to as the numbered value of a month ranging from *[1-12]*, like 1 for January,2 for February respectively.

c) *year:* this variable is referred to as the value of a year of which our data belongs to, as our data belongs to the years *[2013,2014,2015:: 2019*].

d) *close_norm:* this variable refers to the normalized value for the closing prices which is calculated using the following equation, [(*previous closing price – present closing price)/ previous closing price*].

e) *open_norm*: this variable refers to the normalized value for the opening prices which is calculated using the following equation, *[(previous opening price – present opening price)/ previous opening price]*.

f) *high_norm*: this variable refers to the normalized value for the highest prices which is calculated using the following equation, *[(previous highest price – present highest price)/ previous highest price]*.

g) *low_norm*: this variable refers to the normalized value for the lowest prices which is calculated using the following equation, *[(previous lowest price – present lowest price)/ previous lowest price]*.

h) *volume_norm:* this variable refers to the normalized value for the volume of stocks transacted, which is calculated using the following equation, *[(previous volume value – present volume value)/ previous volume value]*.

i) *range_norm:* this variable refers to the normalized value for the range of price within the high and low stock prices which are calculated using the following equation, *[((present high price – present low price) - (previous high price - previous low price)) / (previous high price – previous low price)]*.

These are the variables that have been used in the entire work.

The negative values of the normalized values indicate that there has been a fall in the value when compared with its previous day, while the positive values indicate that there has been an upward peak in the current value when compared to its previous day.

For classification, we converted close_norm values to 0 if it was a negative change that is, falling of the stock value; else it is converted to 1 if there was a positive change from the previous value.

V. Performance Results

In this section, we have with a detailed discussion of the results that we have obtained from the different machine learning techniques.

We have used the machine learning techniques for regression as well as classification.

To evaluate our classification models the metrics that we are using are as follows:-

Sensitivity:- It is the proportion of true positives to the total number of positives in the data-set.

Specificity:- It is the proportion of true negatives in the total number of negatives in the data-set.

PPV:-(Positive Predictive Value) It is the proportion of the true positives to the sum of the true positives and the false positives in the data-set.

NPV:-(Negative Predictive Value) It is the proportion of the true negatives to the sum of the true negatives and the false negatives in the data-set.

CA(Classification Accuracy): It is the proportion of the number of cases correctly classified to the total number of case in the data-set.

The tables I - VI shows the classification results for the various machine learning models starting with logistic regression, followed by boosting, bagging, ANN, random forest, SVM.

TABLE I. Logistic Regression

Stock	**Training Data**		**Test Data**	
TCS	Sensitivity	85.39	Sensitivity	80.08
	Specificity	83.35	Specificity	80.17
	PPV	84.73	PPV	81.45
	NPV	84.06	NPV	79.48
	CA	84.41	CA	80.49
CIPLA	Sensitivity	79.66	Sensitivity	70.00
	Specificity	83.35	Specificity	76.15
	PPV	81.72	PPV	71.29
	NPV	81.43	NPV	75.00
	CA	81.57	CA	73.00
HDFC	Sensitivity	82.18	Sensitivity	84.55
	Specificity	79.94	Specificity	83.05
	PPV	81.22	PPV	83.87
	NPV	80.95	NPV	83.79
	CA	81.09	CA	83.81

TABLE II. Boosting

Stock	**Training Data**		**Test Data**	
TCS	Sensitivity	100	Sensitivity	82.90
	Specificity	100	Specificity	83.90
	PPV	100	PPV	85.60
	NPV	100	NPV	83.90
	CA	100	CA	83.40
CIPLA	Sensitivity	100	Sensitivity	70.29
	Specificity	100	Specificity	71.94
	PPV	100	PPV	64.50
	NPV	100	NPV	76.92
	CA	100	CA	71.25
HDFC	Sensitivity	100	Sensitivity	81.96
	Specificity	100	Specificity	80.67
	PPV	100	PPV	81.30
	NPV	100	NPV	81.35
	CA	100	CA	81.32

We can see from the tables, Among all the techniques which have produced fairly good results, Boosting produces the best result by-far.

TABLE III. BAGGING

Stock	Training Data		Test Data	
TCS	Sensitivity	82.5	Sensitivity	79.2
	Specificity	87.8	Specificity	81.88
	PPV	88	PPV	82.5
	NPV	82.2	NPV	78.5
	CA	85.09	CA	80.49
CIPLA	Sensitivity	81.4	Sensitivity	61
	Specificity	93.4	Specificity	72.3
	PPV	92.1	PPV	65.3
	NPV	82.83	NPV	69
	CA	82.5	CA	67.5
HDFC	Sensitivity	81.66	Sensitivity	78.0
	Specificity	83.56	Specificity	78.81
	PPV	83.98	PPV	79.33
	NPV	81.19	NPV	77.5
	CA	82.58	CA	78.42

TABLE IV. ANN

Stock	Training Data		Test Data	
TCS	Sensitivity	84.48	Sensitivity	60.00
	Specificity	84.48	Specificity	89.00
	PPV	85.48	PPV	67.53
	NPV	83.42	NPV	86.20
	CA	84.48	CA	74.27
CIPLA	Sensitivity	78.54	Sensitivity	40.00
	Specificity	84.01	Specificity	96.15
	PPV	82.11	PPV	89.79
	NPV	80.73	NPV	65.44
	CA	81.36	CA	70.12
HDFC	Sensitivity	44.85	Sensitivity	24.39
	Specificity	60.72	Specificity	75.42
	PPV	54.66	PPV	50.84
	NPV	51.05	NPV	48.90
	CA	52.57	CA	49.37

As we keep exploring through the models it has come to light that, among all the techniques Boosting and Random Forest has produced the best result by far.

TABLE V. RANDOM FOREST

Stock	Training Data		Test Data	
TCS	Sensitivity	100	Sensitivity	83.20
	Specificity	100	Specificity	81.03
	PPV	100	PPV	82.53
	NPV	100	NPV	81.73
	CA	100	CA	82.15
CIPLA	Sensitivity	100	Sensitivity	63.63
	Specificity	100	Specificity	78.46
	PPV	100	PPV	71.42
	NPV	100	NPV	71.83
	CA	100	CA	71.38
HDFC	Sensitivity	100	Sensitivity	79.67
	Specificity	100	Specificity	77.11
	PPV	100	PPV	78.40
	NPV	100	NPV	78.44
	CA	100	CA	78.42

TABLE VI. SVM

Stock	Training Data		Test Data	
TCS	Sensitivity	85.88	Sensitivity	80.08
	Specificity	81.86	Specificity	80.17
	PPV	82.52	PPV	81.45
	NPV	85.33	NPV	79.48
	CA	83.87	CA	80.49
CIPLA	Sensitivity	79.66	Sensitivity	71.55
	Specificity	82.96	Specificity	75.57
	PPV	81.37	PPV	70.90
	NPV	81.36	NPV	76.15
	CA	81.57	CA	73.44
HDFC	Sensitivity	82.18	Sensitivity	82.03
	Specificity	79.94	Specificity	84.07
	PPV	81.22	PPV	85.36
	NPV	80.95	NPV	80.50
	CA	81.09	CA	82.98

For regression results, for calculating the accuracies, we have reverse transformed the normalized values by using the formula (RMSE/mean absolute values) %.

The tables from VII -XI shows the regression results.:-

TABLE VII. BAGGING

Stock	Training Data		Test Data	
TCS	Cor RMSE/Mean	0.85 72.165	Cor RMSE/Mean	0.81 74.11
CIPLA	Cor RMSE/Mean	0.826 79.18	Cor RMSE/Mean	0.72 95.21
HDFC	Cor RMSE/Mean	0.83 76.58	Cor RMSE/Mean	0.808 88.68

TABLE VIII. BOOSTING

Stock	Training Data		Test Data	
TCS	Cor RMSE/Mean	0.91 55.59	Cor RMSE/Mean	0.87 61.17
CIPLA	Cor RMSE/Mean	0.89 62.11	Cor RMSE/Mean	0.89 62.11
HDFC	Cor RMSE/Mean	0.89 62.15	Cor RMSE/Mean	0.89 62.153

TABLE IX. RANDOM FOREST

Stock	Training Data		Test Data	
TCS	Cor RMSE/Mean	0.97 31.48	Cor RMSE/Mean	0.86 64.93
CIPLA	Cor RMSE/Mean	0.97 36.39	Cor RMSE/Mean	0.72 94.90
HDFC	Cor RMSE/Mean	0.97 35.98	Cor RMSE/Mean	0.824 85.96

TABLE X. ANN

Stock	Training Data		Test Data	
TCS	Cor RMSE/Mean	0.61 115.31	Cor RMSE/Mean	0.84 87.37
CIPLA	Cor RMSE/Mean	0.84 68.55	Cor RMSE/Mean	0.773 72.26
HDFC	Cor RMSE/Mean	0.13 333.3	Cor RMSE/Mean	0.88 66.11

TABLE XI. SVM

Stock	Training Data		Test Data	
TCS	Cor RMSE/Mean	0.89 59.96	Cor RMSE/Mean	0.85 68.28
CIPLA	Cor RMSE/Mean	0.87 66.97	Cor RMSE/Mean	0.71 100.19 8
HDFC	Cor RMSE/Mean	0.87 66.59	Cor RMSE/Mean	0.70 104.22

We can clearly see that we have results which are not at par with our expected levels of accuracy. While some of the techniques have proposed fairly good results like *Random Forest* and *Boosting* on test data with comparatively low RMSE/mean values.

At the last level, we apply deep learning method of *Long Short-Term Memory* or commonly known as LSTM technique. LSTM is a very efficient method which is a sub-branch of Recurrent Neural Networks (RNN). LSTM technique is emphatically designed to solve the dependency problem for longer period prevailing in other networks. RNN has a feature which LSTM inherits, unlike standard neural networks, is the back-propagation method. Also known as feedback connections, the back propagation helps to resolve the weightage problem (vanishing gradient and exploding gradients) of the nodes and leads us to more impressive results.

We have used *Python* as our tool of choice for the LSTM regression, *Tensorflow 2.0.0* framework and Keras serving as the deep learning platform [20]. The *close_norm* values of the three NSE listed companies were predicted on a daily forecast horizon with *Mean Absolute Percentage Error(MAPE)* and *Root Mean Squared Error* divided by the mean *(RMSE/Mean)* as the loss function. The final result was obtained by several experimentations of the epoch values and changing of the data size sets.

TABLE XII. LSTM REGRESSION

Stock	Training Data		Test Data	
TCS	RMSE/Mean MAPE	0.043 3.177	RMSE/Mean MAPE	0.187 11.62
CIPLA	RMSE/Mean MAPE	0.091 0.457	RMSE/Mean MAPE	0.611 23.65
HDFC	RMSE/Mean MAPE	0.325 30.74	RMSE/Mean MAPE	0.630 60.29

As we can see the final results of the LSTM regression technique, the training data for HDFC, TCS and Cipla were finalized with a number of epochs as follows – 12, 55, 15 respectively. Initially, we started with the number of epochs set as 50 for TCS, 55 for Cipla, 65 for HDFC. With the initial epochs, TCS produced an RMSE of 1.475 for training and 1.391 for test data. Cipla produced RMSE of 0.14 for the training set and 1.147 for the test data. HDFC produced RMSE of 0.064 for the training set and 0.084 for the test.

However, our main focus should be on predicting the closing values of the test data set, therefore we aimed for higher accuracy of the test data set results. Therefore the final epochs turned out to be 12, 55 and 15 to be optimal, which produced augmented results with accuracy range touching 97-98%, by converging the losses.

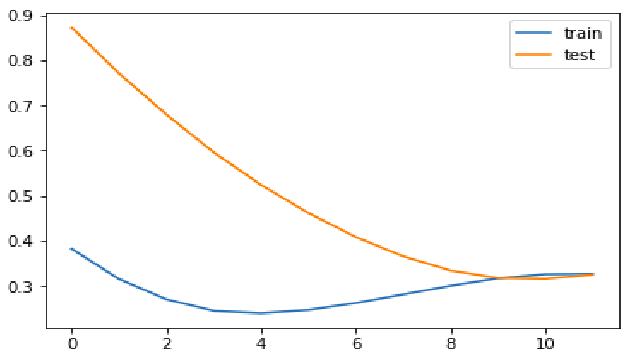

Fig. 1. Convergence Diagram for HDFC Dataset

For HDFC, we started with 65 epochs initially. We changed the number of epochs gradually and we finally stopped at 12. The model has converged, for training and validation, which we have not shown here due to time constraint.

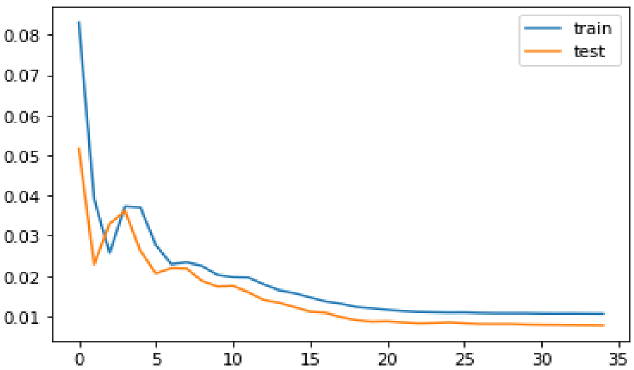

Fig. 2. Convergence Diagram for TCS Dataset

For TCS, we started with 55 epochs initially. We changed the number of epochs gradually and we finally stopped at 50, where the model has produced the best results.

In the case of the Cipla data set, we started with 65 epochs initially. We changed the number of epochs gradually and we finally stopped at 12. The model has converged, for training and validation, which we have not shown here due to time constraint.

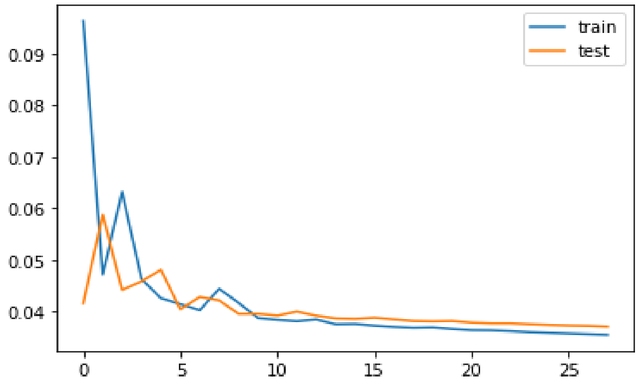

Fig. 3. Convergence Diagram for CIPLA Dataset

VI. Conclusion

In this paper, we have proposed that deep learning techniques like LSTM can provide highly accurate predicted values of stock prices, which are generally non-linear in nature and hence they are very hard to predict using traditional econometrics and mathematical approaches. Our proposition can be found useful for other financial domains also like mutual fund market risk factors and also insurance policies. The fluctuations of the market due to various reasons are very hard to foresee using statistical models. Hence seeking help from deep learning is preferable as it helps to finely scrutinize the cases. We have shown that the conjunction of machine learning and deep learning models together can produce fantastic results which will help us to move ahead with investments in the stock market in a confident and risk-free manner. In normal approach, neural networks using back-propagation can lead to two gradient related problems, the tendency of gradient moving towards zero (also called the vanishing gradient) or their increasing motion towards infinity (called exploding gradient). LSTM partially eliminate this problem and hence provide a much robust solution of the model.

As we know, no model can be classified as a hundred per cent accurate model and we can always work towards improvement. If investors opt for more accurate prediction, we can vouch for more complex models and theories and explore into deeper layers of deep learning which we are planning to undertake in the future.

References

[1] J. Sen and T. Datta Chaudhuri, "An alternative framework for time series decomposition and forecasting and its relevance for portfolio choice - a comparative study of the Indian consumer durable and small-cap sector," Journal of Economics Library, vol. 3, no. 2, pp. 303- 326, 2016.

[2] J. Sen and T. Datta Chaudhuri, "Decomposition of time series data of stock markets and its implications for prediction - an application for the Indian auto sector," In Proceedings of the 2nd National Conference on Advances in Business Research and Management Practices(ABRMP'2016), Kolkata, India, pp. 15-28.

[3] J. Sen and T. Datta Chaudhuri, "An investigation of the structural characteristics of the Indian IT sector and the capital goods sector – an application of the R programming language in time series decomposition and forecasting," Journal of Insurance and Financial Management, vol. 1, no. 4, pp. 68 - 132, 2016.

[4] J. Sen and T. Datta Chaudhuri, "A time series analysis-based forecasting framework for the Indian healthcare sector," Journal of Insurance and Financial Management, vol. 3, no. 1, pp. 66 - 94, 2017.

[5] J. Sen and T. Datta Chaudhuri, "A predictive analysis of the Indian FMCG sector using time series decomposition-based approach," Journal of Economics Library, vol. 4, no. 2, pp. 206 - 226, 2017.

[6] J. Sen, "A time series analysis-based forecasting approach for the Indian realty sector," International Journal of Applied Economic Studies, vol. 5, no. 4, pp. 8 - 27, 2017.

[7] B. Rosenberg, K. Reid, and R. Lanstein, "Persuasive evidence of market inefficiency", Journal of Portfolio Management, vol 11, pp. 9 – 17, 1985.

[8] J. Sen, "A robust analysis and forecasting framework for the Indian mid cap sector using time series decomposition," Journal of Insurance and Financial Management, vol. 3, no. 4, pp. 1- 32, 2017.

[9] J. Sen and T. Datta Chaudhuri, "Understanding the sectors of Indian economy for portfolio choice," International Jouranl of Business Forecasting and Marketing Intelligence, vol. 4, no. 2, pp. 178-222.

[10] J. E. Jarrett and E. Kyper, "ARIMA modelling with intervention to forecast and analyze Chinese stock prices," International Journal of Engineering Business Management, vol 3, no 3, pp. 53 – 58, 2011.

[11] Q. Wu, Y. Chen, and Z. Liu, "Ensemble model of intelligent paradigms for stock market forecasting," In Proceedings of the IEEE.

[12] P Mondal, L. Shit, and S. Goswami, "Study of effectiveness of time series modeling (ARMA) in forecasting stock prices," International Journal of Computer Science, Engineering and Applications, vol. 4,pp. 13 - 29, 2014.

[13] 1st International Workshop on Knowledge Discovery and Data Mining, Washington DC, USA, pp. 205 – 208, 2008.

[14] T. A. Siddiqui and Y. Abdullah, "Developing a nonlinear model to predict stock prices in India: an artificial neural networks approach," IUP Journal of Applied Finance, vol. 21, no. 3, pp. 36 - 39, 2015.

[15] A. Adebiyi, O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," In Proc. of the International Conference on Computer Modelling and Simulation, Cambridge, UK, pp. 105 - 111, 2014.

[16] S. Mishra, "The quantile regression approach to analysis of dynamic interaction between exchange rate and stock returns in emerging markets: case of BRIC nations," IUP Journal of Financial Risk Management, vol 13, no 1, pp 7 -27, 2016.

[17] G. Dutta, P. Jha, A. Laha, and N. Mohan, "Artificial neural network models for forecasting stock price index in the Bombay Stock Exchange," Journal of Emerging Market Finance, vol 5, no 3, pp 283 - 295, 2006.

[18] NSEIndia website - https://www.nseindia.com/.

[19] V. Zoca, G. Spacagna, D. Slater, and P. Roelants, Python Deep Learning, Packt Publishing Ltd., UK, 2017.

[20] A.Geron, Hands-on Machine Learning with Scikit Learn Keras & Tensorflow, O'Reilly Publications, USA, 2019.