NIFTY 50 Value Prediction Using ML & DL Model and Comparison of It

PAPER ID: Abs_124_AppScicon22

Submitted by:
De, Abhik*; Maity, Anubhab; Parvin, Ruksena; Sarkar, Sreeja; Chatterjee, Shreya; Bandyopadhyay, Sayantan

Department of Applied Statistics, School of Natural and Applied Science, MAKAUT WB, India

Acknowledgement

We would like to thank all the people who contributed in some way in this project in the Department of Applied Statistics & Analytics, School of Natural and Applied Science, Maulana Abul Kalam Azad University of Technology. Firstly, we offer our sincere gratitude to our respected supervisor, Mr. Mayukh Bhattacharjee for continuous support for our project, for his patience, motivation and immense knowledge. His guidance helped us in all the time of computing and writing this project.

We owe a debt of gratitude to all of my friends, my professors of Maulana Abul Kalam Azad University of Technology, for giving us support and inspiration in every aspect for completing our project.

Last but not the least; we would like to express our deepest gratitude to our family for their unflagging love and unconditional support throughout our studies and our life in general. This accomplishment would not have been possible without them.

.

CONTENT

Introduction

What is stock?

A stock (also known as equity) is a security that represents the ownership of a fraction of a corporation. This entitles the owner of the stock to a proportion of the corporation's asset and profits equal to how much stock they own.Umits of stock are called "shares".

What is NIFTY50?

The NIFTY50 is a benchmark Indian stock market index that represents the weighted average of 50 of the largest Indian companies listed on the National Stock Exchange. It is one of the two main stock indices used in India, the other being the BSE SENSEX.

OBJECTIVE

1) Develop a model to understand nifty index movement.
2) Future prediction
3) Comparison of performance between ML and DL model

DATA DESCRIPTION

Date:	Date/time of record.
Open:	It is the price at when the financial security opens in the market when trading begins (9:00 am).
Close:	The close is a reference to the end of a trading session in the financial markets when the market close for the day (3:00 pm).
Adj Close:	Closing price after adjustments for all applicable splits and dividend distributions.
High:	The high is the highest price at which a stock traded during a period.
Low:	The low is the amount of an asset or security that changes hand over some period of time, often over the course of a day.

Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Model 1:

Auto-Regressive Integrated Moving Average (ARIMA)

Stapes of Model Devolopment

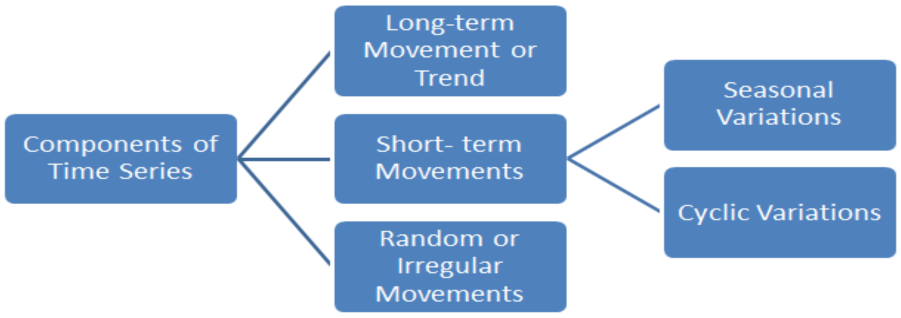
1) For modeling purpose split the data in two category

   i)**Training Data :** -Training data is the data use to train a model to predict the outcome of designed model to predict.

   ii) **Test Data :** - Data selected to satisfy the execution preconditions and inputs to execute or more test cases. Test data is used to measure the performance, such as accuracy or efficiency, of the model using to train the training data.

   FORECAST is the method in which value predicted through modeling based on previous value.

2) ***TIME SERIES:*** *- A time series is a collection of observations made sequentially through time.*
3) ***Components of time series :-***


Components of Time Series
Long-term Movement or Trend
Short- term Movements
Random or Irregular Movements
Seasonal Variations
Cyclic Variations

4) ***Multiplicative model of time series*** *: -*

$$Y = T*S*C*I$$

Where we denote the time series by Y, the secular trend by T, the seasonal or short term periodic

movements by S, the long term cyclical movements by C and the irregular or residual component by I .

5) ***Stationary time series: -*** A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary the trend and seasonality will affect the value of the time series at different times.

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time.

Most statistical forecasting methods are based on the assumption that the time series can be rendered approximately stationary after mathematical transformations.

6) ***Lag: -*** *A "lag" is a fixed amount of passing time; One set observations in a time series is plotted (lagged) against a second, later set of data. The k^{th} lag is the time period that happened "k" time points before time i.*

. The first lag of Y_t is its value in the preceding time period, Y_{t-1}.

• The second lag of Y_t is its value in the two periods preceding, Y_{t-2}.

• The k'th lag of Y_t is Y_{t-k}.

7) ***Augmented Dickey- Fuller test –***

The Augmented Dickey-Fuller test allows for higher-order autoregressive processes by including ΔY_{t-p} in the model. But our test is still if $\gamma=0$

$\Delta Y_t = \alpha + \beta_t + \gamma Y_{t-1} + \delta_1 \Delta Y_{t-1} + \delta_2 \Delta Y_{t-2} + \ldots$

The null hypothesis for both tests is that the data are non-stationary. We want to reject the null hypothesis for this test, if test statistic value less than critical value. The test is left-tailed.

The null hypothesis of the Augmented Dickey-Fuller t-test is

$H_0 : \gamma = 0$ (i.e. series not stationary)

Versus

The alternative hypothesis is $H_1 : \gamma < 0$ (i.e. series stationary)

Test procedure has three type:-

. No constant, no trend : $\Delta Y_t = \gamma Y_{t-1} + \sum_{j=1}^{p} (\delta j \Delta Yt - j) + e_t$

. Constant, no trend : $\Delta Y_t = \alpha + \gamma Y_{t-1} + \sum_{j=1}^{p} (\delta j \Delta Yt - j) + e_t$

. Constant and trend : $\Delta Y_t = \alpha + \beta_t + \gamma Y_{t-1} + \sum_{j=1}^{p} (\delta j \Delta Yt - j) + e_t$

The test statistic is $DF_T = \frac{\gamma\hat{}}{Standard\ Error(\gamma\hat{})}$

If the DF_T statistic is more negative than the table value, reject the null hypothesis. The more negative the DF test statistic the stronger the evidence for rejecting the null hypothesis.

Critical values			
Without trend		With trend	
1%	5%	1%	5%
-3.43	-2.86	-3.96	-3.41

8) **ARIMA(p,d,q) forecasting equation:** ARIMA models are, in theory, the most general class of models for forecasting a time series which can be made to be "stationary" by differencing (if necessary), perhaps in conjunction with nonlinear transformations such as logging or deflating (if necessary). A random variable that is a time series is stationary if its statistical properties are all constant over time. *A stationary series has no trend, its variations around its mean have a constant amplitude, and it wiggles in a consistent fashion*, i.e., its short-term random time patterns always look the same in a statistical sense. The latter condition means that its *autocorrelations* (correlations with its own prior deviations from the mean) remain constant over time, or equivalently, that its *power spectrum* remains constant over time. A random variable of this form can be viewed (as usual) as a combination of signal and noise, and the signal (if one is apparent) could be a pattern of fast or slow mean reversion, or sinusoidal oscillation, or rapid alternation in sign, and it could also have a seasonal component. An ARIMA model can be viewed as a "filter" that tries to separate the signal from the noise, and the signal is then extrapolated into the future to obtain forecasts.

The ARIMA forecasting equation for a stationary time series is a *linear* (i.e., regression-type) equation in which the predictors consist of *lags of the dependent* variable and/or *lags of the forecast errors*. That is:

Predicted value of Y = a constant and/or a weighted sum of one or more recent values of Y and/or a weighted sum of one or more recent values of the errors.

If the predictors consist only of lagged values of Y, it is a pure autoregressive ("self-regressed") model, which is just a special case of a regression model and which could be fitted with standard regression software. For example, a first-order autoregressive ("AR(1)") model for Y is a simple regression model in which the independent variable is just Y lagged by one period (LAG(Y,1) in Statgraphics or Y_LAG1 in RegressIt). If some of the predictors are lags of the errors, an ARIMA model it is NOT a linear regression model, because there is no way to specify "last period's error" as an independent variable: the errors must be computed on a period-to-period basis when the model is fitted to the data. From a technical standpoint, the problem with using lagged errors as predictors is that *the model's predictions are not linear functions of the coefficients*, even though they are linear functions of the past data. So, coefficients in ARIMA models that include lagged errors must be estimated by *nonlinear* optimization methods ("hill-climbing") rather than by just solving a system of equations.

The acronym ARIMA stands for Auto-Regressive Integrated Moving Average. Lags of the stationarized series in the forecasting equation are called "autoregressive" terms, lags of the forecast errors are called "moving average" terms, and a time series which needs to be differenced to be made stationary is said to be an "integrated" version of a stationary series. Random-walk and random-trend models, autoregressive models, and exponential smoothing models are all special cases of ARIMA models.

A nonseasonal ARIMA model is classified as an "ARIMA(p,d,q)" model, where:

- **p** is the number of autoregressive terms,

- **d** is the number of nonseasonal differences needed for stationarity, and
- **q** is the number of lagged forecast errors in the prediction equation.

The forecasting equation is constructed as follows. First, let y denote the d^th^ difference of Y, which means:

If d=0: $y_t = Y_t$

If d=1: $y_t = Y_t - Y_{t-1}$

If d=2: $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

Note that the second difference of Y (the d=2 case) is not the difference from 2 periods ago. Rather, it is the *first-difference-of-the-first difference,* which is the discrete analog of a second derivative, i.e., the local acceleration of the series rather than its local trend.

In terms of y, the general forecasting equation is:

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$$

Here the moving average parameters (θ's) are defined so that their signs are negative in the equation, following the convention introduced by Box and Jenkins. Some authors and software (including the R programming language) define them so that they have plus signs instead. When actual numbers are plugged into the equation, there is no ambiguity, but it's important to know which convention your software uses when you are reading the output. Often the parameters are denoted there by AR(1), AR(2), …, and MA(1), MA(2), … etc..

To identify the appropriate ARIMA model for Y, you begin by determining the order of differencing (d) needing to stationarize the series and remove the gross features of seasonality, perhaps in conjunction with a variance-stabilizing transformation such as logging or deflating. If you stop at this point and predict that the differenced series is constant, you have merely fitted a random walk or random trend model. However, the stationarized series may still have autocorrelated errors, suggesting that some number of AR terms ($p \geq 1$) and/or some number MA terms ($q \geq 1$) are also needed in the forecasting equation.

The process of determining the values of p, d, and q that are best for a given time series will be discussed in later sections of the notes (whose links are at the top of this page), but a preview of some of the types of *nonseasonal* ARIMA models that are commonly encountered is given below.

9) *Akaike Information Criteria –*

Akaike information criteria can be used for selecting the best model among the competing models. This criterion is used for measuring the goodness of fit of the model. This criterion are minimized over the choice of repressors, it will be minimum when the model is good fit and less complex. In comparing two or more models, the best model is the having the least AIC value.

In a regression setting, the estimates of the β_t based on least squares and the maximum likelihood estimates are identical. The difference comes from estimating the common variance σ^2 of the normal distribution for the errors around the true means. It has been using the best unbiased estimator of σ^2,ˆσ^2 = RSS/(n-p), where there are p parameters for the means (p different β_t parameters) and RSS is the residual sum of squares. This estimate does not tend to be too large or too small on average. The maximum likelihood estimate, on the other hand, is RSS/n. This is estimate has a slight negative bias, but also has a smaller variance. Putting all of this together, we it can be written -2 the log-likelihood to be

$$n + n \log (2\Pi) + n \log(RSS/n).$$

In a regression setting. Now, AIC is defined to be -2 times the number of parameters. If there are p different β_t parameters, there are a total of p+1 parameters if we also countσ^2. The correct formula for the AIC for a model with parameters β_0, β_1, . . . ,β_{p-1} and σ^2 is

$$AIC = n + n \log (2\Pi) + n \log(RSS/n) + 2(p+1)$$

DEEP LEARNING

Deep learning is a subset of Machine Learning that achieves great power and flexibility by learning to represent the world as nested hierarchy of concepts

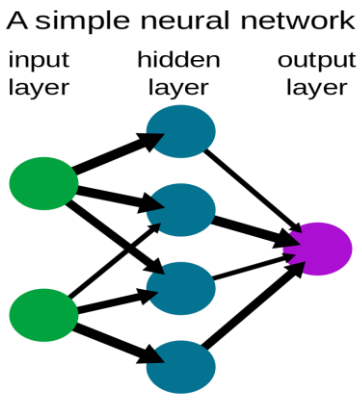
Difference between in ML and DL

When solving a problem using traditional machine learning algorithm, it is generally recommended to break the problem down into different parts, solve them individually and combine them to get the result. Deep Learning in contrast advocates to solve the problem end-to-end.

NEURAL NETWORK

The term "Neural Network" is a very evocative one. This is a backbone of Deep Learning. It suggests machines that are something like brains.

Here neuron is a mathematical function designed to imitate the functioning of a biological neuron.

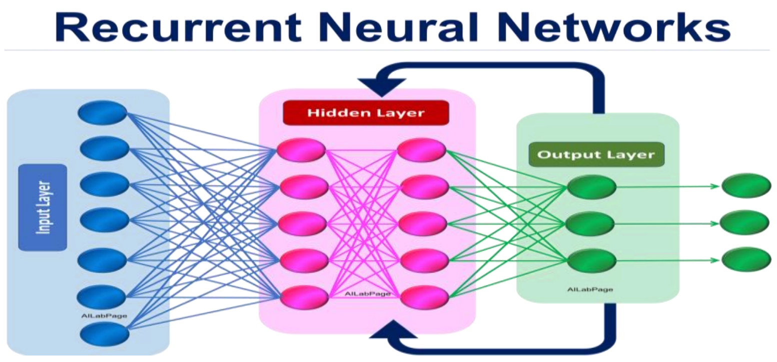


A simple neural network have input layer, hidden layer and output layer, Here, the second output have no any dependency upon first output. Here the learning process starts from scratch again and again.

Take an example,

Humans don't start their thinking from scratch every second, As we read any essay, we understand each word based on our understanding of previous words. We don't throw everything away and start thinking from scratch again. Our thoughts have persistence. Simple neural Network can't do this.

So we introduce some advanced one, named 'Recurrent Neural Network'.



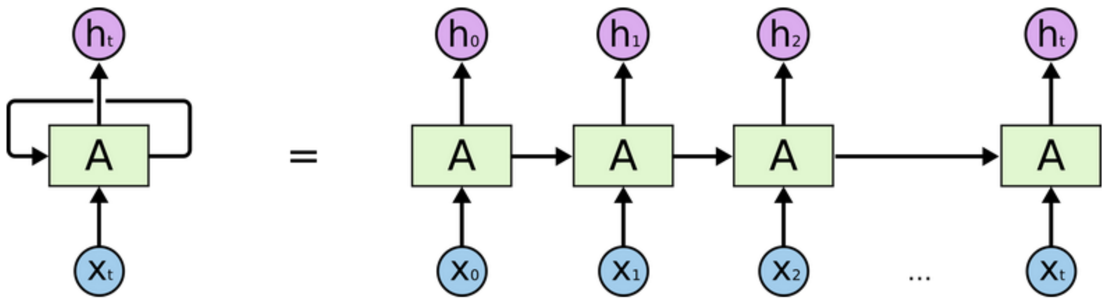
Recurrent Neural Netwoks address the failing issue of simple neural network. They are networks with loops in them, allowing information to persist.

This loop allows information to be passed from one step of the network to the next. So, here the second output have dependency upon first output.

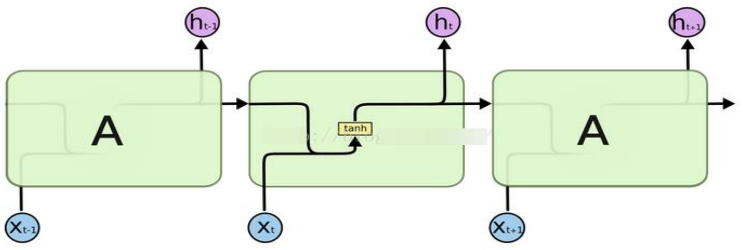
These loops make recurrent neural networks seem kind of mysterious. A recurrent neural network can be thought of as multiple copies of the same network can be thought of as multiple copies of the same network, each passing a message to a successor.

This RNN have some memory to store some previous memory this is a short term memory.

If we unroll the loop of Recurrent Neural Network it's look like



This chain like nature reveals that recurrent neural networks are intimately related to sequences and lists.

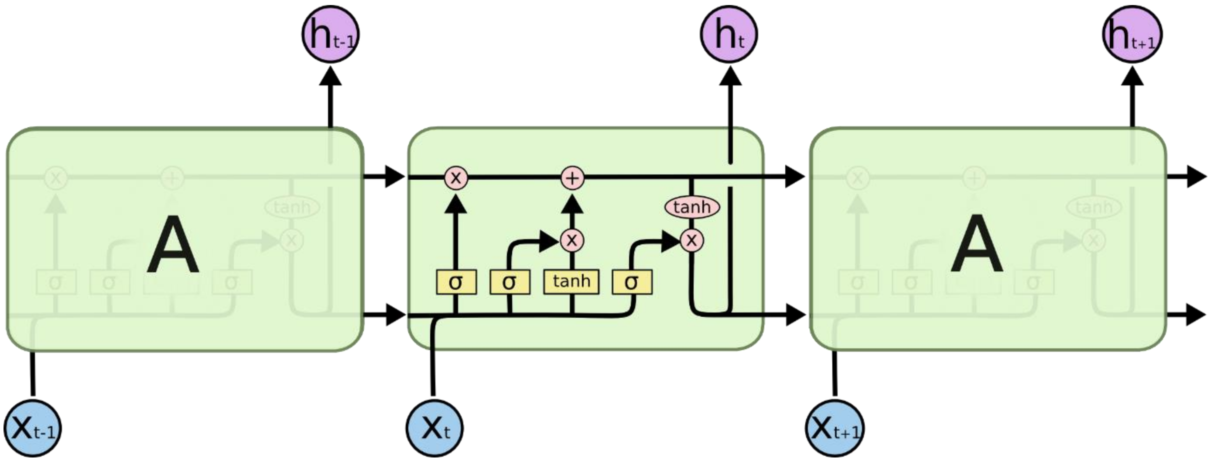


In standard RNN, this repeating module will have a very simple structure, such as single "tan h" layer, tan h gives the weight age to data on -1 to 1.

Model 2:

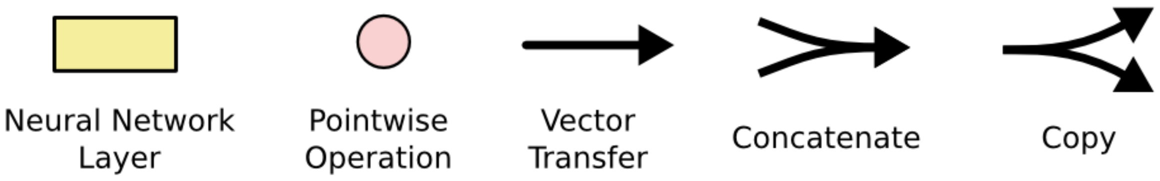
Long –Short Term Memory network (LSTM)

Long Short Term Memory networks – usually just called “LSTMs”. It’s a special kind of RNN, which is capable of learning long-term dependencies. It was introduced by Hochreiter & Schmidhuber in 1997. LSTMs are explicitly designed to avoid the long-term dependency problem. LSTMs also have 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.



The repeating module in an LSTM contains four interacting layers.

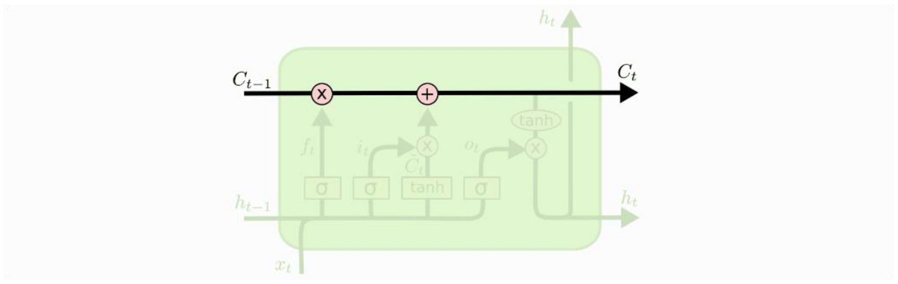
These are some notation we’ll use:



In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

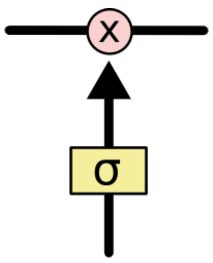
The Core Idea Behind LSTMs:

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It’s very easy for information to just flow along it unchanged.



The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

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

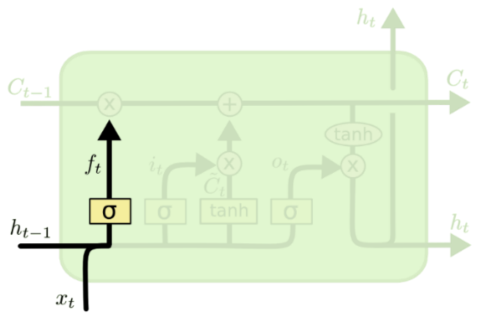


The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”

An LSTM has three of these gates, to protect and control the cell state.

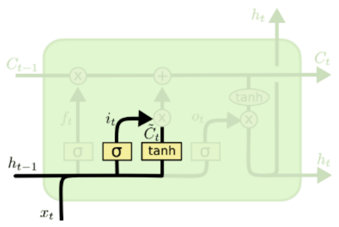
Step-by-Step LSTM Walk Through:

The first step in our LSTM is to decide what information we’re going to throw away from the cell state. This decision is made by a sigmoid layer called the “forget gate layer.” It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. A 1 represents “completely keep this” while a 0 represents “completely get rid of this.”



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

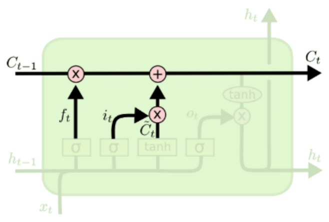
The next step is to decide what new information we’re going to store in the cell state. This has two parts. First, a sigmoid layer called the “input gate layer” decides which values we’ll update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state. In the next step, we’ll combine these two to create an update to the state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

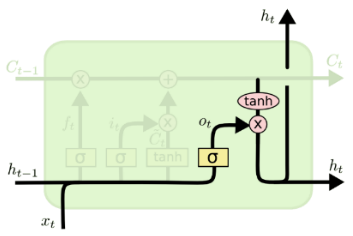
It’s now time to update the old cell state, C_{t-1}, into the new cell state C_t. The previous steps already decided what to do, we just need to actually do it.

We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then we add $i_t * \tilde{C}_t$. This is the new candidate values, scaled by how much we decided to update each state value.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

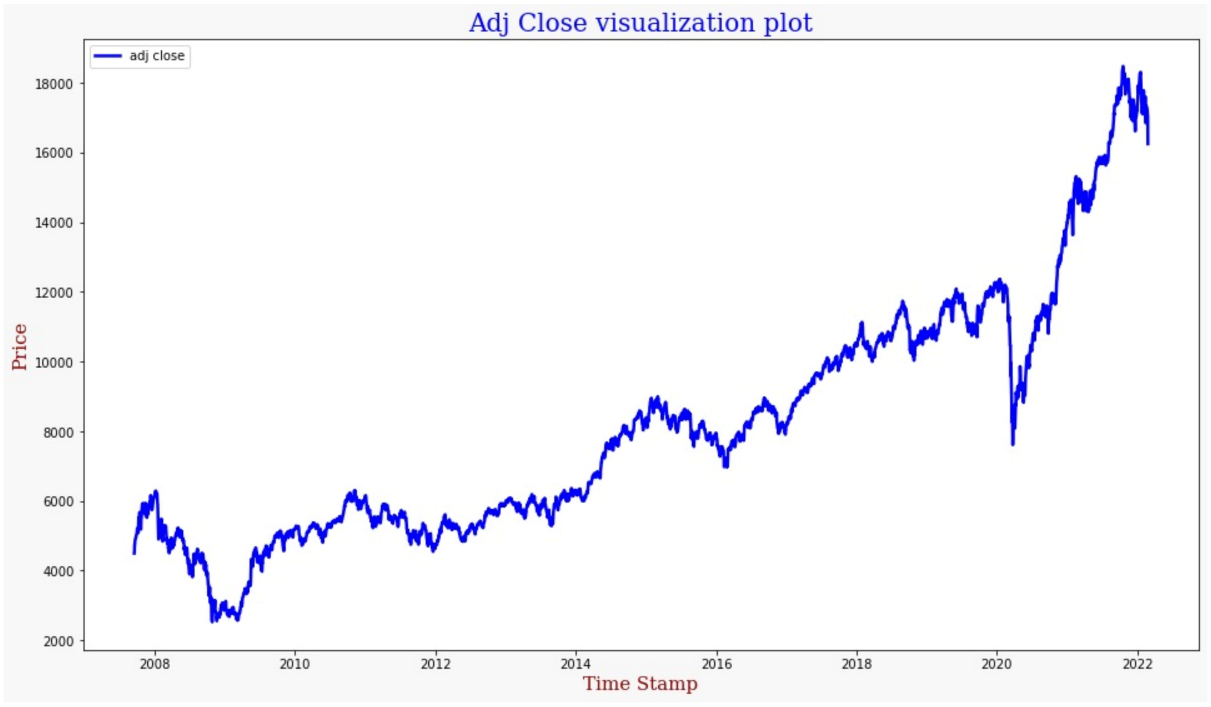
Finally, we need to decide what we’re going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we’re going to output. Then, we put the cell state through tanh (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

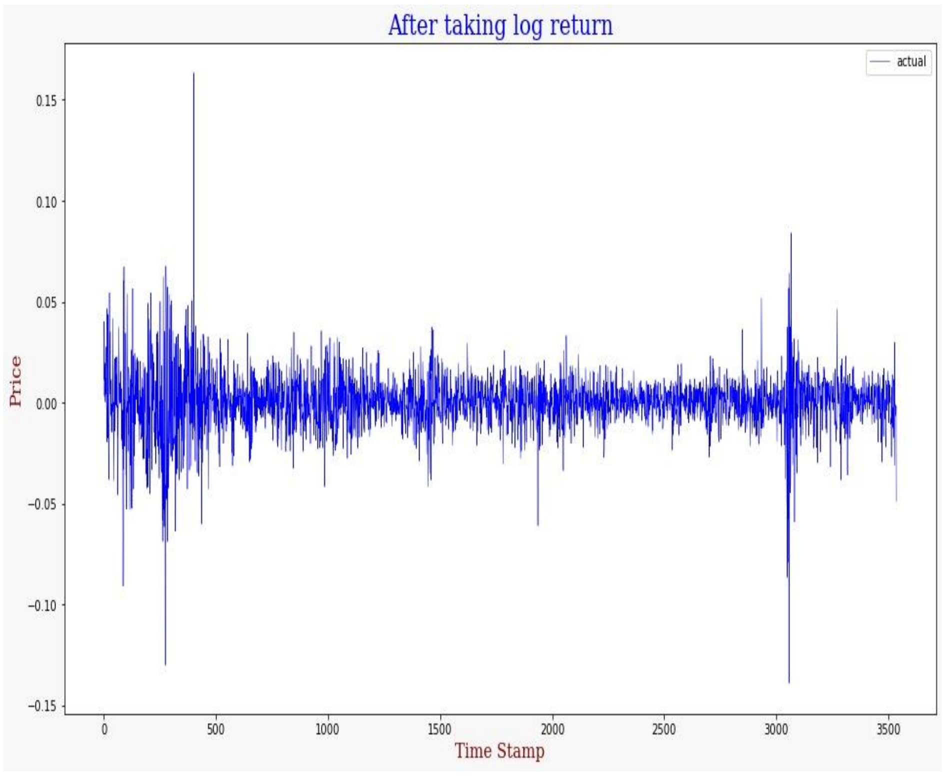
Findings:

Data visualization:



Model 1 (ARIMA):

Stationary data visualization:



ADF Test Result:

ADF test on original value: p- value: 0.978

ADF test value after taking log return: p- value: < 0.01

This is strong evidence against the null hypothesis, reject the null hypothesis. Data has no unit root and is stationary

SELECTION OF MODEL:

So after comparing AIC values of the models, ARIMA(3,1,2) is the best model.

ARIMA (p,d,q)	**AIC**
ARIMA(2,2,4)	-14987.0577
ARIMA(2,3,0)	-12631.4844
ARIMA(3,1,0)	-15022.9812
ARIMA(3,1,1)	-15025.2543
ARIMA(3,1,2)	-15041.7477
ARIMA(3,1,3)	-15041.2042
ARIMA(3,2,2)	-14988.1116

LJUNG BOX TEST

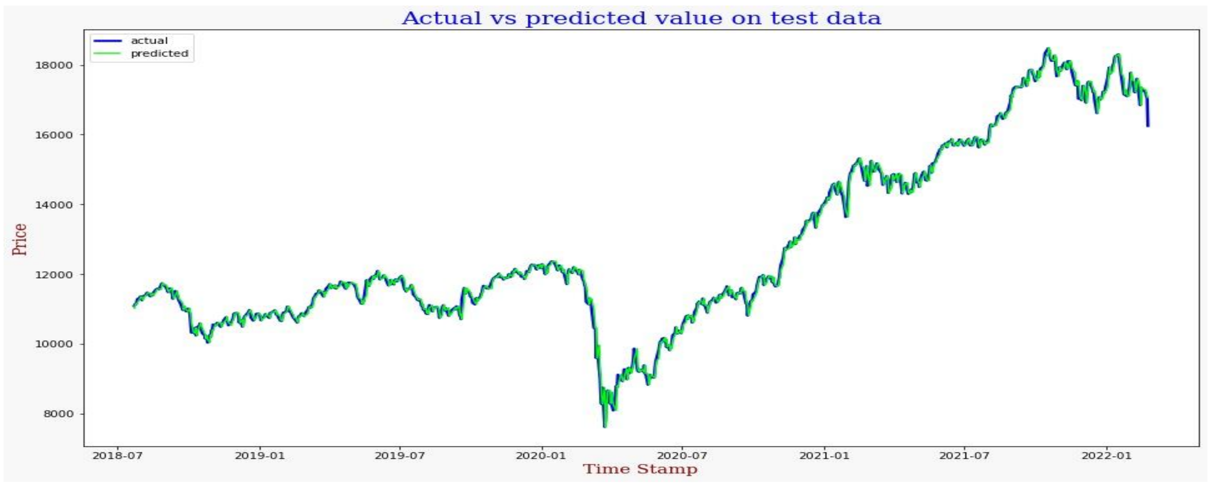
The test examines whether the residuals are random or not.

H_0 : The residuals are independently distributed.

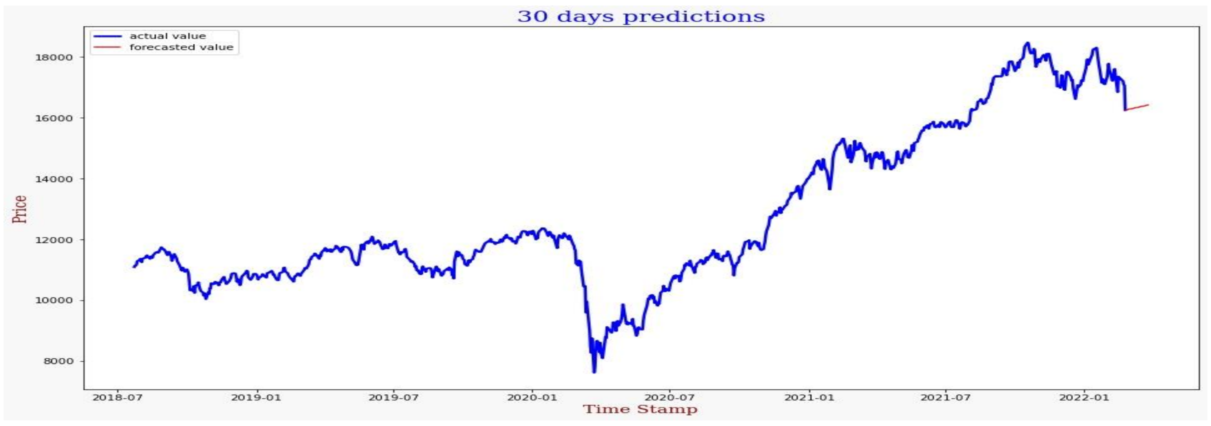
H_1 : The residuals are not independently distributed; they exhibit serial correlation.

Lag	**Test statistics**	**p-value**	**Decision**
1	0.011168	0.915837	Independent
2	2.07808064	0.353797	independent

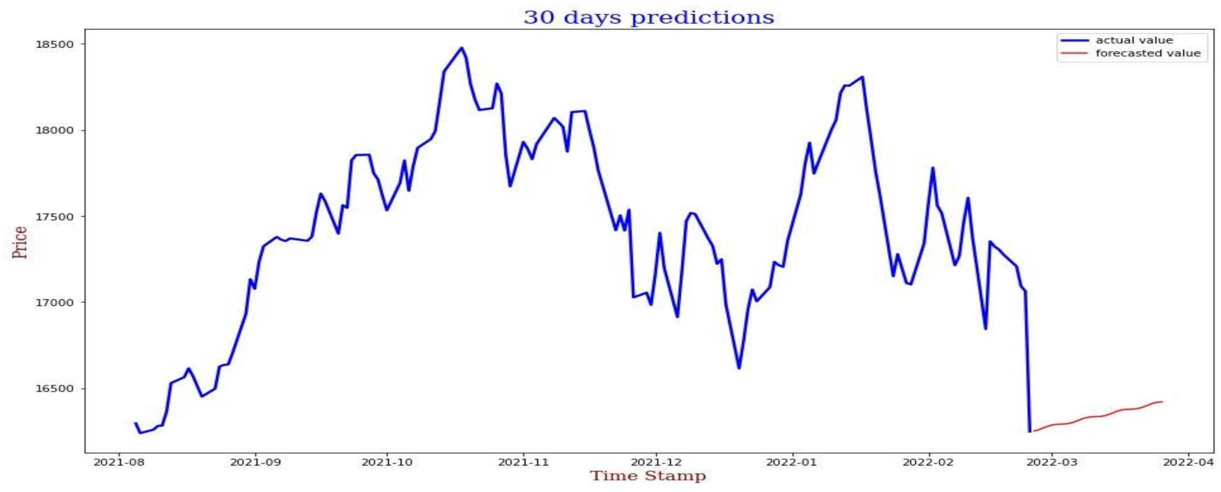
COMPARISION OF ACTUAL TEST DATA AND PREDICTED VALUE ON TEST DATA



FORECASTING

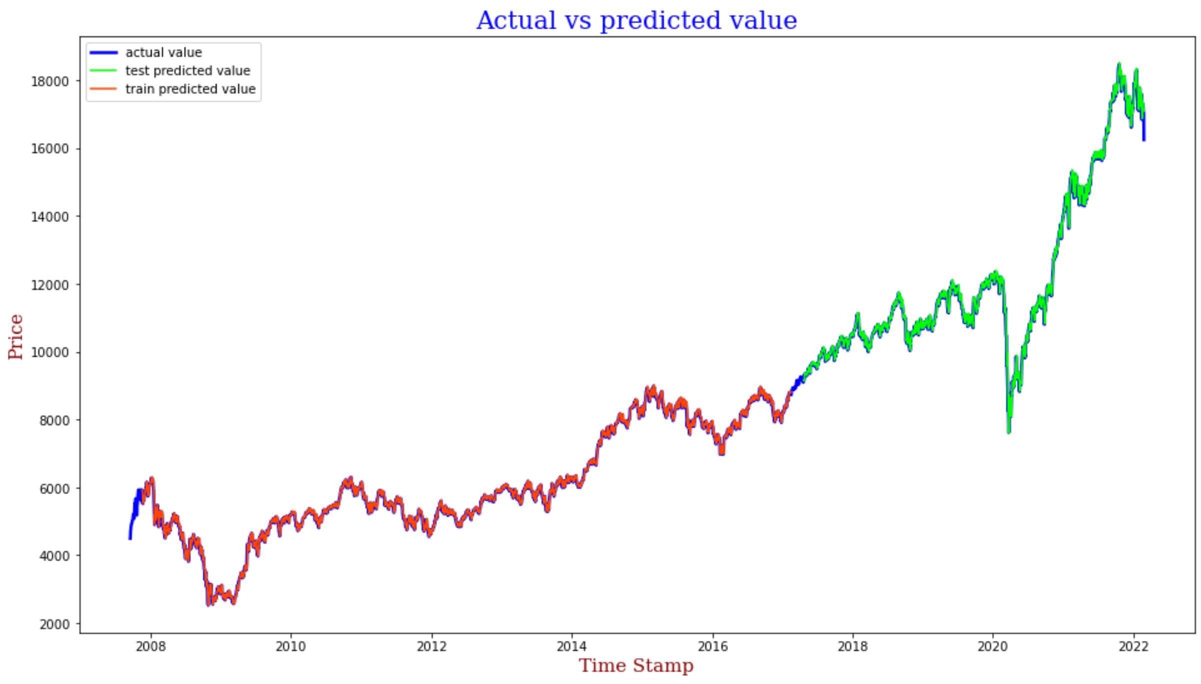


30 DAYS PREDICTION (ZOOM IN)

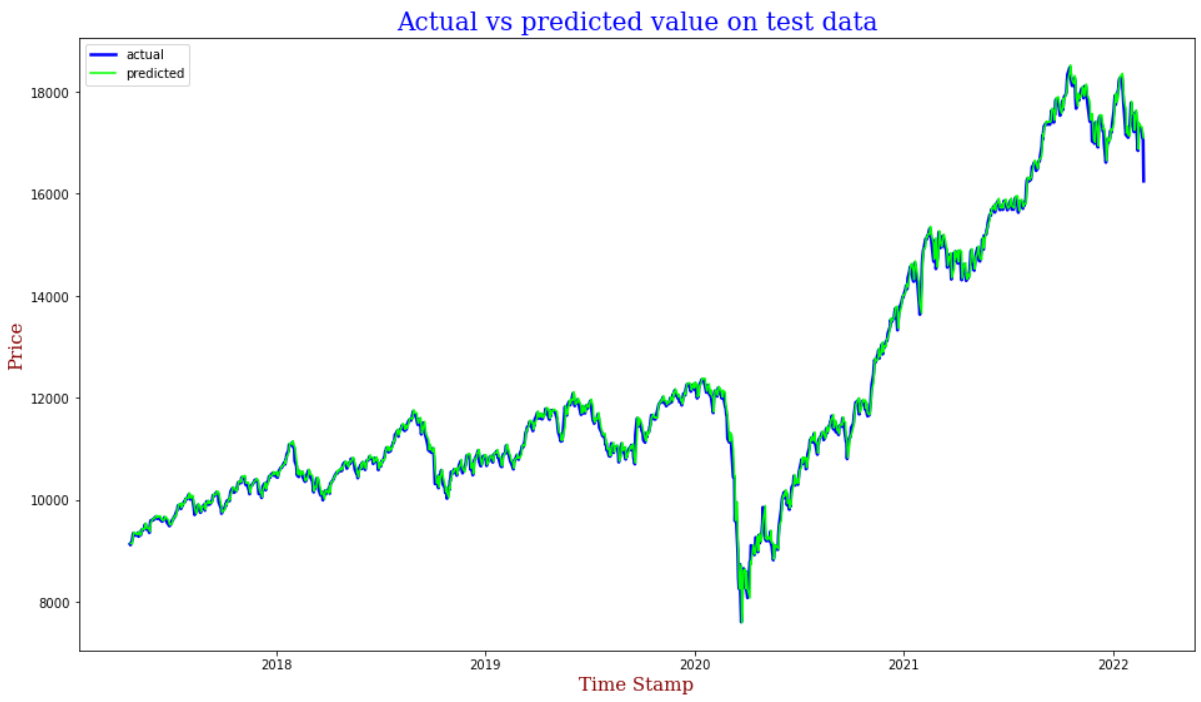


Model 2 (LSTM):

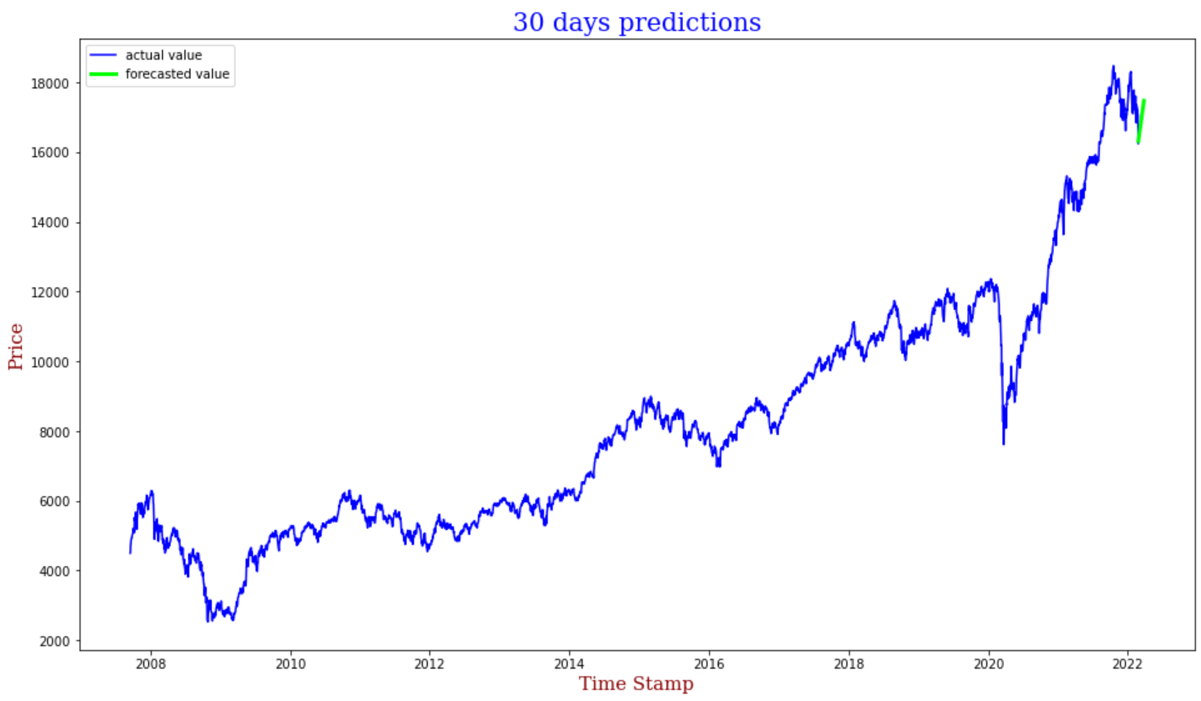
ACTUAL vs PREDICTED VALUE



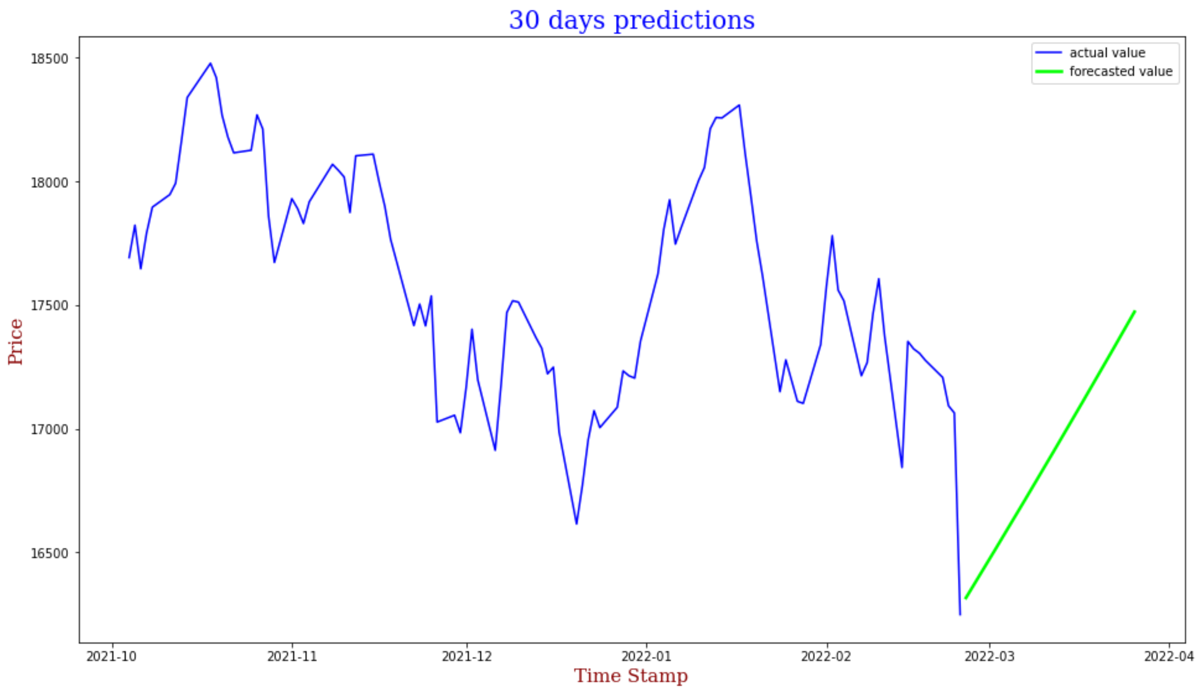
ACTUAL vs PREDICTED VALUE ON TEST DATA



30 DAYS PREDICTION



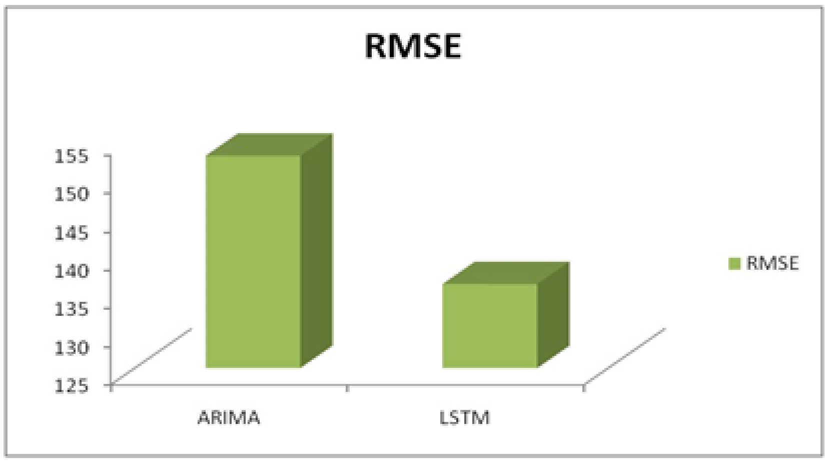
30 DAYS PREDICTION (ZOOM IN)



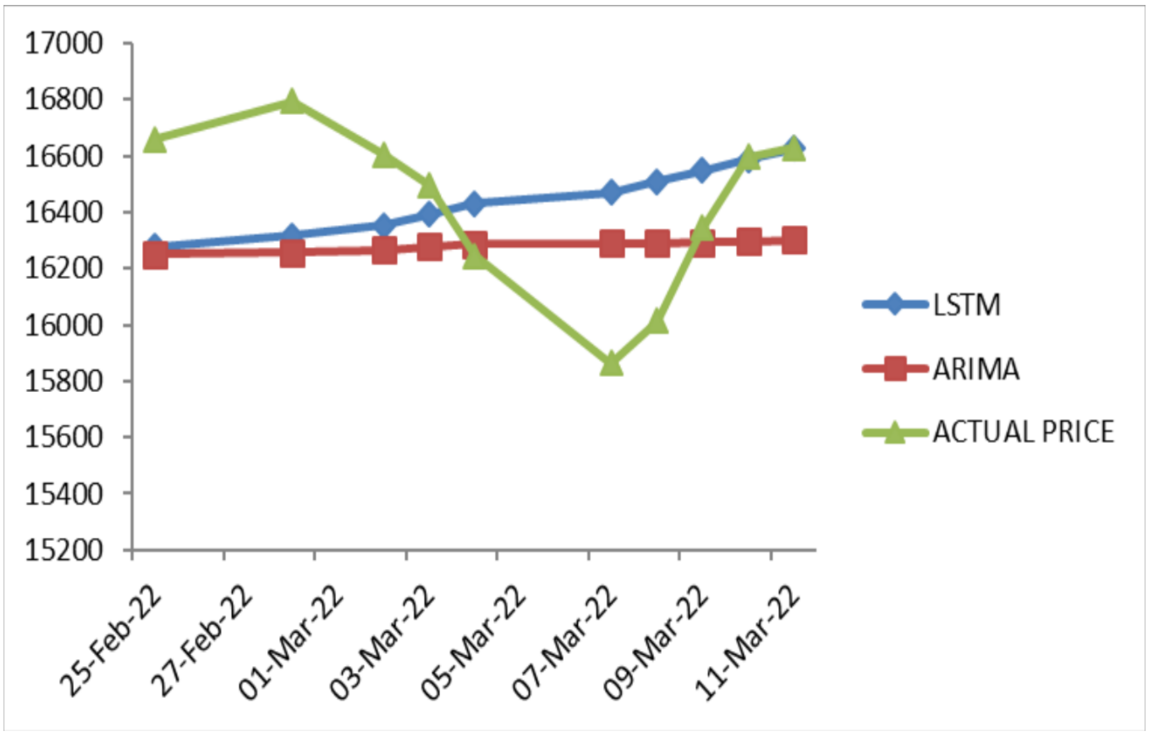
MODEL COMPARISION

RMSE :- Root Mean Square Error is a standard way to measure the error of a model in predicting quantitative data. RMSE indicates average model prediction error.

Lower RMSE indicates best performance of a model.



CONCLUSION



Based on performance 0f these two model on test data LSTM has less RMSE score than ARIMA.

LIMITATIONS

Market is highly volatile . the models may fail to provide good predictions in adverse scenario.

Due to lack of data other factors such as emotional and social factors are not incorporated in the model.

REFERENCE:

- DATA SOURCES - https://finance.yahoo.com/ (yahoo finance).
- colah's blog - https://colah.github.io/
- Robert H.Shumway, David S. Stoffer, Time Series Analysis and Its Applications, (Springer New York, 2000).
- Aurélien Géron, Hands-on Machine Learning with Scikit-Learn and TensorFlow, (O'Reilly Media, Inc., 2017).
- Rehan Guha, Machine Learning Cookbook with Python: Create ML and Data Analytics Projects Using Some Amazing Open Datasets, (BPB PUBN, 2020).