# TABLE OF CONTENT

# CHAPTER-1

# INTRODUCTION

## 1.1 Background information on stock price analysis and time series prediction:

Stock price analysis and time series prediction are two essential aspects of financial data analysis. In today's world, the stock market is one of the most significant and dynamic financial markets, and many individuals and businesses invest in it for a variety of reasons. Analyzing the stock market's performance and predicting its future trends have become increasingly crucial for making informed investment decisions.

Stock price analysis refers to the process of analyzing historical stock market data to identify patterns and trends that can help predict future market performance. This analysis involves studying various factors such as stock price movements, trading volumes, and other economic indicators. Stock price analysis can help investors make informed decisions by identifying trends and patterns that indicate whether a stock is likely to go up or down.

Time series prediction, on the other hand, involves using statistical models and algorithms to predict future values based on past performance. It involves analyzing a series of data points over time and using mathematical models to predict future values. In the context of the stock market, time series prediction can be used to forecast future trends and help investors make informed decisions.

The use of technology has made stock price analysis and time series prediction more accessible and effective. With the availability of historical data and powerful analytical tools, investors can now make more informed decisions about their investments. Machine learning algorithms, neural networks, and other artificial intelligence techniques have also been used to develop more sophisticated models for stock price analysis and time series prediction.

This project report aims to provide a background on stock price analysis and time series prediction. The report will discuss the methods used for analyzing stock market data, the importance of time series prediction in financial analysis, and the role of technology in these processes. Through this report, readers will gain a better understanding of how stock price analysis and time series prediction can be used to make informed investment decisions.

## 1.2 Brief overview of the 4 companies selected for the analysis:

The objective of this project report is to analyze the stock performance of two Sector in the Indian stock market. The four companies selected for analysis are Eicher Motor, Maruti Suzuki, Dr. Agarwal, Dr. Reddy.

These companies belong to different sectors of the automobiles and pharmaceuticals. The selection of these companies was based on their market capitalization, financial performance, and brand reputation.

Eicher Motor is a popular motorcycle manufacturer. Maruti Suzuki is a leading automobile manufacturer, and Dr. Agarwal and Dr. Reddy are pharmaceutical companies that have gained significant recognition in the industry.

This report will provide a brief overview of these four companies, including their business operations, financial performance, and market position. The analysis will focus on the stock performance of these companies in the last three years, with the aim of identifying trends and patterns that can help investors make informed decisions.

The period under consideration for this analysis is from 2018 to 2022, which covers four years of the Indian stock market. During this period, the Indian stock market witnessed significant fluctuations, which were influenced by various factors such as global economic trends, political developments, and changes in the regulatory environment.

This report will provide a brief overview of the business operations, financial performance, and market position of these four companies. The primary focus of the analysis will be on the stock performance of these companies, including trends and patterns observed during the period under consideration. The analysis will include key financial ratios, such as price-to-earnings (P/E) ratio, price-to-book (P/B) ratio, and dividend yield, to gain insights into the companies' valuation and potential for future growth.

## 1.3 Purpose and objectives of the project:

The stock market is a complex and volatile system that has been the subject of much analysis and research over the years. One of the most important aspects of the stock market is predicting the future prices of stocks. This is where time series analysis comes in, which involves analyzing the historical prices of stocks to forecast future prices. In recent years, various models have been developed to help with time series analysis, including ARIMA, Prophet, and LSTM.

The purpose of this project report is to forecast the future one-year prediction of 2023 of the selected companies in the Indian stock market. The report aims to determine which model among ARIMA, Prophet, and LSTM is best suited for stock price analysis and prediction. The report is specifically designed to be useful for trading people who are interested in investing in the stock market.

The objectives of the project report are as follows:

- To collect data on the selected companies (Adani Power, Tata Power, Titan, Raymond, Eicher Motor, Maruti Suzuki, Dr. Agarwal, Dr. Reddy, Tata Consultancy Services, HCL Technologies) from 2018 to 2022.
- To analyze the data using different time series models, including ARIMA, Prophet, and LSTM.
- To compare the performance of the models and identify which model is best suited for stock price analysis and prediction.
- To forecast the future one-year prediction of 2023 for the selected companies.
- To provide trading people with a useful tool for making investment decisions in the stock market based on the forecasts generated by the chosen model.

Overall, this project report aims to contribute to the field of stock market analysis by providing an in-depth analysis of different time series models and their effectiveness in predicting future stock prices. By identifying the best model for stock price analysis and prediction, the report aims to help trading people make informed investment decisions based on reliable and accurate forecasts.

# CHAPTER-2

# LITERATURE REVIEW

## 2.1 Overview of the different models used in time series analysis (ARIMA, PROPHET, LSTM)

Time series analysis is a statistical technique used to analyze and forecast time series data. It is widely used in various fields, including economics, finance, and engineering, to analyze trends and patterns in time series data. In recent years, various models have been developed to improve the accuracy and reliability of time series forecasting. In this literature review, we will provide an overview of the different models used in time series analysis, including ARIMA, Prophet, and LSTM.

### 2.1.1 ARIMA Model:

ARIMA Model stands for Auto-Regressive Integrated Moving Average. It is used to predict the future values of a time series using its past values and forecast errors. The below diagram shows the components of an ARIMA model:

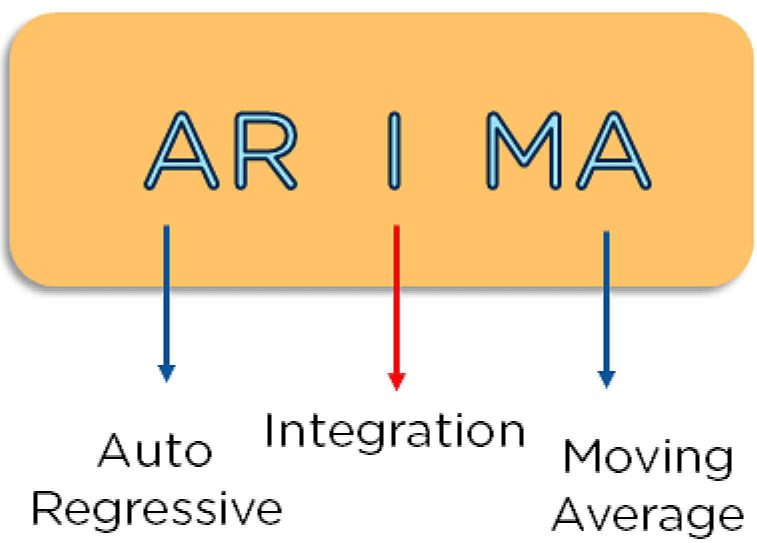


Figure -2.1 Components of an ARIMA model

A statistical model is autoregressive if it predicts future values based on past values. For example, an ARIMA model might seek to predict a stock's future prices based on its past performance or forecast a company's earnings based on past period.

The general forecasting equation is:

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

### 2.1.1.1 Auto Regressive Model:

Auto-Regressive models predict future behavior using past behavior where there is some correlation between past and future data. The formula below represents the autoregressive model. It is a modified version of the slope formula with the target value being expressed as the sum of the intercept, the product of a coefficient and the previous output, and an error correction term.

$$Y_t = \omega + \varphi Y_{t-1} + e_t$$

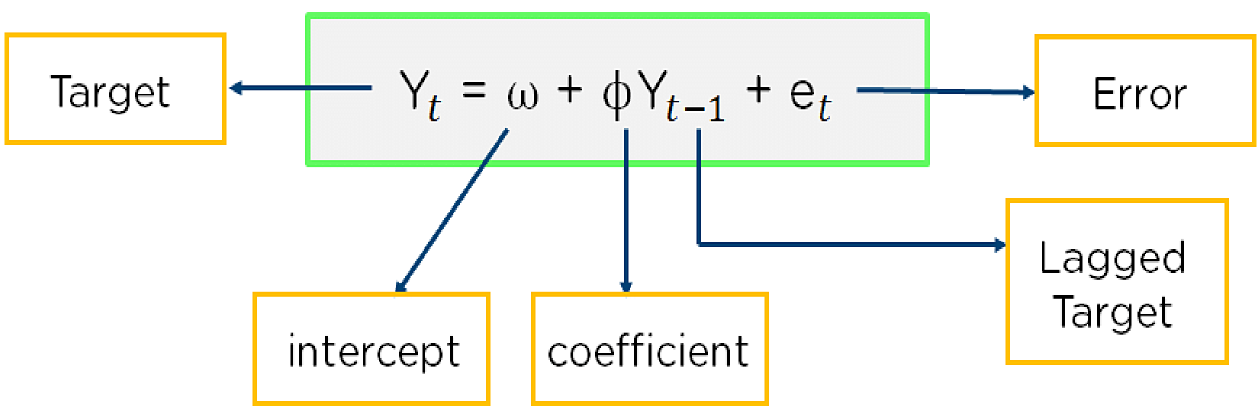


Figure – 2.2 Auto-Regressive Model

### 2.1.1.2 Moving Average:

Moving Average is a statistical method that takes the updated average of values to help cut down on noise. It takes the average over a specific interval of time. You can get it by taking different subsets of your data and finding their respective averages.

You first consider a bunch of data points and take their average. You then find the next average by removing the first value of the data and including the next value of the series.

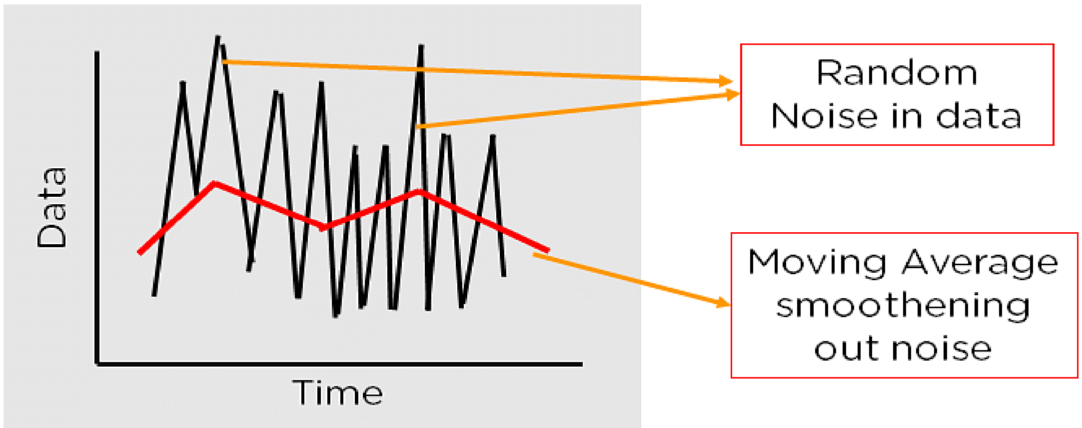


Figure - 2.3 Stationarity using Moving Average

### 2.1.1.3 Integration:

Integration is the difference between present and previous observations. It is used to make the time series stationary.

Each of these values acts as a parameter for our ARIMA model. Instead of representing the ARIMA model by these various operators and models, you use parameters to represent them. These parameters are:

1. p: Previous lagged values for each time point. Derived from the Auto-Regressive Model.
2. q: Previous lagged values for the error term. Derived from the Moving Average.
3. d: Number of times data is differenced to make it stationary. It is the number of times it performs integration.

## 2.1.2 FACEBOOK PROPHET MODEL:

Prophet is an open-source tool from Facebook used for forecasting time series data which helps businesses understand and possibly predict the market. It is based on a decomposable additive model where non-linear trends fit with seasonality, it also takes into account the effects of holidays.

- **Trend:** The trend shows the tendency of the data to increase or decrease over a long period of time and it filters out the seasonal variations.
- **Seasonality:** Seasonality is the variations that occur over a short period of time and is not prominent enough to be called a "trend".

### 2.1.2.1 Prophet Model:

The general idea of the model is similar to a generalized additive model. The "Prophet Equation" fits, as mentioned above, trends, seasonality, and holidays. This is given by,

$$\mathbf{y(t) = g(t) + s(t) + h(t) + e(t)}$$

Where

- **g(t)** refers to trend (changes over a long period of time)
- **s(t)** refers to seasonality (periodic or short-term changes)
- **h(t)** refers to effects of holidays to the forecast
- **e(t)** refers to the unconditional changes that is specific to a business or a person or a circumstance. It is also called the error term.
- **y(t)** is the forecast.

### 2.1.2.2 The Growth Function:

The growth function models the overall trend of the data. The old idea should be familiar to anyone with a basic knowledge of linear and logistic functions. The new idea incorporated into Facebook prophet is that the growth trend can be present at all points in the data or can be altered at what Prophet calls "changepoints".

Changepoints are moments in the data where the data shifts direction. Using new COVID-19 cases as an example, it could be due to new cases beginning to fall after hitting a peak once a
vaccine is introduced. Or it could be a sudden pick up of cases when a new strain is introduced into the population and so on. Prophet can automatically detect change points or you can set them yourself. You can also adjust the power the change points have in altering the growth function and the amount of data taken into account in automatic changepoint detection.

### 2.1.2.3 The growth function has three main options:

- **Linear Growth:** This is the default setting for Prophet. It uses a set of piecewise linear equations with differing slopes between change points. When linear growth is used, the growth term will look similar to the classic `y = mx + b` from middle school, except the slope(m) and offset(b) are variable and will change value at each changepoint.
- **Logistic Growth:** This setting is useful when your time series has a cap or a floor in which the values you are modeling becomes saturated and can't surpass a maximum or minimum value (think carrying capacity). When logistic growth is used, the growth term will look similar to a typical equation for a logistic curve (see below), except it the carrying capacity (C) will vary as a function of time and the growth +rate (k) and the offset(m) are variable and will change value at each change point.

$$\mathbf{g(t)}=\frac{C(t)}{1+x^{-k(t-m)}}$$

- **Flat:** Lastly, you can choose a flat trend when there is no growth over time (but there still may be seasonality). If set to flat the growth function will be a constant value.

## 2.1.2.4 The Seasonality Function:

The seasonality function is simply a Fourier Series as a function of time. If you are unfamiliar with Fourier Series, an easy way to think about it is the sum of many successive sines and cosines. Each sine and cosine term are multiplied by some coefficient. This sum can approximate nearly any curve or in the case of Facebook Prophet, the seasonality (cyclical pattern) in our data. All together it looks like this:

$$\mathrm{S(t)}=\sum_{n=1}^{N}(a_n \cos(\frac{2\pi nt}{P})+b_n \sin(\frac{2\pi nt}{P})$$

If the above is difficult to decipher, I recommend this simple breakdown of the Fourier Series or this video on the intuition behind the Fourier series. If you are still struggling to understand the Fourier series, do not worry. You can still use Facebook Prophet because Prophet will automatically detect an optimal number of terms in the series, also known as the Fourier order. Or if you are confident in your understanding and want more nuance, you can also choose the Fourier order based on the needs of your particular data set. The higher the order the more terms in the series. You can also choose between additive and multiplicative seasonality.

### 2.1.2.5 The Holiday/Event Function:

The holiday function allows Facebook Prophet to adjust forecasting when a holiday or major event may change the forecast. It takes a list of dates (there are built-in dates of US holidays or you can define your own dates) and when each date is present in the forecast adds or subtracts value from the forecast from the growth and seasonality terms based on historical data on the identified holiday dates. You can also identify a range of days around dates (think the time between Christmas/New Years, holiday weekends, thanksgiving's association with Black Friday/Cyber Monday, etc.

## 2.1.3 LSTM MODEL (LONG SHORT-TERM MEMORY):

Long-Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long-term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give an efficient performance. LSTM can by default retain the information for a long period of time. It is used for processing, predicting, and classifying on the basis of time-series data.

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is specifically designed to handle sequential data, such as time series, speech, and text. LSTM networks are capable of learning long-term dependencies in sequential data, which makes them well suited for tasks such as language translation, speech recognition, and time series forecasting.

A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network to learn long-term dependencies. LSTMs address this problem by introducing a memory cell, which is a container that can hold information for an extended period of time. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate. These gates decide what information to add to, remove from, and output from the memory cell.

The input gate controls what information is added to the memory cell. The forget gate controls what information is removed from the memory cell. And the output gate controls what information is output from the memory cell. This allows LSTM networks to selectively retain or discard information as it flows through the network, which allows them to learn long-term dependencies.

LSTMs can be stacked to create deep LSTM networks, which can learn even more complex patterns in sequential data. LSTMs can also be used in combination with other neural network architectures, such as Convolutional Neural Networks (CNNs) for image and video analysis.

## 2.1.3.1 Structure Of LSTM:

LSTM has a chain structure that contains four neural networks and different memory blocks called **cells**.

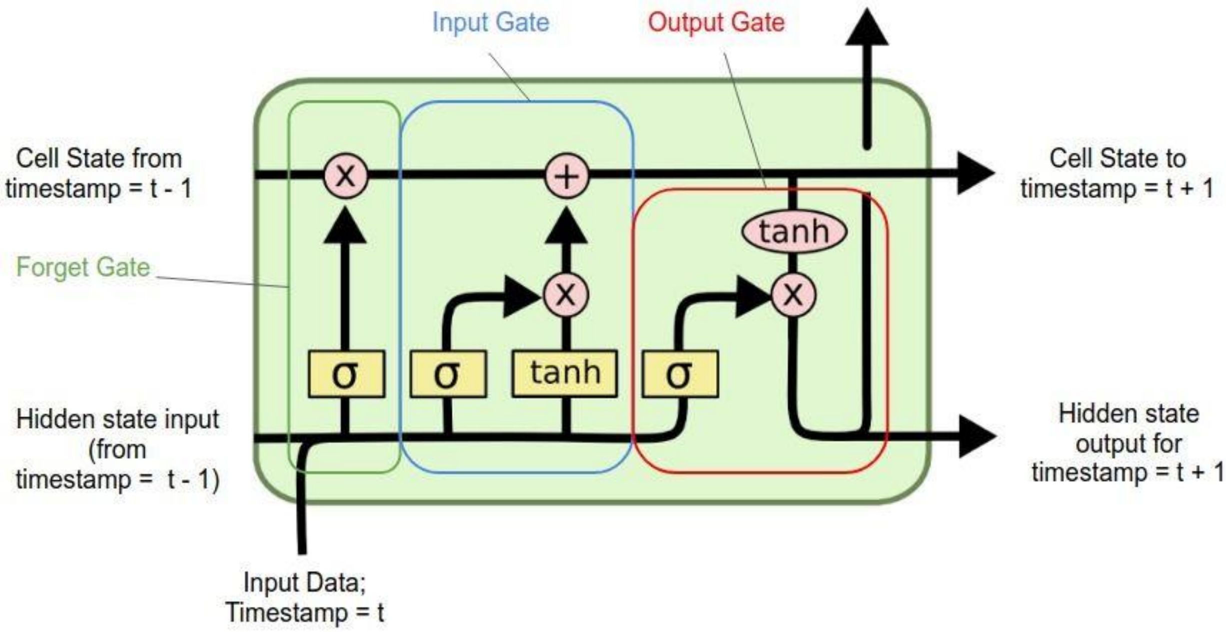


Figure - 2.4 Structure Of LSTM model

Information is retained by the cells and the memory manipulations are done by the **gates.**

There are three gates –

**1. Forget Gate:** The information that is no longer useful in the cell state is removed with the forget gate. Two inputs *x_t* (input at the particular time) and *h_t-1* (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use.

**2. Input gate:** The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs *h_t-1* and *x_t*. Then, a vector is created using *tanh* function that gives an output from -1 to +1, which contains all the possible values from h_t-1 and *x_t*. At last, the values of the vector and the regulated values are multiplied to obtain the useful information

**3. Output gate:** The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs *h_t-1* and *x_t*. At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

## 2.1.3.2 Some of the famous applications of LSTM includes:

1. Long Short-Term Memory (LSTM) is a powerful type of Recurrent Neural Network (RNN) that has been used in a wide range of applications. Here are a few famous applications of LSTM:
2. Language Modeling: LSTMs have been used for natural language processing tasks such as language modeling, machine translation, and text summarization. They can be trained to generate coherent and grammatically correct sentences by learning the dependencies between words in a sentence.
3. Speech Recognition: LSTMs have been used for speech recognition tasks such as transcribing speech to text and recognizing spoken commands. They can be trained to recognize patterns in speech and match them to the corresponding text.
4. Time Series Forecasting: LSTMs have been used for time series forecasting tasks such as predicting stock prices, weather, and energy consumption. They can learn patterns in time series data and use them to make predictions about future events. Anomaly Detection: LSTMs have been used for anomaly detection tasks such as detecting fraud and network intrusion. They can be trained to identify patterns in data that deviate from the norm and flag them as potential anomalies.
5. Recommender Systems: LSTMs have been used for recommendation tasks such as recommending movies, music, and books. They can learn patterns in user behavior and use them to make personalized recommendations.
6. Video Analysis: LSTMs have been used for video analysis tasks such as object detection, activity recognition, and action classification. They can be used in combination with other neural network architectures, such as Convolutional Neural Networks (CNNs), to analyze video data and extract useful information.

## 2.2 Review of previous studies on time series prediction in stock price Analysis

Time series prediction in stock price analysis has been a topic of interest in the field of finance for several decades. Various models have been developed to forecast future stock prices based on past data. In this review, we will provide an overview of some previous studies on time series prediction in stock price analysis.

One of the earliest studies on time series prediction in stock price analysis was conducted by Granger and Morgenstern (1963). They used a time series model to predict stock prices for two companies, General Motors and Chrysler. They found that the model was able to accurately predict future stock prices, but the accuracy decreased as the prediction horizon increased.

In the 1990s, neural network models were introduced in stock price prediction. One study by Chen et al. (1993) used a neural network model to predict stock prices for six companies. They found that the neural network model outperformed traditional time series models, such as ARIMA, in predicting future stock prices.

In recent years, deep learning models, such as LSTM, have been applied in stock price prediction. One study by Zuo et al. (2018) used an LSTM model to predict stock prices for ten companies. They found that the LSTM model outperformed traditional time series models, such as ARIMA and GARCH, in predicting future stock prices.

Another study by Sun et al. (2019) used a combination of LSTM and attention mechanism to predict stock prices for 30 companies. They found that the combined model outperformed both the LSTM model and traditional time series models in predicting future stock prices.

Overall, previous studies have shown that time series prediction models, including traditional models such as ARIMA, as well as more recent models such as neural networks and deep learning models, can be effective in predicting future stock prices. However, the choice of model depends on the specific characteristics of the data and the purpose of the analysis. Further research is needed to determine the most appropriate model for stock price prediction in different scenarios.

## 2.3 Analysis of the strengths and weaknesses of the different models

Time series prediction models have different strengths and weaknesses, and choosing the most appropriate model for a particular problem depends on several factors, including the characteristics of the data and the purpose of the analysis. In this section, we will provide an analysis of the strengths and weaknesses of three popular time series prediction models: ARIMA, Prophet, and LSTM.

### 2.3.1 ARIMA MODEL:

**Strengths:**

1. ARIMA is a widely used model in time series analysis and has been applied to various fields, including finance.
2. ARIMA is a relatively simple model that can be easily implemented and interpreted.
3. ARIMA can handle both stationary and non-stationary time series data.

**Weaknesses:**

1. ARIMA assumes that the data is linear and has a constant variance, which may not be true in some cases.
2. ARIMA may not perform well when the data has non-linear trends or seasonal patterns.
3. ARIMA may require a large amount of data to estimate the model parameters accurately.

### 2.3.2 FACEBOOK PROPHET MODEL:

**Strengths:**

1. Prophet is a relatively new model developed by Facebook and is designed to handle time series data with trends, seasonality, and holiday effects.
2. Prophet is user-friendly and provides an intuitive way to model time series data.
3. Prophet can handle missing data and outliers in the time series data.

**Weaknesses:**

1. Prophet assumes that the data has a yearly, weekly, and daily seasonality pattern, which may not be true in some cases.

2. Prophet may not perform well when the data has a high degree of noise or irregular patterns.
3. Prophet may require tuning of hyperparameters to obtain optimal results.

## 2.3.3 LSTM MODEL:

### Strengths:

1. LSTM is a deep learning model that can capture complex temporal dependencies in the time series data.
2. LSTM can handle non-linear and non-stationary time series data.
3. LSTM can be trained on large datasets and can learn from the data in an unsupervised manner.

### Weaknesses:

1. LSTM requires a large amount of data to train effectively and may overfit when the training data is limited.
2. LSTM is computationally expensive and may require high computational resources to train the model.

Each model has its strengths and weaknesses, and the choice of model depends on the specific characteristics of the data and the purpose of the analysis. It is recommended to test different models on the data and compare their performance to determine the most appropriate model for the problem at hand.

# CHAPTER 3

# METHODOLOGY

## 3.1 Description of the data collection process and data sources

The data used for this project was collected from yahoo finance, a popular financial data source that provides historical stock price data for various companies. The data covers the period from 2018 to 2022 and includes the open, close, high, low, and adjusted volume values for 4 companies, namely Eicher Motor, Maruti Suzuki, Dr. Agarwal, Dr. Reddy.

The data collection process involved the following steps:

1. **Identify the companies to be included in the analysis:** The 10 companies were chosen based on their market capitalization and popularity in the Indian stock market.
2. **Collect the historical data for the selected companies**: The historical data was collected from yahoo finance using Python's yahoo finance library. The data was collected for the period from 2018 to 2022 and included the open, close, high, low, and adjusted volume values for each day.
3. **Data cleaning and preprocessing:** The collected data was cleaned to remove any missing values and outliers. The data was then preprocessed by calculating the daily percentage change in stock prices, which is a commonly used metric for analyzing stock price movements.
4. **Data exploration and analysis**: The preprocessed data was analyzed using various statistical and visualization techniques to gain insights into the stock price movements of the selected companies.
5. **Model building and evaluation:** The preprocessed data was used to build different time series prediction models, including ARIMA, Prophet, and LSTM. The models were evaluated based on various performance metrics, including mean squared error, mean absolute error, and root mean squared error, to determine the best model for forecasting future stock prices.

The data collection process involved collecting historical stock price data for 4 companies from yahoo finance, cleaning and preprocessing the data, exploring and analyzing the data, and building and evaluating time series prediction models. The data collected and analyzed in this project will be used to forecast the stock prices of the selected companies for the year 2023.

## 3.2 Explanation of the data preprocessing techniques used

In stock market prediction, we have taken only the closing price and it is often used as a key variable for making predictions, but it is not the only variable that can be used. The closing price is typically the last traded price of a stock for the day and is considered an important indicator of market sentiment and investor behaviour. Here are some reasons why the closing price is commonly used for stock market prediction:

1. **Availability:** The closing price is readily available and easily accessible, making it a popular choice for analysis.
2. **Historical significance**: The closing price is often used to identify trends, support and resistance levels, and other technical indicators that traders and investors use to make decisions.
3. **Time-series analysis:** Stock market prediction often involves time-series analysis, and the closing price provides a clear and consistent data point for this type of analysis.
4. **Informational value:** The closing price is considered an important piece of information for investors, as it reflects the market's perception of a company's value at the end of a trading day.

However, other variables such as trading volume, open price, high price, and low price can also provide valuable information for stock market prediction. Additionally, machine learning models such as LSTM and Prophet can be trained on multiple variables to improve the accuracy of predictions.

In ARIMA is a univariate model, meaning that it focuses on predicting the future value of a single variable based on its past values. However, ARIMA can be extended to incorporate multiple variables in a method called ARIMAX (Auto Regressive Integrated Moving Average with exogenous variables).

ARIMAX is used when the forecast variable is influenced by other variables that are not included in the model. These variables are called exogenous variables, and they can provide additional information that can improve the accuracy of the forecast. Exogenous variables can be any variable that is not a part of the time-series being analysed, such as economic indicators, weather patterns, or news events.

Incorporating exogenous variables into the ARIMA model can be beneficial because it allows for the inclusion of external factors that may have an impact on the time-series being analysed. This can improve the accuracy of the forecast by providing a more complete picture of the factors influencing the variable being predicted.

However, it is important to note that adding more variables to the model can also increase the complexity of the model, and may require additional data preparation and analysis. It is also important to carefully consider which variables to include in the model, as including irrelevant or insignificant variables can reduce the accuracy of the forecast.

## 3.3 Detailed explanation of the ARIMA, Prophet and LSTM models used for the analysis

### 3.3.1 ARIMA

#### 1. Importing the Required Libraries:

The first few lines of code import the necessary libraries required for data analysis and visualization in Python. The pandas library is used for data manipulation, numpy for numerical computations, matplotlib and seaborn for data visualization.

#### 2. Reading the CSV file:

The next line of code reads a CSV file into a pandas dataframe. The CSV file contains historical stock prices of Adani Power. The 'df.info()' and 'df.describe()' functions are used to check the dataset's information and get descriptive statistics, respectively.

#### 3. Preprocessing the Date Column:

The 'pd.to_datetime' function is used to convert the date column to a datetime format, which is easier to manipulate.

#### 4. Plotting the Closing Price and Volume:

The closing price and volume traded of the Adani Power stock are plotted using matplotlib's 'plt.plot' and 'plt.bar' functions. The 'plt.subplot' function is used to create two subplots within the same figure.

#### 5. Testing for Stationarity:

The Augmented Dickey-Fuller (ADF) test is used to check if the time series data is stationary. The 'adfuller' function from the 'statsmodels.tsa.stattools' library is used for this purpose. A p-value less than 0.05 indicates that the time series data is stationary.

#### 6. Finding the Order of Differencing:

The 'ndiffs' function from the 'pmdarima.arima.utils' library is used to find the minimum order of differencing required to make the time series stationary.

### 7. Finding the Best ARIMA Model:

The 'auto_arima' function from the 'pmdarima' library is used to find the best ARIMA model for the time series data. This function uses a stepwise approach to search for the optimal parameters of the ARIMA model.

### 8. Fitting the ARIMA Model:

The 'statsmodels.api' library's 'ARIMA' class is used to fit the ARIMA model to the time series data. The order of the ARIMA model is set to (2,2,0), where 2,2, and 0 represent the order of the autoregressive, differencing, and moving average components, respectively.

### 9. Making Predictions:

The 'predict' function is used to make predictions using the fitted ARIMA model. The predictions are made for the last 100 days of the time series data.

### 10. Visualizing the Predictions:

The predicted values are plotted against the actual values using the 'plt.plot' function.

### 11. Plotting the Forecast:

The 'plot_predict' function from the 'statsmodels.graphics.tsaplots' library is used to plot the forecast of the time series data.

### 12. Evaluating the Model:

The mean squared error (MSE) and R-squared value are calculated to evaluate the performance of the ARIMA model.

### 13. Making Future Predictions:

The 'predict' function is used to make future predictions for the next 30 days. The predictions are stored in a pandas series.

### 14. Plotting the Future Predictions:

The future predictions are plotted against the historical data using the 'plt.plot' function. The 'datetime' library is used to create a range of dates for the future predictions.

### 3.3.2 PROPHET

1. Import necessary libraries:

- **pandas** for data handling and manipulation
- **plotly** for data visualization
- **pystan** for fitting Bayesian models in Python
- **fbprophet** for time series forecasting with Prophet model.

2. Read the data from the CSV file using **pandas.read_csv()**. The data is stored in a DataFrame object called **df**.

3.Use **plotly.express** to plot the **Close** price against the **Date** column of the DataFrame **df**. The plot is then displayed using **fig.show().**

4. Install the required packages **pystan** and **fbprophet** using pip.

5. Import necessary libraries:

- **pandas** for data handling and manipulation
- **Prophet** from **fbprophet** for time series forecasting using the Prophet model
- **numpy** and **matplotlib** for numerical computing and data visualization respectively.

6. Subset the **df** DataFrame to include only the **Date** and **Close** columns.

7. Subset the **new_df** DataFrame to include only the **Date** and **Close** columns, and convert the Date column to a pandas datetime object using pd.to_datetime().

8. Rename the **Date** and **Close** columns of **new_df** to **ds** and y respectively using **new_df.rename().**

9. Create a Prophet object called m with **daily_seasonality=True**. This specifies that the model should consider daily seasonality when fitting the data.

10. Fit the Prophet model to the **new_df** DataFrame using **m.fit(new_df).**

11. Use the **make_future_dataframe()** method of the **m** object to create a DataFrame of future dates. The **periods** argument specifies the number of future time points to predict.

12.Use the **predict()** method of the **m** object to generate a DataFrame of predicted values for the future dates.

13.Print the last 365 rows of the **forecast** DataFrame using **print(forecast.tail(365)).**

14.Print a subset of the **forecast** DataFrame containing the columns **ds,yhat, yhat_lower,** and **yhat_upper** using **print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']]).**

15.Import the **plot_plotly** function from **prophet.plot** and use it to plot the forecast. This generates an interactive plotly plot of the forecast.

## 3.3.3 LSTM

### 1.Importing necessary libraries:

The first few lines of the code import the necessary libraries like pandas_datareader for fetching the data, numpy, matplotlib, and scikit-learn for data processing, and keras for building lstm models.

### 2.Loading the data:

The code then loads the stock price data from a csv file using pandas read_csv() function.

### 3.Data Pre-processing:

The 'close' column is selected from the loaded data and the minmaxscaler from scikit-learn is used to scale the data between 0 and 1. The training data length is calculated as 80% of the total length of the data, and the data is split into training and testing sets.

### 4.Creating training data:

The training data is created by using a sliding window of 60 days for each day in the training set. This means that for each day in the training set, the model is fed data from the previous 60 days to predict the next day's stock price. The x_train and y_train are then created by splitting the data into input and output sequences.

### 5.Building the lstm model:

The lstm model is built using the keras sequential model. It consists of two lstm layers with 50 neurons each, followed by two dense layers with 25 and 1 neuron respectively. The model is compiled with the 'adam' optimizer and the mean squared error loss function.

## 6.Training the model:

The model is trained on the x_train and y_train data for two epochs with a batch size of 3.

## 7.Testing the model:

The model is tested on the last 60 days of the scaled data. The x_test data is created by using a sliding window of 60 days, and the model is used to predict the stock prices for the next day.

## 8.Inverse scaling and calculating the root mean squared error (RMSE):

The predicted stock prices are then inverse scaled using the minmaxscaler and the rmse is calculated by comparing the predicted prices with the actual prices.

## 9.Visualizing the results:

The training data and the testing data along with the predicted prices are then plotted using the matplotlib library.

## 10.Forecasting future prices:

Finally, the model is used to forecast the next 365 days of stock prices. The last 60 days of the original data is taken as the initial input and the model is used to predict the next day's price. This process is repeated for 365 days to get the forecasted prices. The predicted prices are then plotted using the matplotlib library.

# CHAPTER 4

# ANALYSYIS AND INTERPRETATIONS

## 4.1 Presentation and interpretation of the results obtained from the analysis

### 4.1.1 DR. REDDY DATA


Figure 4.1 Closing price of DR. Reddy from 2018 to 2022.

### ARIMA MODEL:

1. ADF -1.3370660617983308

2. P-Value: 0.6120486341747752

3. Num of Lags: 0

4. Num of Observations used for ADF Regression: 1235

5. Critical Values:

   1%: -3.4356560275160835

   5%: -2.8638831211270817

10%: -2.568017509711682

The p-value is obtained is greater than significance level of 0.05 and the ADF statistic is higher than any of the critical values.

Clearly is no reason to reject the null hypothesis. So, the time series is in fact non-stationary., there

Here, above data is non-stationary

## 4.1 TABULATION OF SARIMA:

| SARIMAX Results | | | |
|---|---|---|---|
| **Dep. Variable:** | y | **No. Observations:** | 1236 |
| **Model:** | ARIMA (3, 2, 0) | **Log Likelihood** | -7031.039 |
| **Date:** | Fri, 31 Mar 2023 | **AIC** | 14070.078 |
| **Time:** | 10:26:43 | **BIC** | 14090.550 |
| **Sample:** | 0 | **HQIC** | 14077.779 |
| | - 1236 | | |
| **Covariance Type:** | opg | | |

| | **Coeff** | **std err** | **z** | **P>\|z\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| **ar. L1** | -0.7407 | 0.019 | -39.239 | 0.003 | -0.778 | -0.704 |
| **ar. L2** | -0.5118 | 0.021 | -24.120 | 0.000 | -0.553 | -0.470 |
| **ar. L3** | -0.2450 | 0.020 | -12.368 | 0.000 | -0.284 | -0.206 |
| **sigma2** | 5203.4570 | 103.880 | 50.091 | 0.000 | 4999.857 | 5407.057 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 2.63 | **Jarque-Bera (JB):** | 2156.34 |
| **Prob(Q):** | 0.10 | **Prob (JB):** | 0.00 |
| **Heteroskedasticity (H):** | 2.36 | **Skew:** | -0.11 |
| **Prob(H) (two-sided):** | 0.00 | **Kurtosis:** | 9.47 |

Table 4.1 shows the results of ARIMA (3,2,0)

## Basic rule for significant:

### P-value ≤ α: The term is statistically significant

- If the p-value is less than or equal to the significance level, you can conclude that the coefficient is statistically significant.

### P-value > α: The term is not statistically significant

If the p-value is greater than the significance level, you cannot conclude that the coefficient is statistically significant. You may want to refit the model without the term

- Here p-value is less than or equal to the significance level, you can conclude that the coefficient is statistically significant

### Key Results: (P, Coeff)

The autoregressive term has a p-value that is less than the significance level of 0.05. You can conclude that the coefficient for the autoregressive term is statistically significant, and you should keep the term in the model.

## FINAL GRAPHICAL OUTPUT:

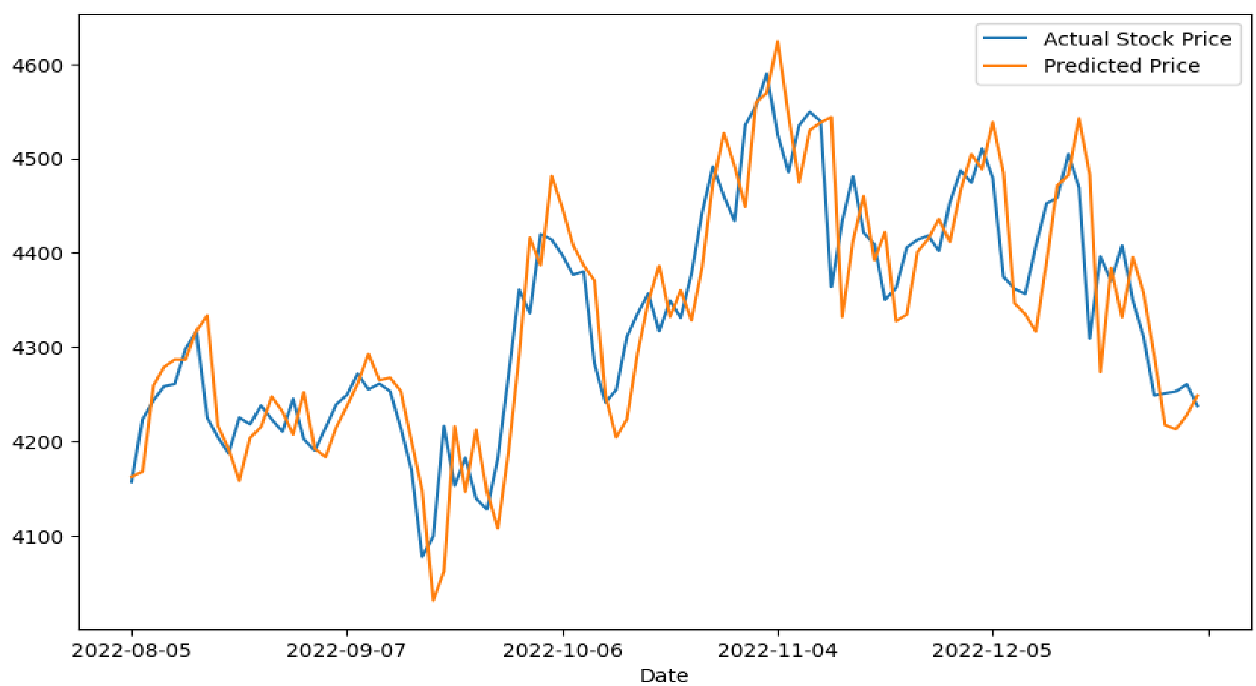


Figure 4.2 Actual price and Predicted price of Dr Reddy for ARIMA Model

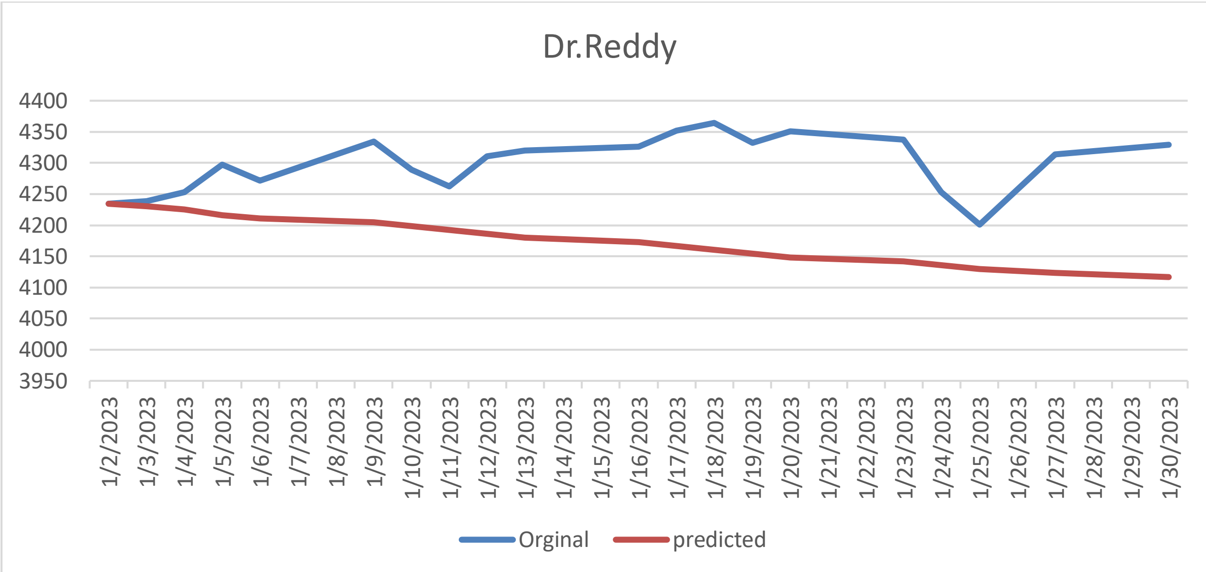


Figure 4.3 One-month predicted price and original price of Dr Reddy for ARIMA Model.

## FACEBOOK PROPHET MODEL:

| | ds | y hat | y hat_ lower | y hat_ upper |
|---|---|---|---|---|
| | | | | |
| 0 | 2018-01-01 | 2493.873607 | 2281.939688 | 2696.058624 |
| 1 | 2018-01-02 | 2501.202859 | 2315.530704 | 2704.962636 |
| 2 | 2018-01-03 | 2501.414925 | 2296.725754 | 2712.946637 |
| 3 | 2018-01-04 | 2503.744979 | 2305.574861 | 2727.899403 |
| 4 | 2018-01-05 | 2499.611796 | 2298.167695 | 2705.606716 |
| ... | ... | ... | ... | ... |
| 1596 | 2023-12-26 | 4099.857427 | 3198.154474 | 4998.703306 |
| 1597 | 2023-12-27 | 4100.901335 | 3222.809354 | 5004.218384 |
| 1598 | 2023-12-28 | 4104.837545 | 3138.684316 | 5010.016668 |
| 1599 | 2023-12-29 | 4103.103737 | 3212.481152 | 5025.128305 |
| 1600 | 2023-12-30 | 4076.519426 | 3165.438494 | 4980.394926 |

Table 4.2 One-year Prediction of DR. Reddy

## FINAL GRAPHICAL OUTPUT:

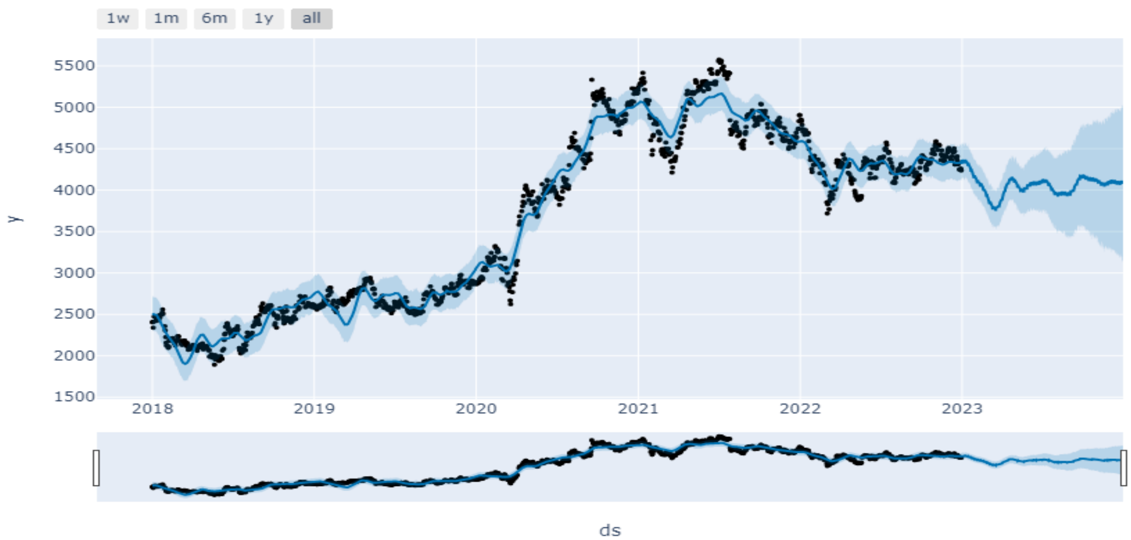


Figure 4.4 Actual price and predicted price of Dr Reddy for FB PROPHET Model.

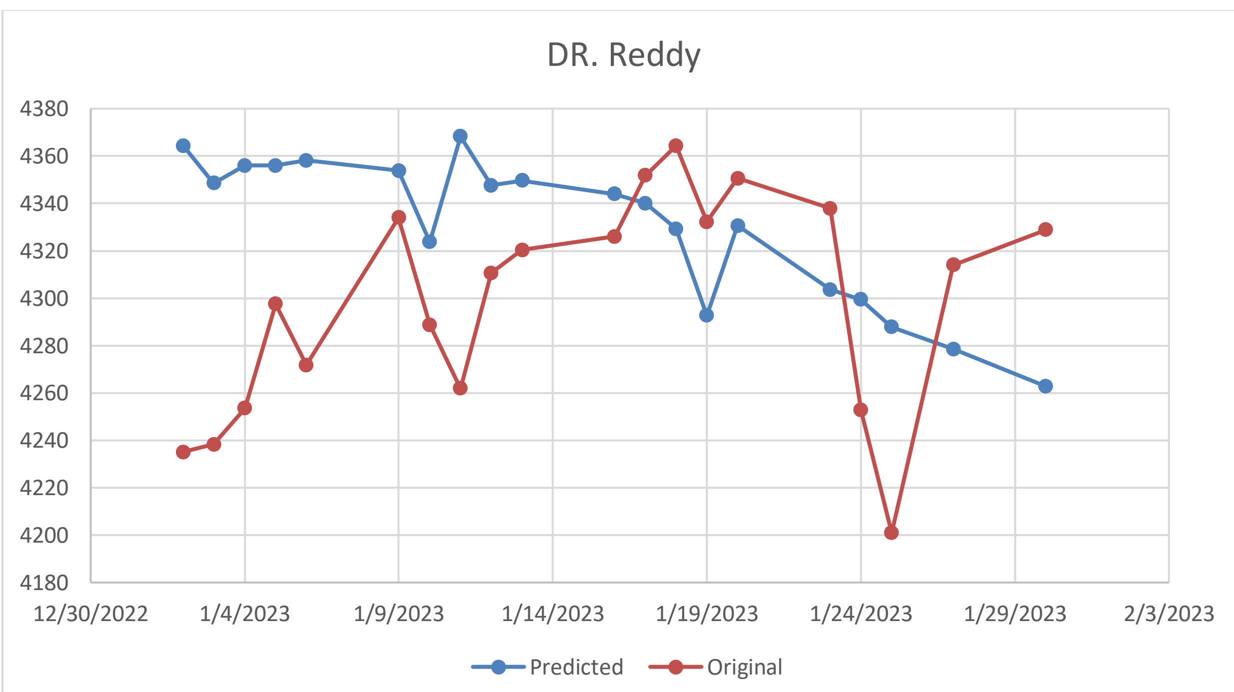


Figure 4.5 One-month predicted price and original price of Dr Reddy for FB PROPHET Model.

## FINAL GRAPHICAL OUTPUT FOR LSTM MODEL:

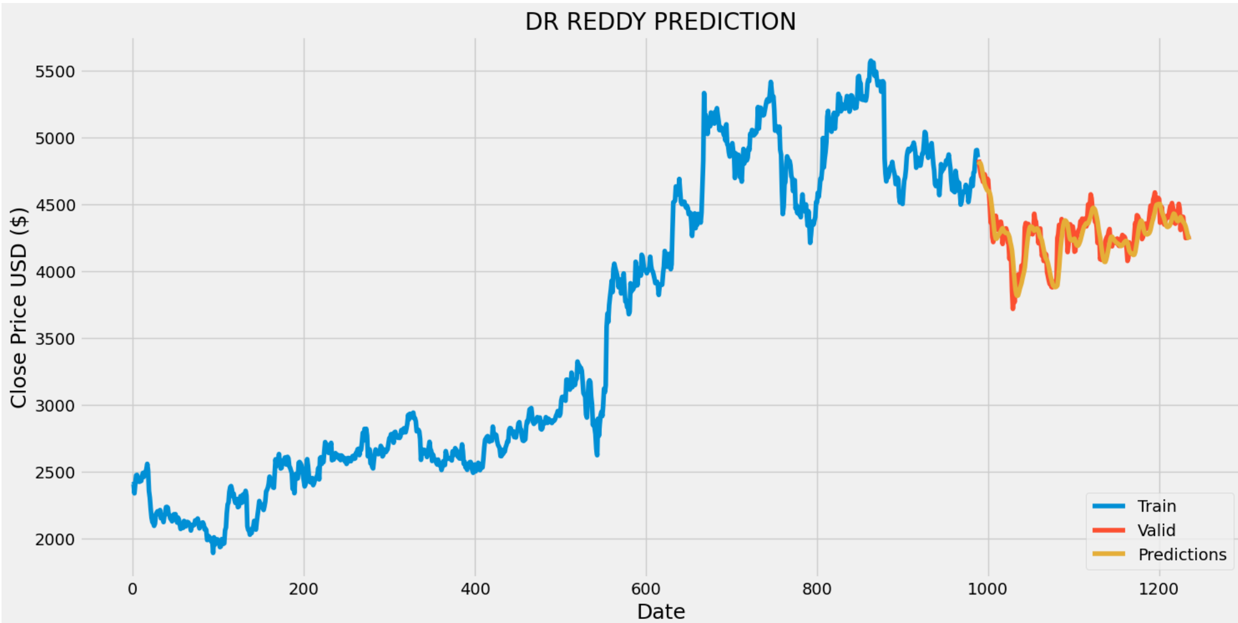


Figure 4.6 Actual price and prediction price with validation of train data of Dr Reddy for LSTM model.

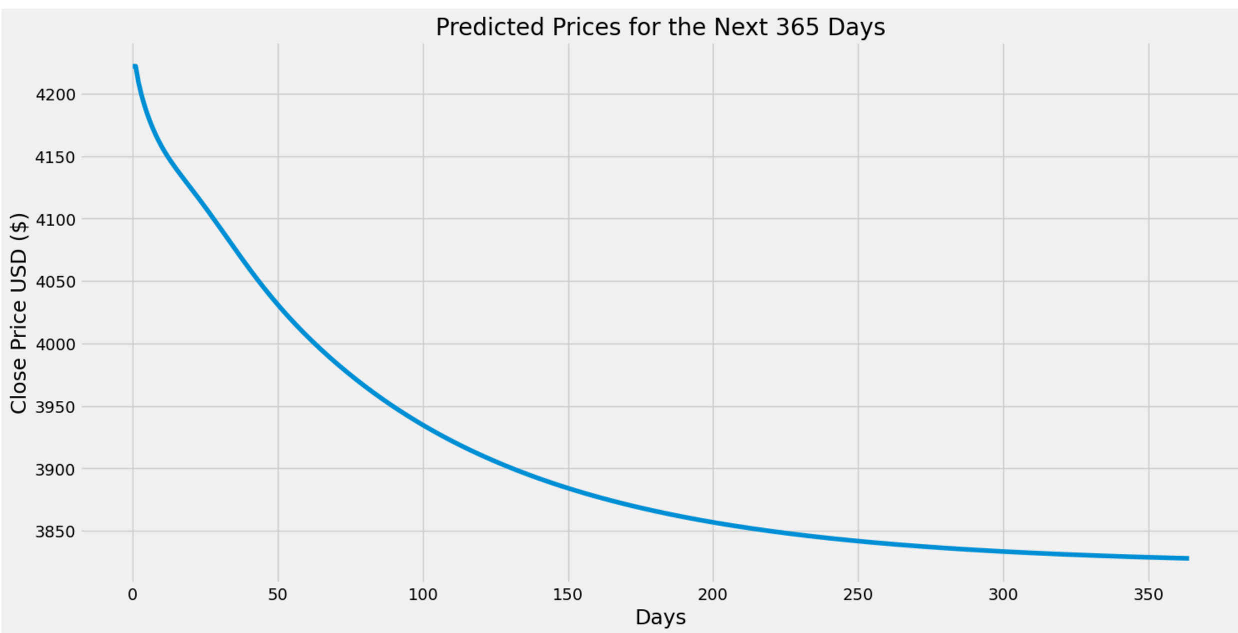


Figure 4.7 One-year predicted price of Dr Reddy for LSTM model.

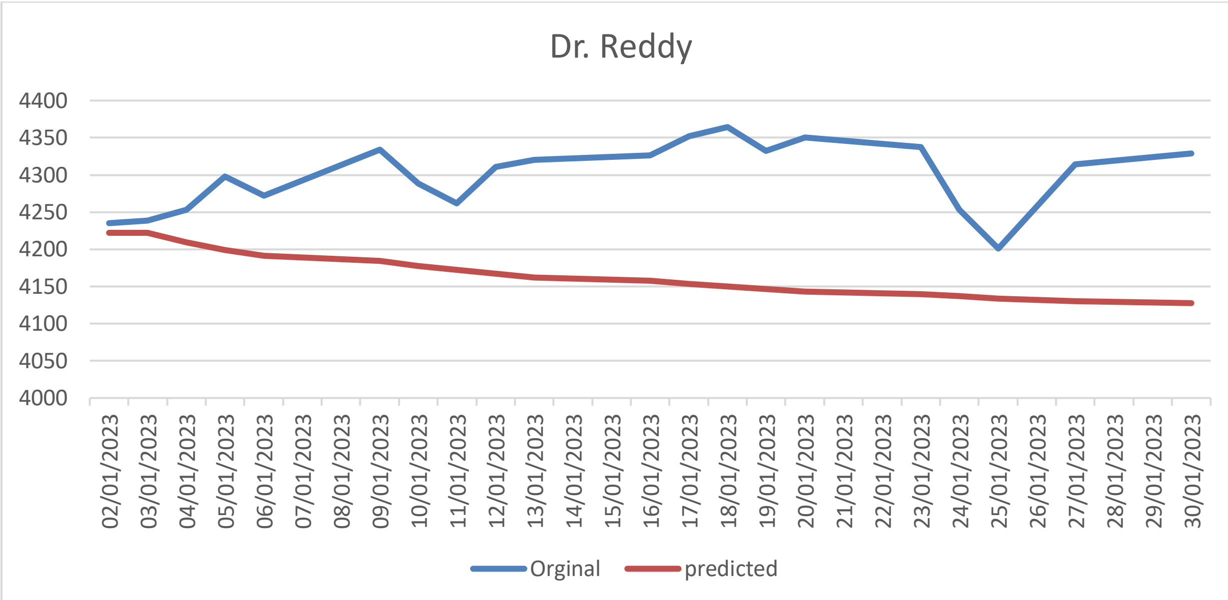


Figure 4.8 One-month predicted price and original price of Dr Reddy for LSTM model.

## 4.1.2 DR. AGARWAL DATA

### ARIMA MODEL

1. ADF: 0.06519564404797677
2. P-Value: 0.9636029153471399
3. Num of Lags: 19
4. Num of Observations Used for ADF Regression: 1216
5. Critical Values:
    1%: -3.435739110194116
    5%: -2.863919777127088
    10%: -2.5680370312770515

The p-value is obtained is greater than significance level of 0.05 and the ADF statistic is higher than any of the critical values.
Clearly, there is no reason to reject the null hypothesis. So, the time series is in fact non-stationary.

## 4.3 TABULATION OF SARIMA:

| SARIMAX Results | | | |
|---|---|---|---|
| **Dep. Variable:** | Y | **No. Observations:** | 1236 |
| **Model:** | ARIMA (5, 2, 0) | **Log Likelihood** | -5377.594 |
| **Date:** | Fri, 31 Mar 2023 | **AIC** | 10767.188 |
| **Time:** | 10:15:55 | **BIC** | 10797.896 |
| **Sample:** | 0 | **HQIC** | 10778.739 |
| | - 1236 | | |
| **Covariance Type:** | Opg | | |

| | **coeff** | **std err** | **z** | **P>\|z\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| **ar. L1** | -0.8902 | 0.015 | -60.090 | 0.000 | -0.919 | -0.861 |
| **ar. L2** | -0.6325 | 0.025 | -25.438 | 0.000 | -0.681 | -0.584 |
| **ar. L3** | -0.5446 | 0.027 | -20.508 | 0.000 | -0.597 | -0.493 |
| **ar. L4** | -0.3691 | 0.023 | -16.279 | 0.000 | -0.414 | -0.325 |
| **ar. L5** | -0.2080 | 0.014 | -14.561 | 0.000 | -0.236 | -0.180 |
| **sigma2** | 356.6899 | 5.537 | 64.421 | 0.000 | 345.838 | 367.542 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 1.22 | **Jarque-Bera (JB):** | 9342.55 |
| **Prob(Q):** | 0.27 | **Prob (JB):** | 0.00 |
| **Heteroskedasticity (H):** | 3.23 | **Skew:** | 0.17 |
| **Prob(H) (two-sided):** | 0.00 | **Kurtosis:** | 16.48 |

Table 4.3 shows the results of ARIMA (5,2,0)

# Basic rule for significant:

### P-value ≤ α: The term is statistically significant

If the p-value is less than or equal to the significance level, you can conclude that the coefficient is statistically significant.

### Key Results: (P, Coeff)

The autoregressive term has a p-value that is less than the significance level of 0.05. You can conclude that the coefficient for the autoregressive term is statistically significant, and you should keep the term in the model.

## FINAL GRAPHICAL OUTPUT:

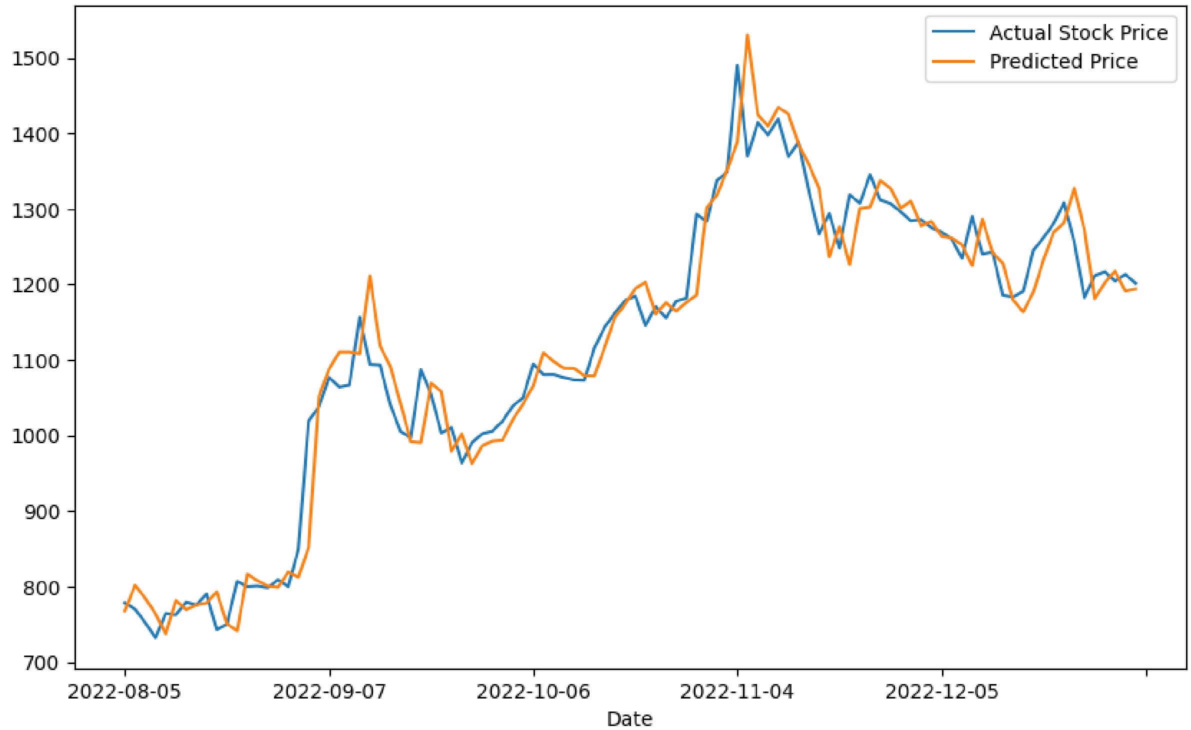


Figure 4.9 Actual price and Predicted price of Dr Agarwal for ARIMA Model

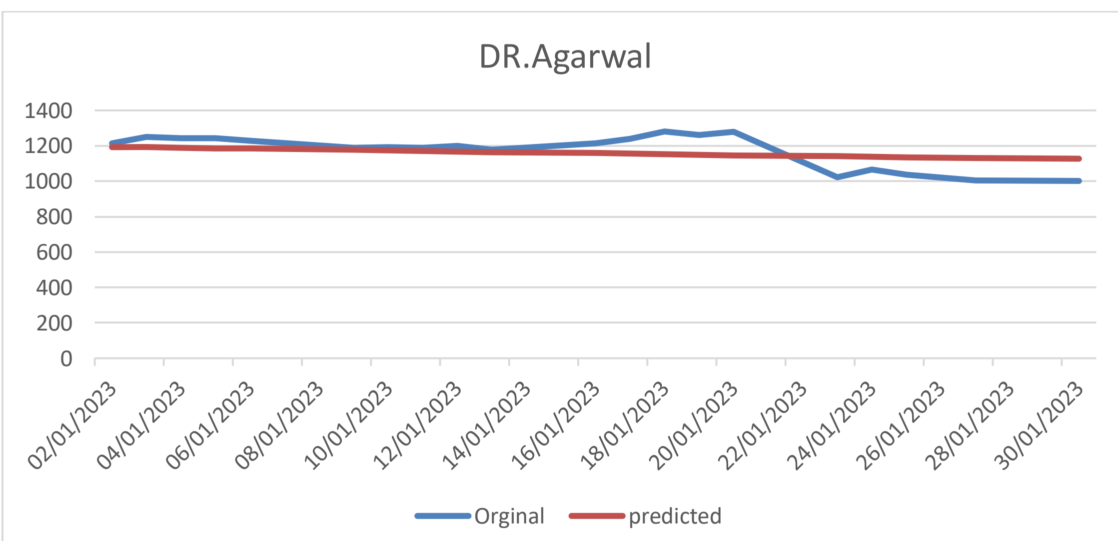


Figure 4.10 One-month predicted price and original price of Dr Agarwal for ARIMA model.

## FACEBOOK PROPHRT MODEL:

| | ds | y hat | y hat_ lower | y hat_ upper |
|---|---|---|---|---|
| 0 | 2018-01-01 | 709.541172 | 622.729440 | 798.474013 |
| 1 | 2018-01-02 | 708.850978 | 616.372158 | 797.770823 |
| 2 | 2018-01-03 | 708.396379 | 616.955599 | 788.052037 |
| 3 | 2018-01-04 | 709.592419 | 622.377642 | 801.745858 |
| 4 | 2018-01-05 | 710.640004 | 633.784862 | 797.856783 |
| ... | ... | ... | ... | ... |
| 1596 | 2023-12-26 | 1738.477583 | 1621.831718 | 1852.929511 |
| 1597 | 2023-12-27 | 1738.353573 | 1629.904829 | 1851.558157 |
| 1598 | 2023-12-28 | 1739.812046 | 1622.779027 | 1858.089342 |
| 1599 | 2023-12-29 | 1741.090477 | 1621.750035 | 1859.698911 |
| 1600 | 2023-12-30 | 1695.716143 | 1584.802926 | 1821.934556 |

Table 4.4 One-year Prediction of DR. Agarwal.

## FINAL GRAPHICAL OUTPUT:


Figure 4.11 Actual price and predicted price of Dr Agarwal for for FB PROPHET Model.

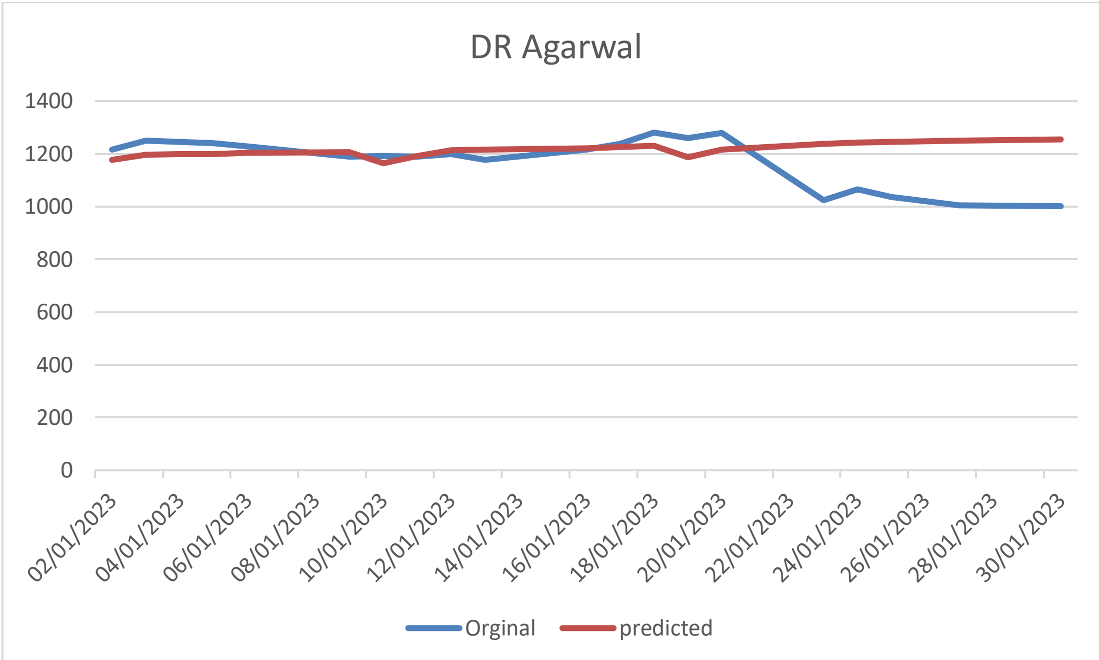


Figure 4.12 One-month predicted price and original price of Dr Agarwal for FB PROPHET Model.

## FINAL GRAPHICAL OUTPUT FOR LSTM MODEL:

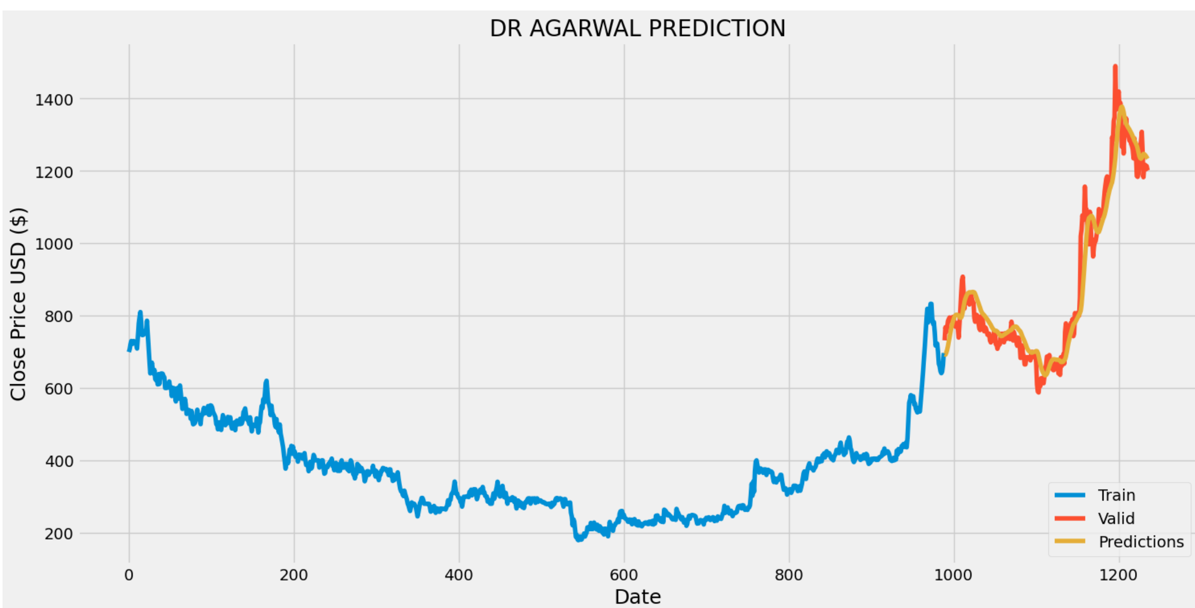


Figure 4.13 Actual price and prediction price with validation of train data of Dr Agarwal

for LSTM model.

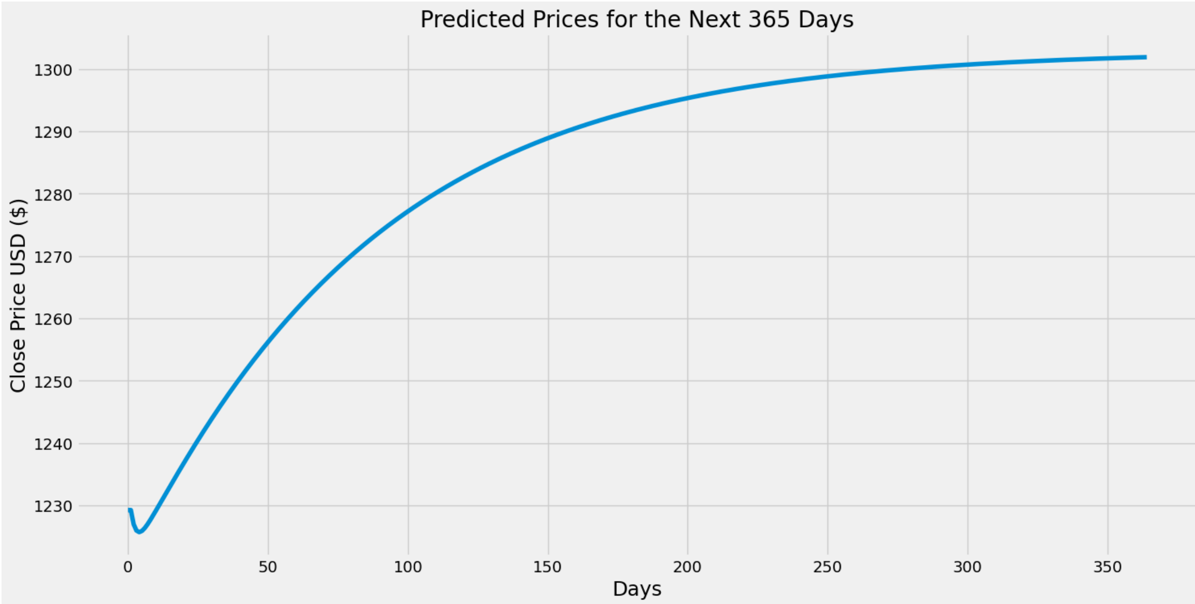


Figure 4.14 One-year predicted price of Dr Agarwal for LSTM model.

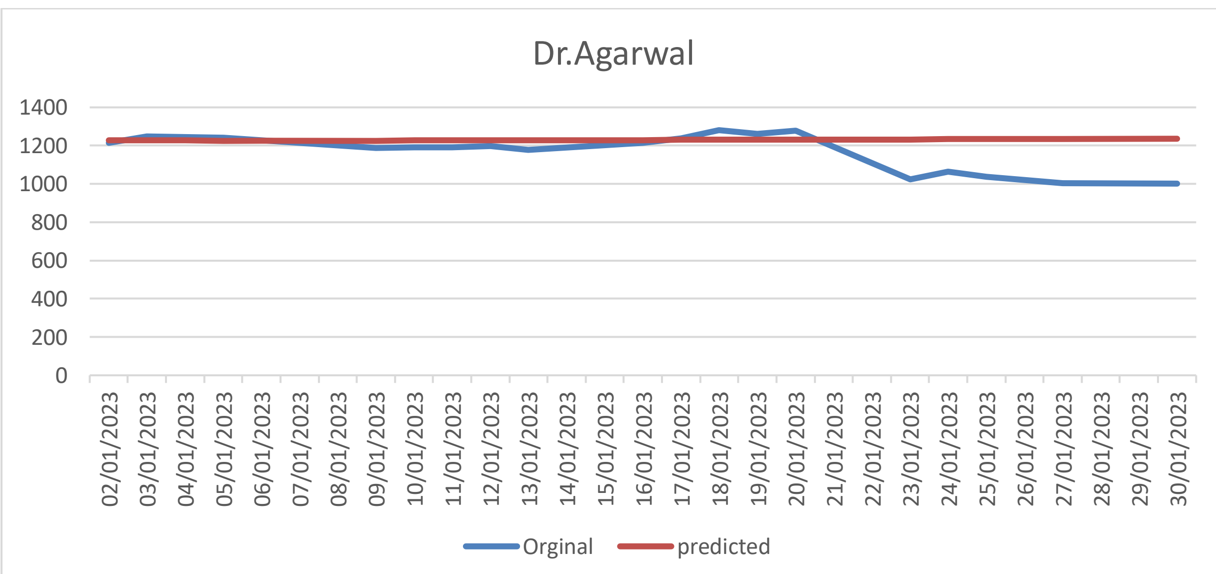


Figure 4.15 One-month predicted price and original price of Dr Agarwal for LSTM model.

## 4.1.3 EICHER MOTOR DATA

### ARIMA MODEL

1. ADF: -1.5542620462067818
2. P-Value: 0.5065185617938253
3. Num of Lags: 0
4. Num of Observations Used for ADF Regression: 1235
5. Critical Values:
   - 1%: -3.4356560275160835
   - 5%: -2.8638831211270817
   - 10%: -2.568017509711682

The p-value is obtained is greater than significance level of 0.05 and the ADF statistic is higher than any of the critical values.
Clearly, there is no reason to reject the null hypothesis. So, the time series is in fact non-stationary.

## 4.5 TABULATION OF SARIMAX

| SARIMAX Results | | | |
|---|---|---|---|
| **Dep. Variable:** | y | **No. Observations:** | 1236 |
| **Model:** | ARIMA (4, 2, 0) | **Log Likelihood** | -6688.121 |
| **Date:** | Fri, 31 Mar 2023 | **AIC** | 13386.243 |
| **Time:** | 10:59:07 | **BIC** | 13411.833 |
| **Sample:** | 0 | **HQIC** | 13395.869 |
| | - 1236 | | |
| **Covariance Type:** | opg | | |

| | **Co eff** | **std err** | **z** | **P>\|z\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| **ar.L1** | -0.7992 | 0.025 | -31.628 | 0.000 | -0.849 | -0.750 |
| **ar.L2** | -0.5890 | 0.033 | -18.009 | 0.000 | -0.653 | -0.525 |
| **ar.L3** | -0.3944 | 0.030 | -13.034 | 0.000 | -0.454 | -0.335 |
| **ar.L4** | -0.2149 | 0.024 | -8.988 | 0.000 | -0.262 | -0.168 |
| **sigma2** | 2977.0464 | 97.620 | 30.496 | 0.000 | 2785.714 | 3168.378 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 0.93 | **Jarque-Bera (JB):** | 54.76 |
| **Prob(Q):** | 0.33 | **Prob (JB):** | 0.00 |
| **Heteroskedasticity (H):** | 1.03 | **Skew:** | 0.06 |
| **Prob(H) (two-sided):** | 0.78 | **Kurtosis:** | 4.02 |

Table 4.5 shows the results of ARIMA (4,2,0)

## Basic rule for significant:

### P-value ≤ α: The term is statistically significant

If the p-value is less than or equal to the significance level, you can conclude that the coefficient is statistically significant.

**Key Results: (P, Coeff)**

The autoregressive term has a p-value that is less than the significance level of 0.05. You can conclude that the coefficient for the autoregressive term is statistically significant, and you should keep the term in the model.

## FINAL GRAPHICAL OUTPUT

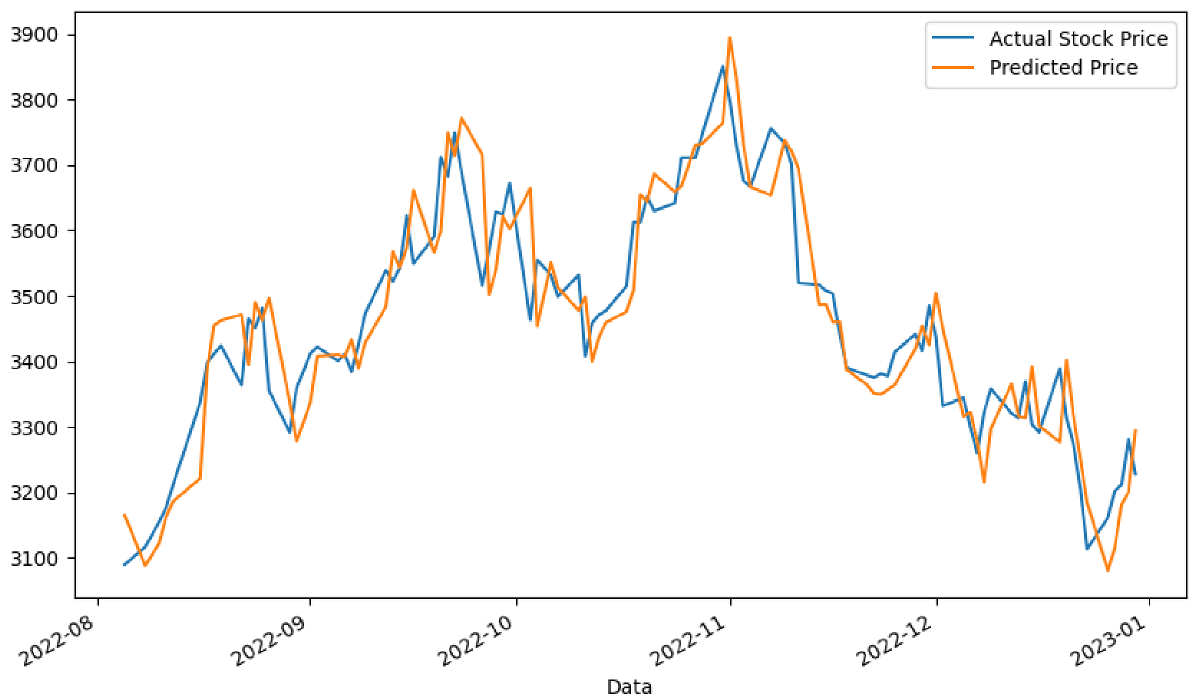


Figure 4.16 Actual price and Predicted price of Eicher Motor for ARIMA Model

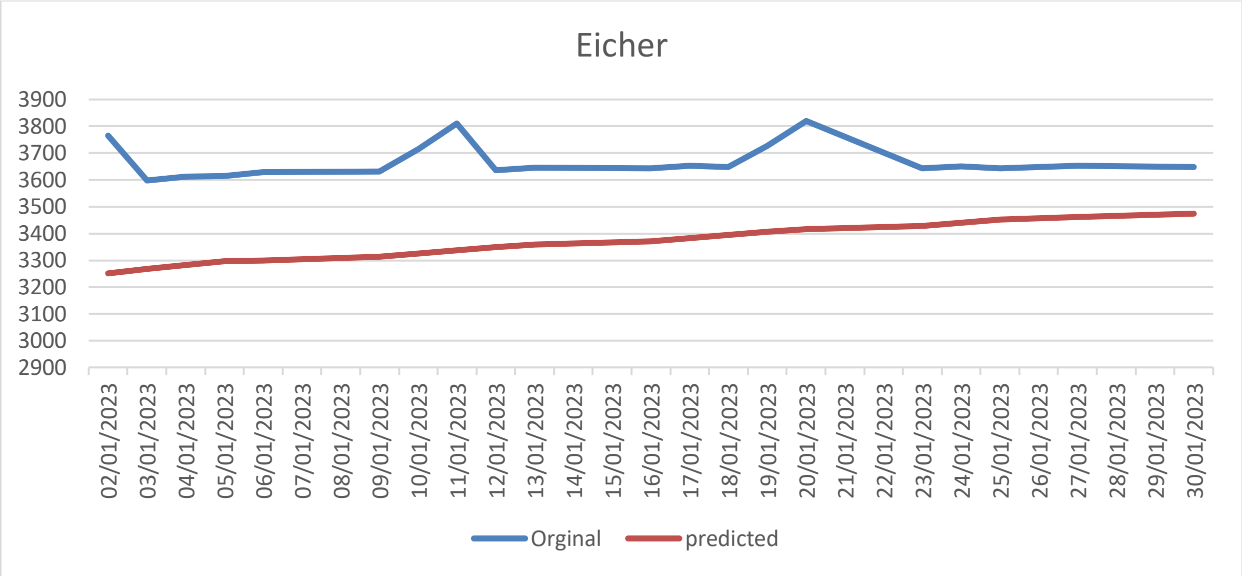


Figure 4.17 One-month predicted price and original price of Eicher Motor for ARIMA model.

## FACEBOOK PROPHET MODEL

| | ds | y hat | y hat_ lower | y hat_ upper |
|---|---|---|---|---|
| 0 | 2018-01-01 | 2877.874151 | 2660.640167 | 3082.331247 |
| 1 | 2018-01-02 | 2891.634487 | 2679.508083 | 3093.840960 |
| 2 | 2018-01-03 | 2894.066680 | 2676.639900 | 3094.620244 |
| 3 | 2018-01-04 | 2908.857395 | 2705.545152 | 3129.842577 |
| 4 | 2018-01-05 | 2910.479177 | 2708.549935 | 3127.152448 |
| ... | ... | ... | ... | ... |
| 1596 | 2023-12-26 | 4530.281569 | 3714.371857 | 5275.072601 |
| 1597 | 2023-12-27 | 4535.624070 | 3802.860064 | 5232.939298 |
| 1598 | 2023-12-28 | 4554.165925 | 3801.507833 | 5296.831570 |
| 1599 | 2023-12-29 | 4560.275211 | 3804.935516 | 5310.972146 |
| 1600 | 2023-12-30 | 4648.677592 | 3860.748222 | 5412.860812 |

Table 4.6 One-year Prediction of Eicher Motor.

## FINAL GRAPHICAL OUTPUT:


Figure 4.18 Actual price and predicted price of Eicher Motor for FB PROPHET.

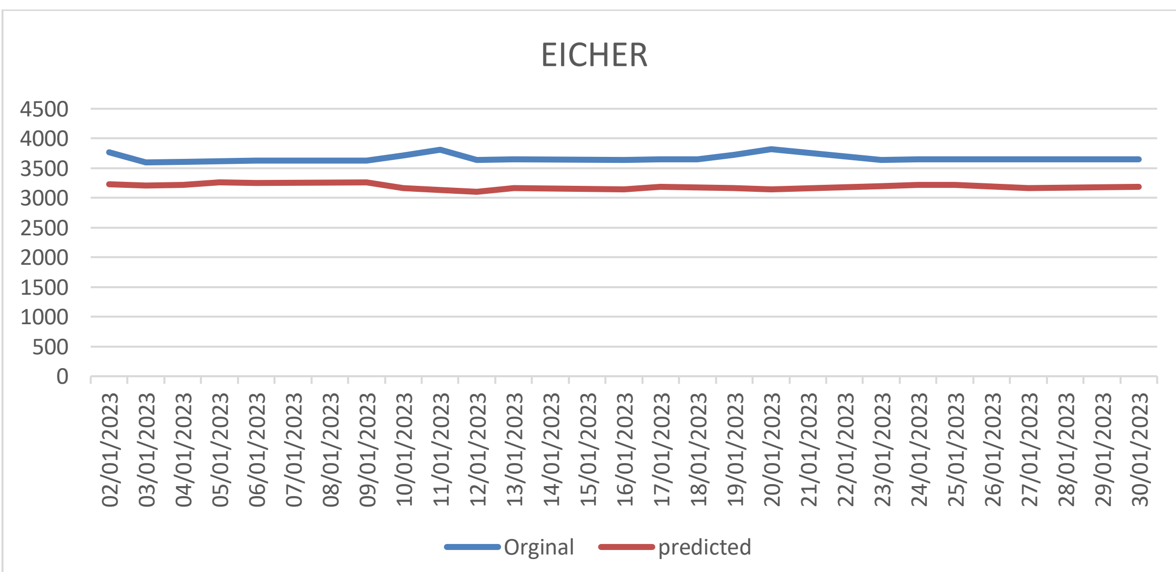


Figure 4.19 One-month predicted price and original price of Eicher Motor for fb prophet model

## FINAL GRAPHICAL OUTPUT FOR LSTM MODEL:

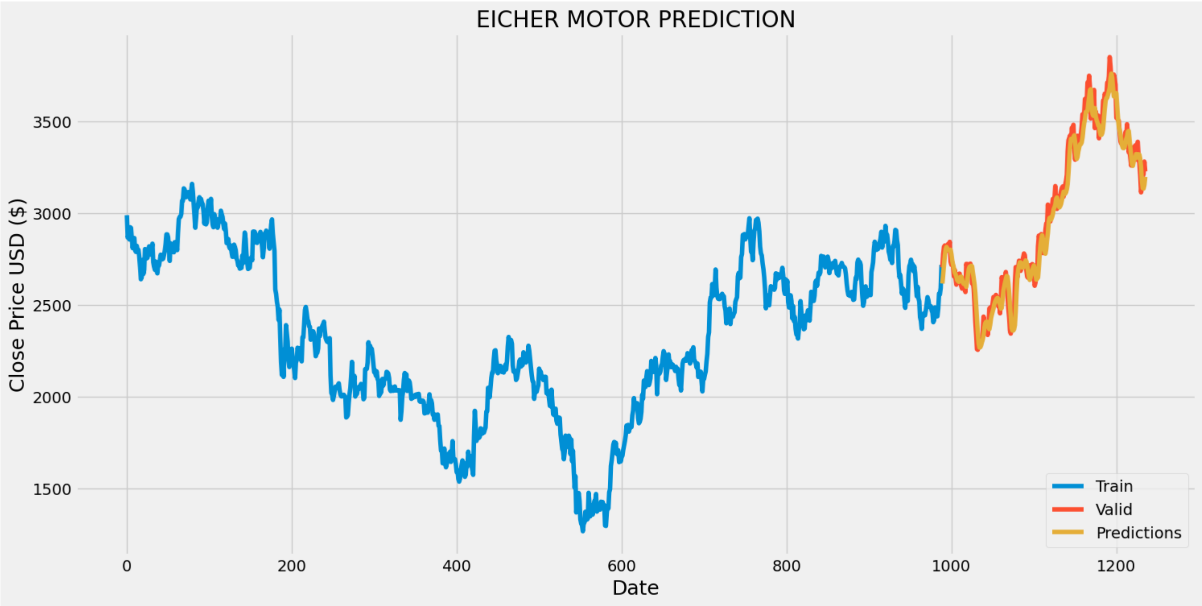


Figure 4.20 Actual price and prediction price with validation of train data of Eicher Motor for LSTM model.

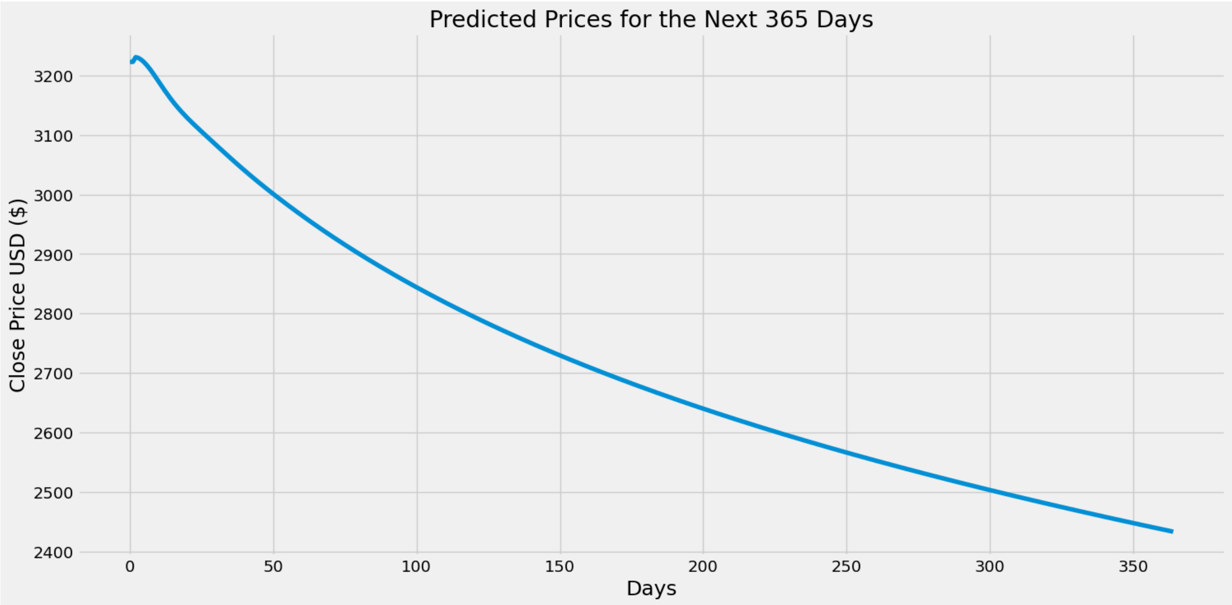


Figure 4.21 One-year predicted price of Eicher Motor for LSTM model.

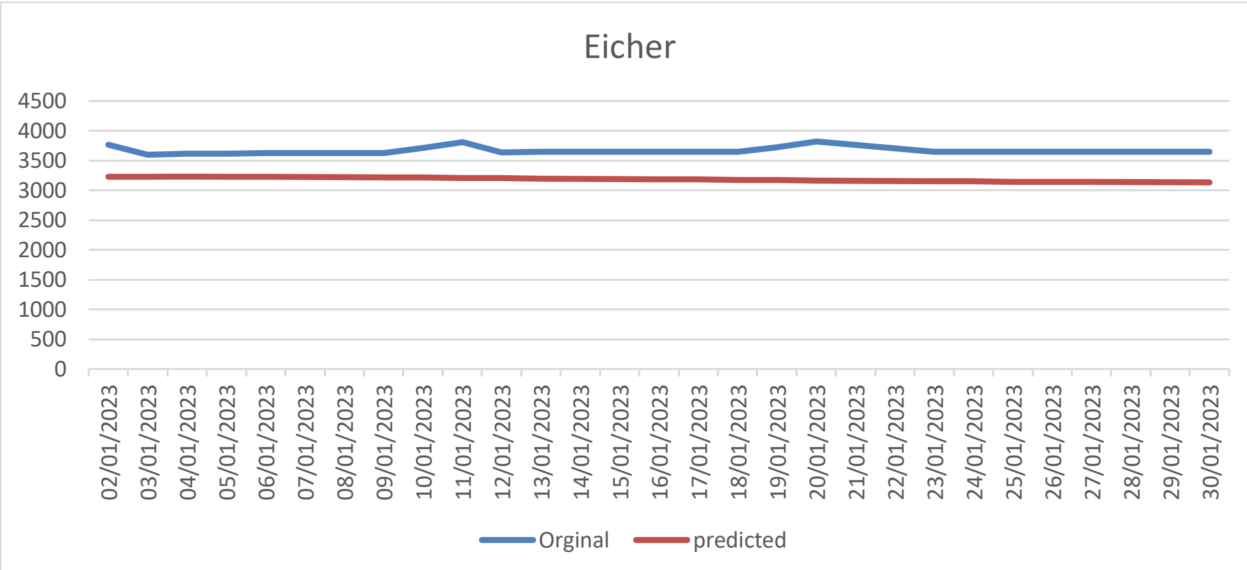


Figure 4.22 One-month predicted price and original price of Eicher Motor for LSTM model.

## 4.1.4 MARUTI MOTOR

### ARIMA MODEL

1. ADF: -2.6025489596431814
2. P-Value: 0.0924715928860656
3. Num of lags: 0
4. Num of Observations Used for ADF Regression: 1235
5. Critical Values:
   - 1%: -3.4356560275160835
   - 5%: -2.8638831211270817
   - 10%: -2.568017509711682

The p-value is obtained is greater than significance level of 0.05 and the ADF statistic is higher than any of the critical values.

Clearly, there is no reason to reject the null hypothesis. So, the time series is in fact non-stationary.

## 4.7 TABULATION OF SARIMAX

| SARIMAX Results | | | |
|---|---|---|---|
| **Dep. Variable:** | y | **No. Observations:** | 1236 |
| **Model:** | ARIMA (2, 2, 0) | **Log Likelihood** | -8006.518 |
| **Date:** | Sat, 01 Apr 2023 | **AIC** | 16019.036 |
| **Time:** | 06:55:15 | **BIC** | 16034.390 |
| **Sample:** | 0 | **HQIC** | 16024.812 |
| | - 1236 | | |
| **Covariance Type:** | opg | | |

| | **Co eff** | **std err** | **z** | **P>\|z\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| **ar. L1** | -0.6478 | 0.020 | -33.140 | 0.000 | -0.686 | -0.610 |
| **ar. L2** | -0.3162 | 0.021 | -14.961 | 0.000 | -0.358 | -0.275 |
| **sigma2** | 2.531e+04 | 702.698 | 36.017 | 0.000 | 2.39e+04 | 2.67e+04 |

| | | | |
|---|---|---|---|
| **Ljung-Box (L1) (Q):** | 7.49 | **Jarque-Bera (JB):** | 268.91 |
| **Prob(Q):** | 0.01 | **Prob (JB):** | 0.00 |
| **Heteroskedasticity (H):** | 1.19 | **Skew:** | 0.17 |
| **Prob(H) (two-sided):** | 0.08 | **Kurtosis:** | 5.26 |

Table 4.7 shows the results of ARIMA (2,2,0)

## Basic rule for significant:

### P-value ≤ α: The term is statistically significant

If the p-value is less than or equal to the significance level, you can conclude that the coefficient is statistically significant.

**Key Results: (P, Coeff)**

The autoregressive term has a p-value that is less than the significance level of 0.05. You can conclude that the coefficient for the autoregressive term is statistically significant, and you should keep the term in the model.

## FINAL GRAPHICAL OUTPUT:

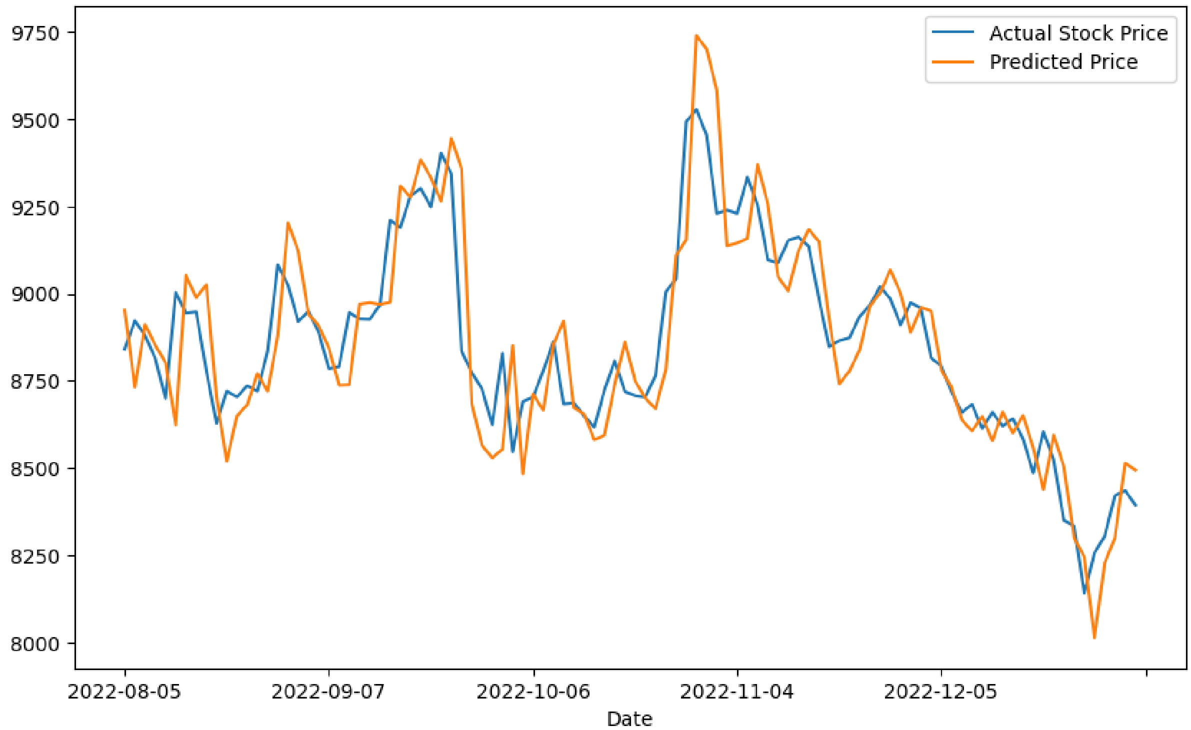


Figure 4.23 Actual price and Predicted price of Maruti Motor for ARIMA Model

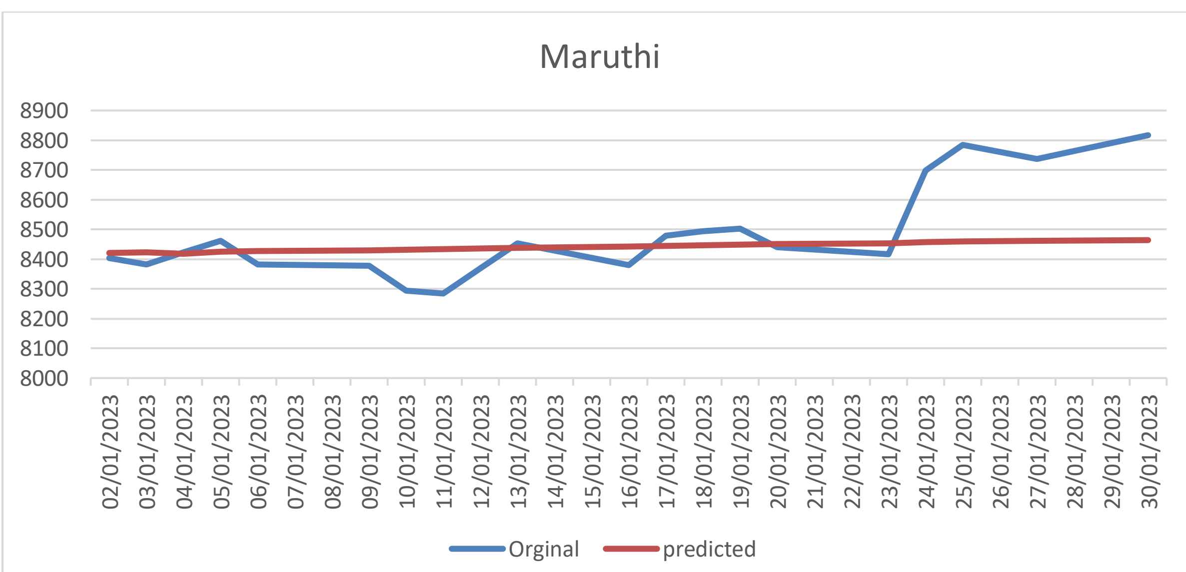


Figure 4.24 One-month predicted price and original price of Maruti Motor for ARIMA

model.

## FOR FACEBOOKPROPHRT MODEL

| | ds | y hat | y hat_ lower | y hat_ upper |
|---|---|---|---|---|
| 0 | 2018-01-01 | 9295.445357 | 8779.072404 | 9822.259742 |
| 1 | 2018-01-02 | 9327.675366 | 8757.159705 | 9863.346834 |
| 2 | 2018-01-03 | 9329.243509 | 8786.039448 | 9808.380645 |
| 3 | 2018-01-04 | 9369.665422 | 8865.597465 | 9880.195809 |
| 4 | 2018-01-05 | 9366.137457 | 8878.678578 | 9889.111839 |
| ... | ... | ... | ... | ... |
| 1596 | 2023-12-26 | 10573.168671 | 8740.187569 | 12398.686903 |
| 1597 | 2023-12-27 | 10573.140962 | 8652.945283 | 12457.676237 |
| 1598 | 2023-12-28 | 10613.815788 | 8827.152062 | 12534.150276 |
| 1599 | 2023-12-29 | 10612.405997 | 8785.466756 | 12569.404379 |
| 1600 | 2023-12-30 | 10368.374084 | 8555.784829 | 12279.125964 |

Table 4.8 One-year Prediction of Maruti Motor.

## FINAL GRAPHICAL OUTPUT:

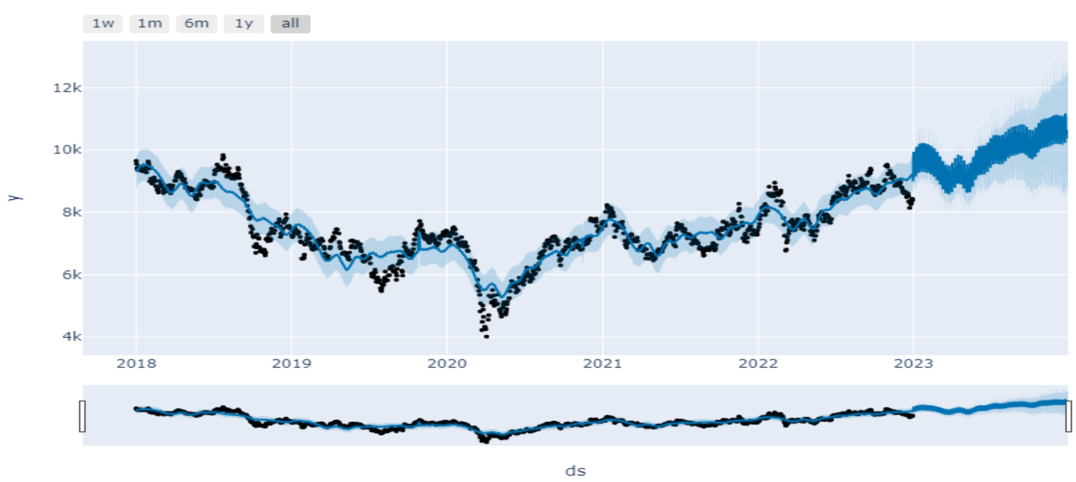


Figure 4.25 Actual price and predicted price of Maruti Motor for FB PROPHET.

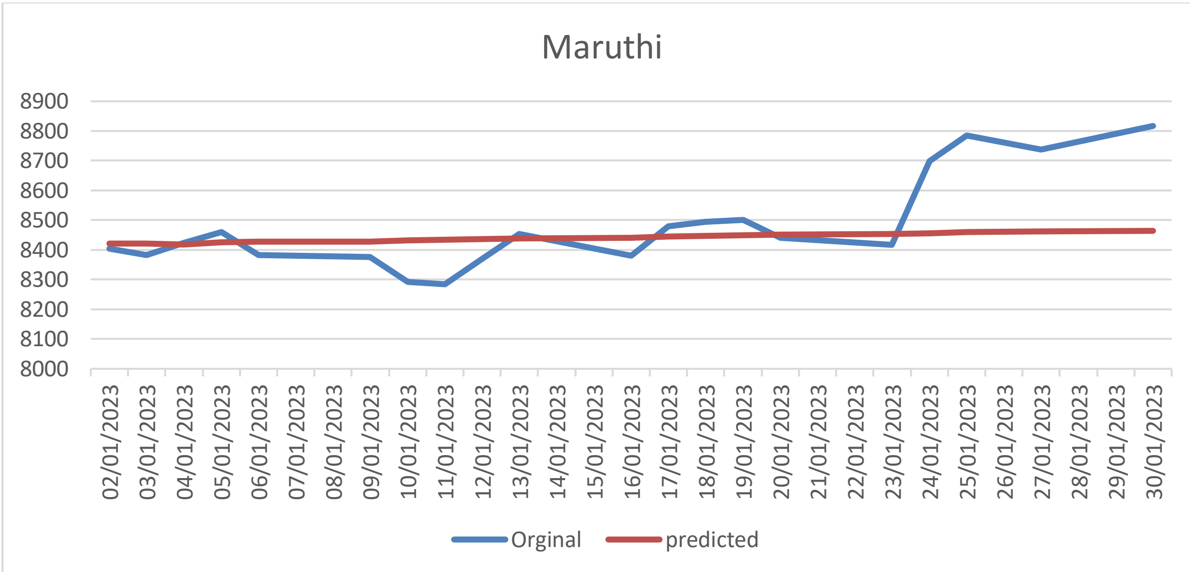


Figure 4.26 One-month predicted price and original price of Maruti Motor for fb prophet model

## FINAL GRAPHICAL OUTPUT FOR LSTM MODEL:

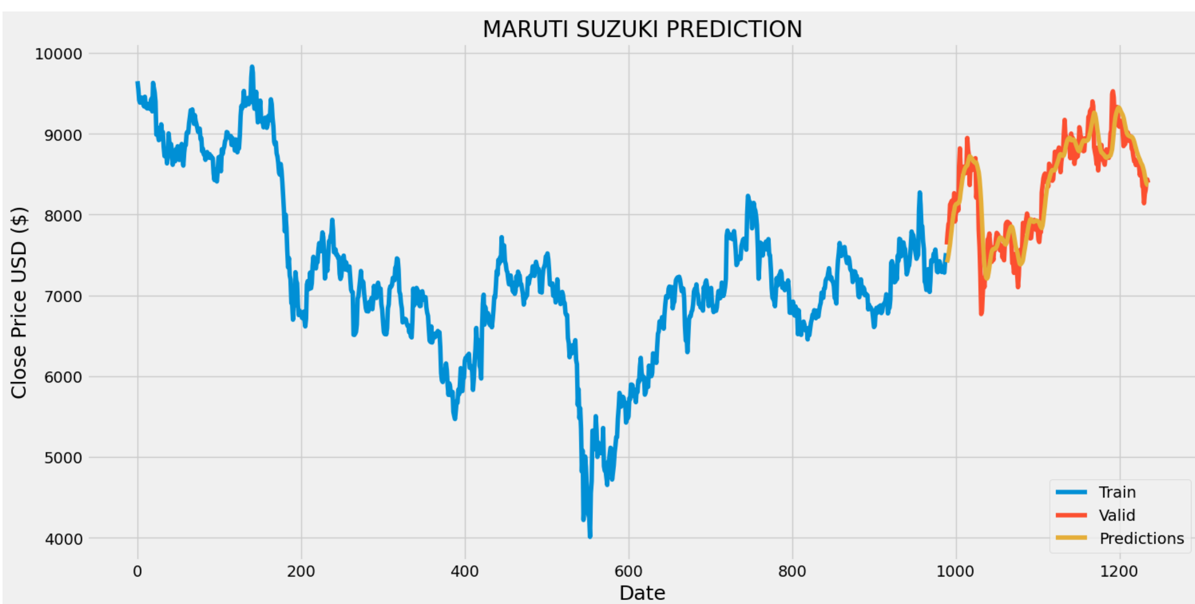


Figure 4.27 Actual price and prediction price with validation of train data of Maruti Motor for LSTM model

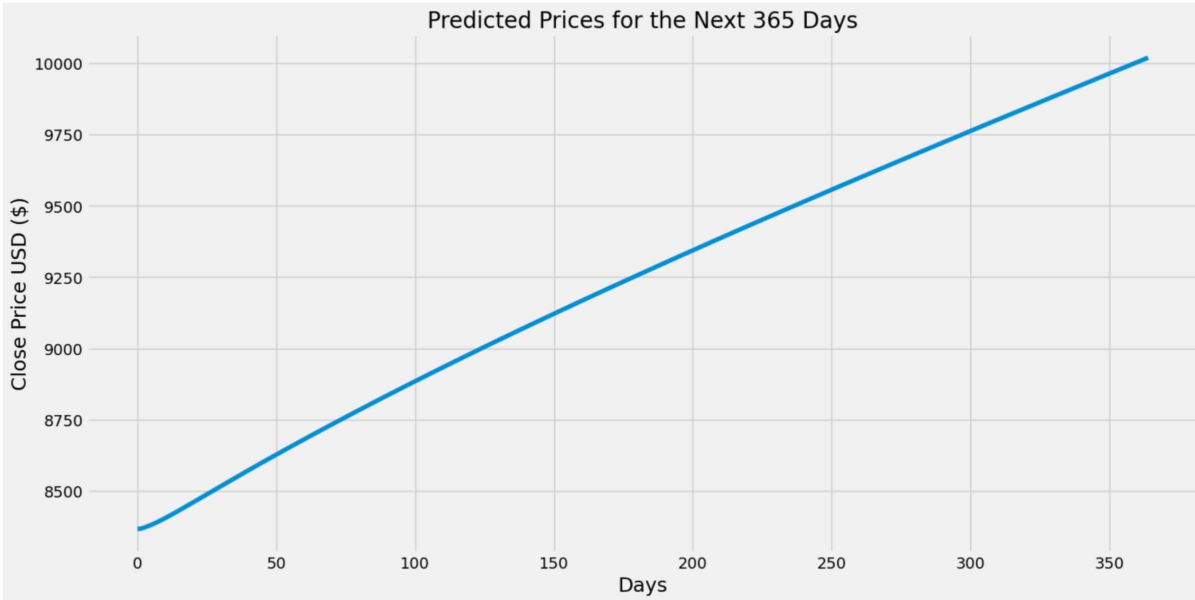


Figure 4.28 One-year predicted price of Maruti Motor for LSTM model.

## 4.2. Comparison of the accuracy of the three models

### DR. REDDY

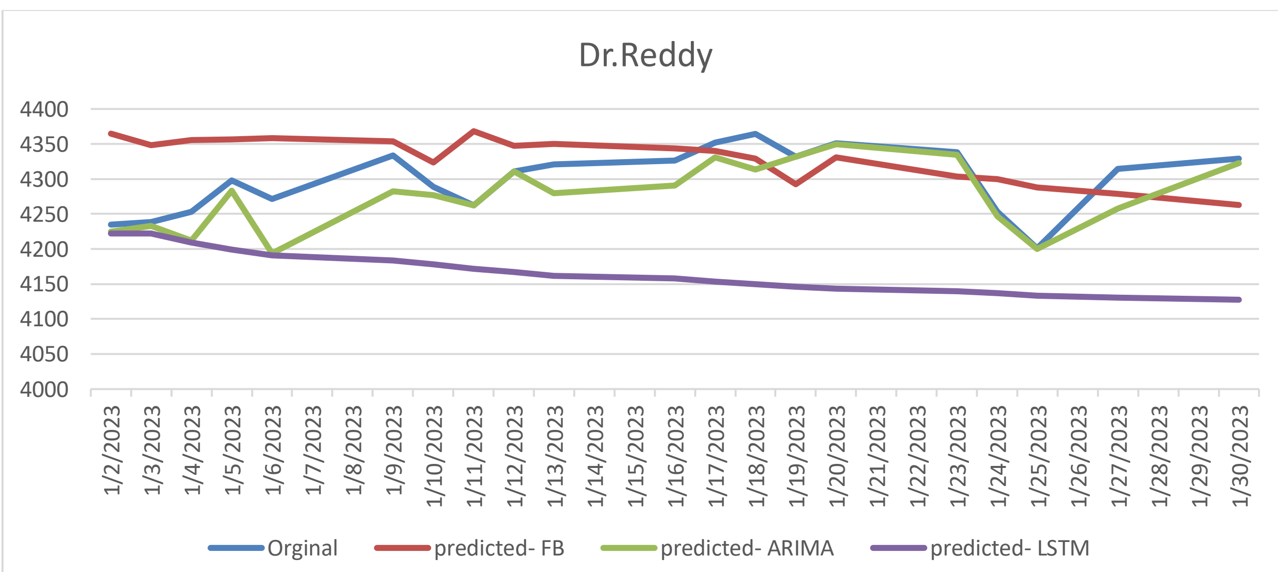


Figure 4.29 Comparison of three model for Dr Reddy

Based on the predicted values obtained from ARIMA, Prophet, and LSTM models, it appears that DR REDDY stock price is expected to show a mixed trend over the next few days, with some days showing an increase and others showing a decrease. However, it is important to note that the accuracy of these predictions is subject to limitations between the range 4150-4350, almost it could have a significant impact on the actual stock price. Therefore, investors should exercise caution and consider a price difference -100 to 100 in between the factors having assurance to make a further investment decision in future.

## DR. AGARWAL

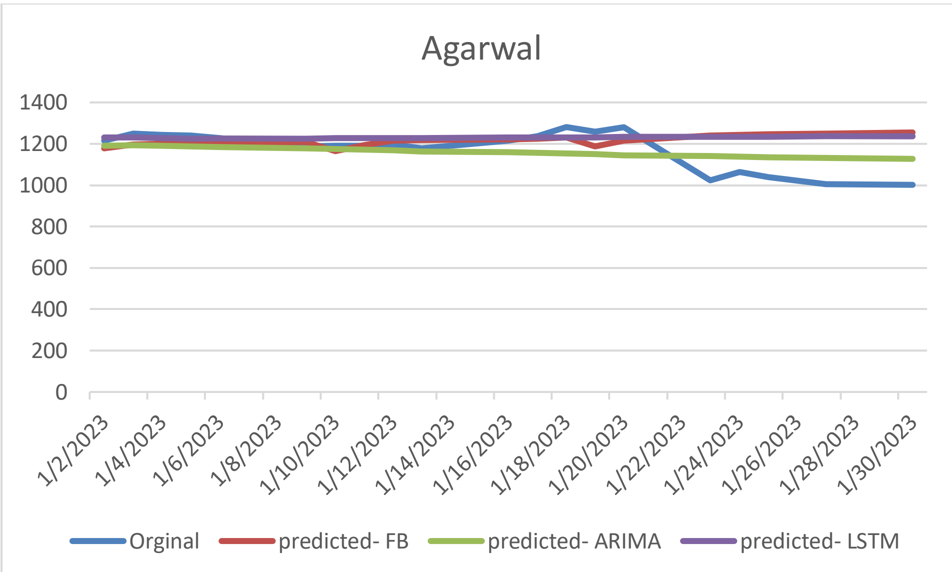


Figure 4.30 Comparison of three model for Dr Agarwal.

Based on the predicted values obtained from ARIMA, Prophet, and LSTM models, it appears that DR AGARWAL stock price is expected to show a mixed trend over the next few days, with some days showing an increase and last eight days showing a decrease. However, it is important to note that the accuracy of these predictions is subject to limitations between the range 1180-1280, almost it could have a significant impact on the actual stock price. Therefore, investors should exercise caution and consider a Price difference -50 to 80 in between the factors having assurance to make a further investment decision in future.

## EICHER MOTOR

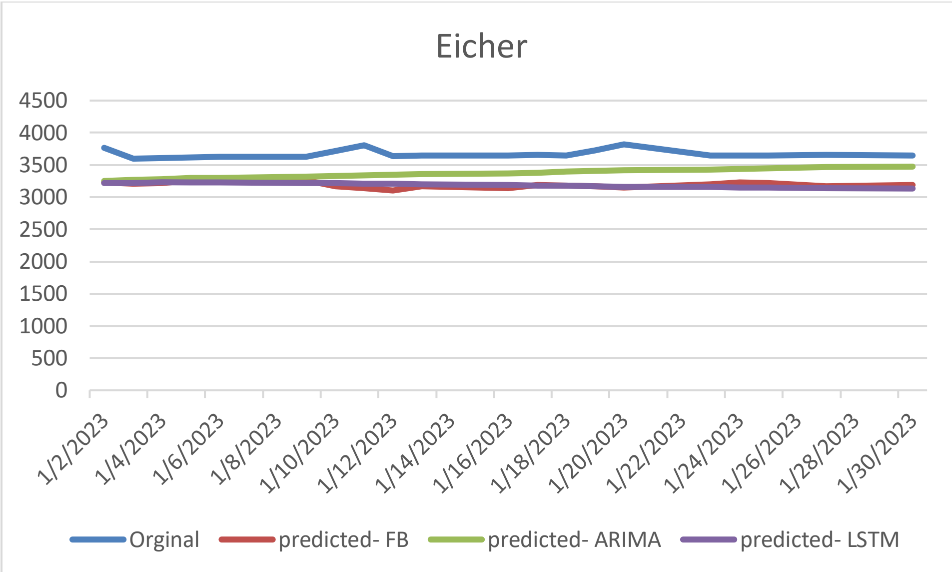


Figure 4.31 Comparison of three model for Eicher Motor.

Based on the predicted values obtained from ARIMA, Prophet, and LSTM models, it appears that EICHER MOTOR stock price is expected to show a mixed trend over the next few days showing a constant flow. However, it is important to note that the accuracy of these predictions is subject to limitations between the range 3000-3500, almost it could have a significant impact on the actual stock price. Therefore, investors should exercise caution and consider a price difference 0 to 150 in between the factors having assurance to make a further investment decision in future.

## MARUTI MOTOR

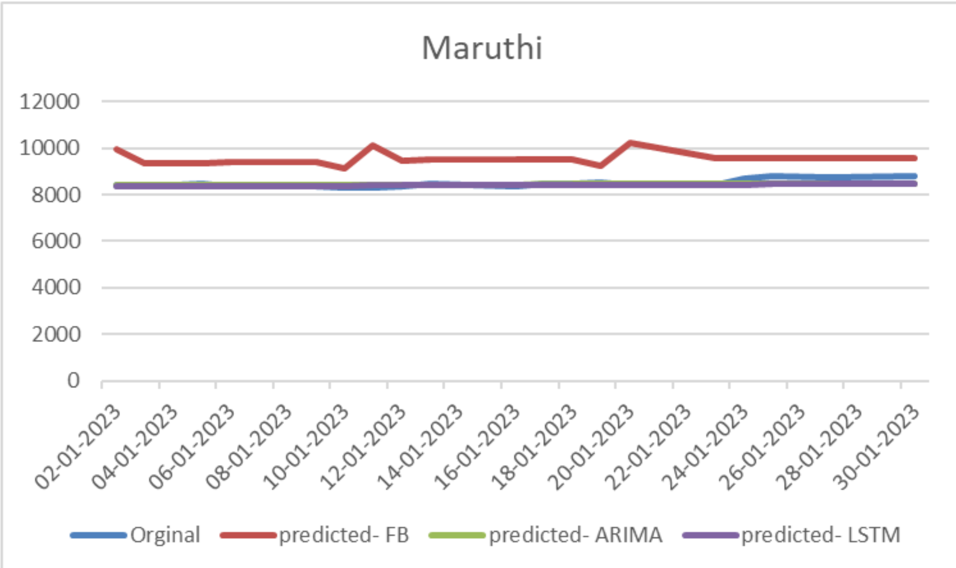


Figure 4.32 Comparison of three model for Maruti Motor.

Based on the predicted values obtained from ARIMA, Prophet, and LSTM models, it appears that MARUTI MOTOR stock price is expected to show a constant trend over the days and almost the fluctuate through the original price. However, it is important to note that the accuracy of these predictions is subject to limitations between the range 8300-8450, almost it could have a significant impact on the actual stock price. Therefore, investors should exercise caution and consider a price difference -10 to 80 in between the factors having assurance to make an further investment decisions in future.

# CHAPTER 5
# CONCLUSION

In this report, we presented and compared three different algorithms for time series prediction. Each algorithm has its own advantages and limitations. Below we summarize our observations for each algorithm:

ARIMA is a powerful model and as we saw it achieved the best result for the stock data. A challenge is that it might need careful hyperparameter tuning and a good understanding of the data.

Prophet is specifically designed for business time series prediction. It achieves very good results for the stock data but, it can fail spectacularly on time series datasets from other domains. In particular, this holds for time series and we can learn seasonal patterns. Prophet's advantage is that it requires less hyperparameter tuning as it is specifically designed to detect patterns in business time series.

LSTM-based recurrent neural networks are probably the most powerful approach to learning from sequential data and time series are only a special case. The potential of LSTM based models is fully revealed when learning from massive datasets where we can detect complex patterns. Unlike ARIMA or Prophet, they do not rely on specific assumptions about the data such as time series stationarity or the existence of a Date field.  A disadvantage is that LSTM based RNNs are difficult to interpret and it is challenging to gain intuition into their behaviour. Also, careful hyperparameter tuning is required in order to achieve good results.

With help of RMSE have to conclude that FB prophet model performed good for stock market prediction with approximately 80% accuracy whereas the ARIMA and LSTM model gives 75% and 78% accuracy. By adding up of all these three models result. We will get the confidence interval of the prediction was in between 3 percent of lower and upper limit. Through this analysis, the two sectors are major leading companies so the investor have faith to invest their money and we can assure them with this accuracy we can say that how long you can hold the shares. Investors have ideas when they have to buy and at which time, they have to sell it. Through this we conclude our prediction on this company investor can make decision for the future use.

# CODING FOR ARIMA MODEL:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("/content/NAME OF THE COMPANY.csv")
df
df.info()
df.describe()
df.Data= pd.to_datetime(df.Data)
df2 =df.set_index('Data')
plt.figure(figsize=(20,12))
plt.subplot(2, 1, 1)
plt.title('Eicher motor Stock Price')
plt.plot(df2.Close,label='Close price')
plt.legend()
plt.subplot(2, 1, 2)
plt.title('Volume Traded')
plt.bar(x=df2.index,height=df2['Volume'])
plt.show()
data = list(df2["Close"])
from statsmodels.tsa.stattools import adfuller

result = adfuller(data)
print("1. ADF : ",result[0])
print("2. P-Value : ", result[1])
print("3. Num Of Lags : ", result[2])
print("4. Num Of Observations Used For ADF Regression:", result[3])
print("5. Critical Values :")
for key, val in result[4].items():
  print("\t",key, ": ", val)
  !pip install pmdarima
from pmdarima.arima.utils import ndiffs
d_value = ndiffs(data,test = "adf")
print("d value:", d_value)
from statsmodels.tsa.arima.model import ARIMA
from pmdarima import auto_arima
x_train= data[:-100]
x_test = data[-100:]
print(len(x_train),len(x_test))
stepwise_fit = auto_arima(data,trace=True,suppress_warnings=True)
```

```
print(stepwise_fit.summary())
import statsmodels.api as sm
model = sm.tsa.arima.ARIMA(data, order=(4,2,0))
from statsmodels.tsa.arima.model import ARIMA
model = model.fit()
model.summary()
start=len(x_train)
end=len(x_train)+len(x_test)-1
pred = model.predict(start=start,end=end)
pred
s = pd.Series(pred, index =df2.index[-100:])
s
plt.figure(figsize=(10,6), dpi=100)
df2['Close'][-100:].plot(label='Actual Stock Price', legend=True)
s.plot(label='Predicted Price', legend=True,)
from statsmodels.graphics.tsaplots import plot_predict
plot_predict(model, start = len(data)-500, end = len(data)+10, dynamic = False);
from sklearn.metrics import mean_squared_error
np.sqrt(mean_squared_error(x_test,pred))
from sklearn.metrics import r2_score
r2_score(x_test,pred)
pred_future = model.predict(start=end,end=end+30)
pred_future
import datetime
start_date = datetime.datetime(2022,12,30)
dates = [start_date + datetime.timedelta(days=idx) for idx in range(31)]
pred_future2 = pd.Series(pred_future, index = dates)
pred_future2
plt.figure(figsize=(10,6), dpi=100)
df2['Close'][-200:].plot(label='Actual Stock Price', legend=True)
pred_future2.plot(label='Future Predicted Price', legend=True)
```