Practicum Report

Project Title:	Stock Market prediction using Statistical & Machine Learning Techniques
Student ID:	18210686
Student name:	Paritosh Gupta
Student email	paritosh.gupta3@mail.dcu.ie
Chosen major:	Computing [Data Analytics]
Supervisor	Marija Bezbradica
Date of Submission	11-Aug-2019

Disclaimer

An essay submitted to Dublin City University, School of Computing for module CA685 Data Analytics Practicum, 2018/2019. I understand that the University regards breaches of academic integrity and plagiarism as grave and serious. I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the sources cited are identified in the assignment references. I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work. By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name: Paritosh Gupta
Date: August 11, 2019

Stock Market prediction using Statistical & Machine Learning Techniques

Paritosh Gupta
Dublin City University, School of Computing
Dublin, Ireland
paritosh.gupta3@mail.dcu.ie

***Abstract*— Stock market prediction is considered as a hot topic in the financial world which also pulled the attention of various researchers and data scientists to develop a predictive model. Being highly noisy data makes stock prices trend prediction a big challenge. In this paper, the main goal was to perform the time series analysis over the Indian stock market, National Stock Exchange (NSE) data, and compare the results of the Statistical methods and Machine learning techniques. Daily market price movement from the past are used as technical indicators to identify the closing price of the stock on very next day to weeks price. The focus will be mainly on to make a short-term prediction or identifying the trend for the upcoming week and compare the results of various methods like ARIMA for classical time-series analysis, an advanced method like LSTM for sequence forecasting and regression model, LASSO.**

Keywords*— *Time Series, Stock Market Prediction, Short-term prediction, long-term prediction, Machine learning techniques, Statistical techniques, ARIMA, LSTM, and LASSO.

I. INTRODUCTION

Stock Market Prediction will continue to be an interesting area for Data Scientist, not only for the desire for material gain but also for the challenge to understand the hidden dynamics of the volatile stock market. As a Data Scientist, looking at daily ups and downs of the market and thinking if there is any hidden pattern that we or our models can learn to empower the individuals to make effective financial decisions. Prediction of the stock price is considered to be one of the most challenging tasks in the financial world due to its nonlinearity and complex behavior. It is one of the oldest problems which is still completely not solved because of its highly volatile behavior and driven by a high degree of noise from various factors [10].

Stock price movement consists of hidden dynamics, which also not constant over time. It depends on several factors which include- (a) economic variables, such as interest rates, exchange rates, monetary growth rates, commodity prices, and general economic conditions; (b) industry-specific variables, such as growth rates of industrial production and consumer prices; (c) company-specific variables, such as changes in company policies, income statements, and dividend yields; (d) psychological variables of investors, such as investors' expectations, investors sentiments, and institutional investors' choices; (e) political variables, such as the occurrence and the release of important political events[7], but frequently traders rely on technical indicators, based on stock data from the past to predict the stock price. Depending on the number on the number of days for which prediction is made, there are three types of stock price prediction can be done [4]-

a) Short-term prediction, (prediction for days or months).

b) Medium-term prediction, (prediction up to 2 years).

c) Long-term prediction, (prediction beyond 2 years).

Most of the forecasting problem involves time series analysis. A time-series data can be defined as the sequence of observations for any variable in chronological order of time[1]. Time Series Analysis consists of methods for the analysis of time series data and to obtain meaningful information, by analyzing the data from the past to predict future values. It can be classified into three types depending on the number of variables used for the analysis [11]-

(i) Univariate, only one variable used.

(ii) Bivariate, only two variables are used.

(iii) Multivariate, more than two variables are used.

The aim of this work will be to perform univariate time series analysis for performing short-term forecasting. Here, we will be trying to perform a few of the classical time series analysis, starting with the statistical approaches and moving towards the advanced Machine Learning approaches for the sequence learning task.

Autoregressive Integrated Moving Average (ARIMA), and deep learning model, long short-term memory (LSTM), which were quite popularly used in past as well but latest linear models were not explored in this area. I tried to convert the time series problem into supervised learning problem by sliding window approach, to apply the linear model like 'Least Absolute Shrinkage and Selection Operator' (LASSO). Predicting the stock price always challenging but if we can understand the trend at least for a week, then it will allow the investor to make better investment decisions for the short term.

II. RELATED WORKS

There are many models and techniques developed in past several years for time-series predictions, such as moving average (MA), ARIMA, LSTM, multilayer perceptron

(MLP), recursive neural network (RNN), ANN, etc. Among these models, from the statistical perspective, ARIMA has shown strong potential for short-term prediction, and from the deep learning perspective, LSTM is quite popular due to its ability to learn pattern on its own and to understand the hidden dynamics [2]. Although, there were supervised learning methods like RFC and SVM were explored but hasn't shown any improvement over advanced models like LSTM.

ARIMA model was introduced in 1970 by Box and Jenkins. It is a robust model which was considered to be the most promising method for financial forecasting. It continuously outperformed the complex models in short-term prediction. In ARIMA, the future value is calculated by the linear combination of past values and the past error. In [2], data from two countries stock exchange were used as the input variable. These data consisted of four elements, open price, low price, high price and close price, were close price was considered to be the price for that day. Nokia and Zenith bank stocks were considered for the experiment, where autocorrelation functions (ACF) and partial autocorrelation functions (PACF) were obtained from the data set. These ACF and PACF's are converted to stationary, i.e. no significant difference between them, by differentiating. Once ACF and PACF have no significant spikes and considered to be a white noise model, which does not have any significant noise left in time series. Both the stocks were becoming stationary after first difference and considered to be the best-fitted model. These models had a smaller standard error of regression as compared to other models. There were some instances were predicted and actual prices were very close. Therefore, it was concluded that the best ARIMA model shown satisfactory results on short-term and has a high potential for short-term prediction.

LSTM network, which is a specific kind of RNN is considered as one of the advanced deep learning architectures for the sequence learning task. In [3], the author applied LSTM on large scale financial market prediction task on 'S&P 500' to contrast the findings with different benchmarks from literature, a random forest (RAF), standard deep neural network (DNN) and standard logistic regression. The methodology consists of five steps, first splitting the data in the training set and trading set. Second, considering the features need to be considered for prediction. Third, an in-depth discussion of the LSTM network. Fourth, briefly describing RAF, DNN, and logistic regression. Finally, developed the trading approach [3]. Data handling was done by various python packages such as NumPy, Pandas, Sci-Kit learn, and LSTM network was developed by Keras and Google Tensor flow and finally, for statistical computing, R is used. As per the concluded results, LSTM outperformed RAN, DNN, and logical regression, statistically and economically. Only during the global crisis, RAF outperformed.

Further comparison between LSTM, RNN and the convolutional neural network was discussed in [4], where the author made a multivariate model, to include the price of different stocks as input. For the purpose, two stocks from different sectors were selected from the Indian stock market, NSE. The primary objective of the author was on CNN -- sliding window approach for the short-term prediction. CNN, are a specialized kind of neural network for processing data that has a known, grid-like topology was time-series data is thought as 1D and image data which can be thought as a 2D grid of pixels [4]. The authors motive behind comparison between these three models to validate if there is any long-term dependency on given data for future prediction, whereas CNN focused on a given sequence instead of previous data. This would be a good approach to capture the hidden dynamics of the stock market, as these dynamics also keep on changing over time. Mean square error was calculated from the difference in actual price and predicted the price. Results from CNN model outshined the result of LSTM and RNN, possibly due to its ability to forecast on the basis of the current window, whereas RNN and LSTM require the previous indicators to predict future prices [4].

In the ANN model, if a number of hidden layers increased more from than two then it's quite difficult to have an insight of the model on backpropagation, that's why it's considered as the black box. ANN has multiple layers, such as an input layer, hidden layer and output layer which has neurons. The connection between these neurons of different layers is assigned with weights. Jonathan L.Ticknor in his study [5], assigned probabilistic nature to the network weights allowing the network to automatically penalize the complex model. One of the major drawbacks for the backpropagation algorithm is model overfitting, whereas to overcome this drawback related to noise fitting, the Bayesian regularized artificial neural network is considered to be the novel method for predicting the stock market trend [5]. The model combines the Levenberg–Marquardt algorithm with Bayesian regularization to forecast the stock price movement. The author tried to train the model by Microsoft and Goldman Sachs stock prices, where the error was calculated by the mean absolute percentage error. The results indicated that this model reduces the potential for overfitting, due to its probabilistic nature. As a future scope, the inclusion of other technical indicators would have improved the performance of a model.

David, Manfred, and Nijat in their study [6] discussed on a three-stage hybrid model. The first phase, multiple regression analysis was applied to define the financial and economic variables, the second phase, type-2 fuzzy clustering was implemented to create the prediction model, third phase, a type-2 neural network used to perform the reasoning for future stock prediction [6]. Considering the non-linearities and discontinuity of the stock market, the selection of input data of its manageable amount is considered as a necessary initial stage. The input dataset 'The Standard & Poor's 500' American stock market was used to train the model. This model was expected to perform well but at a later stage, the percentage error was not as global minima. Further refinement attempts were made by supplying the differential evolution optimization to the fuzzy neural network inference system and implementing in stock market prediction, it resulted in lower percentage error when used Fuzzy type-2 as compared to fuzzy type-1[6]

There were more recent works were conducted in [20], [21], [23] on LSTM, and were compared with machine learning models like Random Forest, Support Vector Machine and ARIMA but the LSTM has shown significant results.

III. PROPOSED PLAN

The main objective of this paper will be to predict the stock price on very next day if that goes up or down, and if feasible, for the long-term as well. There are various approaches which are popular for forecasting the time-series as mentioned in the above review. From frequentist perspective ARIMA shown a potential method, whereas from deep learning perspective LSTM and ANN can be considered a good approach to start with. I want to conduct my research on ARIMA & LSTM methods and one from linear models, in different phases. The first phase will consist of identifying the financial and economical parameter needed to be included in training the models. The second phase includes identifying the models that are going to be used for stock prediction. In this case, the plan is to go with one statistical model and one or more deep learning model and will also try to hybridize them to see if the results are affected. The third phase will consist of the methods for calculating the error and comparing among these methods. Possible error calculation methods, in this case, will be, mean square error, mean absolute percentage error, determination coefficient. The method with the least error would be considered as a novel method. Apart from the mentioned approach, I would also like to include news as a parameter, which can be fetched from any stock market news website, and to see if it has any effect on stock price movement. Post model training, I would like to implement the model at an online mode by deploying it in the cloud, that means the model keeps on training itself from the daily live data as it will allow the model to learn the new dynamics that keeps on changing over time.

IV. APPROACH

A. *Data understanding & pre-processing*

Stock price data of TCS company from Indian National Stock Exchange has been fetched through yahoo finance API. It had the details like Open Price, High Price, Low Price, Close Price, Volume. Here, we used the Close Price to perform our analysis, which is the end of the day price (at the time of market close) which represents the final movement of the stock price for that day. Interestingly, these Close Prices are adjusted closing price, which was adjusted upon the split of the stock. A stock split or stock divide is the process of incrementing the shares. The adjustment in price happens such that there is no change in market capital of the company [14]. It lowers the price of the shares and provides the opportunity to small investors for buying the shares. Now further exploring the data, I have plotted the line graph for the closing price for the past six years.

Fig 1: TCS close price movement

Here, we see the upward trend in stock price. Important to note that, there is sudden fluctuation in price from the start of 2018, so we might need to normalize the data for some of the algorithms, that will be discussed within algorithm sections. Further, the closing price of five working days in a week are available except any other public holiday, so I have treated them as a sequence of points for these prices. The data is divided into two, train and test set of 80:20. The model evaluation will be performed over the test set containing 20% of data. To understand other components in time series, it has been decomposed as below.

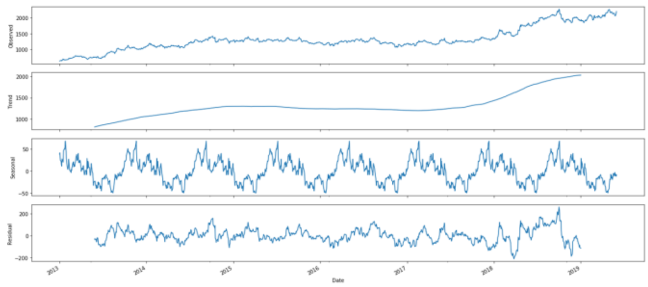


Fig 2: Time series decomposition

Here time series has been decomposed into three components:

a) Trend, which shows upwards movement.
b) Seasonality, which can be seen in this time series.
c) Residual, anything unexplained falls into residual.

B. *Infrastructure & software requirement*

While I was trying various algorithm, I preferred to use Kaggle kernel, which runs on the cloud and provide us GPU support and 16GB of RAM. It provides the python notebook, which runs over the browser and allows us to maintain the version control for our code. Data handling and preparation are conducted on python 3.6, with the help of packages like pandas and numpy. To fetch the data from Yahoo finance API, pandas_datareader package was used. Matplotlib & Seaborn were used to perform the visualization. Sci-kit learn was used to implement the time series and machine learning algorithms. Deep learning LSTM networks were developed over Keras on top of google tensor flow, which is one of the powerful libraries for a large-scale machine learning problem.

C. Algorithm 1: Autoregressive Moving Average (ARIMA)

ARIMA is a very popular algorithm in time-series forecasting. It is one of the classical and powerful algorithms for time-series analysis and good in short-term prediction. ARIMA is divided into three components, AR (autoregressive term), I (differencing term), MA (moving average term). To further understand these components [15]:

a) AR: term signifies the past value used in predicting the new value. It is also known as lag order, denoted by ‘p’ in ARIMA and its value is identified using PACF plot.
b) I: It refers to differencing of the raw observations by subtracting it with the previous time step to make the time series stationary. It is denoted by ‘d’. Test like KPSS and ADF can be used to identify if time-series is stationary or not.
c) MA: term signifies the dependency between observations and residual errors of moving average model applied to lagged observations [15]. It is also known as the order of moving average, denoted by ‘q’, and its value is identified using the ACF plot.

Looking at the data, the challenge was to make the time series stationary first. Differencing is used to remove the level of time-series by removing seasonality and trend and stabilizing the time series. It is done by subtracting the time series with the lagged time series.

$$y'_t = y_t - y_{t-1}$$

Fig 3: First order differencing
Source: [16]

Sometimes the time series doesn’t stabilize after first differencing, then we require to perform the second differencing which is also known as second-order differencing and below is its mathematical interpretation [16].

$$\begin{aligned} y^*_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \end{aligned}$$

Fig 4: Second order differencing
Source: [16]

I have performed the second-order differencing to stabilize the time series. To further identify the order of moving average, I have plotted ACF plot as below, where it was concluded as 2 spikes were out of threshold limit.

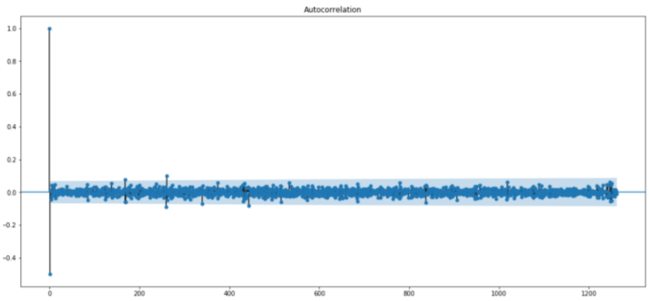


Fig 5: ACF plot

Similarly, the PACF was also plotted to identify the order of AR. Now, the target was to predict the price for very next as the long-term prediction may lose the volatility of the market for this algorithm. Here, the model was recursively trained on a daily basis to update its parameter and predict future one day price. Usually, from the investor's point of view, the prediction for the stock price for very next day is not that useful. To overcome the scenario, we have some of the advanced algorithms which were discussed in later sections.

D. Algorithm 2: Long-Short Term Memory (LSTM)

LSTM is one of the most advanced deep learning architecture & considered as state-of-the-art techniques in the sequence learning task. Although this algorithm is popular in image processing problems, it has also gained researchers attention for the time series forecasting problem and had shown a significant result. It is the special case of Recursive Neural Network (RNN) and will try to understand the domination of LSTM in some of the sequence learning tasks. To understand more on the concept of LSTM, consider an example of reading an essay, we see each word is the comprehension of past words. We don’t ignore everything and start from inception. In contrast, traditional neural networks have this drawback, where it starts processing from scratch. RNN has the ability to overcome this flaw by persisting the information from the past. It can be represented as a neural network with loop. But, RNN also had some drawback when information required to pertain for a longer period [12].

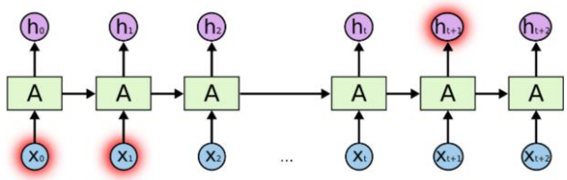
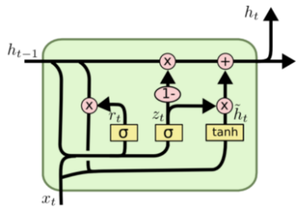


Fig 6: RNN flaw to pertain the information as the gap grows
Source : [12]

LSTM has the capability to learn long-term dependencies. They have the default tendency to learn the information for a longer period of time. Similar to RNN, it also has chain-like structure but instead of single neural network in repeating layer, it has four nodes (also known as single tanh layer) which interacts with each other in a unique way.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Fig 7: LSTM Neuron Architecture
Source: [10]

The neuron in LSTM uses three gates, input gate, output gate and forget gate, each having different functionality.

In the process of implementing the LSTM in stock price prediction, there are few parameters that need to be discussed, as this algorithm works differently as compared to the other discussed algorithms. We are looking to predict the price for one week (5 working days) in the future by looking back 30 days closing price from the past. The test dataset will be composed of 40 periods, in which every period will have a series of 5 days prediction. It looks like a sliding window of 5 days. Below is the example of predicting one day in the future by looking 3 days of data in the past.

	Open	High	Low	Close	Volume
0	0.6277	0.6362	0.6201	0.6201	2575579
1	0.6201	0.6201	0.6122	0.6201	1764749
2	0.6201	0.6201	0.6037	0.6122	2194010
3	0.6122	0.6122	0.5798	0.5957	3255244
4	0.5957	0.5957	0.5716	0.5957	3696430
5	0.5957	0.6037	0.5878	0.5957	2778285
6	0.5957	0.6037	0.5957	0.5957	2337096

Fig 8: Sliding window approach
Source: [13]

This approach should be good to understand the changing dynamics of stock price movement, which we will discuss in the evaluation section. Above mentioned type of series data is used to train the model and performed the testing on 40 periods of test data. During the compilation of the model, the mean squared error was used for the evaluation of loss function. For optimizing function, stochastic gradient descent has a single learning rate for all the weight updates, whereas 'ADAM' optimizer which maintains the learning rate for each network parameter separately and adapted as learning unfolds. As the data was highly volatile and non-stationary, I preferred to use ADAM. Fifty epochs were used to converge the algorithm to minimize loss. Usually, Neural Network performs better on normalized and same was the case with LSTM. I have used min-max normalization to bring the price on smaller ranges.

E. *Algorithm 3: Least Absolute Shrinkage and Selection Operator (LASSO)*

Linear models are usually ignored. However, they have shown significant results as compared to non-linear models [17]. Consequently, it is important to understand linear models generalization and its extensibility. But the challenge was how to implement the linear model on the time-series data set. The time-series data can be converted into supervised learning data, similar to the approach that was discussed in LSTM. From the given time-series dataset, we can reframe the data to appear like a supervised learning problem. We can achieve this by taking previous time steps as an input variable and output variable as next time step [18].

LASSO regression is also a type of linear regression which uses shrinkage, where data points contracts towards central point like mean. Usually, this type of regressions is quite useful in data with high multicollinearity. As the learning data is converted from the sliding window approach, it shows multi-collinearity among the feature space. It uses L1 regularization, where penalty on model is equivalent to the absolute value of coefficients magnitude. These type of regularization results in a sparse model with lesser coefficients [19]. Larger penalties will result in parameter values closer to zero or become zero and can be eliminated from the model, which will help in producing a simpler model. Whereas in L2 regularization which is used in Ridge regression doesn't result in the elimination of the parameter. In this way, L1 regularization provides the variable selection ability in LASSO regression.

The regression equation for LASSO regression is quadratic in nature and the aim is to minimize the below equation [19]-

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Fig 9: LASSO Optimization function
Source: [19]

Here, in the second half of the equation, we can see that the absolute value of the parameter is used for regularization. Some of the values of β's will converge to zero which makes the model easier to interpret. The tuning parameter (λ) which regulates the magnitude of the penalty on parameters or the shrinkage. Below are the observations on the effect of λ value on the model.

a) If λ=0, then no parameters are eliminated.

b) If λ tends towards ∞, then almost all coefficients are eliminated.

c) When λ increases, bias also increases.

d) When λ decreases, variance also increases.

As I was more interested in viewing the trend for the week, I performed the weekly sampling of the data and then performed the train and test split (80:20). In the implementation of the algorithm, I used GridSearchCV to find the best parameter for the model. It allowed us to perform 5-fold cross-validation in a range of different regularization parameter to obtain the optimal value of alpha.

V. EVALUATION

In this section, the results obtained from the above three algorithms implemented on time series data are compared. There are various popular statistical measures such as 'Mean Absolute Percentage Error' (MAPE), 'Root Mean Square Error' (RMSE), etc to evaluate the forecasting model. But I have preferred MAPE over others as it is not depended on the scale of the data and gives the error results in percentage term which was effective in comparing the results when different scaling for same data was done while applying the algorithms. The mathematical formulae for calculating the MAPE is given as below-

$$\mathrm{M} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

Fig 10: MAPE
Source: [22]

Where A_t is the actual value, F_t is the forecasted value and 'n' is the number of observations.

Starting with the ARIMA evaluation, which is one of the classical and powerful algorithms for time-series analysis and was expected to show the good results in short-term prediction. In this algorithm, instead of predicting for all the test dataset in a single go, I tried to predict only step ahead and then recursively training and predicting the data from that point. It helped in achieving good accuracy as compared to the prediction of test data in a single go. Below are the training and prediction plot for ARIMA model.

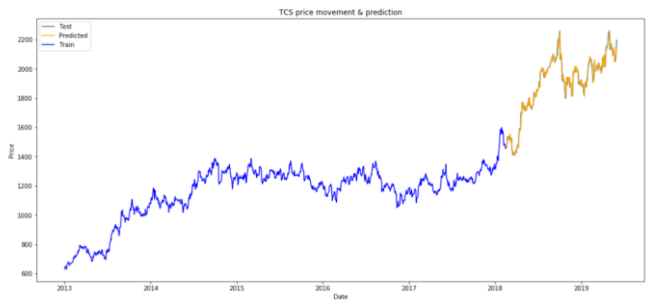


Fig 11: ARIMA's training, test & prediction plot

Here, the blue line represents actual training data, the green line is actual test data which is almost covered by an orange line which is predicted data. From the above-shown plot, it can be seen that prediction and actual test results are almost overlapping or at least able to predict the trend movement of the stock price.

On further evaluation of the LASSO algorithm, which was simple to execute and have no constraint while applying it, has shown a significant improvement over the ARIMA model. Here, the weekly data sampling was done, and the past ten data points were looked to predict the next week trend. Below is the plot for actual test data represented in blue and predicted results for test points, represented in orange. The predicted line is almost following the test line. Since it is looking a few data points in the past and predicting the result, this method can be considered to be a good approach as it is quick in execution and not much of difference in test and predicted results.

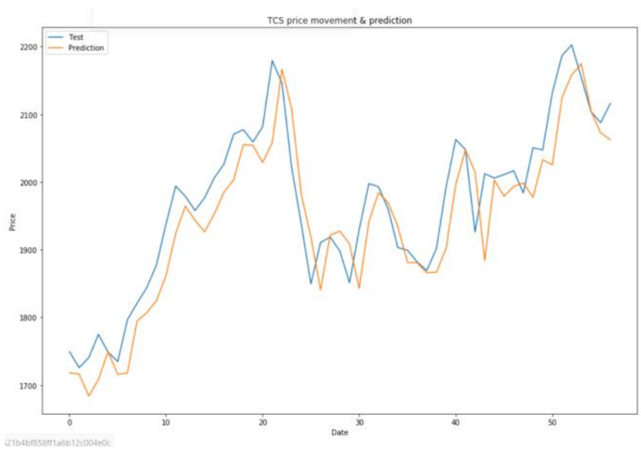


Fig 12: LASSO's test & prediction plot

On evaluating the LSTM algorithm, which was considered as one of the best algorithms for stock forecasting tasks, has shown further improvement over the other two algorithms. Below is the graph between loss validation loss, which can be seen that it almost converged in 5o epochs.

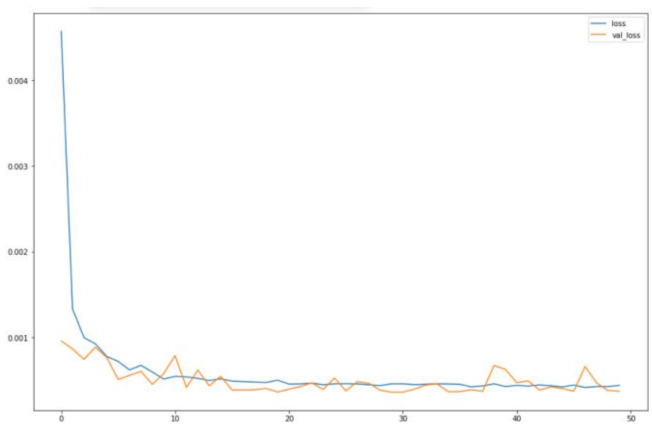


Fig 13: Loss vs Validation loss plot

Further plotting the training and prediction data as below.

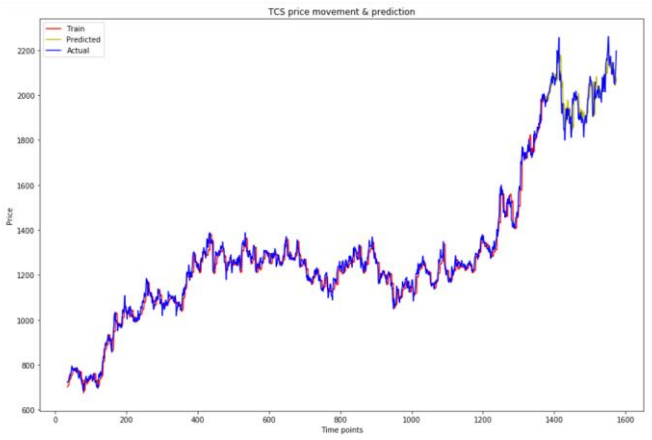


Fig 14: LSTM's training, test & prediction plot

The predicted results are very close to the actual test results. Here, the data was not sampled but the consecutive five working days (considered to be a week) price is predicted in future. It looked price of past 30 days to predict the 5 days future price. On looking at the test set more closely, the discrete graph of five days interval for 40 periods is shown below.

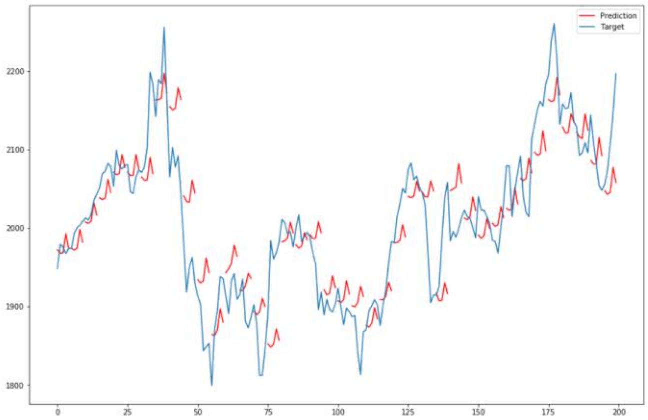


Fig 15: LSTM test & discrete interval plot

Here each red lines represent 5 day's prediction on the basis of 30 days in the past. Also, there are 40 red lines, representing 40 periods for the test sample.

Below table represents the Mean Absolute Percentage Error which is evaluated on all the mentioned algorithms.

Algorithm	Mean Absolute Percentage Error
ARIMA	12.18%
LASSO	2.16%
LSTM	1.97%

From the above table, it can be said the LSTM has outperformed the other two, ARIMA & LASSO, in weeks trend prediction. From the previous works as well, the LSTM was always highlighted as the best-preferred method. In this case, LASSO was close enough to LSTM but the LSTM requires much more time for training and validating the model, whereas the LASSO was simpler and was very quick in execution. So, the LSTM can be considered as the best algorithm when it compared to the other two for this dataset but in the time and computational cost constraint environment, LASSO has also shown the good results.

VI. Conclusion

There are many methods that are developed in recent time for the time series prediction. In this paper, the comparison among one statistical model, and two supervised learning including one deep learning model was conducted and the LSTM has shown the best results but in the time-critical environment, LASSO has shown good results. Further from the future scope perspective, if we can include other factors like stock market news, and companies portfolio information to predict the stock price, then it might show improvement in prediction results.

Stock market prediction is a quite demanding topic in the finance world and many of the researchers are looking for the model which can automatically learn the latent dynamics of the complex stock market. This research will encourage the individual and the investors to make a sturdy decision regarding their investment plan and future endeavors.

VII. References

[1] "Time Series," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Time_series.

[2] A. O. A. C. K. A. Ayodele A. Adebiyi, "Stock Price Prediction Using the ARIMA Model," in *UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014.

[3] C. K. Thomas Fischer, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 664-659, 2018.

[4] V. R. G. E. V. K. M. S. K. Sreelekshmy Selvin, "Stock market prediction using LSTM, RNN and CNN-sliding window model," September 2017. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8126078.

[5] J. L.Ticknor, "Science direct- A Bayesian regularized artificial neural network for stock market forecasting," April 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417413002509.

[6] M. G. N. M. David Enke, "Stock Market Prediction with Multiple Regression, Fuzzy Type-2 Clustering and Neural Networks," *Procedia Computer Science*, vol. 6, pp. 201-206, 2011.

[7] D. E. Xiao Zhong, "Forecasting daily stock market return using dimensionality reduction," *Expert Systems & Applications*, vol. 67, pp. 126-139, Jan 2017.

[8] H.-m. Z. Z. Y. L. Y. Feng Zhou, "EMD2FNN: A strategy combining empirical mode decomposition and factorization machine-based neural network for stock market trend prediction," *Expert Systems with Applications*, vol. 115, pp. 136-151, Jan 2019.

[9] S. R. K. S. Rohan Pimprikar, "Use of machine learning algorithms and twitter sentiment analysis for stock market prediction," *International Journal of Pure and Applied Mathematics*, vol. 115, pp. 521-526, 2017.

[10] A. Xavier, "Predicting stock prices with LSTM," Jan 2019. [Online]. Available: https://medium.com/neuronio/predicting-stock-prices-with-lstm-349f5a0974d4.

[11] "GeeksforGeeks," [Online]. Available: https://www.geeksforgeeks.org/univariate-bivariate-and-multivariate-data-and-its-analysis/.

[12] "Understanding LSTM Networks," 27 Aug 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[13] A. Nayak, "Predicting Stock Price with LSTM," Mar 2018. [Online].

Available: https://towardsdatascience.com/predicting-stock-price-with-lstm-13af86a74944.

[14] "Wikipedia - Stock Split," [Online]. Available: https://en.wikipedia.org/wiki/Stock_split.

[15] J. Brownlee, "Machine Learning - ARIMA," 09 Jan 2017. [Online]. Available: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/.

[16] [Online]. Available: https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average.

[17] M. Peixeiro, Jan 2019. [Online]. Available: https://towardsdatascience.com/intro-to-linear-model-selection-and-regularization-d47bd2c5d54.

[18] J. Brownlee, "Time Series Forecasting as Supervised Learning," 05 Dec 2016. [Online]. Available: https://machinelearningmastery.com/time-series-forecasting-supervised-learning/.

[19] Stephanie, "Statistics How To," 24 Sep 2015. [Online]. Available: https://www.statisticshowto.datasciencecentral.com/lasso-regression/.

[20] X. Z. Y. W. P. e. a. J. S. Pang, "An innovative neural network approach for stock market prediction," 2018. [Online]. Available: https://link.springer.com/article/10.1007/s11227-017-2228-y.

[21] M. R. F. K. Asghar, "Development of stock market trend prediction system using multiple regression," 2019. [Online]. Available: https://doi.org/10.1007/s10588-019-09292-7.

[22] "MAPE," [Online]. Available: https://en.wikipedia.org/wiki/Mean_absolute_percentage_error.

[23] HiranshaM, Gopalakrishnan.E.A, V. K. Menon and SomanK.P, "NSE Stock Market Prediction Using Deep-Learning Models," *Procedia Computer Science,* vol. 132, pp. 1351-1362, 2018.