**ABSTRACT**

In this project, the basic aim is to get the available stock market dataset and perform predictive models on it to check which model performs best and gives accurate results. The models used for this comparison is AR model, MA model,ARIMA model and LSTM model. ARIMA is the combination of AR and MA models, so both models were also executed on the same data while performing ARIMA model. To see the performance of the models Mean Squared Error was used in order to get the average squared difference between the estimated values and the actual value. At the end, all the models were compared with each other and the best model for predicting the stock prices for a particular company was identified. It was our hypothesis that the ARIMA model would work best suitable for non-seasonal short-term data prediction and LSTM model was suitable otherwise. But from the experimentation with our dataset, the LSTM model works the best when both models were used on the same data.

**KEYWORDS**

Stock Market Prediction, Prediction Model, ARIMA Model, AR Model, MA Model, LSTM Model, MSE, Regressive Model, Statistical Model, ACF, PACF, tensorflow, seasonal differentiation, log transformation

**1. INTRODUCTION**

Stock market is the most influential part in the financial sector of any nation. An aggregation of buyers and sellers with a representation of ownership claim on a specific business, as preferred, is how the market runs. Investments in the stock market can be done is various ways, some of which include electronic trading platforms and brokerages. Rates of participation and the value of the holdings significantly differ across strata of income. On a study conducted in 2003, it has been told that participation rates have been shown to strongly correlate with educational levels, promoting the hypothesis that information and transaction costs of market participation are better absorbed by more educated households. As the raw data is not possible to be read straight away, the investors depend largely on the predictions and interpretation of the current stock market state and data. To analyse the stock market data and predict the future stock prices, many factors come into the picture. They include[3]:

* Demand and Supply: When demand for shares exceeds supply, which means the buyers are more than sellers, the prices increase. When demand is less than supply, meaning that buyers are less than sellers, the prices decrease.
* Investors: Market players have an impact on share prices. With more bulls than bears, the prices increase. With more bears than bulls, share prices decline. This also includes, Long-term and Short-term investors.
* Political Factors: Political factors that range from relations with other nations to government policies can affect share prices.
* Previous Stock Market Prices: The current stock price is also dependant on the previous trend in the data.

The stock market price is not solely dependant on the previous values and trend, but it is the combination of the above mentioned factors. For our project, the main goal is not to predict the future stock market trends and stock prices but it is to learn the different prediction models and find the best models based on the prediction error and the accuracy of the models.

Therefore, we have assumed that the future values of the stocks are dependant on the current and previous values and also the trend of the stock prices over a period of time.

The models we used for the prediction of the data are:

**ARIMA:**

ARIMA or Auto Regressive Integrated Moving Average model captures a suite of different standard temporal structures in time series data. It is a statistical model and works on time series data. It is the combination of two different models, AR Model, i.e., Auto Regressive Models and MA Model, i.e., Moving Average Window Model.

**LSTM:**

LSTM or Long Short Term Memory model is based in the Recurrent Neural Network architecture. But the difference between the standard feed forward neural network and the LSTM is that LSTM has feedback connections. These connections are known as LSTM connections and these are suitable for many applications such as [classifying](https://en.wikipedia.org/wiki/Classification_in_machine_learning), [processing](https://en.wikipedia.org/wiki/Computer_data_processing) and [making predictions](https://en.wikipedia.org/wiki/Predict) based on [time series](https://en.wikipedia.org/wiki/Time_series) data.

Both the methods are described in details in the following sections.

**2. RELATED WORK**

**2.1 Using Autoregressive Modelling and Machine Learning for Stock Market Prediction and Trading[1]**

The paper describes and compares different types of predictive models on the stock market data to analyze it. The following models were discussed: Multiple Linear Regression (MLR), Autoregressive Integrated Moving Average (ARIMA), Multivariate Vector Autoregressive (VAR) models, Long Short-Term Memory (LSTM), Nonlinear Autoregressive exogenous (NARX).

It was concluded that, for short term, NARX was the best model and VARM model predicts the trend accurately. But for the first year, LSTM and ARIMA were the most accurate models.

**2.2 High-Frequency Stock Trend Forecast Using LSTM Model[2]**

The paper discusses the use of LSTM(Long Short-Term Memory) network which is a variant of RNN(Recurrent Neural Network) to predict price movement of a short-term stock data. RNNs have a short-term memory capability and this makes it difficult to capture long-term dependencies during long time transmission of information. To improve upon this limitation, LSTM introduces a memory cell and gate structure which effectively associates memories and input remote in time. The LSTM uses linear units called Constant Error Carousels(CECs) to overcome gradient vanishing which is another improvement over previous RNNs.

**2.3 Stock Price Prediction Using the ARIMA Model [9]**

The paper presents extensive process of building the ARIMA model for the time series prediction of stock prices. The ARIMA model proves to be very effective in the short term prediction. In ARIMA model, the future value of a variable is a linear combination of past values and past errors and it is based on ARMA(Autoregressive and Moving Average (ARMA) model except that it converts a non-stationary data to a stationary data(the mean, variance and autocorrelation structure do not change over time) before working on it.

**2.4 New Approach of Moving Average Method[6]**

The paper discusses different methods introduces a new method for calculation of Moving Average. The different methods include Simple Moving Average, Exponential Moving Average, Weighted Moving Average, Weighted Exponential Moving Average, Mean Square Error, Mean Absolute Percentage Error.

**3. PROPOSED METHODS**

**3.1 LSTM**

LSTM stands for Long Short-Term Memory model which is based in the Recurrent Neural Network architecture. But the difference between the standard feed forward neural network and the LSTM is that LSTM has feedback connections.

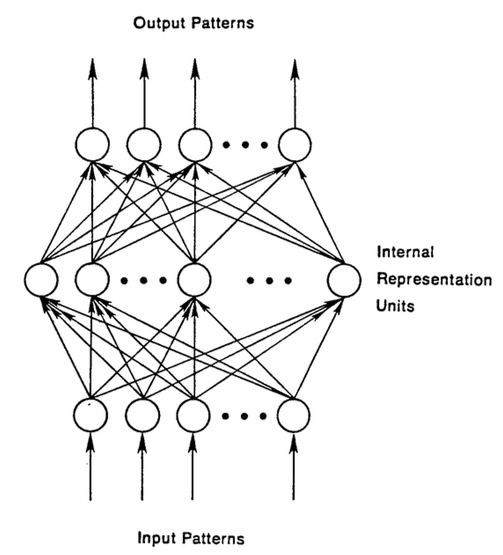


Fig. 1: Feed Forward Network

LSTMs help preserve the error that can be backpropagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps (over 1000), thereby opening a channel to link causes and effects remotely.

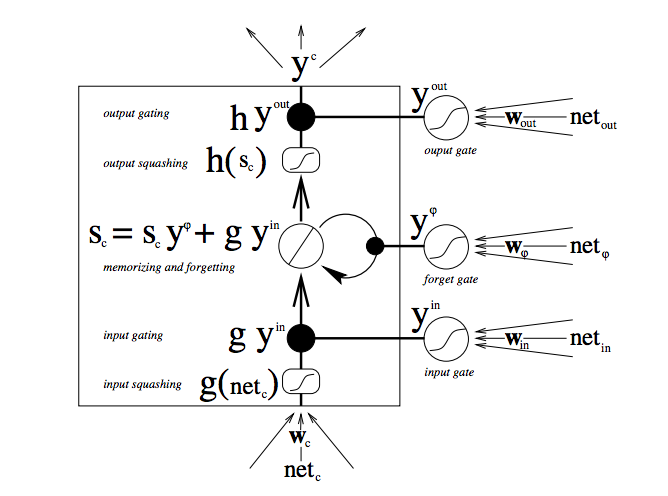


Fig 2. Working of LSTM

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”

An RNN using LSTM units can be trained in a supervised fashion, on a set of training sequences, using an optimization algorithm, like [gradient descent](https://en.wikipedia.org/wiki/Gradient_descent), combined with [backpropagation through time](https://en.wikipedia.org/wiki/Backpropagation_through_time) to compute the gradients needed during the optimization process, in order to change each weight of the LSTM network in proportion to the derivative of the error (at the output layer of the LSTM network) with respect to the corresponding weight.

**3.2 ARIMA**

ARIMA, short for ‘Auto Regressive Integrated Moving Average’ is actually a class of models that ‘explains’ a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values.

Any ‘non-seasonal’ time series that exhibits patterns and is not a random white noise can be modeled with ARIMA models.

An ARIMA model is characterized by 3 terms: p, d, q

where,

p is the order of the AR term

q is the order of the MA term

d is the number of differencing required to make the time series stationary.

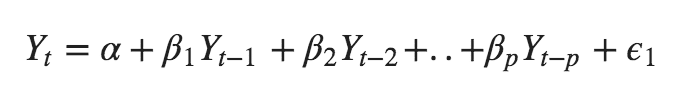
If a time series, has seasonal patterns, then you need to add seasonal terms and it becomes SARIMA, short for ‘Seasonal ARIMA’.

As discussed above that the ARIMA model is the combination of two models, namely, Auto-Regressive Models and Moving Averages Model.

**3.2.1 Auto-Regressive Model**

A pure Auto Regressive (AR only) model is one where the current value depends only upon its own lags and not on the forecast errors.

The following equation represents the AR model:

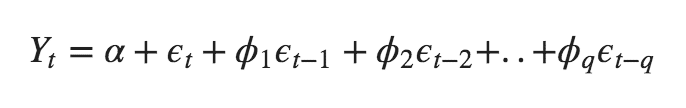


Here, Yt depends only on its own lags. That is, Yt is a function of the ‘lags of Yt’.

The value of p in AR model can be estimated by plotting the PACF graph.

**3.2.1 Moving Averages Model**

A pure Moving Average (MA only) model is one where the current value depends only upon the lagged forecast errors and not on the own lags.

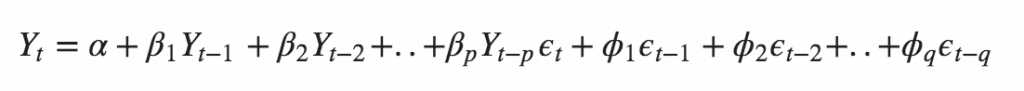
Here, Yt depends only on the lagged forecast errors, i.e., Yt is a function of the ‘lagged forecast errors’.

The value of q in MA model can be estimated by plotting the ACF graph.

**3.2.1 ARIMA Model**

The ARIMA, i.e., Auto-Regressive Integrated Moving Average Model, is the combination of above two models.

The AR term represents the lag of the current values, the MA term represents the lagged errors. The ARIMA model requires the data to be stationary. So the last term, i.e., I stands for Integrated. It is responsible for making the data stationary. It does it by performing differencing. Differencing in statistics is a transformation applied to time-series data in order to make it stationary.

****

**4 EXPERIMENTAL EVALUATION**

**4.1 Dataset Used**

The dataset used was acquired from Kaggle Datasets[12]. The dataset contains 6 attributes, namely, date, open, high, low, close, volume, name. Among these attributes only two attributes are considered, i.e., Date and the Closing price of the dataset. the data was split between two sets, i.e., training and testing datasets in the ratio 4:1. The final dataset looks like this:

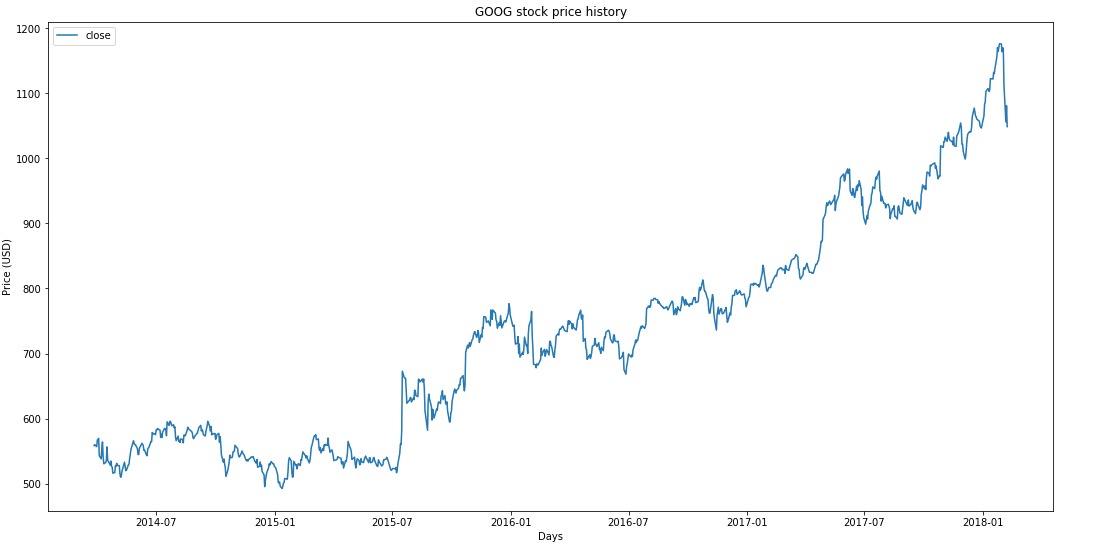
****

Fig 3: Data Plot

**4.2 LSTM Implementation**

In the first part of the project, we have implemented the LSTM model and performed the prediction for 10 days in the future.

Step 1:

* First the non-required data attributes were eliminated in order to work with only the date and the closing price of the stock.
* The data was scaled down to lie between 0 and 1. Both the test and train data was normalized with respect to training data because we are not supposed to have access to test data.
* The above scaled data was trained and smooth data. The LSTM smoothing window size was taken as 10. The last bit of the remaining data was normalized again.
* To avoid the ragged data, we performed the exponential moving average smoothing. This will give us a smoother curve rather than the original ragged data.

Step 2:

* The first step to define the hyperparameter: D: It stands for dimensionality. We took the previous stock price as the input and predict the next one, which should be 1.

num\_unrollings: related to the backpropagation through time (BPTT) that is used to optimize the LSTM model. This denotes how many continuous time steps you consider for a single optimization step. It was selected as 50.

batch\_size: It is how many data samples you consider in a single time step. It was selected as 500.

num\_nodes: It represents the number of hidden neurons in each cell. It was selected as: [200,200,150].

n\_layers: It denotes the number of layers in the number of layers. It is initialized as the size of the num\_nodes.

dropout: Denotes the number of units excluded from the network. It is initialized as 0.2.

* After initializing the hyperparameters, the cell state and the hidden state variables were created in order to maintain the state of the LSTM.
* Several tensor transformations were performed in order to get the output to be of a specific format.
* Then the LSTM model was executed and the results were stored in order to be analyzed.

Step 3:

* This step involves the loss calculation and optimization of the model.
* We calculated the Mean Square Errors for each batch of predictions and true outputs. Then the sum of all the mean squared losses was taken.
* Then the optimizer chosen in order to optimize the neural network. The chosen optimizer chosen is AdamOptimizer as it is a well performing optimizer.

Step 4:

* To perform the prediction, we first create a placeholder for feeding the input and state variables in order to maintain the state of the network.
* Finally, the values are predicted using tf.nn.dynamic\_rnn function. Number of epochs used are 50.

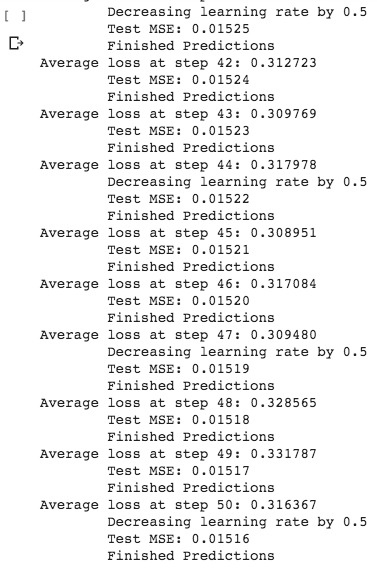


Fig 4: Epochs

* The visualizations for the prediction looks as following:

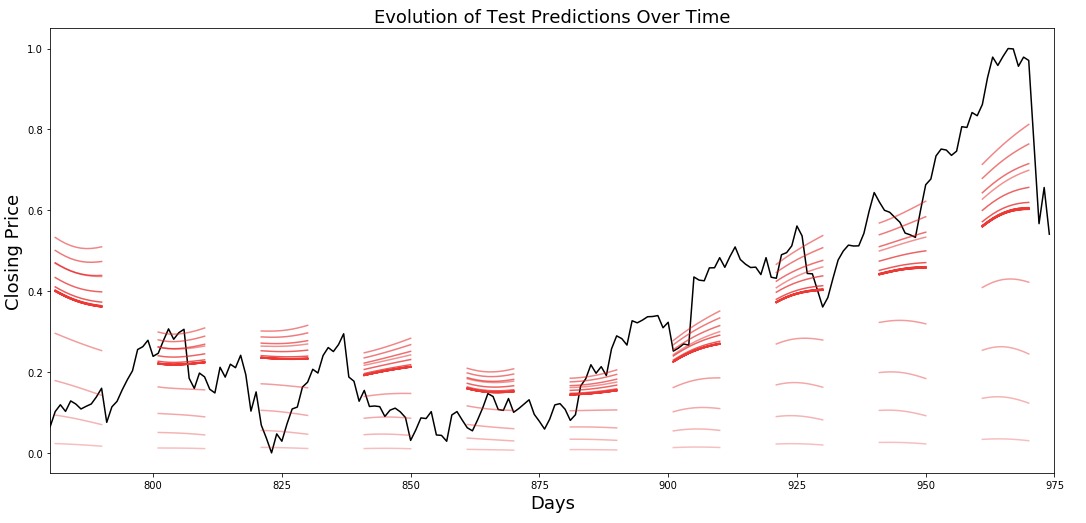


Fig 5: Predictions

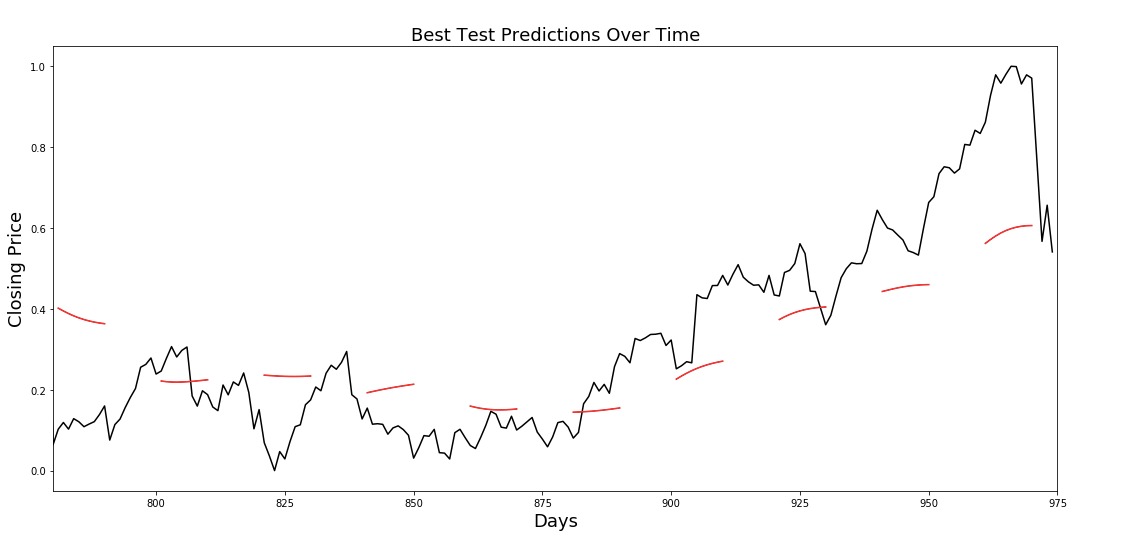


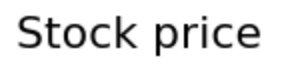
Fig 6: Best Test Predictions

**4.3 ARIMA Implementation**

In this part of the project, we implemented the ARIMA model and did the prediction for one year(taking monthly data- means we did the prediction for 12 points) We used the Google stock dataset for our model. The stepwise implementation is as follows:

Step 1:

Loading and and plotting the source data.(Check if the data has any seasonal patterns, cyclic patterns, general trends and dealing with missing values: ARIMA models don’t work on data that have NAs)





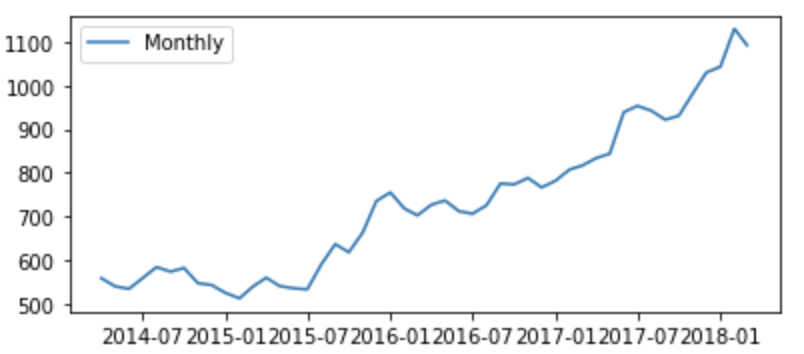


Fig 7: Daily and Monthly Stock Price

Note: We will proceed with the analysis by resampling the data into monthly domain so as to easily visualise the results. Predicting for 1 year (12 points in monthly domain) than 365 points in daily data domain is much feasible for ARIMA and easy to visualise given our dataset for 5 years.

Checking for Trend and stationarity:

* Rolling mean and deviation for trend check

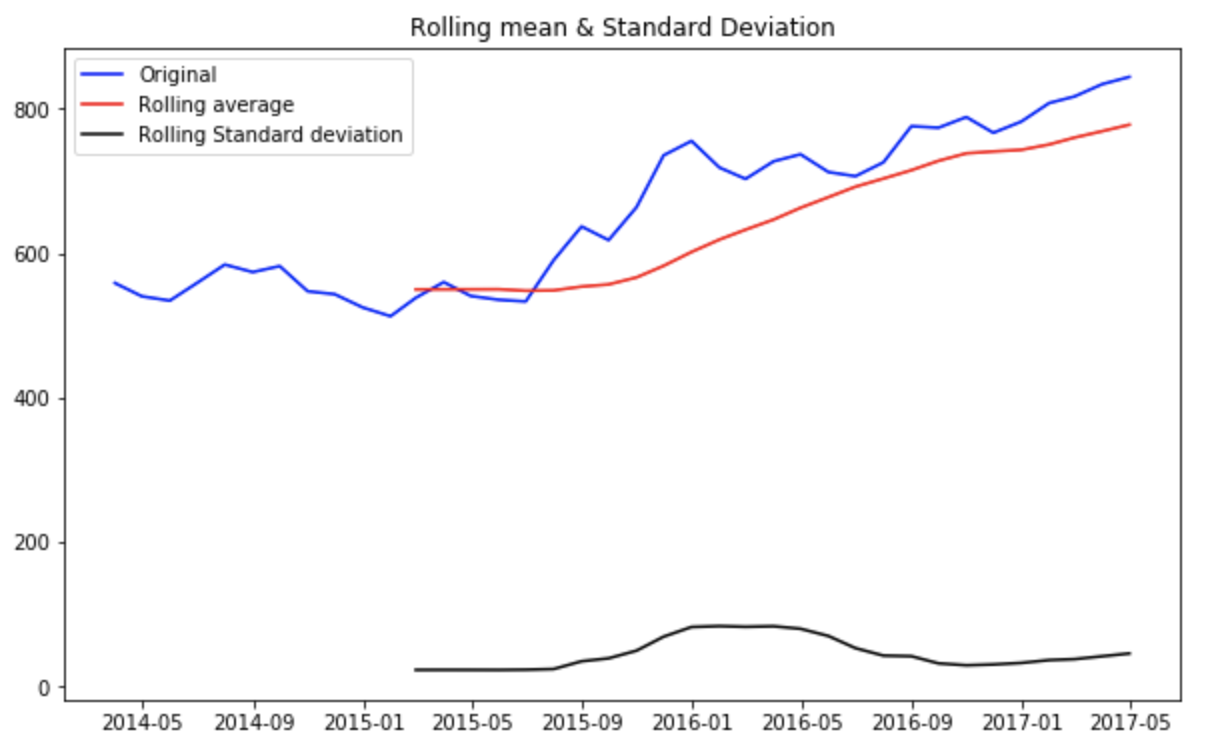


Fig 8: Rolling Mean and Standard Deviation

* Apply the Augmented Dickey Fuller Test (to confirm the stationarity of data)

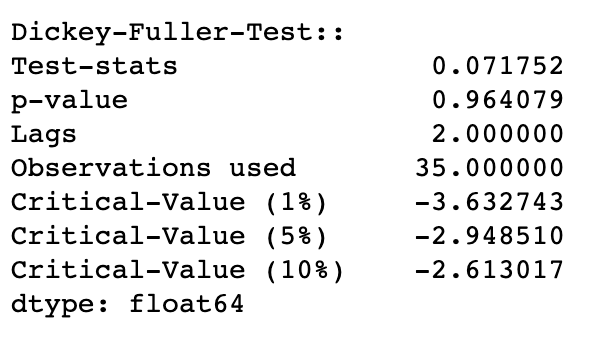


Fig 9: Results of Dickey-Fuller-Test

If the data is stationary, proceed with ARIMA.

If the data is not stationary, data needs to be differenced to make it stationary: ‘I’ component of ARIMA does this.

We can see the upward trend in our data, so we normalized the data on log scale and did the one lag differencing on the data to make the data stationary.

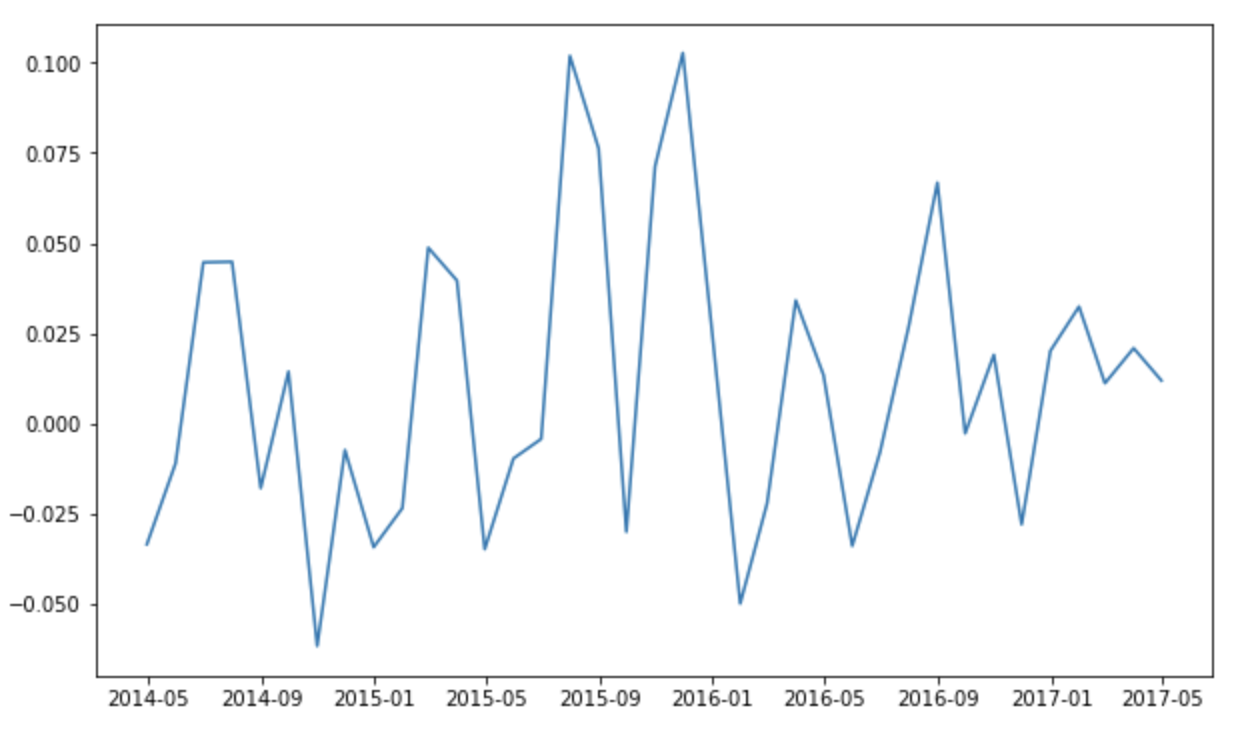


Fig 10: Log Normalized and differenced data

Augmented dickey fuller test also confirms stationarity (very small p-value):

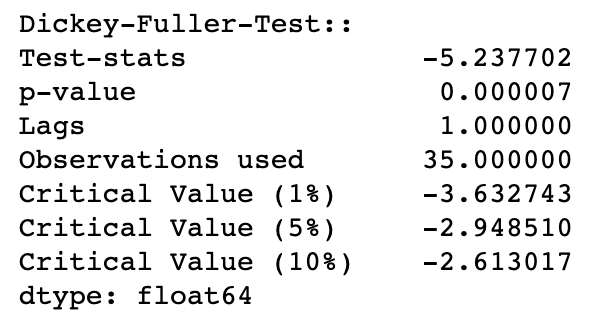


Fig 11: Results of Dickey-Fuller-Test

Step 3: Run ETS Decomposition on data (To check the seasonality in data)

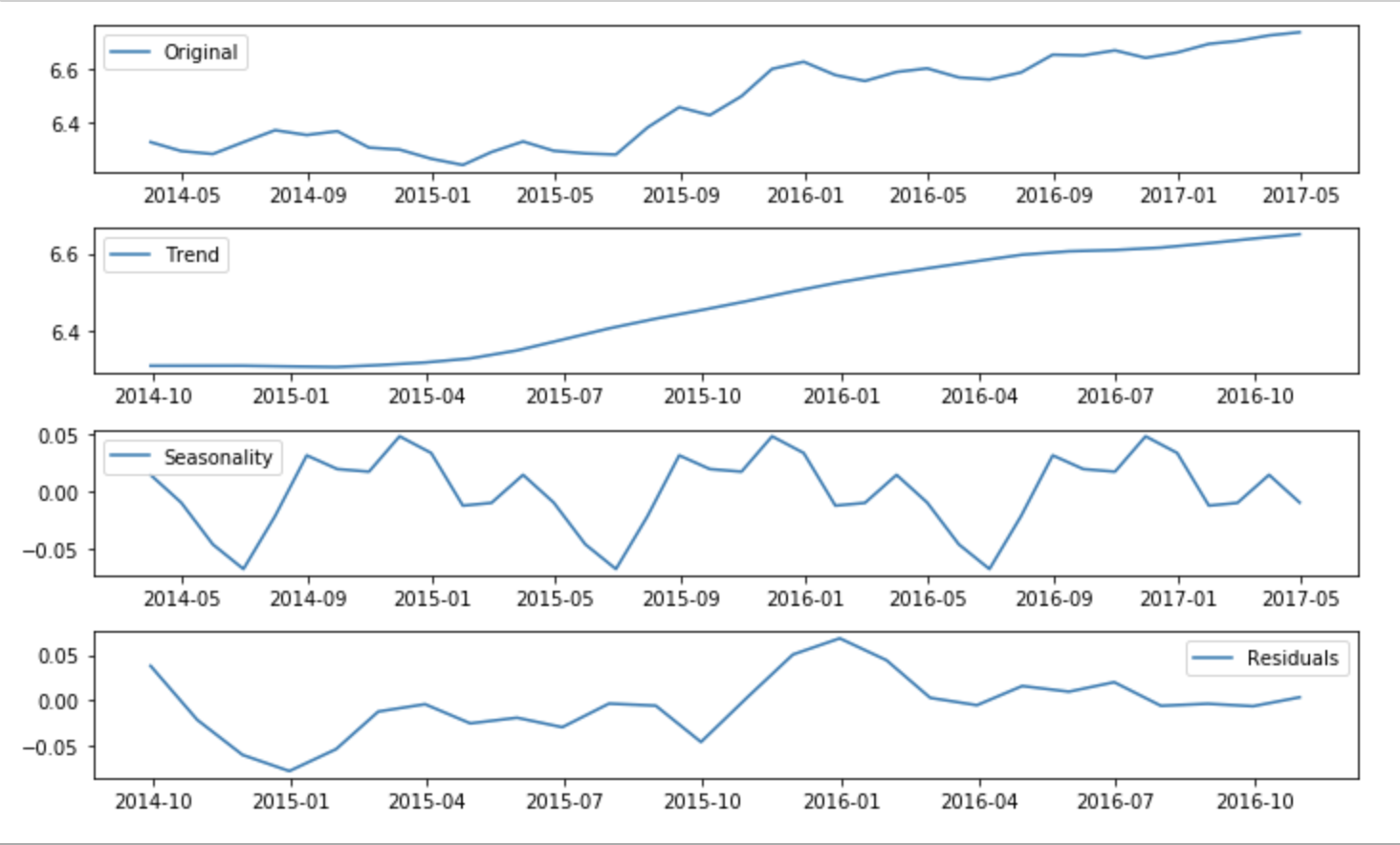


Fig 12: Seasonal Decomposition

If the seasonality is minute compared to the trend component, we need not consider it. In this case, we can proceed with the non-seasonal ARIMA model.

* ACF - can be used to determine the MA order (q).
* PACF - can be used to determine the AR order (p).
* Likewise, if the seasonality is significant in the data, proceed with Seasonal ARIMA or SARIMA.

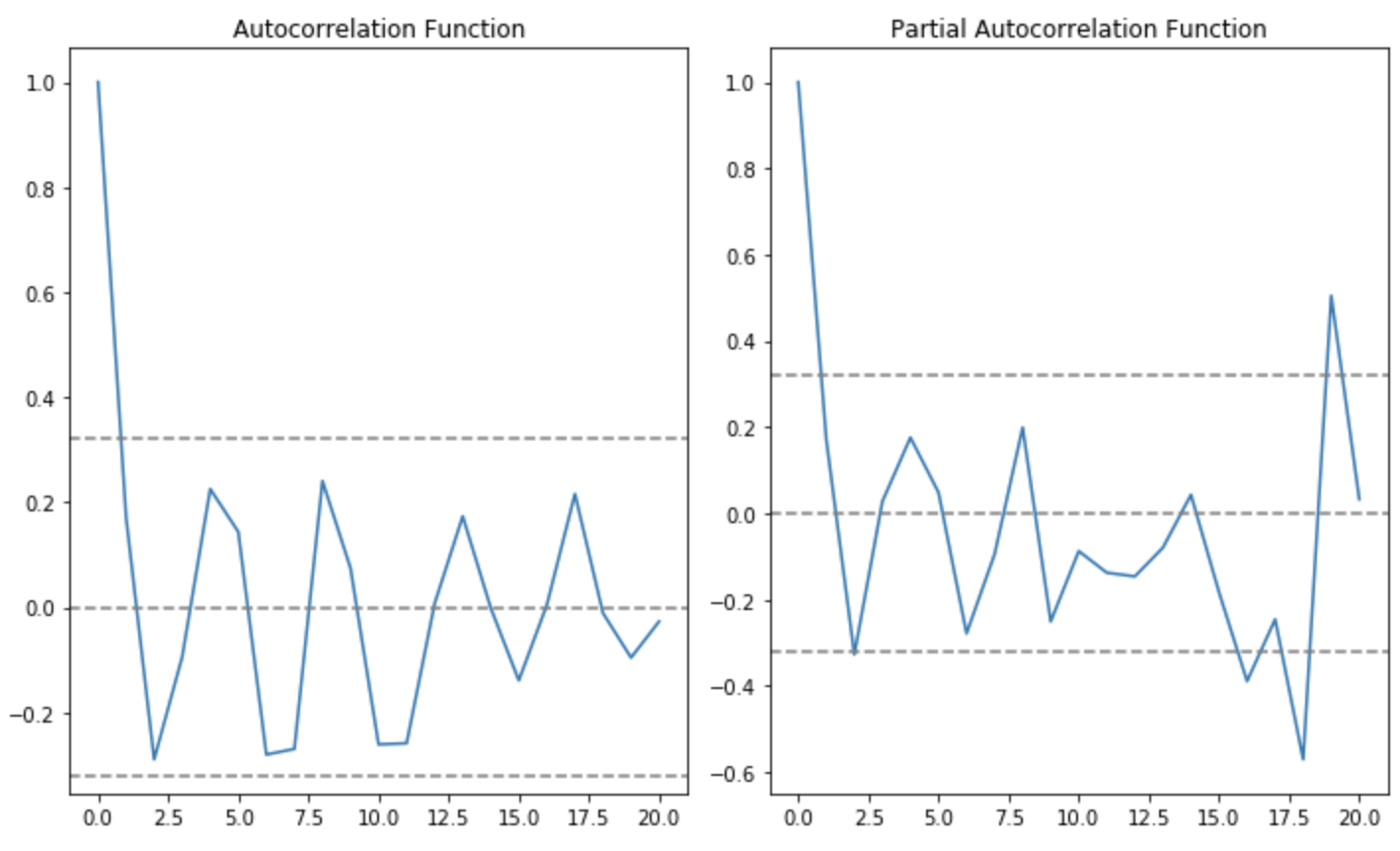


Fig 13: ACF and PACF

We can estimate values of p,q to be less than 2 each. We calculated the mean square error for both values and found that (p=2,q=2) fitted best.

Model fitting and prediction:

We imported the ARIMA model from statsmodels library and implemented the AR model, MA model and ARIMA model using the library and found that ARIMA was most accurate among them.

AR model:

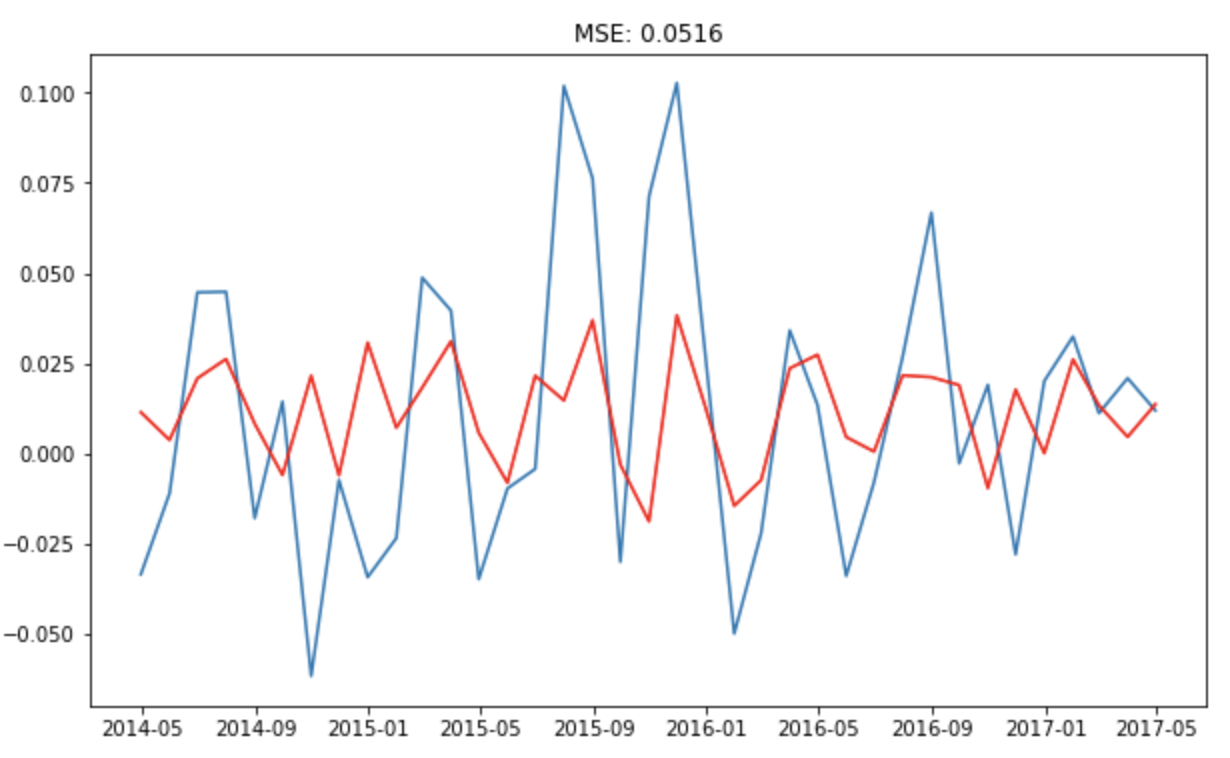


Fig 14: AR model prediction(over normalized data)

MA model:

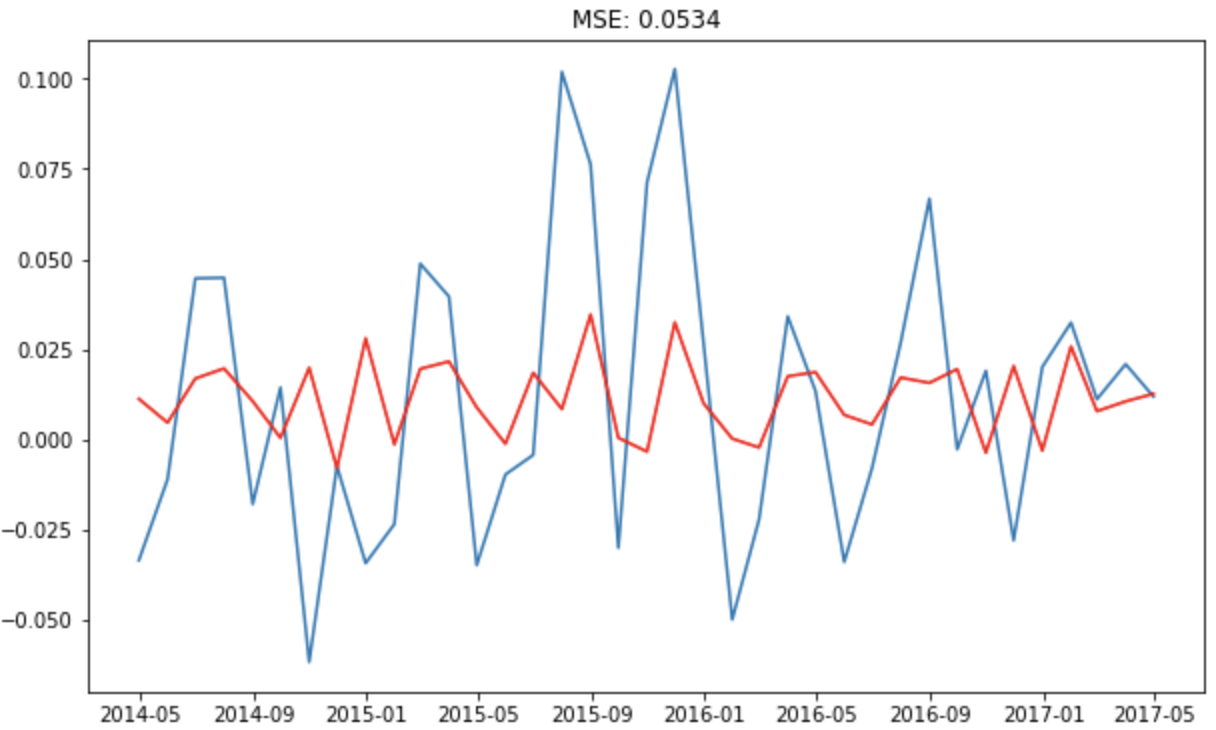


Fig 15: MA model prediction(over normalized data)

ARIMA:

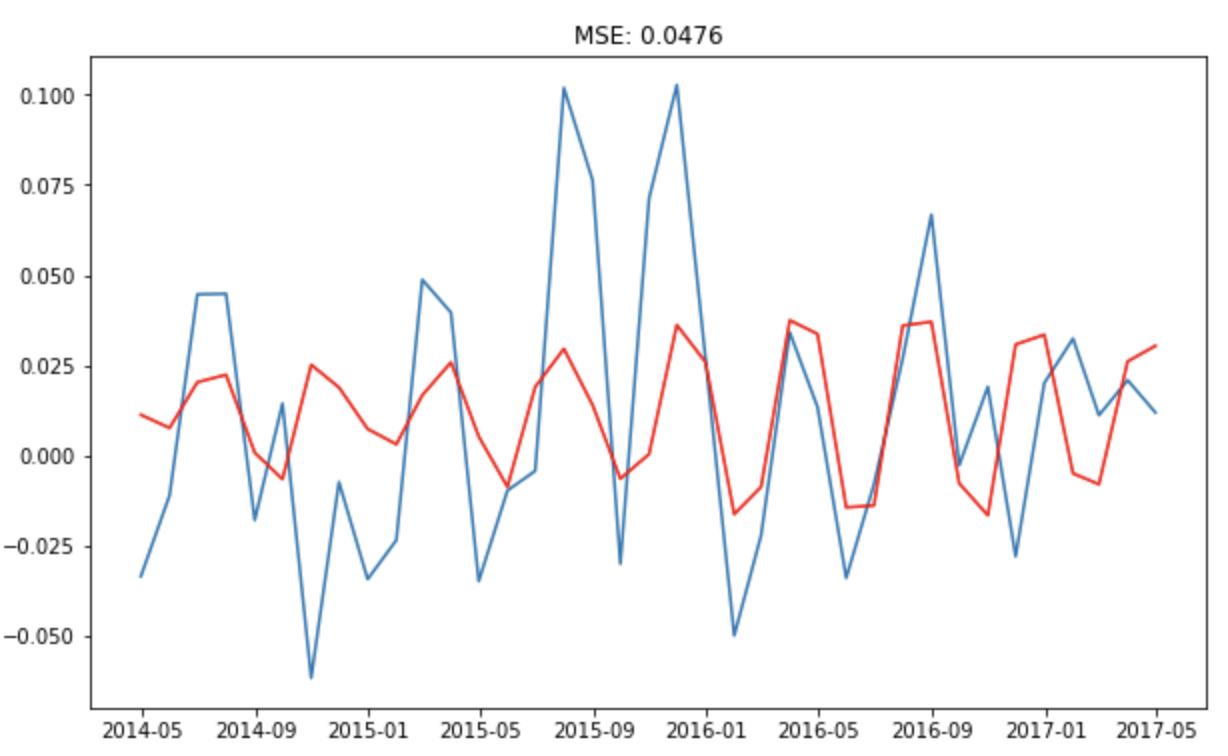


Fig 16: ARIMA model prediction(over normalized data)

Final plot on original scale:

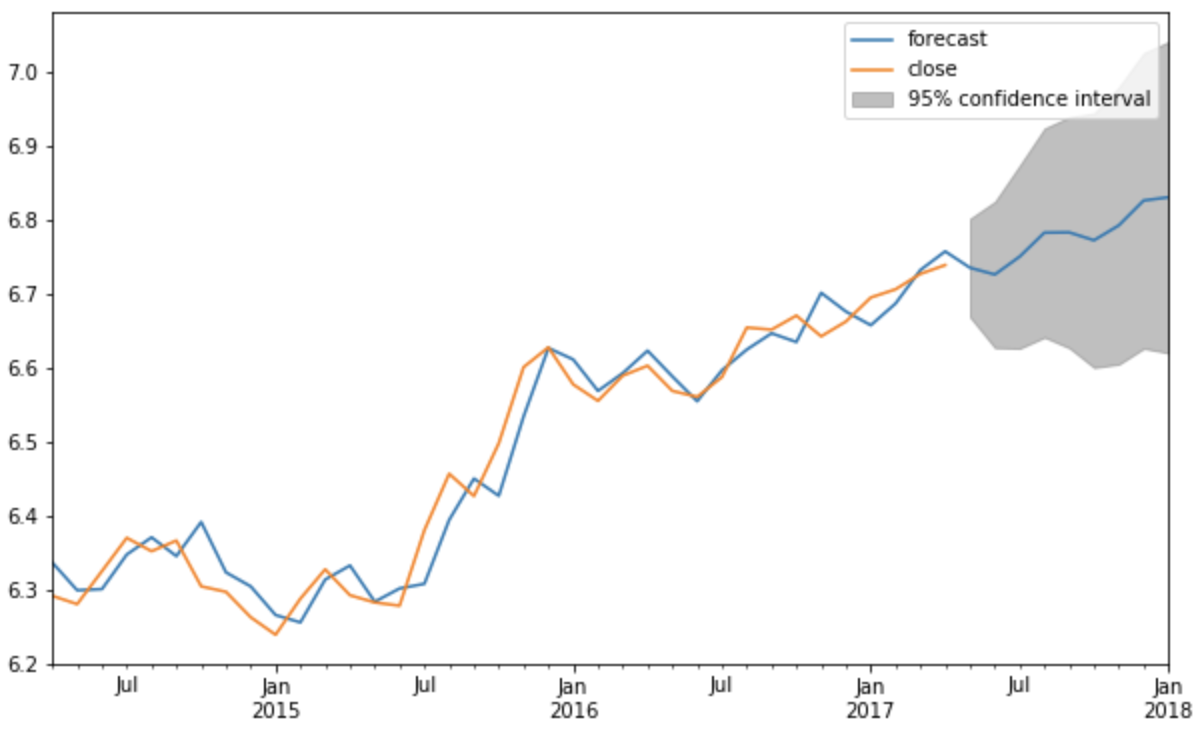


Fig 17: ARIMA model prediction(over original data)

**CONCLUSION**

As discussed in the previous section, both the models were implemented successfully and following results were identified:

1. The LSTM model predicted the values with the MSE of 0.015
2. The ARIMA model predicted values with the MSE of 0.047
3. The AR model predicted the values with the MSE of 0.051
4. The MA model predicted the values with the MSE of 0.053

The Above results are summarized in the table as follows:

|  |  |
| --- | --- |
| **Model** | **MSE** |
| LSTM | 0.015 |
| ARIMA | 0.047 |
| AR | 0.051 |
| MA | 0.053 |

Table 1: Comparison of Models

From the above table, it is clear that the MSE of the LSTM model is lesser than the MSE of the ARIMA model intuitively AR and MA models.

In conclusion, it is experimentally identified that the LSTM model works best with predicting the time series data.

**REFERENCES**

[1]Hushani, Phillip. (2018). Using Autoregressive Modelling and Machine Learning for Stock Market Prediction and Trading: ICICT 2018, London.767-774.10.1007/978-981-13-1165-9\_70.

[2] S. Yao, L. Luo and H. Peng, "High-Frequency Stock Trend Forecast Using LSTM Model," 2018 13th International Conference on Computer Science & Education (ICCSE), Colombo, 2018, pp. 1-4.

[3]<https://wall-street.com/trading-basics-factors-influence-share-prices/>

[4] P. Tongkhow and N. Kantanantha, "Bayesian model for time series with trend, autoregression and outliers," 2012 Tenth International Conference on ICT and Knowledge Engineering, Bangkok, 2012, pp. 90-94.

[5] K. Ichinose and K. Shimada, "Stock Market Prediction from News on the Web and a New Evaluation Approach in Trading," 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Kumamoto, 2016, pp. 77-81.

[6] S. Hansun, "A new approach of moving average method in time series analysis," 2013 Conference on New Media Studies (CoNMedia), Tangerang, 2013, pp. 1-4.

[7] R. Li, D. Fu and Z. Zheng, "An Analysis of the Correlation between Internet Public Opinion and Stock Market," 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, 2017, pp. 150-153.

[8] G. Shyamala and N. Pooranam, "A survey on online Stock forum using subspace clustering," 2016 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2016, pp.1-6.

[9] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, 2014,

pp. 106-112.

[10]<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[11]<https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>

[12]<https://www.kaggle.com/camnugent/sandp500#ADBE_data.csv>

[13]K. Chen, Y. Zhou and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, 2015, pp. 2823-2824.

[14]Jung-Hua Wang and Jia-Yann Leu, "Stock market trend prediction using ARIMA-based neural networks," Proceedings of International Conference on Neural Networks (ICNN'96), Washington, DC, USA, 1996, pp. 2160-2165 vol.4.

[15]S. M. Idrees, M. A. Alam and P. Agarwal, "A Prediction Approach for Stock Market Volatility Based on Time Series Data," in IEEE Access, vol. 7, pp. 17287-17298, 2019.