Stock Market Forcasting using ARIMA and LSTM Models.

Malita Dodti
School of Computer Science
University of Windsor
Windsor, Canada
dodti@uwindsor.ca

Khushboo Ladha
School of Computer Science
University of Windsor
Windsor, Canada
ladhak@uwindsor.ca

Dhairya Chauhan
School of Computer Science
University of Windsor
Windsor, Canada
chauha51@uwindsor.ca

Kushal Manvar
School of Computer Science
University of Windsor
Windsor, Canada
manvark@uwindsor.ca

Abstract:

Today, the stock market attracts much attention from various big Investment firms through its high-level risks and high returns. Stock Exchange markets typically involve securities investments like NYSE, NSE, DJIA, that are beneficial to the global economy. This project proposed a prediction model based on the analytics for forecasting Stock markets' movements using time series forecasting for a short-term prediction like the traditional ARIMA model and the state-of-art deep learning sequential model, namely Long Short-Term Memory Model (LSTM). These models will be evaluated, analyzed and compared, following the main project directions. This project focuses on quantitative forecasting involving our variable to forecast (close price), applying statistical principles analysis and advanced concepts to a given historical data. The correlations and forecasts obtained to conclude the model's good potential for prediction of stock market Forecasting.

Introduction:

In financial stock markets, it is considered to be an impossible task to predict the most-likely stock market movement. There are generally two approaches to predicting future market movements. One approach [1] involves using Time Series Forecasting methods on previous historical data to find patterns in short-term seasonal intervals predict the most likely bounded price change and its accuracy. This approach uses Historical stock market changes in the dataset for forecasting by assuming that specific patterns have bearings on the future for short-term linear intervals, making the ARIMA prediction model for time series forecasting famous. Another challenging approach to ARIMA assumes using Historical data as it is like a random series and use it for short-term forecasting. Thus, ARIMA is the basis for another popular model known as Random Walk. These approaches can be combined with the prediction of Stock returns to uncover market patterns, plan strategies for future investments, and forecast the value of the next day's closing price. The quantitative trading industry has shifted into the 'deep learning era.' Additionally, this project, we predict stock

return using the deep learning model with TensorFlow, which uses the Keras back-end in R for predicting Time Series Analysis, implementing the LSTM model.

Motivation:

Macro-economic factors, international events, and human behavior govern Stock markets Future. Hence, forecasting stock returns can become a challenging task—higher accuracy for Stock market prediction guarantees the profitability of investments in stock markets. Thus, having a forecasting model or technique to predict the market's direction lowers investment risk promptly and uncertainty. Thereby enhancing investment flows into stock markets and helps policymakers and regulators make appropriate decisions and take corrective measures.

Literature Review:

Fundamental analysis and technical analysis form the basis of stock markets Forecasting, where fundamental aspects deal with financial analysis of the company or industries. In contrast, technical relies on historical data, securities, market trends and assumptions [2]. Many researchers claimed the absence of predictability, which was challenged by the random walk theory [3]. The autoregressive integrated moving average [4] model, one of the time series forecasting techniques, was employed for stock market returns [5]; later, they found it produced inferior forecasts for time-series not address non-linearity problems.
Meanwhile, artificial intelligence (AI) models were introduced for Forecasting purposes; their data-driven, nonlinear, and self-adaptive features made them famous for stock market forecasts.[6] It focuses on selecting the critical and relevant information hence enhancing the predictive accuracy using the LSTM model. Various studies suggest that no single model is suitable for forecasting the returns of all stock markets. It motivates researchers to compare [8] the models to find an optimal solution that forecasts all markets to save time and resources and make better decisions.

Dataset:

Popular sites like Yahoo! News [12] have large chunks of research data that try to predict the stock market in real-time. The stock downloaded is NIFTY 50, (Fig 1) representing the Indian stock market index consisting of the weighted average of Fifty largest Indian companies listed on the National Stock Exchange. The dataset applied on the model is downloaded using the function getsymbols () from the quantmod package. NSEI contains missing values which is removed using na.omit(). The dataset consists of 2193 entries ranging from date 2012-01-03 to date 2020-12-17.

	NSEI.Open	NSEI.High	NSEI.Low	NSEI.Close	NSEI.Volume	NSEI.Adjusted
2012-01-03	4675.80	4773.10	4675.80	4765.30	0	4765.30
2012-01-04	4774.95	4782.85	4728.85	4749.65	0	4749.65
2012-01-05	4749.00	4779.80	4730.15	4749.95	0	4749.95
2012-01-06	4724.15	4794.90	4686.85	4754.10	0	4754.10
2012-01-09	4747.55	4758.70	4695.45	4742.80	0	4742.80
2012-01-10	4771.85	4855.90	4768.25	4849.55	0	4849.55
2012-01-11	4863.15	4877.20	4841.60	4860.95	0	4860.95
2012-01-12	4840.95	4869.20	4803.90	4831.25	0	4831.25
2012-01-13	4861.95	4898.85	4834.20	4866.00	0	4866.00
2012-01-16	4844.00	4880.80	4827.05	4873.90	0	4873.90
2012-01-17	4904.50	4975.55	4904.00	4967.30	0	4967.30
2012-01-18	4977.75	4980.65	4931.05	4955.80	0	4955.80

Fig 1: Dataset downloaded.

Methodology: ARIMA.

George Box and Gwilym Jenkins proposed ARIMA in the '70s. ARIMA full form is for Auto-Regressive Integrated Moving Average. ARIMA models use linear functions based on previous variables and random errors to predict future values. It predicts future short-term values based on changing historical data. They are hence making it a famous linear model.
Auto Regression (p), Integration (d), Moving average (q) are the three statistical notations that make the ARIMA Model.

Auto Regression(p) — Regression equation uses a weighted sum of past values to create time series described by data points regressed on their own lagged values. Here it uses a single predictor that combines past values for autoregression, Unlike a linear combination of predictors.
Integration (d) — It is also known as Differences. Since one of ARIMA's pre-requisite is to have consistent series, unstable time series should be made stationary or differenced to eliminate trends. Differencing subtracts raw data observations in the current period from the previous ones until the data does not grow at an increasing rate. This component also removes seasonal trends.
Moving average (q) — It represents the number of error terms in the regression equation. The differenced data consists of trends or lags; therefore, it creates an indicator to determine upward or downward trends. The longer the period for the moving average, the more significant the lag, the more likely the change.

Implementation: ARIMA.

First, the data from Yahoo Finance is loaded. The close price is considered as a benchmark for future predictions. since they best reflect the stock's changes in real values during the time frame. Refer Fig 3 for better understanding of data.

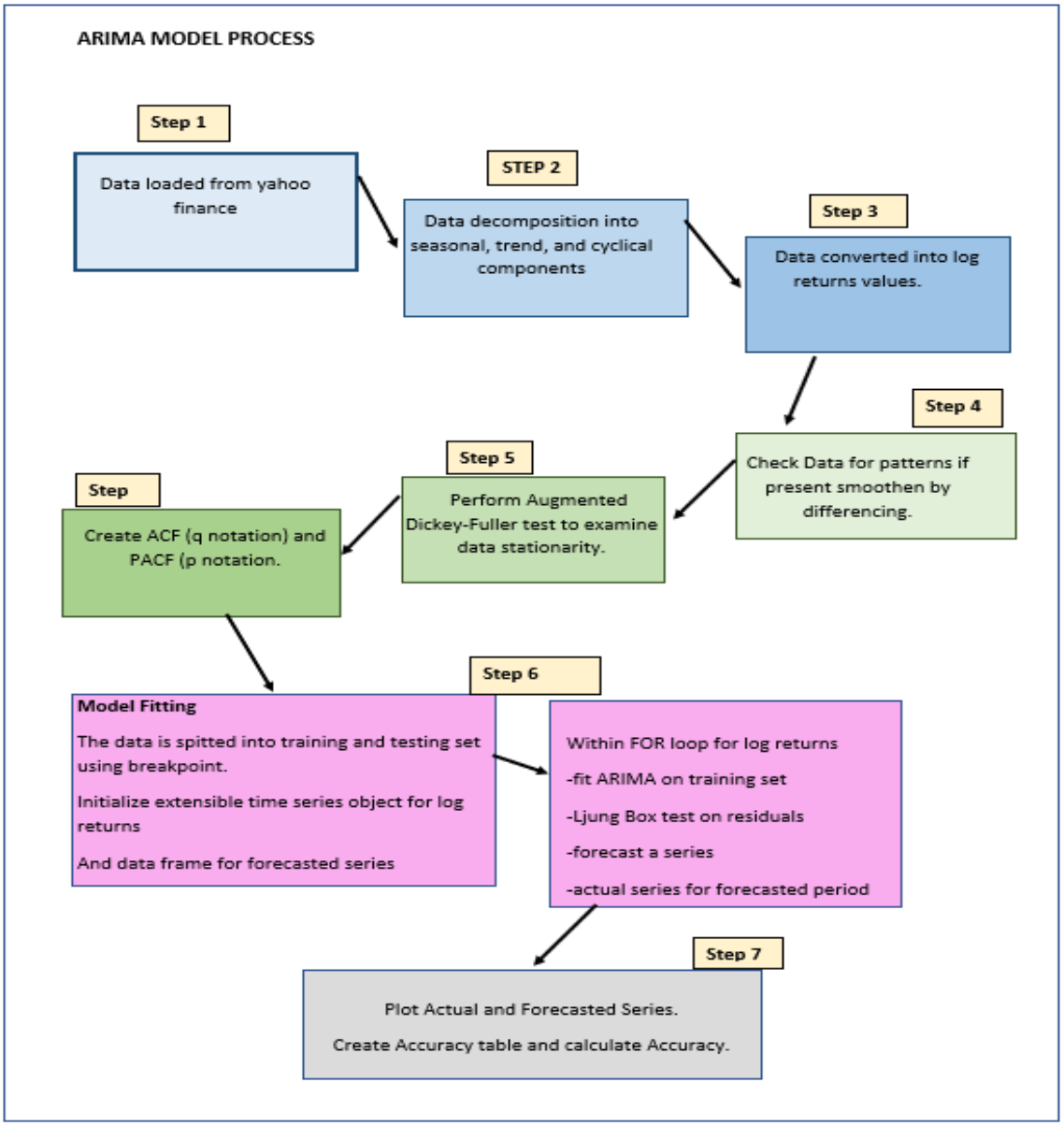

Fig 2: Architecture of ARIMA.

Next, the data is decomposed into the trend and seasonal components to analyze further, forming the foundation of the ARIMA Model building (Fig 2), which provides information on the stock behavior over a past period. Additionally, the residual error component describes the section of data not explained by trend and seasonality components.

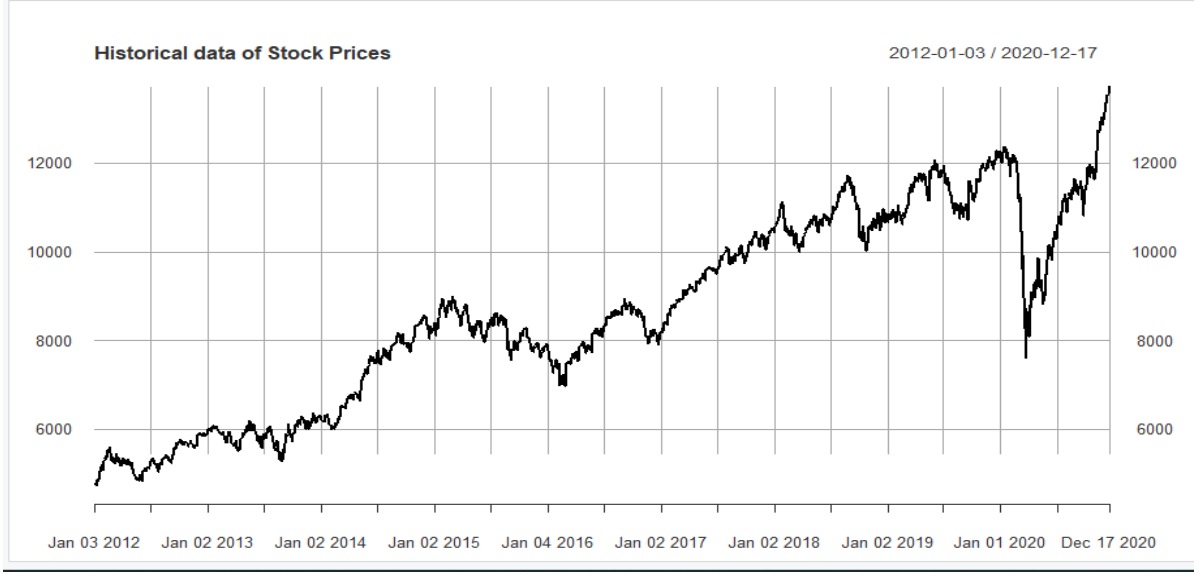

Fig 3: Plot of Historical Data for Stock NSEI.

The data is smoothened to stabilize variance using logarithmic returns. Statistical properties like mean, variance, autocorrelation and Linear properties, y-intercept and slope, are maintained constant over time by integrating data. Refer to the Fig 4 which shows data after smoothening.

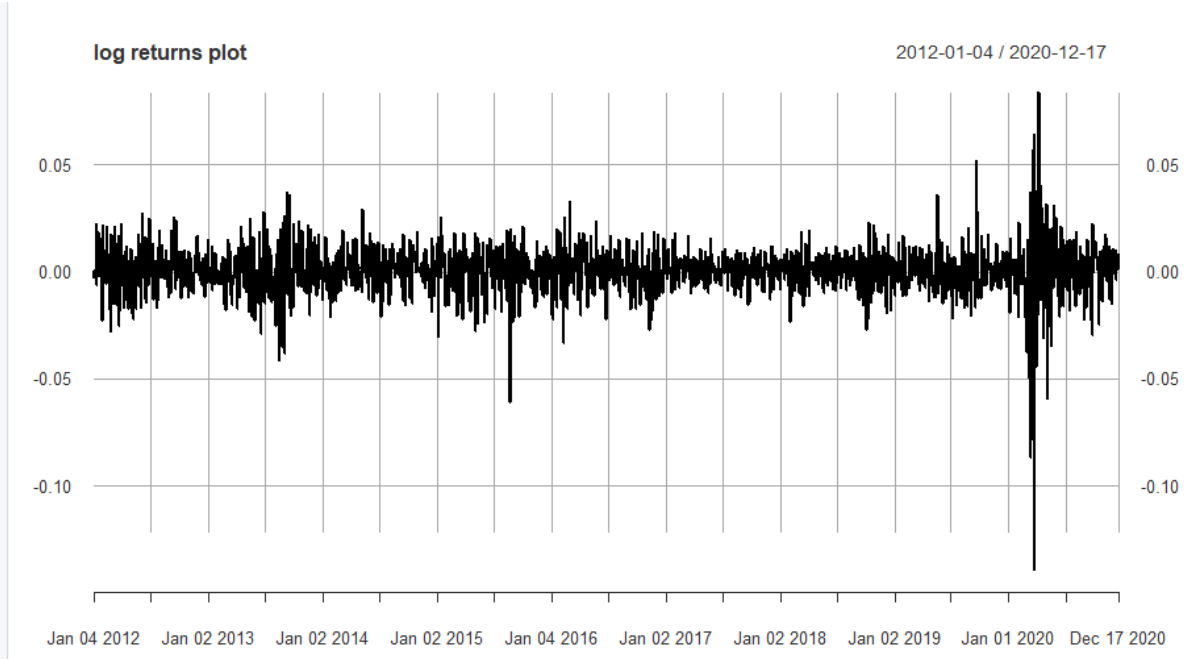

Fig 4: Logarithmic returns

After visual evidence of data usability, the Augmented Dickey-Fuller Test determines if the data is suitable to use. The results imply more negative the value of the ADF test, the higher the chances of Null hypothesis rejection.

In the next step, the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) displays lag values (Fig 5) for getting error term value (q) and Auto-Regression (p), respectively. The values of p,d and q were adjusted automatically in each seasoned interval for improving the accuracy of the overall prediction. The ARIMA testing gave the idea of how well the model fits the data through AIC, AICc and BIC values. All the values obtained were shallow, and the lower the values of these terms, the better the ARIMA model fits. The ARIMA model value obtained is (0,1,0)

The datasets are then split into training and testing sets by establishing a breakpoint where the ARIMA model (0,1,0) is trained on training sets and assessed on the testing set. Lastly, the Forecasted series is visualized to get a pictorial view of the results obtained by the ARIMA model. (Fig 6) It shows the Actual series of datapoints in red color and predicted series in blue color.

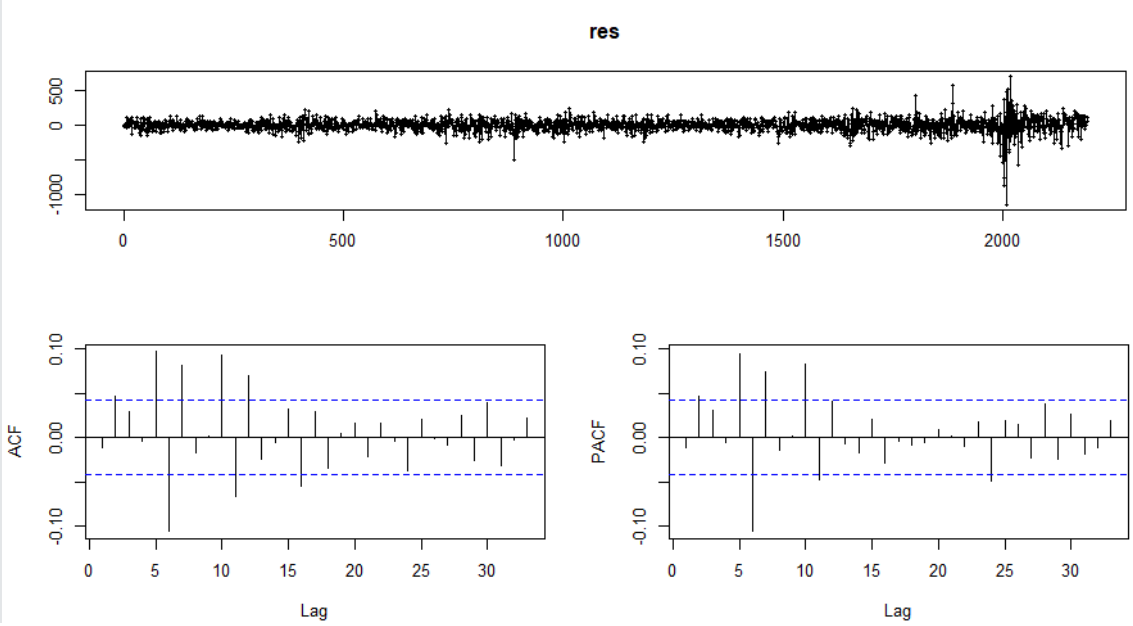

Fig 5.i) Residual component, ii) ACF plot, iii) PACF plot

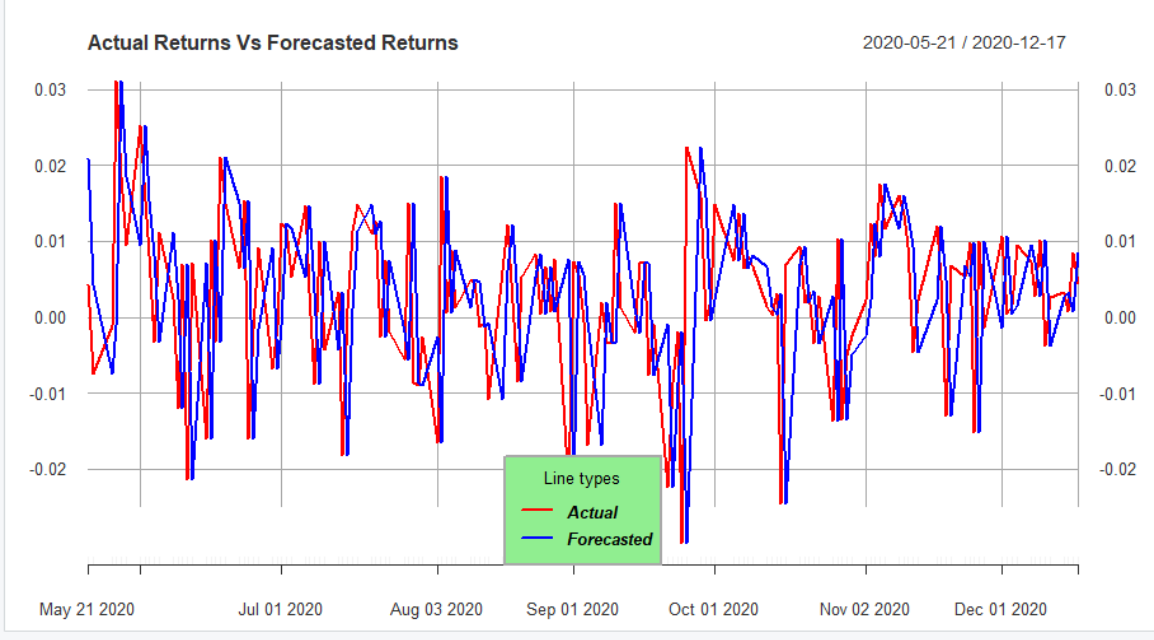

Fig 6. Actual and Forecasted Returns.

Methodology: LSTM.

Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) that overcomes the vanishing gradient problem. Thus, learning long-term dependencies.
LSTM network is made of memory blocks, referred to as cells, connected through layers. The cells' information in cell state Ct and hidden state Ht is modulated by mechanisms, known as gates, through *sigmoid* and *tanh* activation functions. Generally, the gates' input consists of previously hidden layer output H(t-1) and current input I(t) multiplied to weight matrix Wi in addition to the bias Bi.
The three main gates are:

Forget Gate:

- It removes information of less importance from the network.
- The output is either '0' or '1'. '0', meaning remove the information, and '1' means to retain it.
- Sigmoid function discards the information based on its output.

Input gate:

- The sigmoid function acts as a filter to the information included in the cell state.
- The tan activation layer creates a vector of possible values added to the cell state. Its outputs values range from -1 to +1.
- The old cell state is updated by adding previous information to the product of the filter (sigmoid function) and the vector (tan activation function).

Output gate:

- A vector is scaled to the range [-1, +1], after applying tan activation function.

- The sigmoid function is employed to create a filter that outputs values from the vector created above.
- Finally, the scaled vector is multiplied by the filtered output to obtain the hidden cell state, which passes down to the next cell.

Implementation: LSTM.

The data is transformed to static data by getting the difference of consecutive data points in the series, removing the time dependency of data and increasing the predictive power. LSTM operates in a supervised learning mode, which is achieved by lagging (Fig 7) the value at a time (t−v) as the input and value at time t as the output for a v-step lagged dataset.

The dataset is then split into training sets and testing sets. For further processing, the data is normalized by scaling it using sigmoid activation function.

The model is defined to be stateful since the previous sample state is used as input to the next sample. The input is a 3D array [s,t,f] where s is the number of samples in each iteration, t is the timestamp for given samples, and f is the features.

```
> supervised = lag_transform(diffed, 1)
> head(supervised)
                 x-1           x
2012-01-03   0.000000   0.000000
2012-01-04   0.000000 -15.649903
2012-01-05 -15.649903   0.300293
2012-01-06   0.300293   4.149903
2012-01-09   4.149903 -11.300293
2012-01-10 -11.300293 106.750000
```

Fig 7: Output of lagged series.

For evaluating the model, parameters used are mean_squared_error as the loss function and Adaptive Monument Estimation as the optimization algorithm and learning rate. The argument reset_states (Fig 8) is employed, which resets network states for every epoch to fit the model.

```
Epochs = 50
for(i in 1:Epochs ){
  model %>% fit(x_train, y_train, epochs=1, batch_size=batch_size, verbose=1, shuffle=FALSE)
  model %>% reset_states()
}
```

Fig 8: Reset of states for each epoch.

Lastly, we get the prediction for the LSTM model. Fig 9 shows the green series (predicted) overlapping the actual series in red for the test dataset.

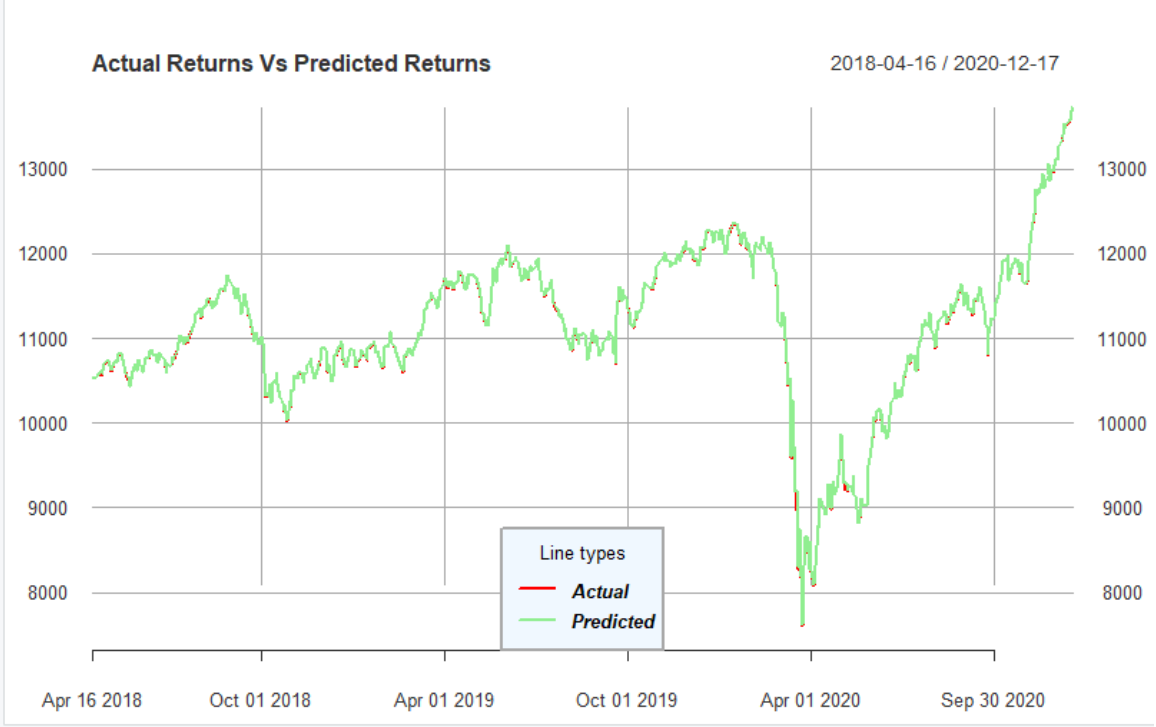

Fig 9: Prediction Using LSTM model.

Results and Discussion:

ARIMA used 80 percent of data for training and 20 percent for testing. Aggregating values calculated the Accuracy Percentage of the forecasted in a table. ARIMA's testing phase compares (Fig 10) the predicted stock returns for testing logarithmic returns data with their actual returns. The accuracy obtained is 58 percent, which is improved when using LSTM for prediction.

```
> print(comparsion)
           Actual_series    Forecasted Accuracy
2020-05-21  0.0043691956  0.0208916389        1
2020-05-22 -0.0073847846  0.0043691956        0
2020-05-26 -0.0011290709 -0.0073847846        1
2020-05-27  0.0311735219 -0.0011290709        0
2020-05-28  0.0186284515  0.0311735219        1
2020-05-29  0.0094597776  0.0186284515        1
2020-06-01  0.0253383527  0.0094597776        1
2020-06-02  0.0154456277  0.0253383527        1
2020-06-03  0.0082283423  0.0154456277        1
2020-06-04 -0.0032303808  0.0082283423        0
2020-06-05  0.0112092176 -0.0032303808        0
```

```
> print(Accuracy_percentage)
[1] 57.82313
```

Fig 10: Accuracy evaluation.

The future (50 days) closing price for a stock (Fig 11) can be predicted using the fitted ARIMA model.

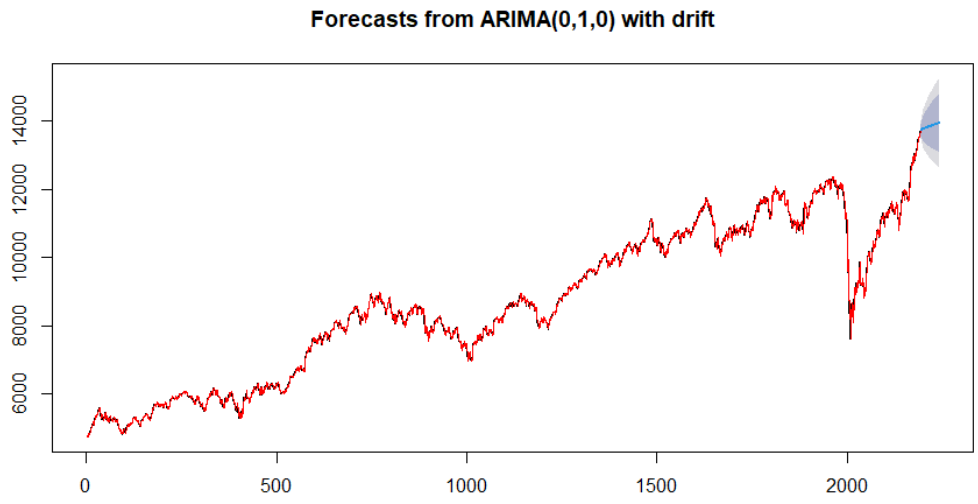

Fig 11: Future predicted stock close price.

The LSTM model employed a tanh activation function for the LSTM layers and a sigmoid activation function for output layers. Thus, showing more superior results than other activation functions. Moreover, the dropout probability applied to each hidden layer as the usual regularization method reduced the overfitting issue. Finally, the ADAM optimization avails learning the parameters, and the mean squared error serves as the loss function.

The Actual values and predicted values are compared to get the accuracy (Fig 12) obtained during Testing phase. The accuracy obtained is 96 percent.

```
> comparsionlstm = merge(Act_ser,predicted_series)
> comparsionlstm$Accuracy1 = sign(comparsionlstm$NSEI.Close)==sign(comparsionlstm$predictions)
> print(comparsionlstm)
           NSEI.Close predictions Accuracy1
2018-04-16   10528.35   10530.460         1
2018-04-17   10548.70   10550.810         1
2018-04-18   10526.20   10528.310         1
2018-04-19   10565.30   10567.410         1
2018-04-20   10564.05   10566.160         1
2018-04-23   10584.70   10586.810         1
2018-04-24   10614.35   10616.460         1
2018-04-25   10570.55   10572.660         1
2018-04-26   10617.80   10619.910         1
2018-04-27   10692.30   10694.410         1
2018-04-30   10739.35   10741.460         1
2018-05-02   10718.05   10720.160         1
2018-05-03   10679.65   10681.760         1
2018-05-04   10618.25   10620.360         1
2018-05-07   10715.50   10717.610         1
2018-05-08   10717.80   10719.910         1
```

Fig 12: Accuracy evaluation for LSTM.

Conclusion:

This project attempts to develop a prediction and forecasting model for finding the future stock market movements and their values using correlation for time series analysis, historical stock market data, data preprocessing, and machine learning algorithms and techniques.

The predicted outputs Display the proposed model's potential to forecast the stock market movements for the short-term analysis of the future, helping investors in their profitable investments in securities of stock markets and decisions related to buying/selling/holding a stock share. They can also contribute to advancements in technology by investing in the best Technology industry and compete successfully with other emerging prediction and forecasting techniques.

Future works:

Improvements such as using sentiment analysis of opinions and emotions expressed online and from news sources can be used for forecasting.

Additionally, one direction of future work will be handling stock time series volatility and the difficulty of predicting the stock market arises from its non-stationary behavior.

References:

[1] Angadi, Mahantesh C., and Amogh P. Kulkarni. "Time Series Data Analysis For Stock MarketPrediction Using Data Mining Techniques With R." International Journal of Advanced Research in Computer Science 6, no. 6 (2015), August 2015.

[2] Levy RA (1967) The theory of random walks: a study of findings. Am Econ 11(2):34–48

[3] Fama EF (1970) Efficient capital markets:a review of theory and empirical work. J Financ 25(2):383–417.

[4] Box GEP, Jenkins GM (1970) Time series analysis: forecasting and control. Holden-Day, San Francisco.

[5] Ojo JF, Olatayo TO (2009) ON the estimation and performance of subset of autoregressive integrated moving average models. Eur J Sci Res 28:287–293

[6] Zhichao Zou, Zihao Qu, Using LSTM in Stock prediction and Quantitative Trading

[7] Kenneth Page, Stock Price Forecasting Using Time Series Analysis, Machine Learning and single layer neural network Models,2019.

[8] Data-based investor predicting the next decade in the stock market, R blogger,2019

[9] Aishwarya Singh, Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes),2018

[10] Pranjal Srivastava ,Essentials of Deep Learning: Introduction to Long Short Term Memory,2017

[11] Quanstart-Article, Autoregressive Integrated Moving Average ARIMA (p, d, q) Models for Time Series Analysis

[12] Yahoo! Finance Stock Details for Nifty