Stock Price Prediction using Machine Learning and Deep Learning Approaches: A Comparative Study

Master Of Computer Application

(MCA)

under

Faculty Of Engineering & Technology

By

Name: **Goutam Biswas**

Examination Roll No.: **MCA2340003**

Registration No.: **160115 of 2021-2022**

Under the Guidance of

Prof. Ram Sarkar

Department of Computer Science & Engineering

Department of Computer Science & Engineering

Jadavpur University

Kolkata – 700032

2023

Certificate

I hereby recommend that the project entitled "***Stock Price Prediction using Machine Learning and Deep Learning Approaches: A Comparative Study"*** prepared under my supervision by **Goutam Biswas**, Exam-Roll: **MCA2340003**, be accepted for the degree of **Master of Computer Application** of **Jadavpur University, Kolkata.**

Supervisor

Prof. Ram Sarkar

Department of Computer Science & Engineering

Jadavpur University, Kolkata - 32

Prof. Nandini Mukherjee

Head of the Department

Department of Computer Science & Engineering

Jadavpur University, Kolkata – 32

Prof. Ardhendu Ghoshal

Dean

Faculty of Engineering & Technology

Jadavpur University, Kolkata – 32

Certificate of Approval*

The foregoing project "***Stock Price Prediction using Machine Learning and Deep Learning Approaches: A Comparative Study***" at instance is hereby approved as a creditable study of an engineering subject carried out and presented in a manner of satisfactory to warrant its acceptance as pre-requisite to the degree for which it has been submitted. It is notified to be understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed and conclusion drawn there in but approve the project only for the purpose for which it has been submitted.

Final Examination for the Evaluation of Project

Board of Examiners

_____

_____

(Signature of the Examiners)

**Only in case project is approved*

Acknowledgement

Despite the fact that my name appears only on the cover, a great many people have contributed to this project. This project is a result of the hard work and dedication of many people, and they have made my postgraduate experience one of the most memorable of my life.

First and foremost, I would like to express my deepest gratitude and sincere thanks to my adviser, Prof. Ram Sarkar, Professor, Department of Computer Science & Engineering, Jadavpur University, for his insightful suggestions, guidance, encouragement and attentive supervision during all stages of my research. Having him as my supervisor was incredibly fortunate since he allowed me the freedom to explore on my own while also guiding me to recover when I faltered. This has been an excellent learning experience for me.

I am grateful to him for allowing me to use the computing facilities of "*Centre for Microprocessor Application for Training Education and Research (CMATER)*" laboratory, Department of Computer Science and Engineering, Jadavpur University.

I am thankful to Prof. Nandini Mukherjee, Head of the Department of Computer Science & Engineering, Jadavpur University for all the necessary help I got during my project work.

I am also thankful to the system staff who maintained all the machines in my lab so efficiently that I never had to worry about viruses, losing files, creating backups or installing software.

Throughout these challenging years, I have received immense support and care from my dear friends, who have played a crucial role in keeping me grounded and motivated as I pursued my postgraduate studies. Their unwavering belief in me has been invaluable, and I cherish their friendship dearly.

Above all, I owe a debt of gratitude to my loving family, to whom I dedicate this dissertation. Their love, support, and unwavering encouragement have been a constant source of strength, enabling me to navigate through difficulties and pursue my academic aspirations. I am deeply grateful for the profound impact they have had on my life.

Date: _____

Place: Kolkata

Goutam Biswas

M.C.A. (C.S.E.)

Roll: **MCA2340003**

Contents

Chapter 1

1. ***Introduction***

Stock Market is characterized as dynamic, unpredictable and non-linear in nature. Predicting stock prices is a challenging task as it depends on various factors including but not limited to political conditions, global economy, company's financial reports and performance etc. Thus, to maximize the profit and minimize the losses, techniques to predict values of the stock in advance by analysing the trend over last few years, prove to be highly useful for making stock market movements [1].

Traditionally, two main approaches have been proposed for predicting the stock price of an organization. Technical analysis method uses historical price of stocks like closing and opening price, volume traded, adjacent close values etc. of the stock for predicting the future price of the stock. The second type of analysis is Qualitative, which is performed on the basis of external factors like company profile, market situation, political and economic factors, textual information in the form of financial new articles, social media and even blogs by economic analyst [1].

Now-a-days, advanced intelligent techniques based on either technical or fundamental analysis are used for predicting stock prices. For stock data, the data size can be unimaginably large & also non-linear. To deal with these, efficient model is needed that can identify the hidden patterns & complex relations in this huge dataset. Machine learning techniques in this area have proved to improve efficiencies by 60-86 percent as compared to the past methods [1].

Several studies have been the subject of using machine learning in the quantitative financial, predicting prices of managing and constricting entire portfolio of assets, as well as, investment process, and many other operations can be covered by machine learning algorithms. In general machine learning is a term used for all algorithm's methods using computers to reveal patterns based only on data and not using any programming instructions. For quantitative finance and specially assets selections several models supply a large number of methods that can be used with machine learning to forecast future assets value [2].

1.1 Problem Definition

Due to the complexity and dynamic nature of financial markets, predicting stock prices is a challenging task. Traditional statistical methods have limitations in capturing the underlying patterns and complexities of financial data, leading to inaccurate predictions. As a result, there has been a growing interest in applying machine learning and neural network techniques to stock price prediction. However, despite the progress made in this field, there are still several challenges to be addressed. These include issues related to data quality, feature selection, model selection, and interpretability. Additionally, the effectiveness of these models may be affected by changes in market conditions and unexpected events, such as natural disasters or political upheavals. So, the purpose of this study is to investigate the effectiveness of Machine Learning and Deep Learning techniques for stock price prediction and to address the challenges associated with them.

1.2 Applications

Stock Price Prediction can be significantly impacted by the application of Machine Learning and Deep Learning. Models like these can be used to generate more accurate predictions of stock prices, thereby improving investment decisions and increasing profitability for investors and financial institutions. Moreover, these techniques can be used for portfolio optimization, risk management, and other applications in finance. Furthermore, machine learning and neural network techniques for stock price prediction can drive innovation and advance finance. Therefore, these techniques could benefit both individual investors and the financial industry in general.

1.3 Motivation

There has been a significant amount of interest in both academia and industry in the application of Machine Learning and Deep Learning techniques to stock price prediction. In addition to the potential profitability of accurately predicting stock prices, these models are efficient in processing and analysing large amounts of financial data, are capable of capturing the complexity of various factors that affect stock prices, and offer the possibility

of developing new techniques that can be innovative. As a result, there is a growing body of research focused on using machine learning and neural network models to improve investment decision-making and increase profits in the financial industry.

1.4 Organisation of Project Work

In this present work, some Machine Learning and Neural Network based Deep Learning technique(s) are adopted for the comparison of Stock Market data and ultimately to find the more precise model for prediction of stock prices.

Chapter 1. *Introduction*

In this chapter, a brief description of Stock Market data & its vastness, prediction techniques of stock price are explained. Then comes the problem definition, application of this particular research work & also includes the motivation for the same. This part ends with the Organisation of the Project.

Chapter 2. *Related Work*

This chapter describes some previous work regarding Stock Price Prediction using some Machine Learning & Deep Learning techniques.

Chapter 3. *Methods and Materials*

This chapter consists of the description of various techniques used throughout this project work.

Chapter 4. *Results & Analysis*

In this chapter, a detailed discussion of the used datasets, description of evaluation metrics and results of the used techniques of the project work are done.

Chapter 5. *Conclusion & Future Scope*

Finally, the project work is concluded in this chapter and the future scope of the work is specified.

Chapter 2

2. ***Related Work***

The prediction of stock market prices has always attracted many researchers due to the enormous factors including but not limited to political conditions, global economy, company's financial reports and performance and many more which makes the prediction of stock pricing a bit more challenging. The present project work focuses on the prediction of stock prices using machine learning & deep learning techniques.

The work [2] proposes RNN based on LSTM to forecast future values using the opening prices for the assets by varying the number of epochs and showed the precision of forecasting by comparing the losses in the processing times of each epoch.

Another work [1] compared the predictions of stock values using Random Forest (RF) and Artificial Neural Network (ANN) for the five companies: Nike, Goldman Sachs, JP Morgan and Co., Johnson & Johnson and Pfizer Inc. For ANN, six new variables have been created for predicting stock closing price: Stock High minus Low price (H-L), Stock Close minus Open price (O-C), Stock price's seven days' moving average (7 DAYS MA), Stock price's fourteen days' moving average (14 DAYS MA), Stock price's twenty-one days' moving average (21 DAYS MA), Stock price's standard deviation for the past seven days (7 DAYS STD DEV); these six variables used as input layers in the ANN. In case of RF, these new variables are provided for each decision tree and training variables predicted the next day closing price of the stock for a company. In case of RNN, the RMSE values are in between 1.10 to 3.30, MAPE values range from 0.70% to 1.09%; but in case of RF, these values range from 0.43 to 3.40 and 0.75% to 1.14% indicating to the fact that ANN has been proved to be a better technique.

In [3], the authors have shown a comparison by changing the number of parameters in LSTM and also varying the number of epochs on the stock returns of NIFTY 50. Their training RMSE and testing RMSE ranged from 0.00983 to 0.01511 and 0.00859 to 0.014 and their experimental results showed that by taking 4

features set (High/Low/Open/Close) with 500 epochs would achieve the best results.

The authors in [4] have utilised some techniques of deep learning like Auto-regressive Integrated Moving Average (ARIMA), Multi-Layer Perceptron (MLP), LSTM and machine learning technique SVR and compared accuracies among them.

The work [5] also showed short term stock prediction on stock data of 10 trading days of 10 stocks listed on the New York Stock Exchange over a period of 1 year using MLP and LSTM. The authors have achieved average case and best case RMSE for LSTM to be 4.8×10^{-2} and 1.88×10^{-2} & for MLP, 2.5×10^{-3} and 9.37×10^{-4} and it has been observed that MLP has outperformed LSTM in predicting short term stock prices.

Lin, Yuling et al. [6] showed a quasi-linear SVM based stock market trend prediction system with various SVM kernels on stock data of these: Hon Hai, Taiwan Semiconductor, Evergreen & Taiwan50 Index.

In [7], the authors also proposed a prediction model based on RBF-SVM algorithm.

Heo, Junyoung et al. [8] utilised a prediction based on financial statements using SVM.

In the work [9], the authors have shown prediction on one year stock price data of Coca-Cola using Linear Regression and SVM. They have achieved RMSE, MAE, MSE and R-Squared value for Linear Regression: 3.22, 2.53. 10.37 & 0.73 and for SVM: 1.58, 1.33, 2.51 & 0.93; by observation they had concluded that SVM performed better than Linear Regression.

Panwar, Bhawna et al. [10], have predicted the stock prices of Amazon and achieved accuracy percentages for Linear Regression and SVR prediction models as 98.76% and 94.32% accordingly, which indicates that Linear Regression is performing better in this case.

Another work [11] also showed a prediction on TCS stock data based on Linear Regression, Polynomial Regression and RBF and achieved Confidence Values of 0.9774, 0.468 & 0.5652 accordingly, indicating that Linear Regression is performing best in this case.

Chapter 3

3. ***Methods and Materials***

In this chapter, the methods used for experimentation in this project work have been described.

3.1 Linear Regression

Linear Regression is a statistical method used to study the relationship between two variables by fitting a linear equation to the data. It is one of the most commonly used methods for modeling the relationship between a dependent variable and one or more independent variables. The goal of linear regression is to find the best fit line that explains the variation in the data.

Fig 3.1: *Illustration of Linear Regression Model* [12]

In simple linear regression, there is only one independent variable, and the relationship between the independent variable and the dependent variable is modeled using a straight line. The equation of the line is given by:

$y = mx + b$

where y is the *dependent variable*, x is the *independent variable*, m is the slope of the line, and b is the y-intercept. The slope of the line indicates the change in the dependent variable for every unit change in the independent variable [12].

3.2. Support Vector Machine (SVM)

SVM based regression is a type of machine learning technique used to build a model that predicts a continuous target variable. Unlike SVM classification, where the goal is to separate data into different classes, SVM regression aims to predict a continuous output value based on input variables. In Support Vector Regression (SVR), the straight line that is required to fit the data is referred to as *Hyperplane* [13].

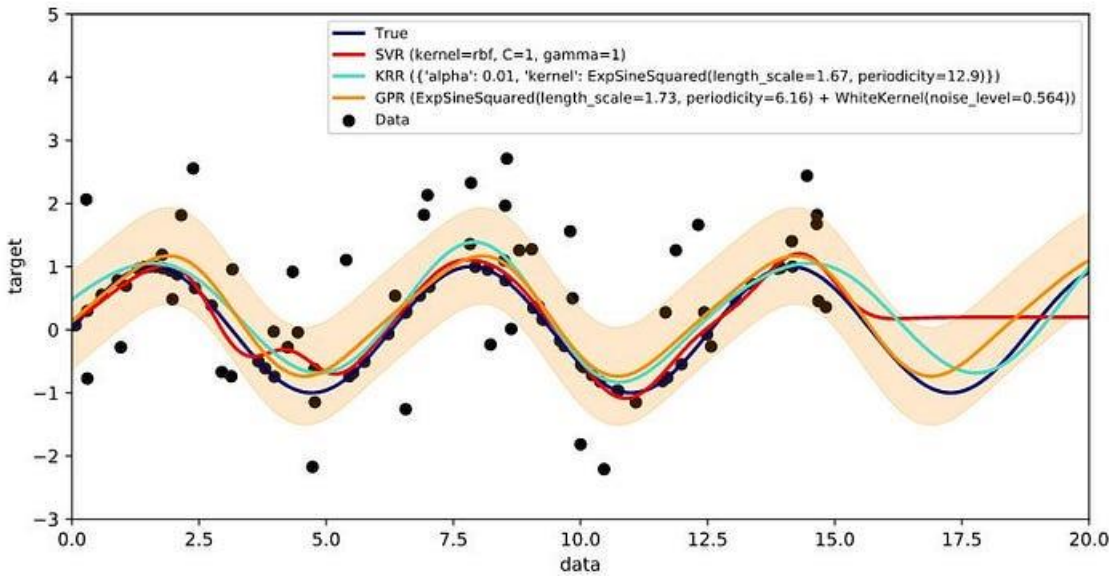


Fig 3.2: Illustration of *Support Vector Machine based Regression Model* [13]

The objective of a support vector machine algorithm is to find a hyperplane in an n-dimensional space that distinctly classifies the data points. The data points on either side of the hyperplane that are closest to the hyperplane are called Support Vectors. These influence the position and orientation of the hyperplane and thus help build the SVM.

The various hyperparameters of SVM are,

1. *Hyperplane*:

   Hyperplanes are decision boundaries that is used to predict the continuous output. The data points on either side of the hyperplane that are closest to the hyperplane are called Support Vectors. These are used to plot the required line that shows the predicted output of the algorithm.

2. *Kernel*:

A Kernel is a set of mathematical functions that takes data as input and transform it into the required form. These are generally used for finding a hyperplane in the higher dimensional space. The most widely used kernels include Linear, Non-Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid. By default, RBF is used as the kernel. Each of these kernels are used depending on the dataset.

3. *Boundary Lines*:

These are the two lines that are drawn around the hyperplane at a distance of ε (epsilon). It is used to create a margin between the data points.

The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points [13].

3.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is one of many types of Recurrent Neural Network RNN, it's also capable of catching data from past stages and use it for future predictions [14].

A RNN is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (nlp), speech recognition, and image captioning; they are incorporated into popular applications such as Siri, voice search, and Google Translate. Like feedforward and convolutional neural networks (CNNs), recurrent neural networks utilize training data to learn. They are distinguished by their "memory" as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depend on the prior elements within the sequence. While future events would also be helpful in determining the output of a given sequence, unidirectional recurrent neural networks cannot account for these events in their predictions [15].

The earlier stages of data should be remembered to predict and guess future values, in this case the hidden layer act like a stock for the past information from the sequential data. The term recurrent is used to describe the process of using elements of earlier sequences to forecast future data [2].

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behaviour, not something they struggle to learn!

As shown in Fig 3.3(a), all RNN(s) have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

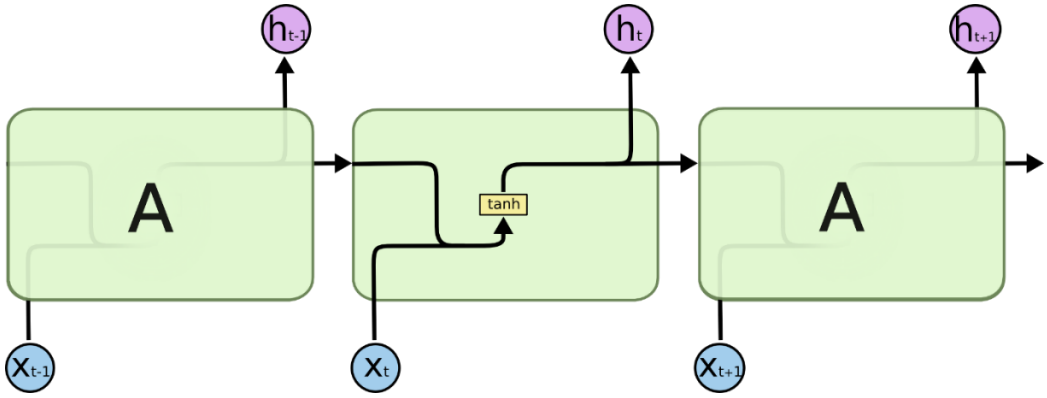


Fig 3.3(a): *Internal Structure of RNN containing a single layer* [16]

As in Fig 3.3(b), LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way [16].

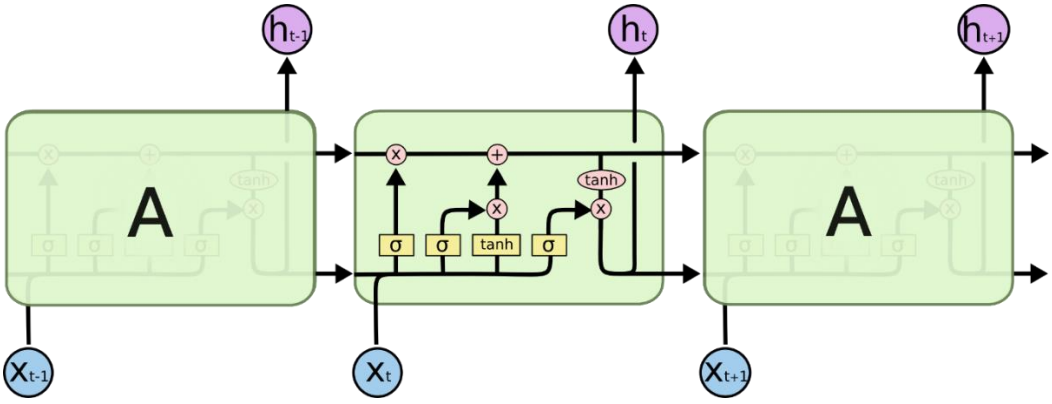


Fig 3.3(b): *Internal Structure of LSTM containing four interacting layers* [16]

The ability of memorizing sequence of data makes the LSTM a special kind of RNNs. Every LSTM node most be consisting of a set of cells responsible of storing passed data streams, the upper line in each cell links the models as transport line handing over data from the past to the present ones, the independency of cells helps the model dispose filter of add values of a cell to another. In the end the sigmoidal neural network layer composing the gates drive the cell to an optimal value by disposing or letting data pass through. Each sigmoid layer has a binary value (0 or 1) with 0 “let nothing pass through”; and 1 “let everything pass through.” The goal here is to control the state of each cell, the gates are controlled as follows:

- *Forget Gate*: It outputs a number between 0 and 1, where 1 illustration “completely keep this”; whereas, 0 indicates “completely ignore this.”

- *Memory Gate*: It chooses which new data will be stored in the cell. First, a sigmoid layer “input door layer” chooses which values will be changed. Next, a tanh layer makes a vector of new candidate values that could be added to the state.

- *Output Gate*: It decides what will be the output of each cell. The output value will be based on the cell state along with the filtered and freshest added data [2].

Chapter 4

4. ***Results and Analysis***

4.1 Description of Data & Feature Selection

The Time-Series Data of the stock prices for these three companies has been collected from *Kaggle*. Table: 4.1 shows the statistics of the databases used in this work,

Dataset	Range
Wipro	03/01/2000 - 30/04/2021
Larsen & Toubro	23/06/2004 - 30/04/2021
Google	02/01/2013 - 27/01/2023

Table 4.1: *Statistics of Datasets Used*

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
0	2000-01-03	WIPRO	EQ	2522.40	2724.00	2724.20	2724.00	2724.20	2724.20	2724.17	1599	4.355942e+11	NaN	NaN	NaN
1	2000-01-04	WIPRO	EQ	2724.20	2942.15	2942.15	2942.15	2942.15	2942.15	2942.15	4392	1.292192e+12	NaN	NaN	NaN
2	2000-01-05	WIPRO	EQ	2942.15	2942.15	3177.55	2715.00	3000.00	2990.10	3063.86	132297	4.053390e+13	NaN	NaN	NaN
3	2000-01-06	WIPRO	EQ	2990.10	3144.70	3183.00	2790.00	2915.00	2932.25	2962.41	72840	2.157822e+13	NaN	NaN	NaN
4	2000-01-07	WIPRO	EQ	2932.25	2751.00	2751.00	2697.70	2697.70	2697.70	2697.95	10110	2.727630e+12	NaN	NaN	NaN

Fig 4.1(a): *Some Samples taken from Wipro Stock Dataset*

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
0	23-06-2004	LT	EQ	2.00	500.0	745.0	150.0	638.00	635.95	627.71	2699293	1.690000e+14	NaN	395612	0.1466
1	24-06-2004	LT	EQ	635.95	630.0	630.0	608.4	612.00	616.00	618.21	913575	5.650000e+13	NaN	305751	0.3347
2	25-06-2004	LT	EQ	616.00	619.0	679.0	616.1	665.00	668.30	658.45	1637383	1.080000e+14	NaN	217950	0.1331
3	28-06-2004	LT	EQ	668.30	660.0	671.9	640.2	653.00	656.40	657.39	913349	6.000000e+13	NaN	194984	0.2135
4	29-06-2004	LT	EQ	656.40	656.0	683.4	653.0	666.85	671.70	675.11	743063	5.020000e+13	NaN	101786	0.1370

Fig 4.1(b): *Some Samples taken from Larsen & Toubro Stock Dataset*

	Unnamed: 0	Date	Open	High	Low	Close	Adj Close	Volume
0	0	2013/01/02	18.003504	18.193193	17.931683	18.099348	18.099348	101550348
1	1	2013/01/03	18.141392	18.316566	18.036036	18.109859	18.109859	92635272
2	2	2013/01/04	18.251753	18.555305	18.210211	18.467718	18.467718	110429460
3	3	2013/01/07	18.404655	18.503002	18.282784	18.387136	18.387136	66161772
4	4	2013/01/08	18.406906	18.425926	18.128880	18.350851	18.350851	66976956

Fig 4.1(c): *Some Samples taken from Google Stock Dataset*

In Fig 4.1(a)(b)(c), among these feature sets of these datasets, the *'Close'* feature set of each database has been used in various model for prediction.

4.2 Evaluation Metrics

4.2.1 RMSE: It stands for *Root Mean Squared Error*. It is a popular performance metric used in regression analysis to measure the difference between the predicted values and actual values of a dataset.

RMSE is calculated by taking the square root of the average of the squared differences between the predicted and actual values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left|\left| y(i) - y_p(i) \right|\right|^2}{N}}$$

Fig 4.1: *Root Mean Squared Error (RMSE), where N is the number of data points, y(i) is the i^{th} measurement, and $y_p(i)$ is its corresponding prediction* [17]

4.2.2 MAPE: It stands for *Mean Absolute Percentage Error*. This is a relative error measure that uses absolute values to keep the positive and negative errors from cancelling one another out and uses relative errors to enable to compare forecast accuracy between time-series models.

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

Fig 4.2: *Mean Absolute Percentage Error (MAPE), where n is the number of fitted points, A_t & F_t are the Actual & Forecast value* [18]

4.3 Using Linear Regression

4.3.1 Wipro

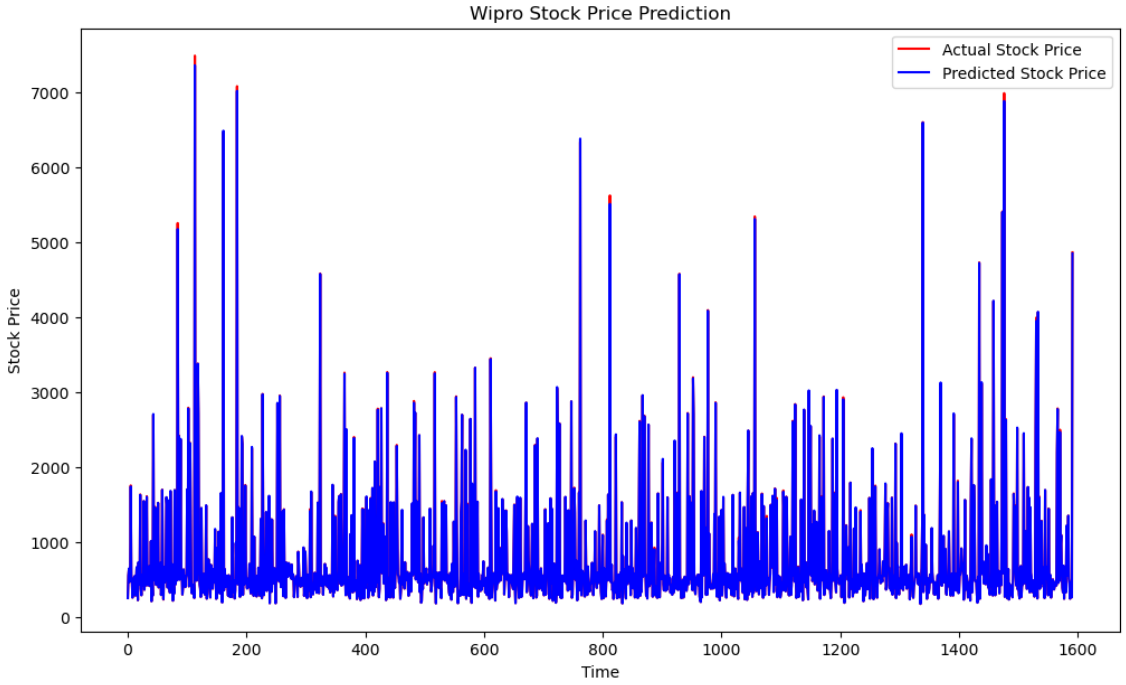


Fig 4.3.1(a): *Graph showing the prediction with 70%-30% split of Training & Testing Data*

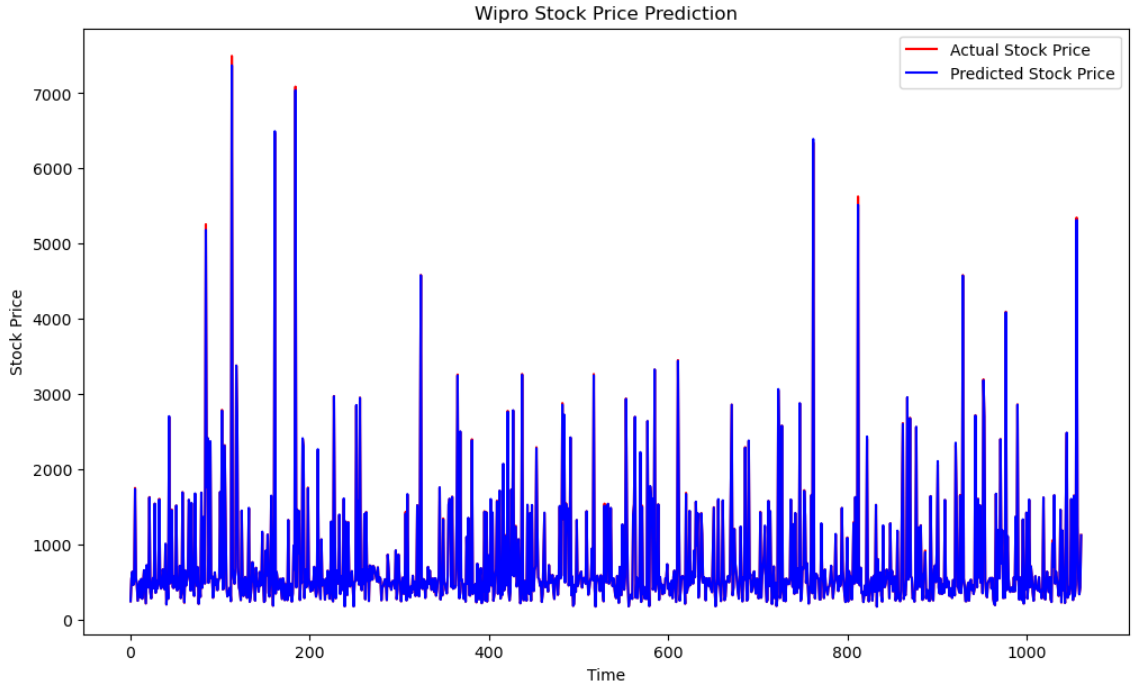


Fig 4.3.1(b): *Graph showing the prediction with 80%-20% split of Training & Testing Data*

Training Data (%)	RMSE	MAPE
70	7.60581	0.00265
80	7.61259	0.00257

Table 4.3.1: *% of Training Data vs RMSE vs MAPE*

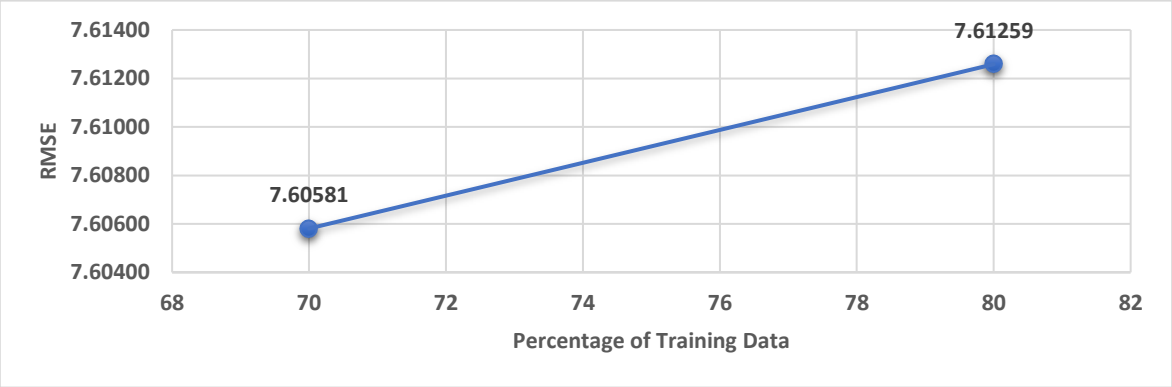


Fig 4.3.1(c): *Plotting of % of Training Data vs RMSE*

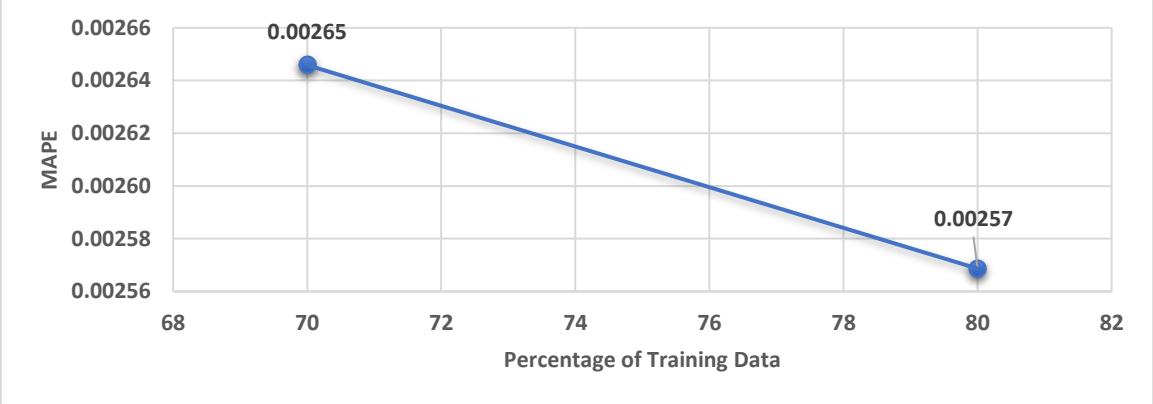


Fig 4.3.1(d): *Plotting of % of Training Data vs MAPE*

By observing the experimental results from Table 4.3.1 and Figures 4.3.1(c)(d), we can conclude that for this dataset, Linear Regression works the best when the dataset is used in 70-30 ratio for Training & Testing purpose.

4.3.2 Larsen & Toubro

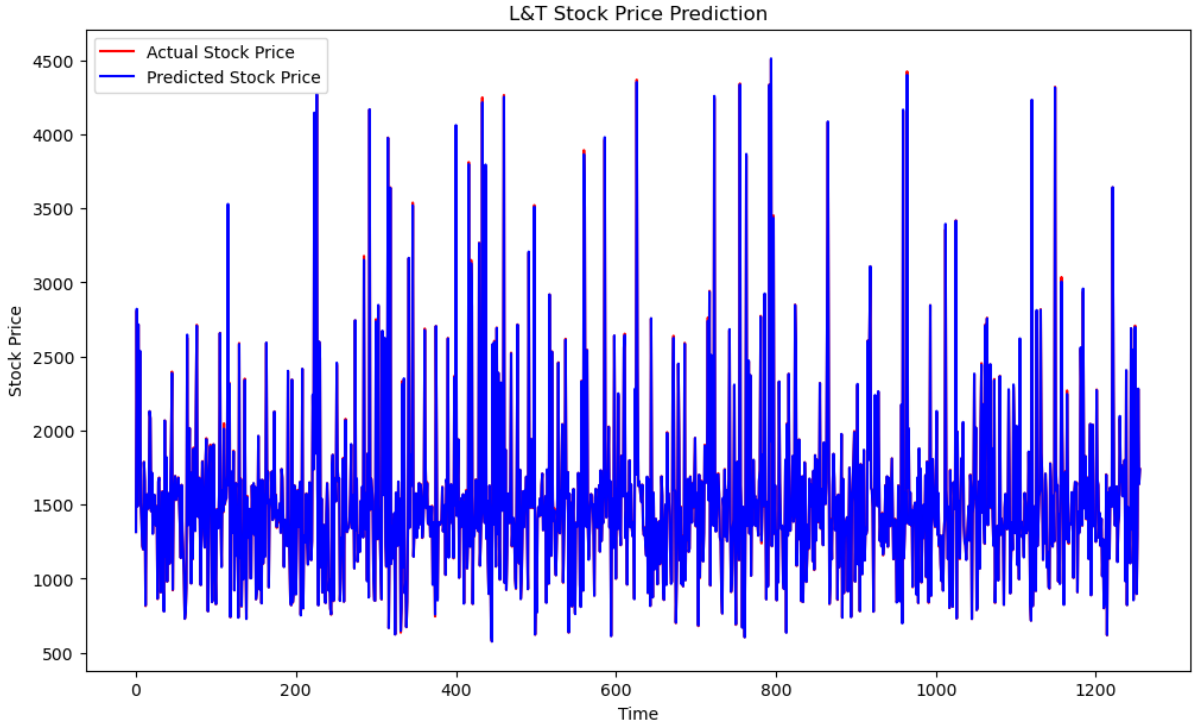


Fig 4.3.2(a): *Graph showing the prediction with 70%-30% split of Training & Testing Data*

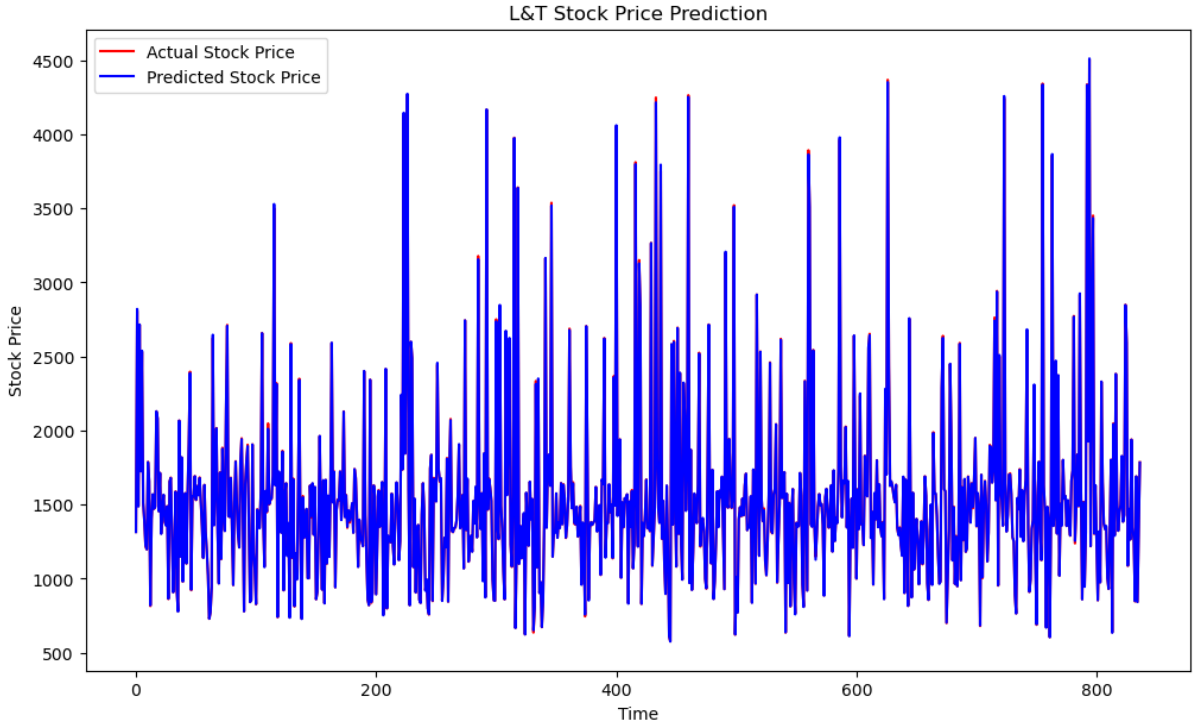


Fig 4.3.2(b): *Graph showing the prediction with 80%-20% split of Training & Testing Data*

Training Data (%)	RMSE	MAPE
70	5.12075	0.00189
80	5.24930	0.00193

Table 4.3.2: *% of Training Data vs RMSE vs MAPE*

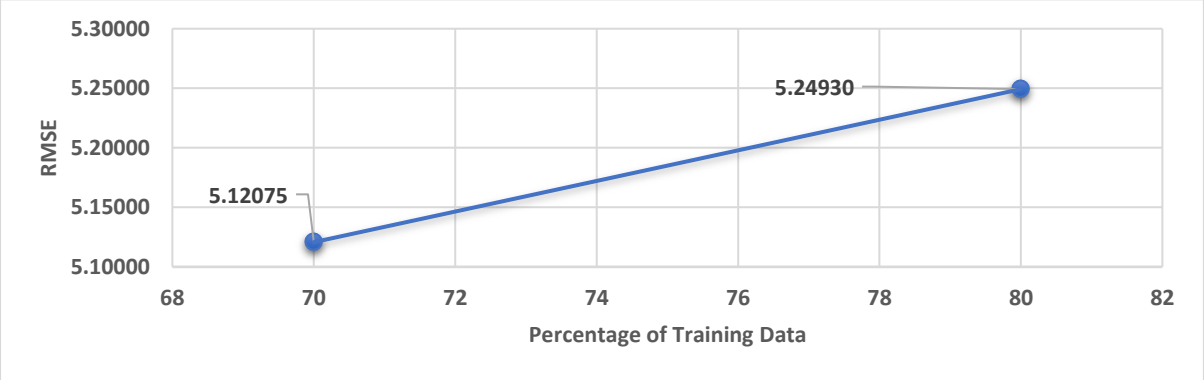


Fig 4.3.2(c): *Plotting of % of Training Data vs RMSE*

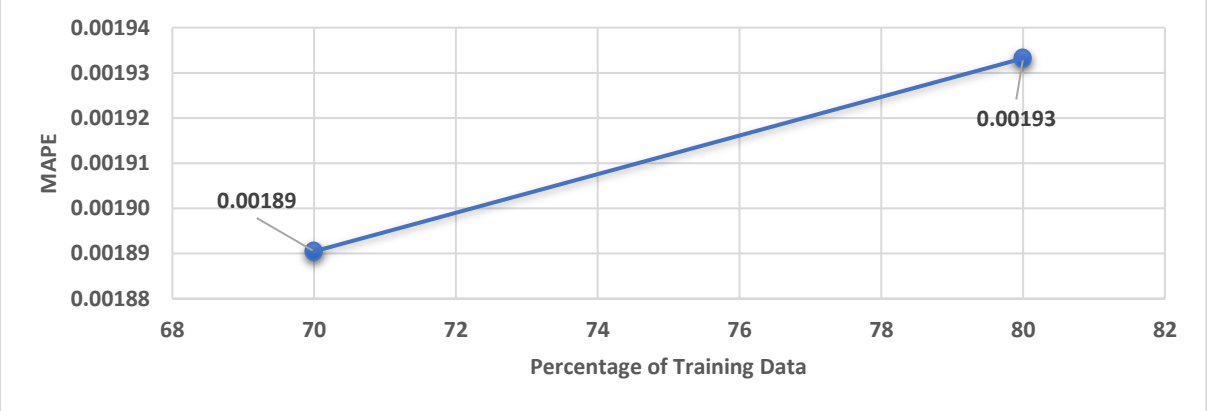


Fig 4.3.2(d): *Plotting of % of Training Data vs MAPE*

By observing the experimental results from Table 4.3.2 and Figures 4.3.2(c)(d), we can conclude that for this dataset, Linear Regression works the best when the dataset is used in 70-30 ratio for Training & Testing purpose.

4.3.3 Google

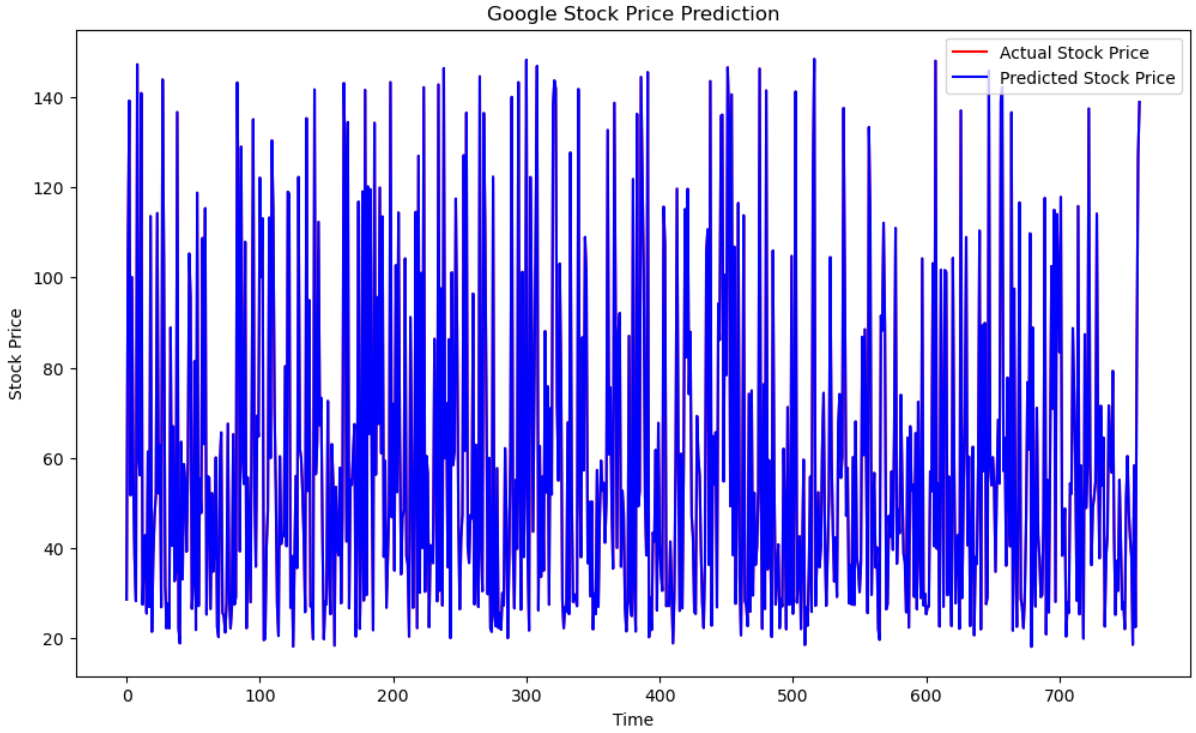


Fig 4.3.3(a): *Graph showing the prediction with 70%-30% split of Training & Testing Data*

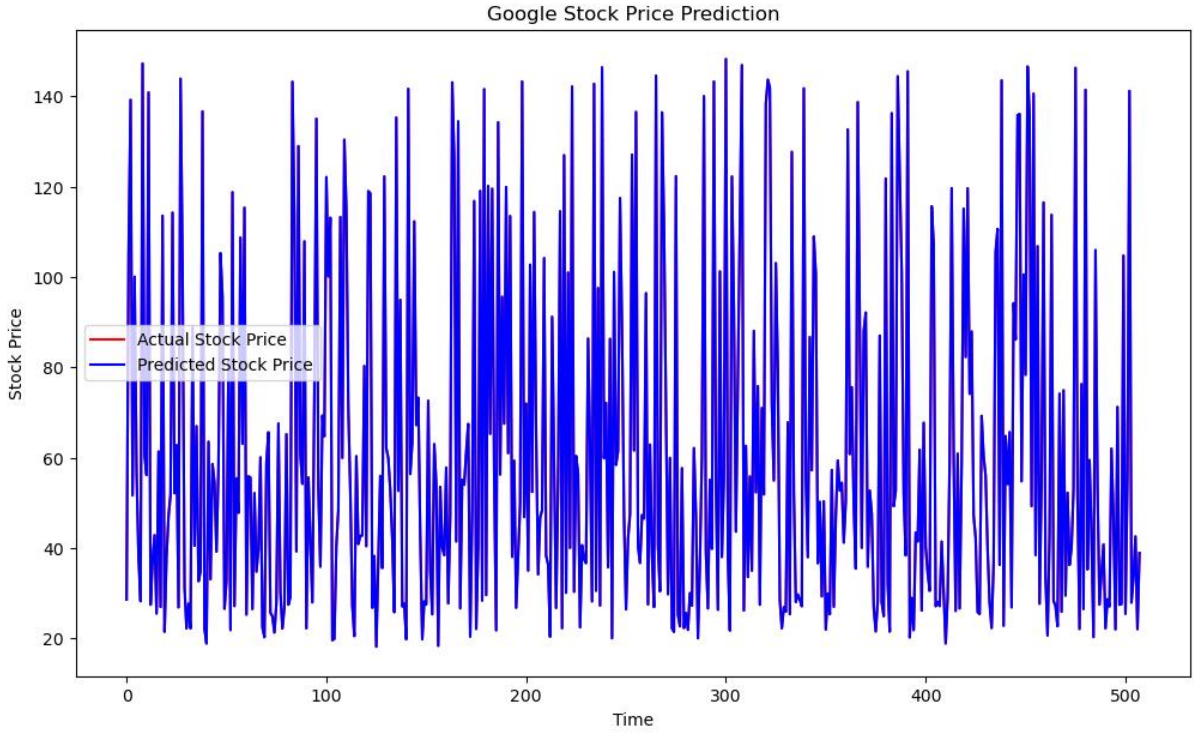


Fig 4.3.3(b): *Graph showing the prediction with 80%-20% split of Training & Testing Data*

Training Data (%)	RMSE	MAPE
70	1.99904E-14	1.94421E-16
80	4.31026E-14	8.39013E-16

Table 4.3.3: *% of Training Data vs RMSE vs MAPE*

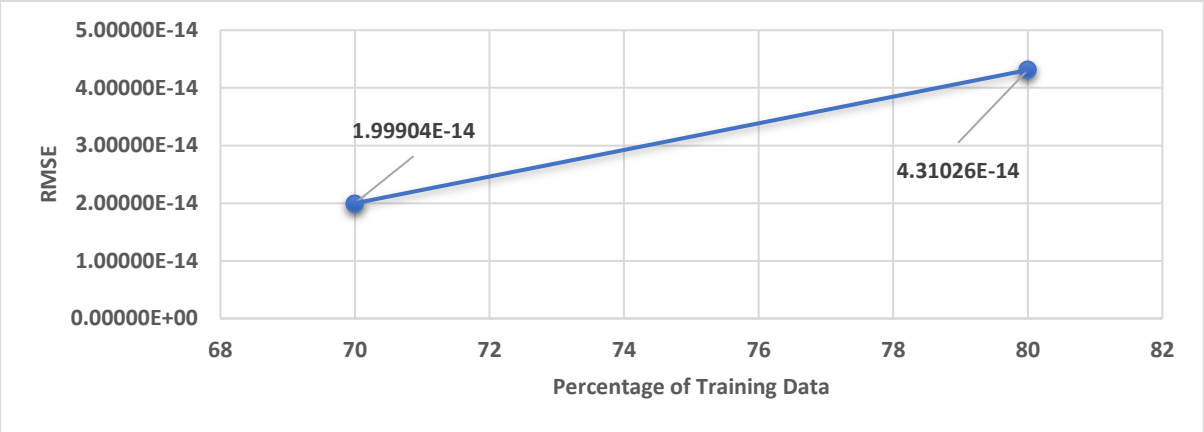


Fig 4.3.3(c): *Plotting of % of Training Data vs RMSE*

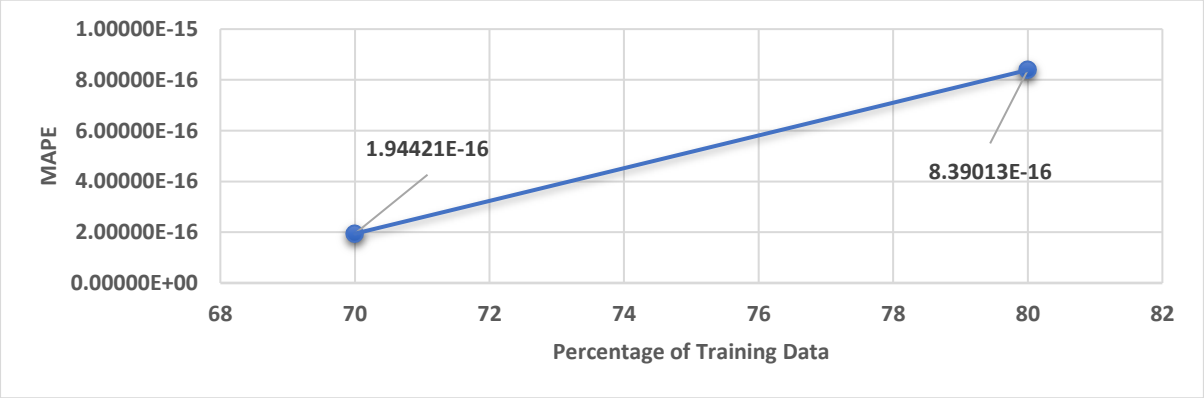


Fig 4.3.3(d): *Plotting of % of Training Data vs MAPE*

By observing the experimental results from Table 4.3.3 and Figures 4.3.3(c)(d), we can conclude that for this dataset, Linear Regression works the best when the dataset is used in 70-30 ratio for Training & Testing purpose.

4.4 Using Support Vector Machine (SVM)

4.4.1 Wipro

Training Data: 70%

Testing Data: 30%

Kernel: *Radial Basis Function (RBF)*

Gamma: 0.001

Epsilon: 0.1

C (Regularization Parameter)

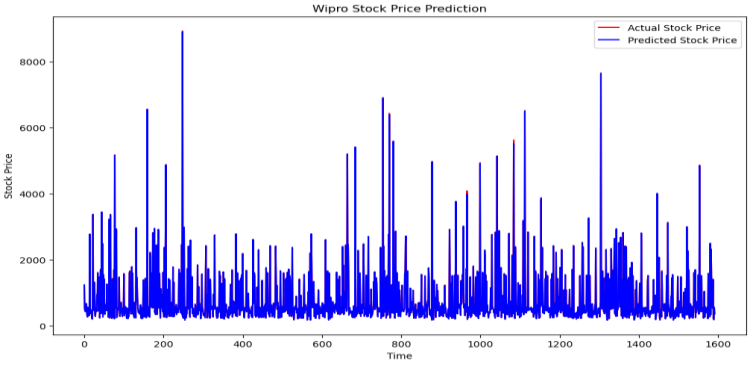


Fig 4.4.1(a): *Graph showing the prediction when C = 10000*

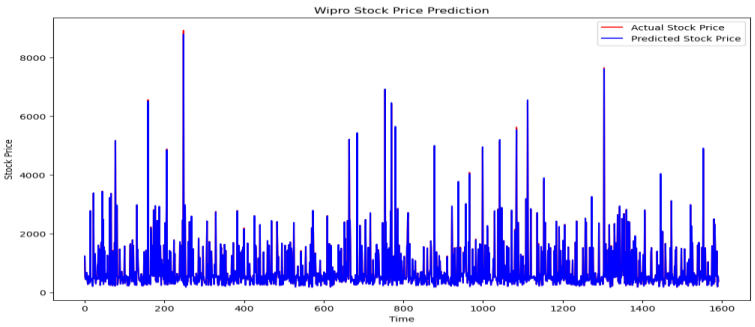


Fig 4.4.1(b): *Graph showing the prediction when C = 100000*

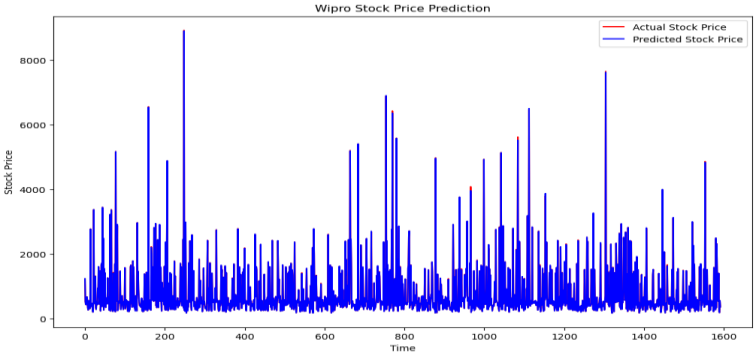


Fig 4.4.1(c): *Graph showing the prediction when C = 1000000*

Gamma	C	Epsilon	RMSE	MAPE
0.001	10000	0.1	9.01481	0.00311
0.001	100000	0.1	7.42278	0.00273
0.001	1000000	0.1	7.57091	0.00279

Table 4.4.1: *Gamma vs C vs Epsilon vs RMSE vs R2*

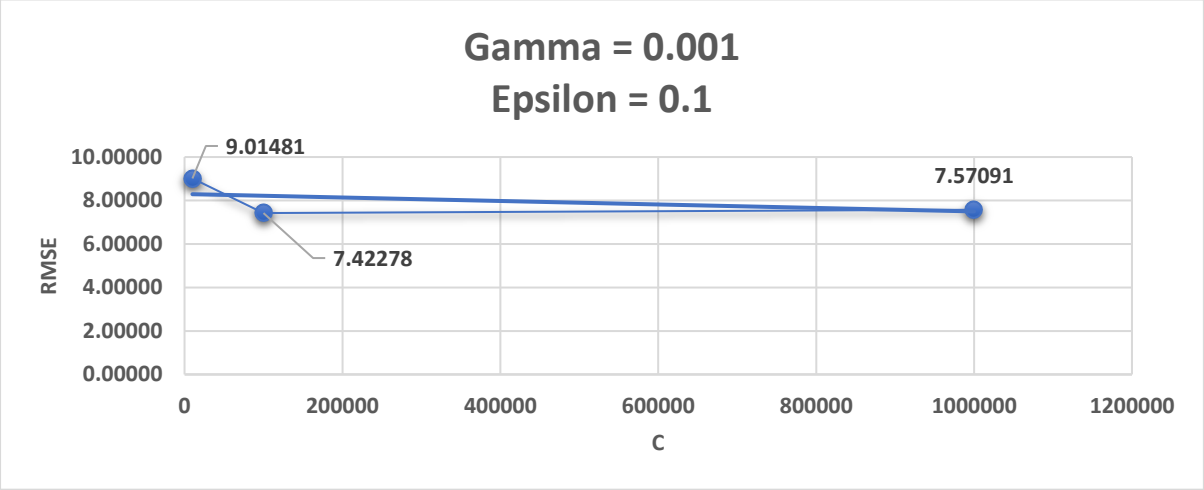


Fig 4.4.1(d): *Plotting of RMSE vs C*

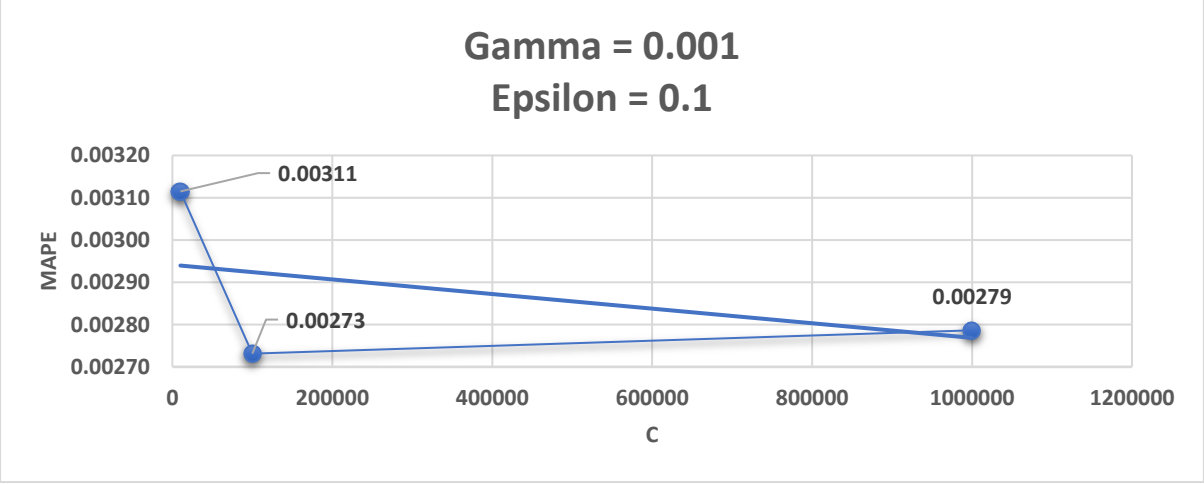


Fig 4.4.1(e): *Plotting of MAPE vs C*

By observing the experimental results from Table 4.4.1 and Figures 4.4.1(d)(e), we can conclude that for this dataset, SVM works the best for value of C = 100000.

4.4.2 Larsen & Toubro

Training Data: 70% C (Regularization Parameter)

Testing Data: 30%

Kernel: *Radial Basis Function (RBF)*

Gamma: 0.001

Epsilon: 0.1

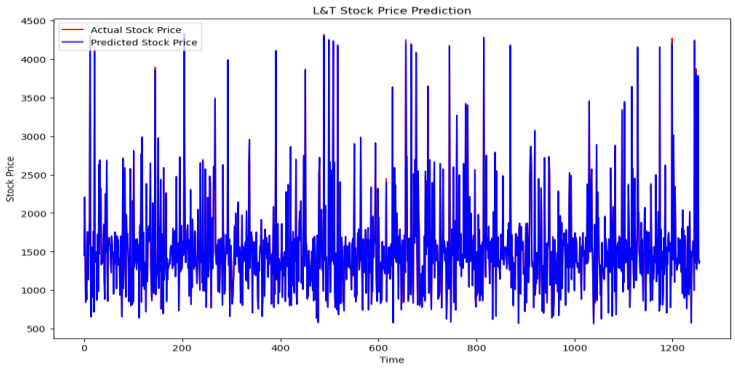


Fig 4.4.2(a): *Graph showing the prediction when C = 10000*

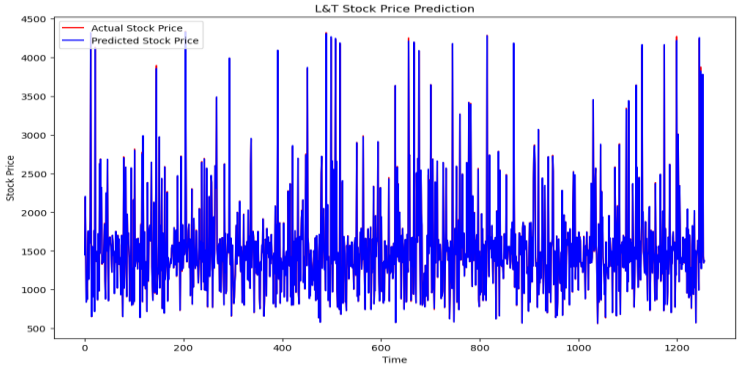


Fig 4.4.2(b): *Graph showing the prediction when C = 100000*

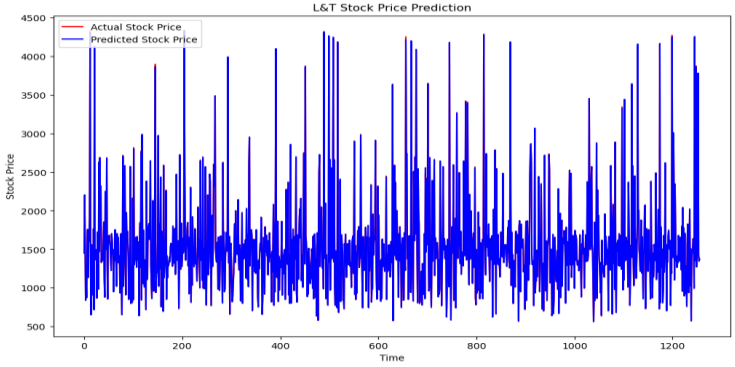


Fig 4.4.2(c): *Graph showing the prediction when C = 1000000*

Gamma	C	Epsilon	RMSE	MAPE
0.001	10000	0.1	6.07010	0.00212
0.001	100000	0.1	5.53083	0.00190
0.001	1000000	0.1	5.02804	0.00187

Table 4.4.2: *Gamma vs C vs Epsilon vs RMSE vs MAPE*

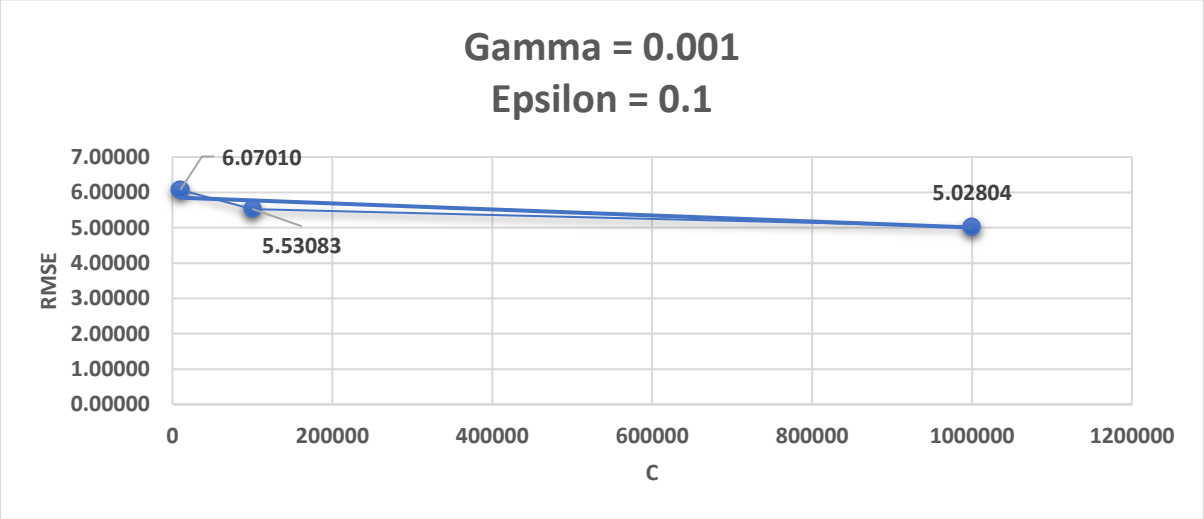


Fig 4.4.2(d): *Plotting of RMSE vs C*

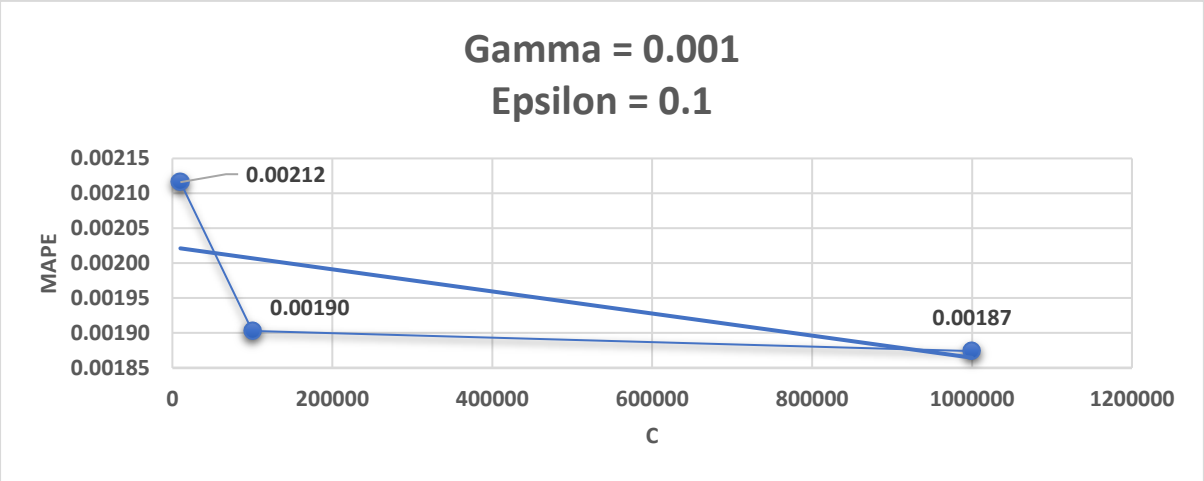


Fig 4.4.2(e): *Plotting of MAPE vs C*

By observing the experimental results from Table 4.4.2 and Figures 4.4.2(d)(e), we can conclude that for this dataset, SVM works the best for value of C = 1000000.

4.4.3 Google

Training Data: 70% C (Regularization Parameter)

Testing Data: 30%

Kernel: *Radial Basis Function (RBF)*

Gamma: 0.001

Epsilon: 0.1

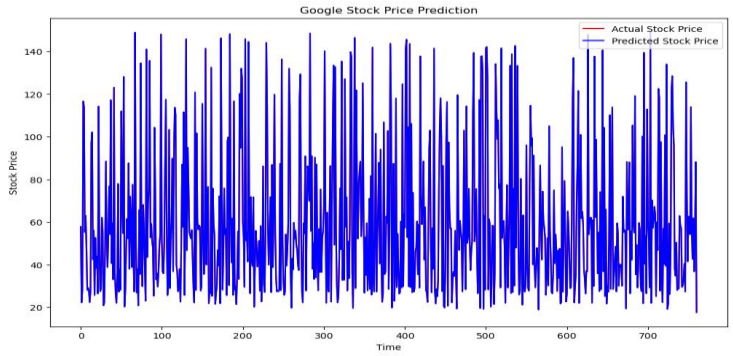


Fig 4.4.3(a): *Graph showing the prediction when C = 10000*

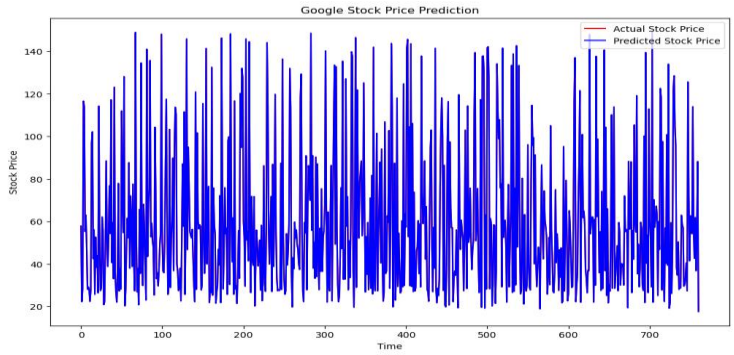


Fig 4.4.3(b): *Graph showing the prediction when C = 100000*

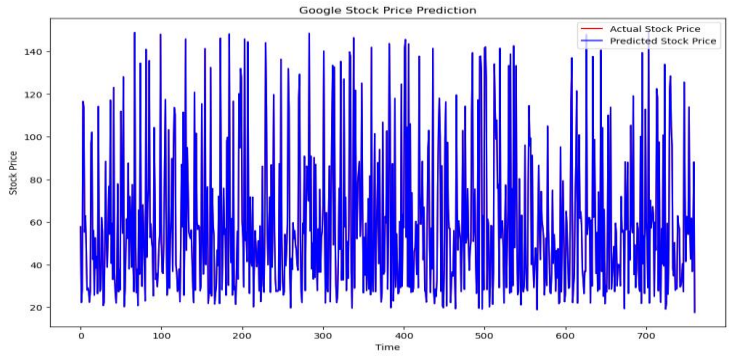


Fig 4.4.3(c): *Graph showing the prediction when C = 1000000*

Gamma	C	Epsilon	RMSE	MAPE
0.001	10000	0.1	0.05751	0.00099
0.001	100000	0.1	0.05071	0.00088
0.001	1000000	0.1	0.04969	0.00086

Table 4.4.3: *Gamma vs C vs Epsilon vs RMSE vs MAPE*

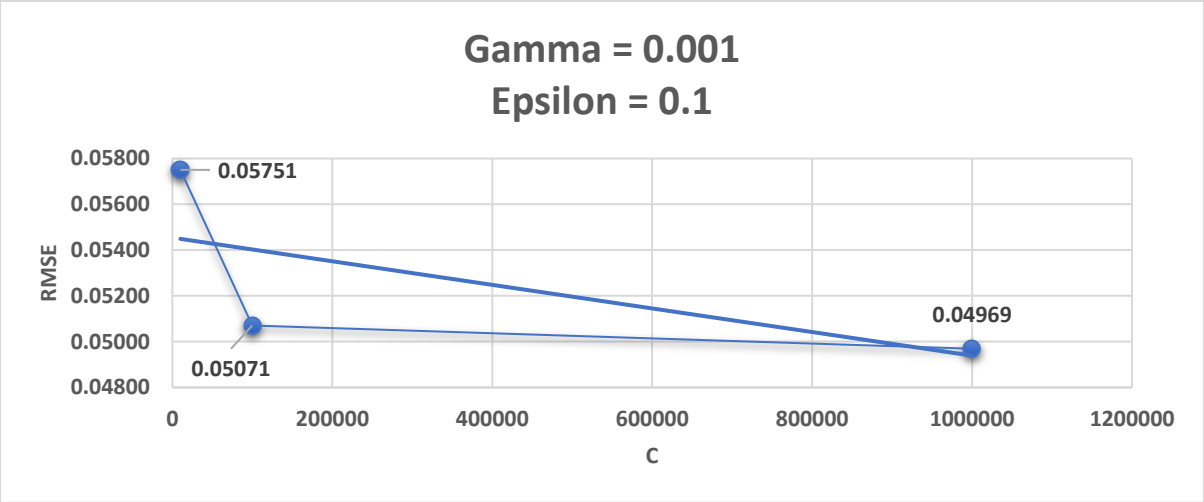


Fig 4.4.3(d): *Plotting of RMSE vs C*

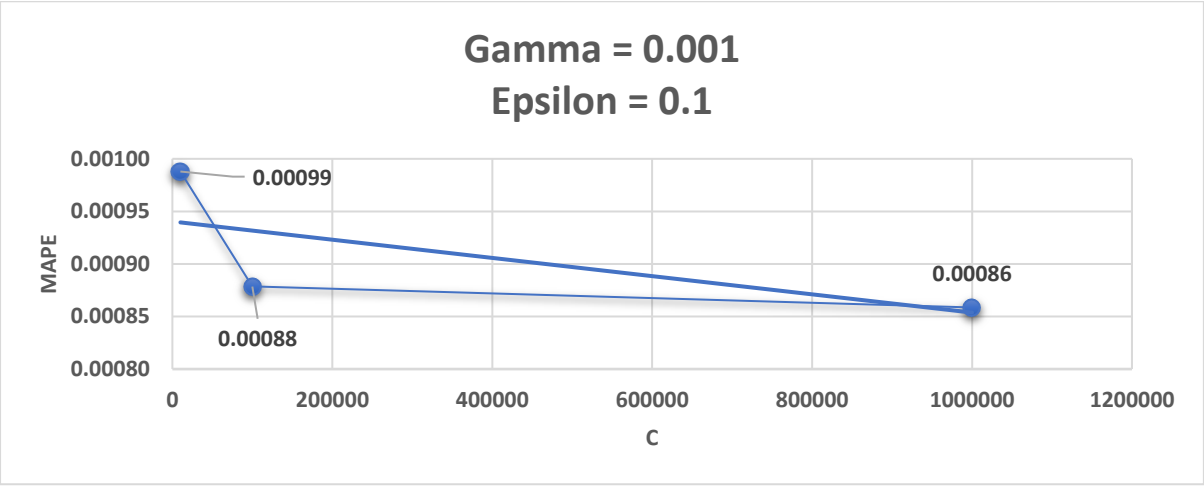


Fig 4.4.3(e): *Plotting of MAPE vs C*

By observing the experimental results from Table 4.4.3 and Figures 4.4.3(d)(e), we can conclude that for this dataset, SVM works the best for value of C = 1000000.

4.5 Using Long Short-Term Memory (LSTM)

4.5.1 Wipro

Training Dataset Length: 3715 (70%)

Testing Dataset Length: 1591 (30%)

For Batch Size = 60,

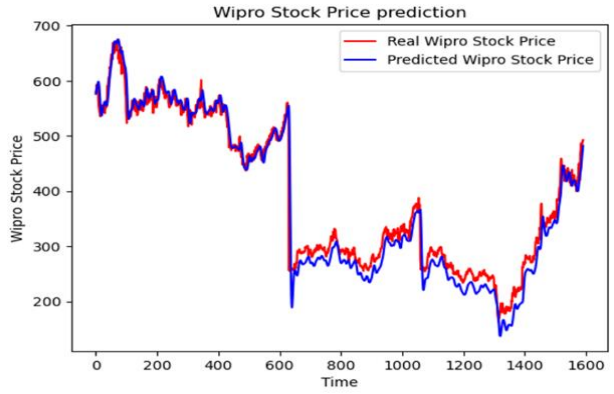


Fig 4.5.1(a): *LSTM Prediction Curve at Epochs = 20*

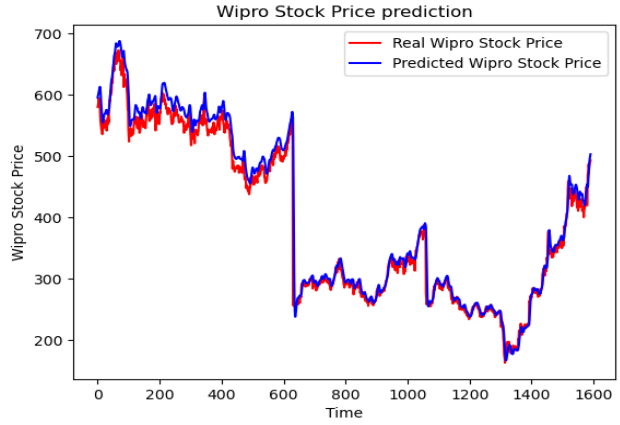


Fig 4.5.1(b): *LSTM Prediction Curve at Epochs = 50*

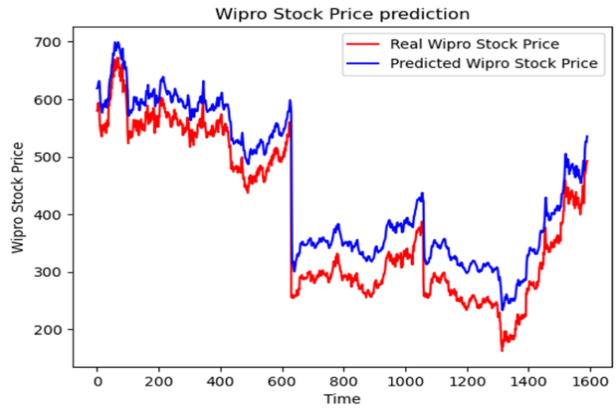


Fig 4.5.1(c): *LSTM Prediction Curve at Epochs = 100*

For Batch Size = 90,

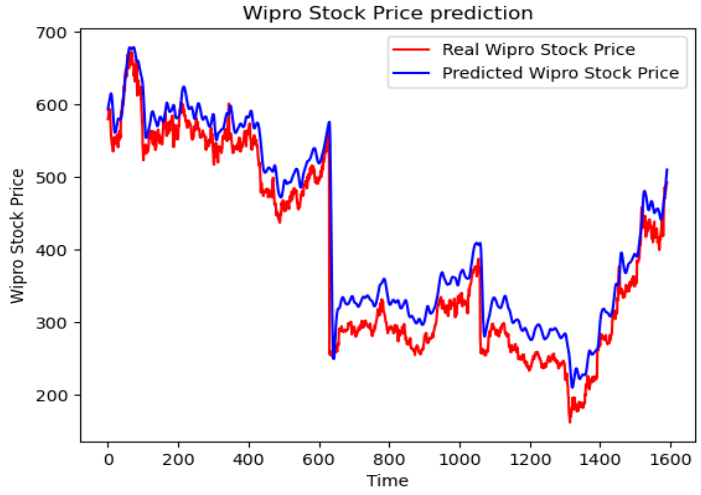


Fig 4.5.1(d): *LSTM Prediction Curve at Epochs = 20*

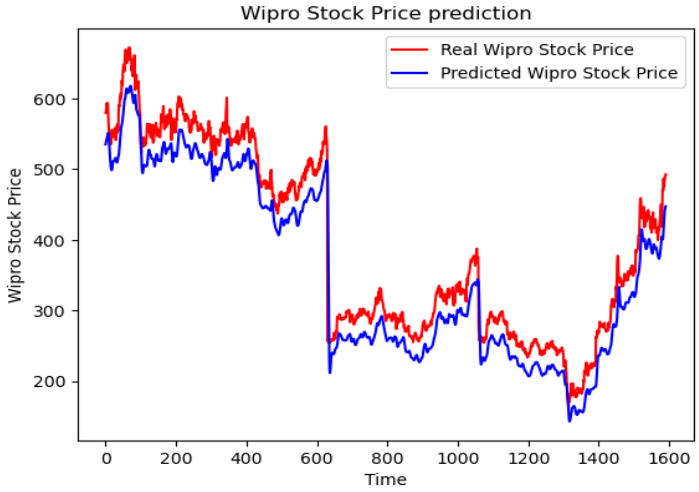


Fig 4.5.1(e): *LSTM Prediction Curve at Epochs = 50*

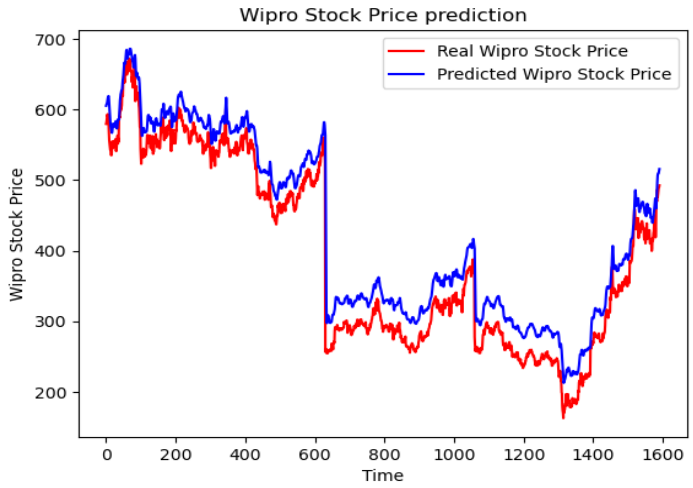


Fig 4.5.1(f): *LSTM Prediction Curve at Epochs = 100*

Batch Size	Number of Epochs	RMSE	MAPE
60	20	5.00159	0.06102
	50	4.44329	0.02890
	100	7.32023	0.13186
90	20	6.25110	0.08909
	50	6.14045	0.10919
	100	6.02030	0.09071

Table 4.5.1: *Batch Size vs Epochs vs RMSE vs MAPE*

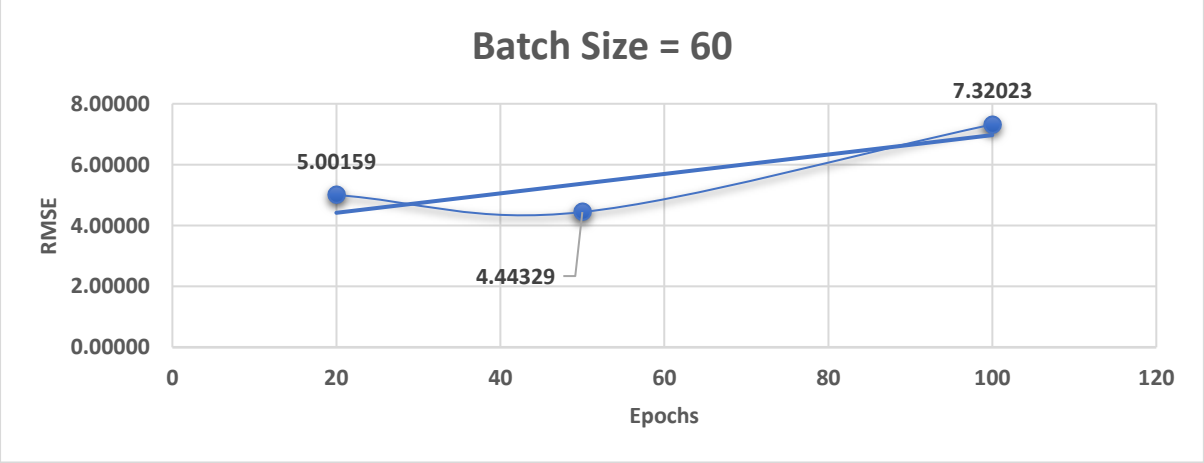
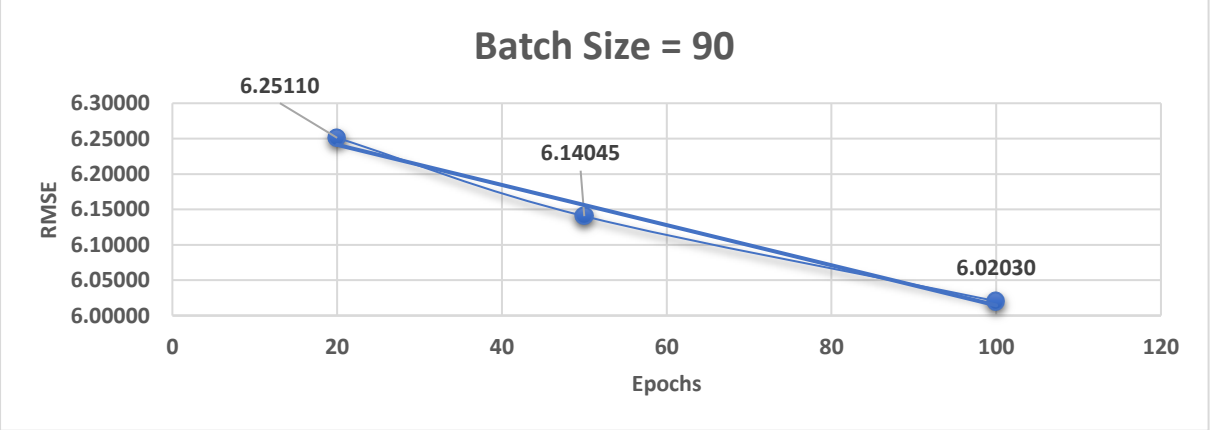


Fig 4.5.1(g)(h): *Plotting of RMSE vs Epochs*



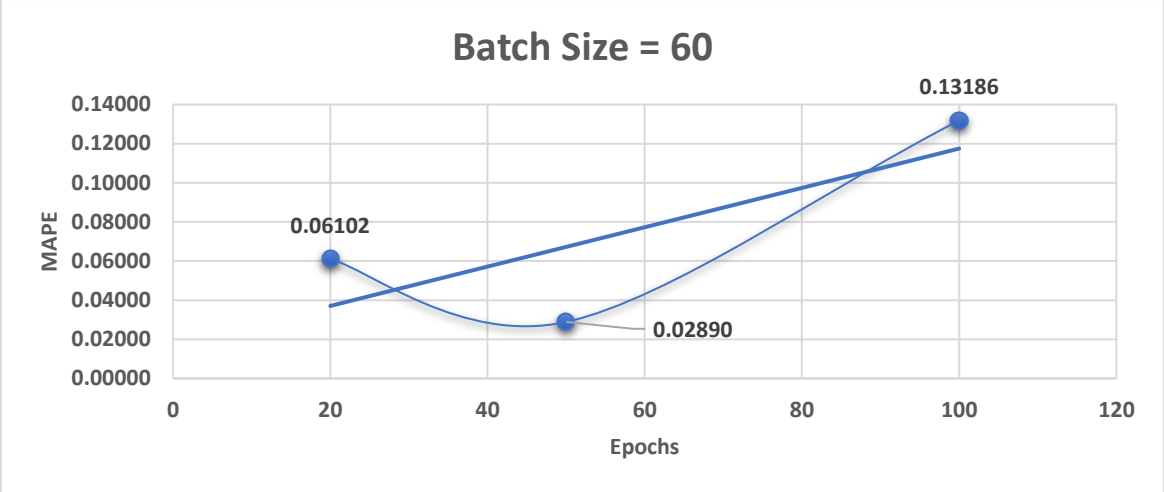
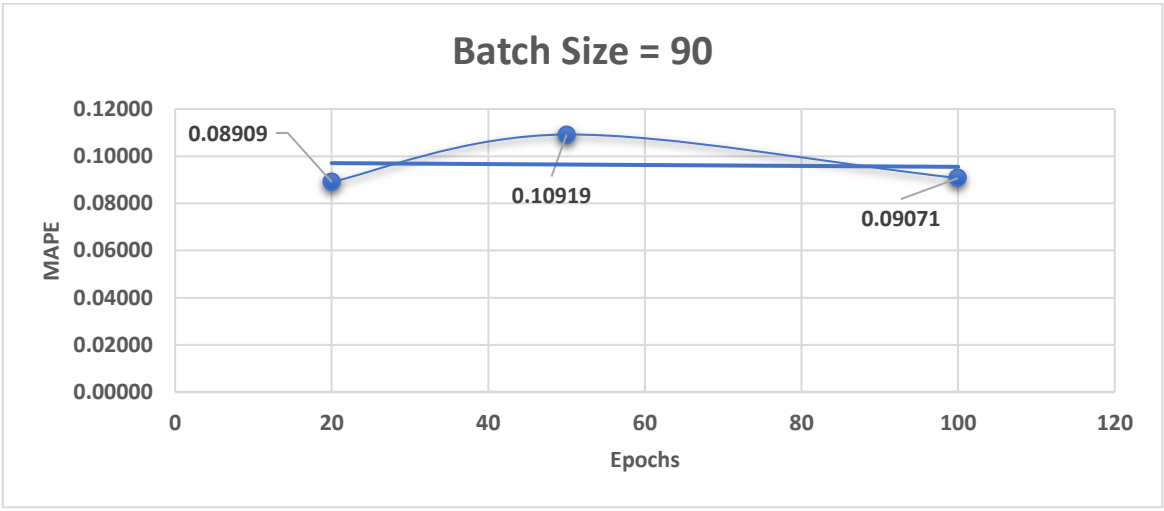


Fig 4.5.1(i)(j): *Plotting of MAPE vs Epochs*



By observing the experimental results from Table 4.5.1 and Figures 4.5.1(g)(h)(i)(j), we can conclude that for this dataset, LSTM works the best for 60 batch size & 50 number of epochs.

4.5.2 Larsen & Toubro

Training Dataset Length: 2929 (70%)

Testing Dataset Length: 1255 (30%)

For Batch Size = 60,

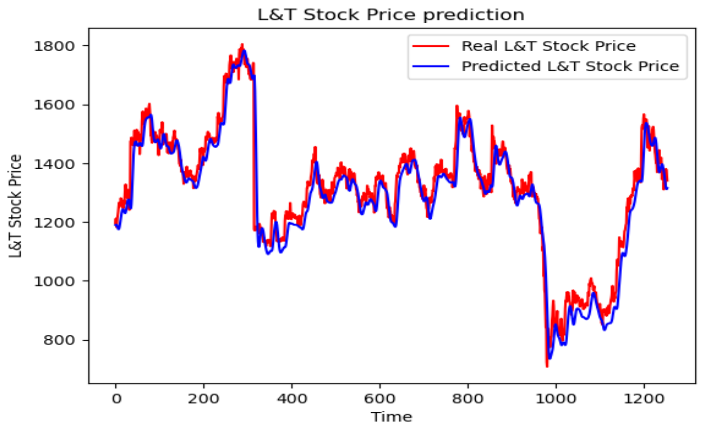


Fig 4.5.2(a): *LSTM Prediction Curve at Epochs = 20*

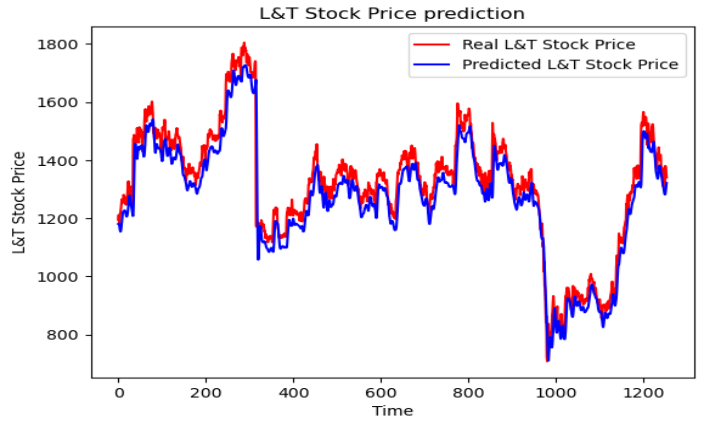


Fig 4.5.2(b): *LSTM Prediction Curve at Epochs = 50*

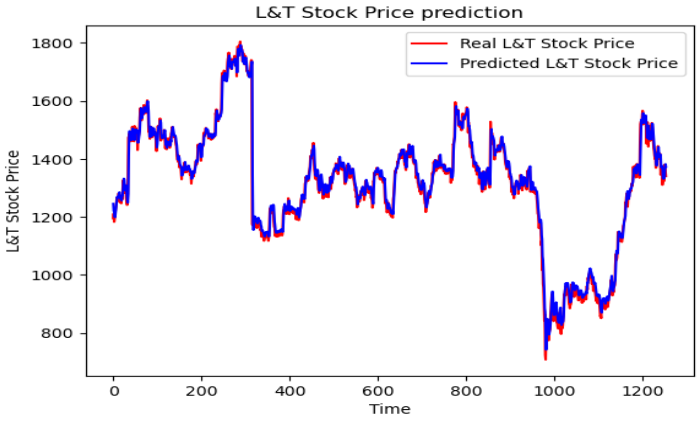


Fig 4.5.2(c): *LSTM Prediction Curve at Epochs = 100*

For Batch Size = 90,

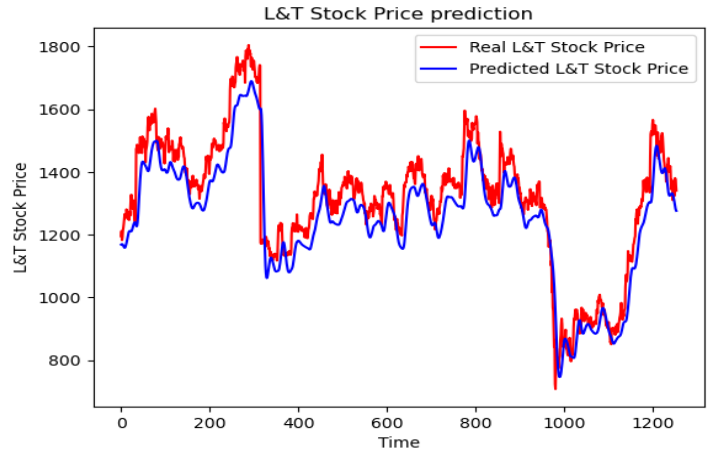


Fig 4.5.2(d): *LSTM Prediction Curve at Epochs = 20*

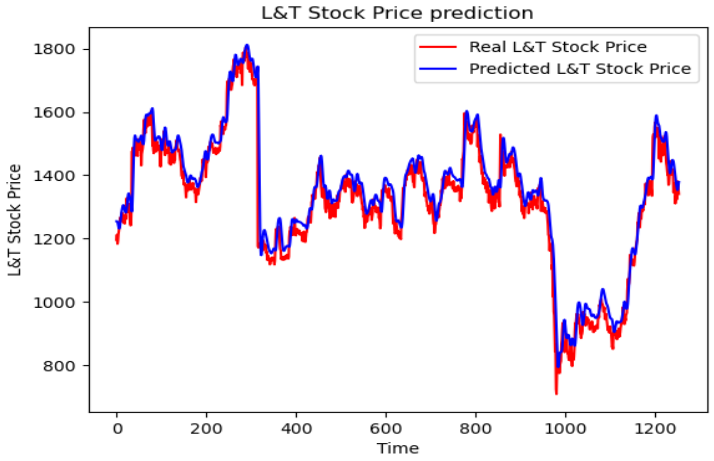


Fig 4.5.2(e): *LSTM Prediction Curve at Epochs = 50*

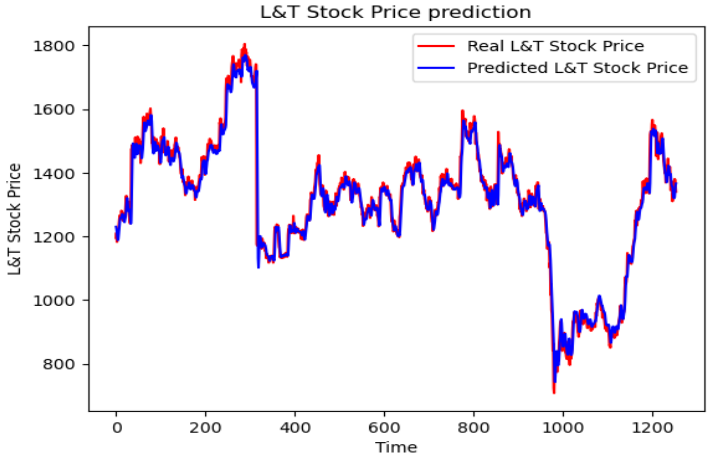


Fig 4.5.2(f): *LSTM Prediction Curve at Epochs = 100*

Batch Size	Number of Epochs	RMSE	MAPE
60	20	7.77664	0.03471
	50	7.69493	0.03905
	100	5.47170	0.01460
90	20	9.25110	0.05582
	50	7.24931	0.02920
	100	5.87720	0.01681

Table 4.5.2: *Batch Size vs Epochs vs RMSE vs MAPE*

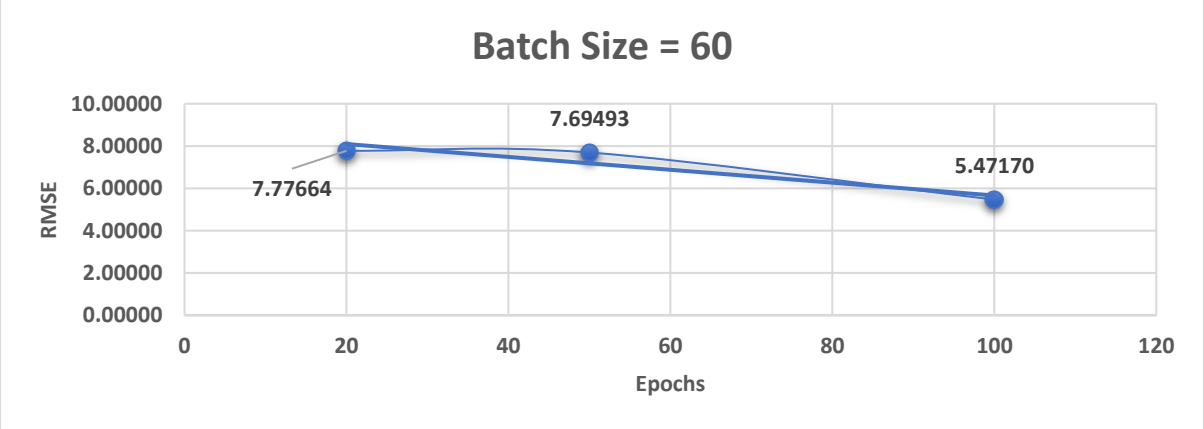
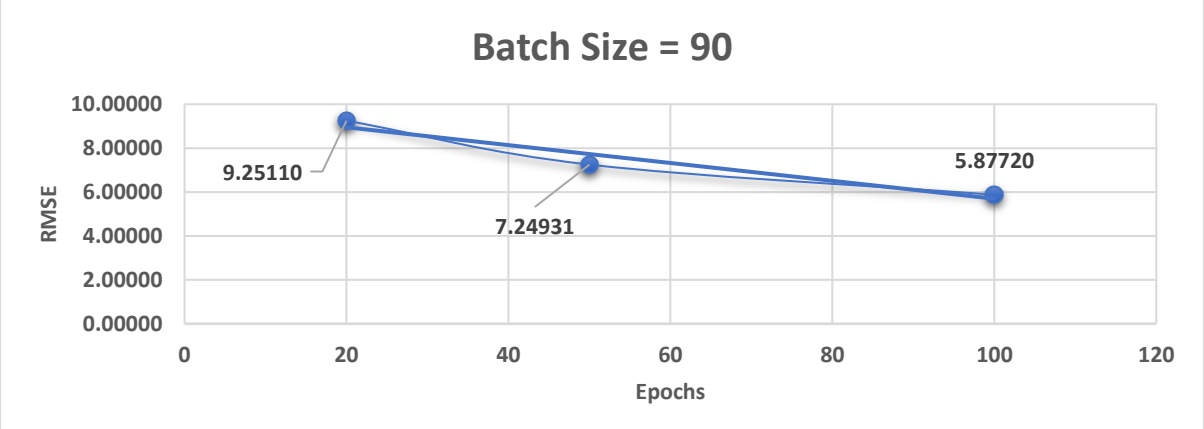


Fig 4.5.2(g)(h): *Plotting of RMSE vs Epochs*



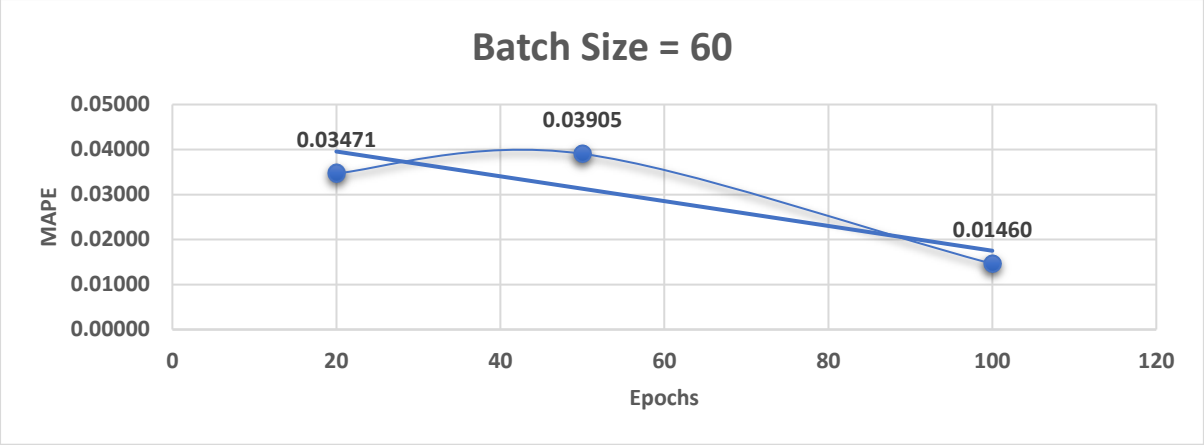
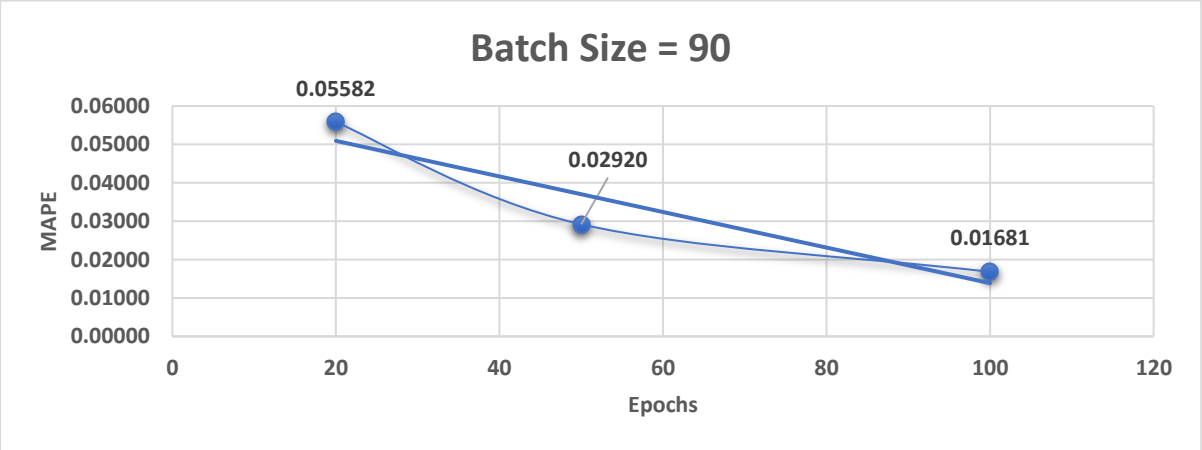


Fig 4.5.2(i)(j): *Plotting of MAPE vs Epochs*



By observing the experimental results from Table 4.5.2 and Figures 4.5.2(g)(h)(i)(j), LSTM works the best for 60 batch size & 100 number of epochs.

4.5.3 Google

Training Dataset Length: 1775 (70%)

Testing Dataset Length: 761 (30%)

For Batch Size = 60,

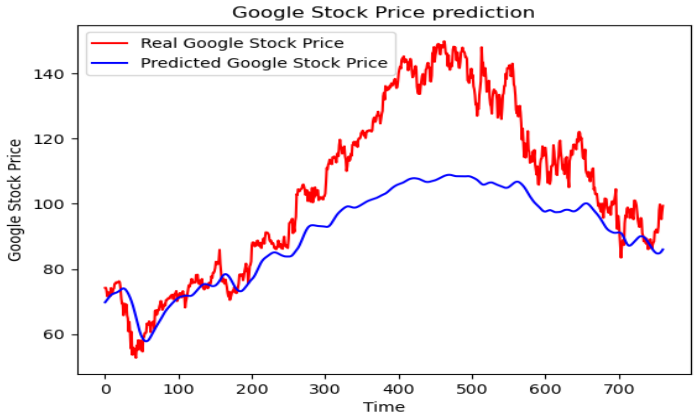


Fig 4.5.3(a): *LSTM Prediction Curve at Epochs = 20*

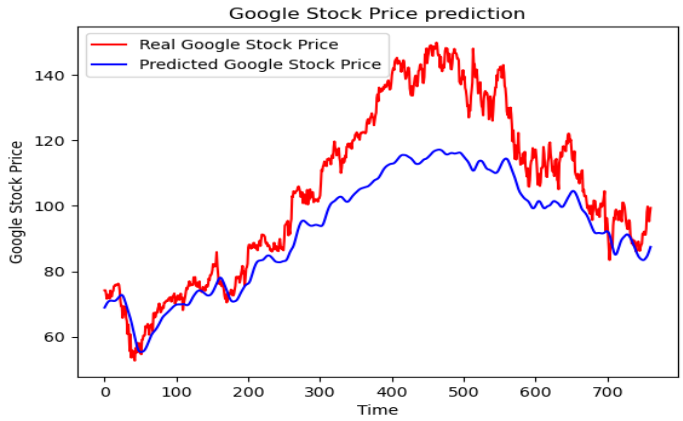


Fig 4.5.3(b): *LSTM Prediction Curve at Epochs = 50*

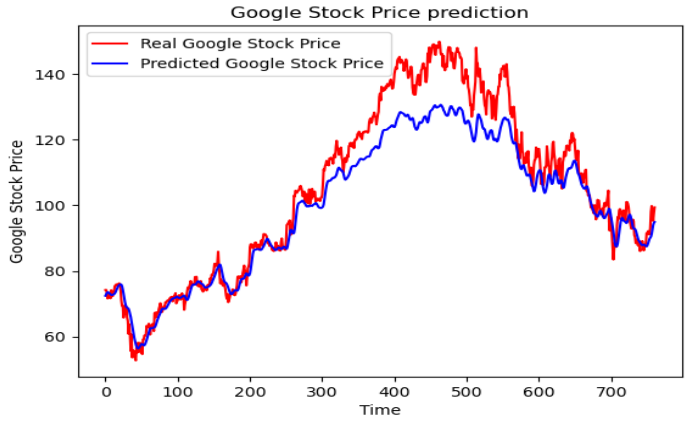


Fig 4.5.3(c): *LSTM Prediction Curve at Epochs = 100*

For Batch Size = 90,

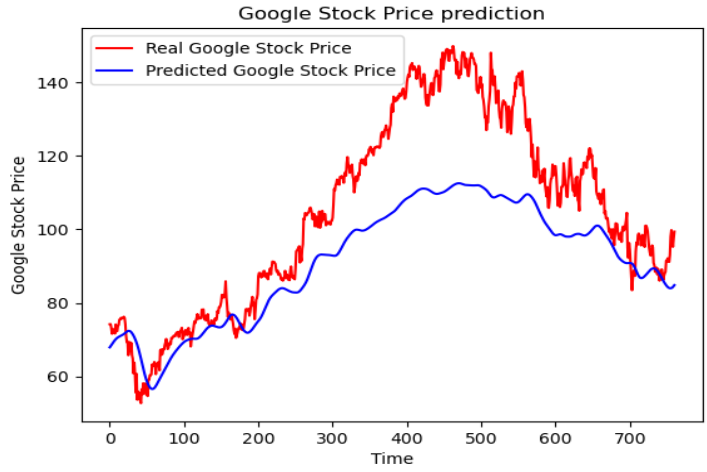


Fig 4.5.3(d): *LSTM Prediction Curve at Epochs = 20*

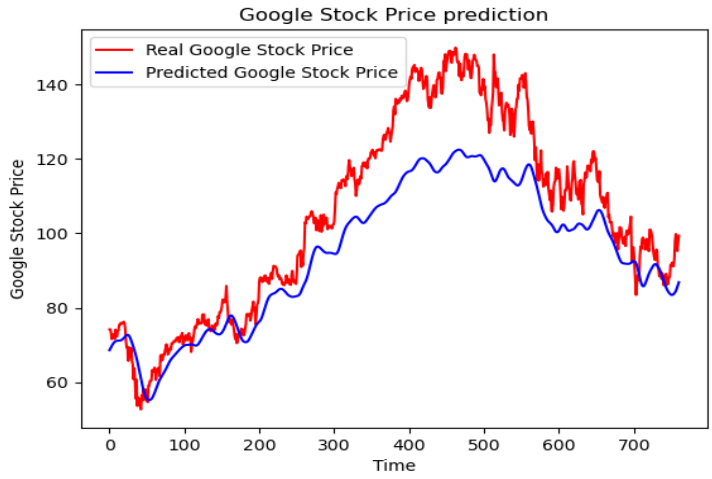


Fig 4.5.3(e): *LSTM Prediction Curve at Epochs = 50*

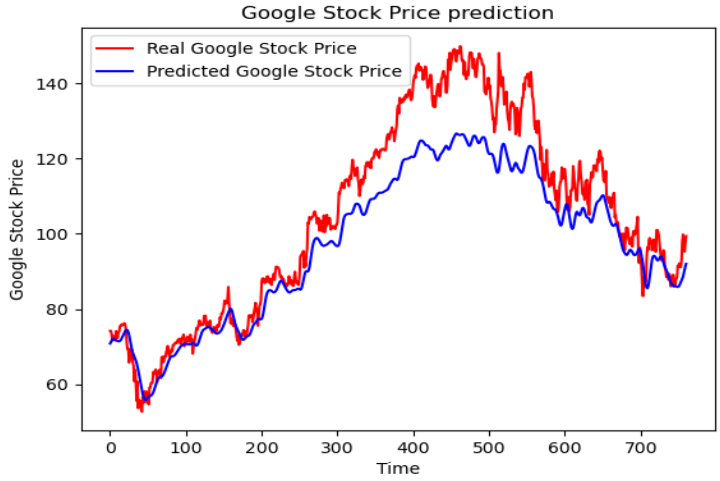


Fig 4.5.3(f): *LSTM Prediction Curve at Epochs = 100*

Batch Size	Number of Epochs	RMSE	MAPE
60	20	4.36012	0.14892
	50	3.91879	0.12103
	100	2.85553	0.05466
90	20	4.21102	0.14378
	50	3.66836	0.10726
	100	3.25152	0.07733

Table 4.5.3: *Batch Size vs Epochs vs RMSE vs MAPE*

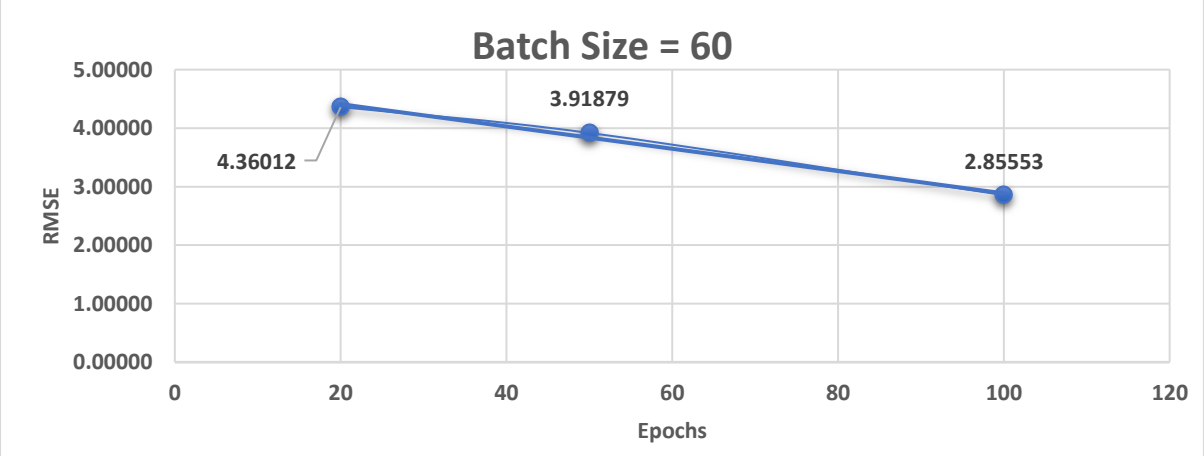
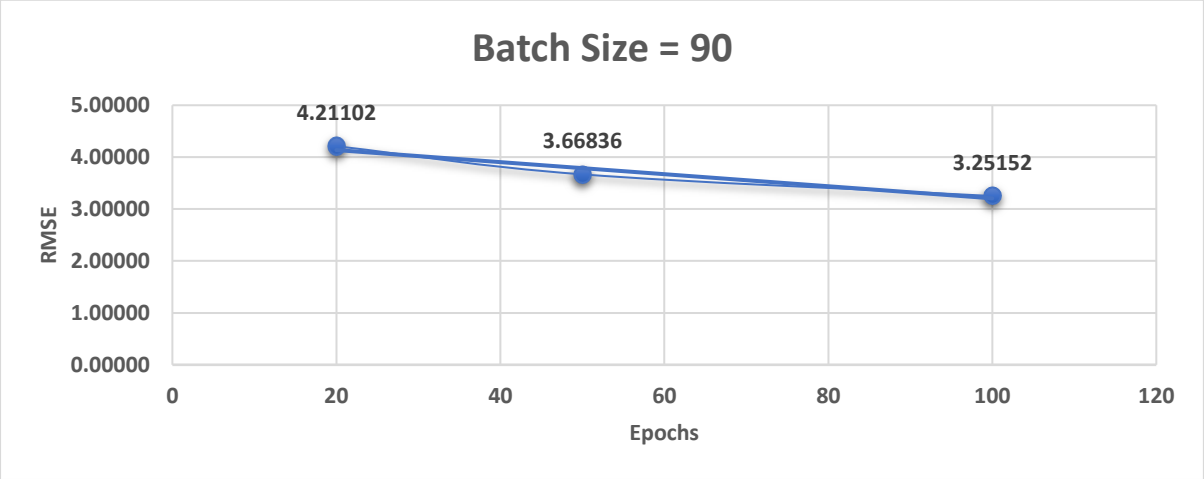


Fig 4.5.3(g)(h): *Plotting of RMSE vs Epochs*



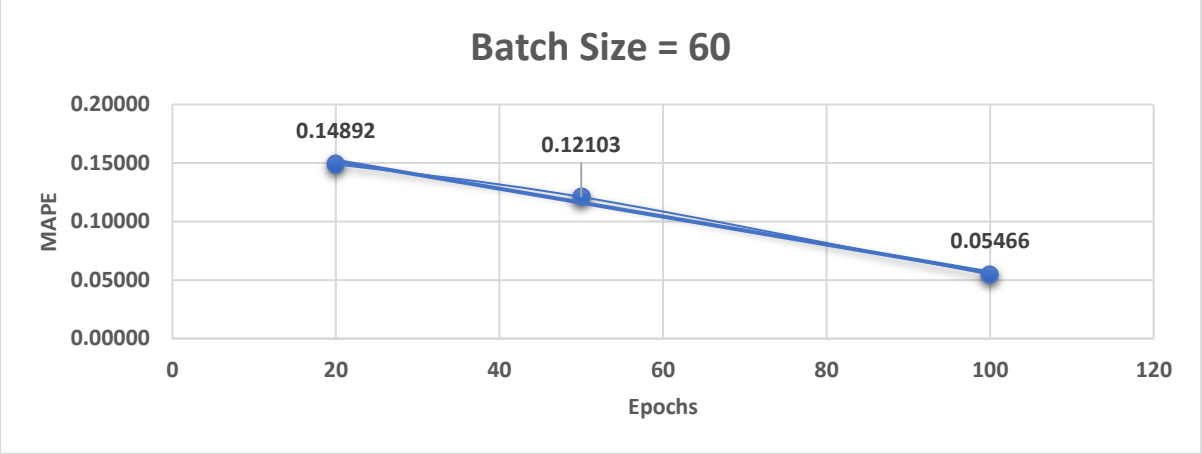
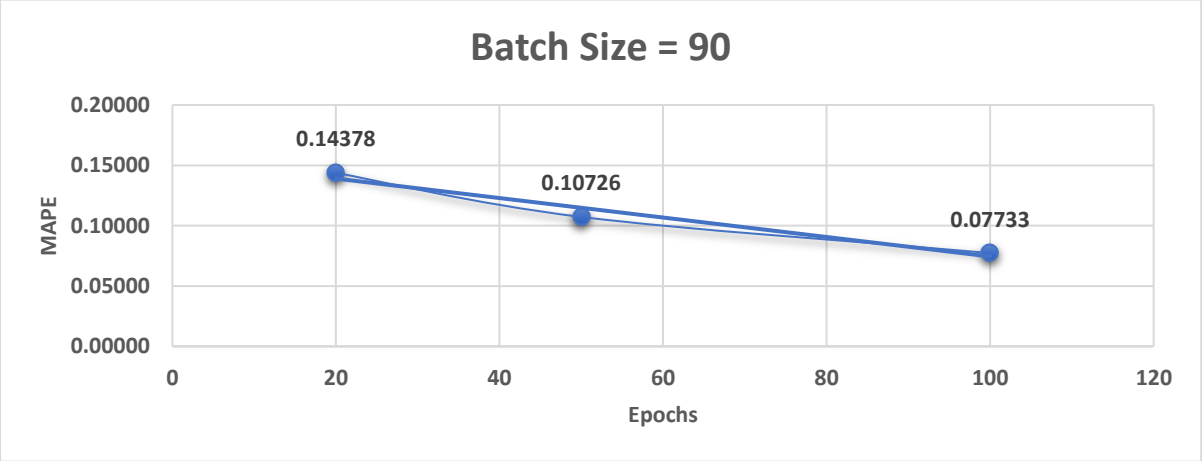


Fig 4.5.3(i)(j): *Plotting of MAPE vs Epochs*



By observing the experimental results from Table 4.5.3 and Figures 4.5.3(g)(h)(i)(j), LSTM works the best for 60 batch size & 100 number of epochs.

Dataset	Model	Best RMSE	Respective MAPE
Wipro	Linear Regression	7.60581	0.00265
	Support Vector Machine	7.42278	0.00273
	Long Short-Term Memory	4.44329	0.02890

Table 4(a): *Wipro Dataset vs Various Model Evaluation Metrics*

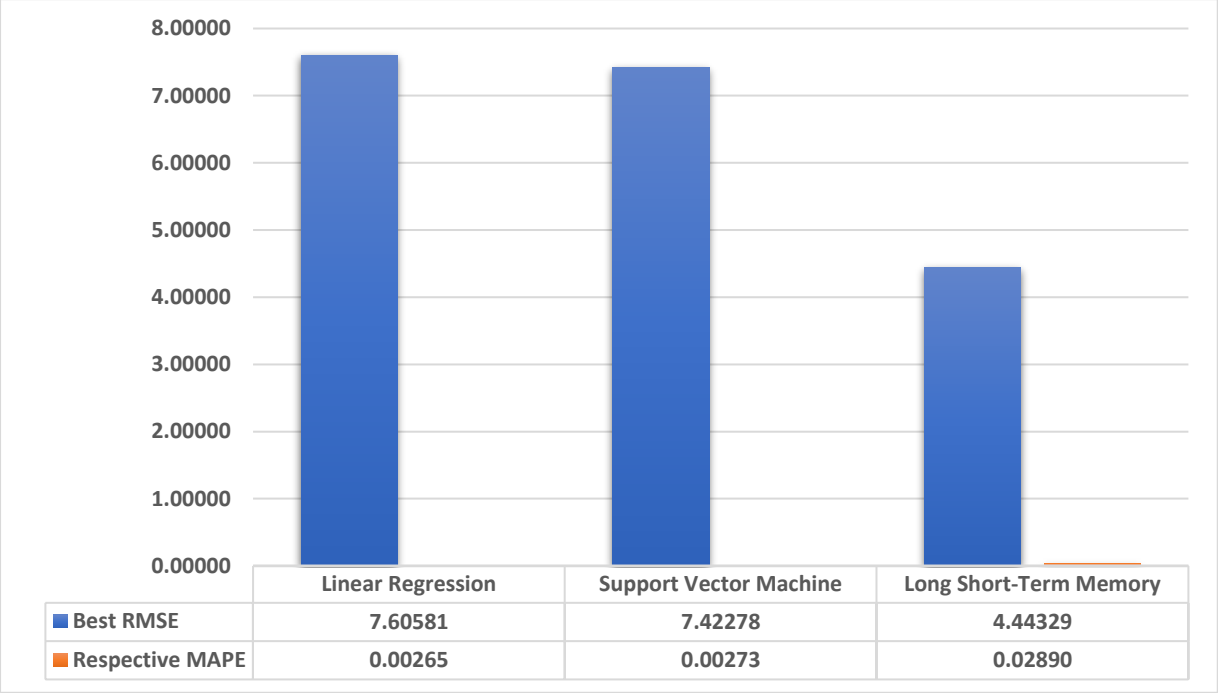


Fig 4(a): *Wipro Dataset vs Best RMSE & MAPE of Various Models*

Dataset	Model	Best RMSE	Respective MAPE
Larsen & Toubro	Linear Regression	5.12075	0.00189
	Support Vector Machine	5.02804	0.00187
	Long Short-Term Memory	5.47170	0.01460

Table 4(b): *Larsen & Toubro Dataset vs Various Model Evaluation Metrics*

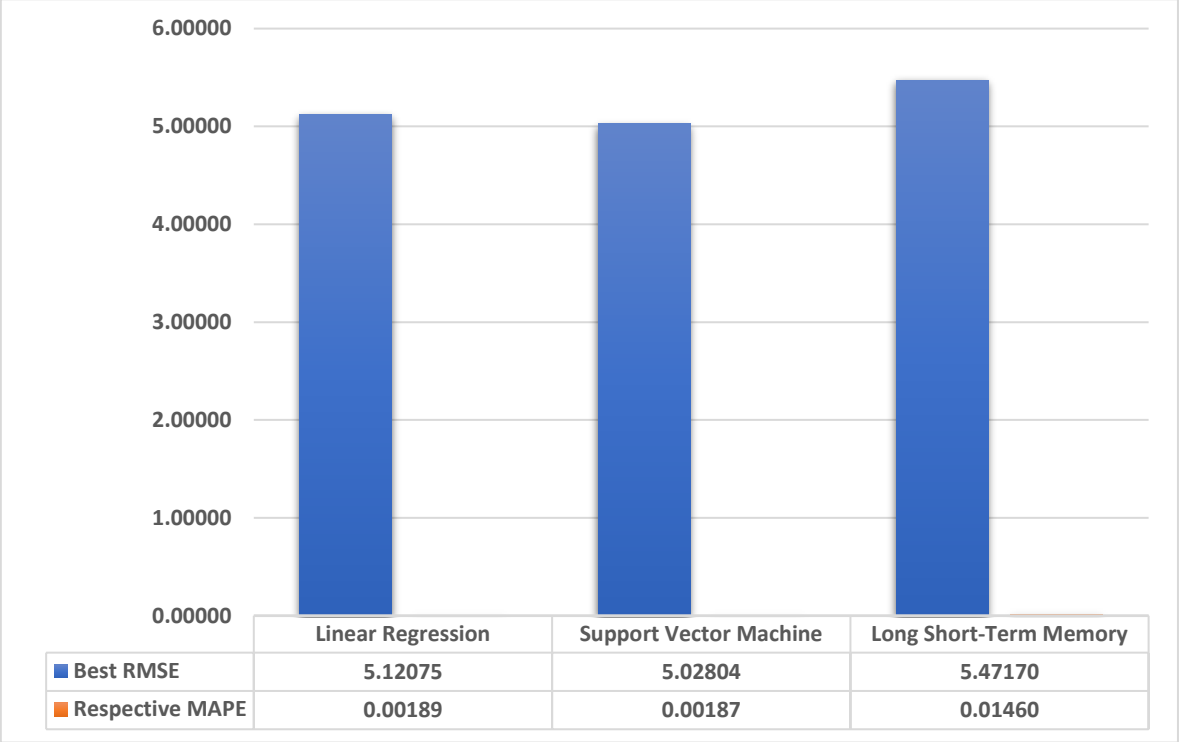


Fig 4(b): *Larsen & Toubro Dataset vs Best RMSE & MAPE of Various Models*

Dataset	Model	Best RMSE	Respective MAPE
Google	Linear Regression	1.99904E-14	1.94421E-16
	Support Vector Machine	0.04969	0.00086
	Long Short-Term Memory	2.85553	0.05466

Table 4(c): *Google Dataset vs Various Model Evaluation Metrics*

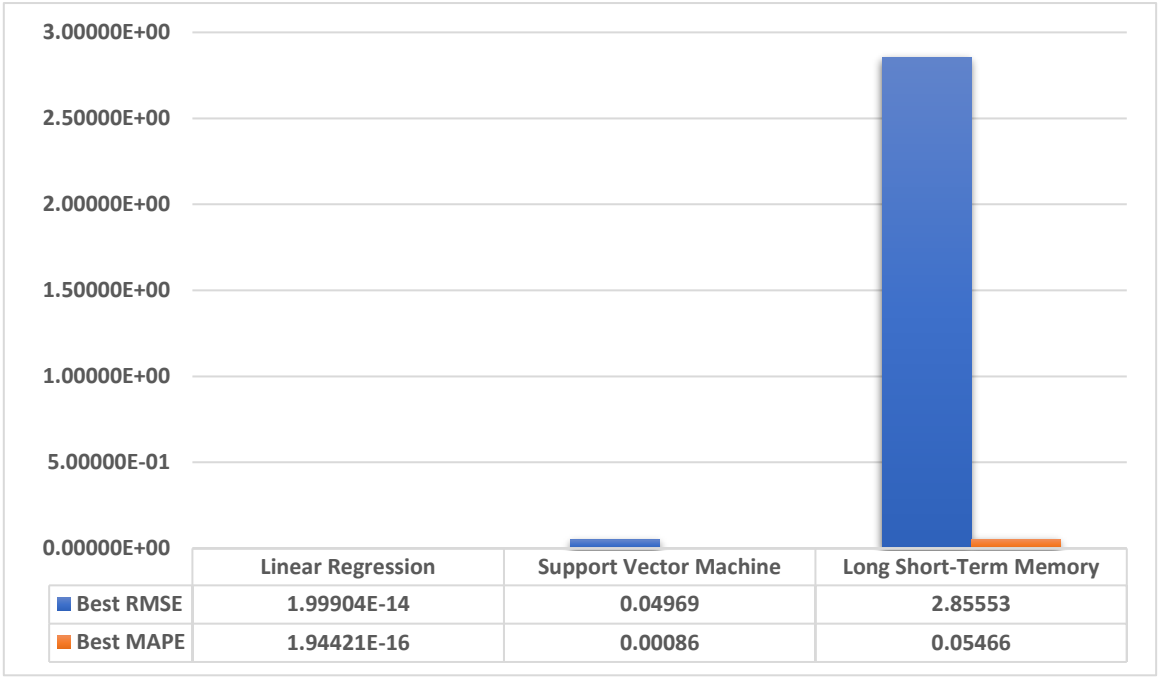


Fig 4(c): *Google Dataset vs Best RMSE & MAPE of Various Model Evaluation*

Datasets	Best Working Model	RMSE	MAPE
Wipro	Long Short-Term Memory	4.44329	0.02890
Larsen & Toubro	Support Vector Machine	5.02804	0.00187
Google	Linear Regression	1.99904E-14	1.94421E-16

Table 4(d): *Each Dataset vs Best Model Evaluation Metrics*

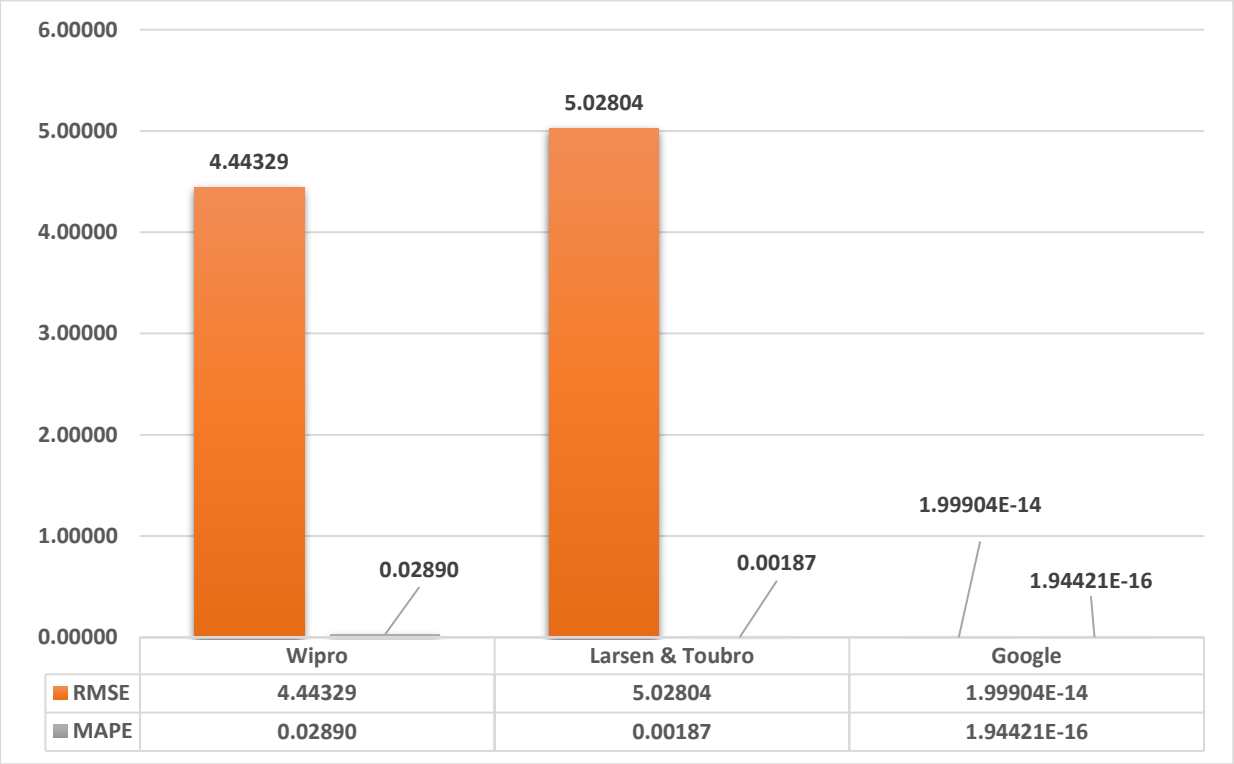


Fig 4(d): *Each Dataset vs Best RMSE & Respective MAPE*

From above experimental results from Tables 4(a)(b)(c)(d) and Figures 4(a)(b)(c)(d), it can be seen that,

- Long Short-Term Memory is best working for Wipro dataset.
- Support Vector Machine is best working for Larsen & Toubro dataset.
- Linear Regression is best working for Google dataset.

Chapter 5

5. ***Conclusion & Future Scope***

A lot of research work have been conducted to develop new methods for predicting the stock market using new techniques as stock market trading is becoming increasingly popular. Forecasting techniques are not only useful to researchers, but also to investors and others who deal with the stock market. The stock indices can be predicted better with a forecasting model with good accuracy.

By experimenting on the various datasets, this project report provides a comparative study using various forecasting models: Linear Regression, SVM and LSTM. It will help investors, analysts, and individuals interested in investing in the stock market by giving them a better understanding of the stock market's future situation.

There are several opportunities for future work in this field. One potential avenue for future research is the development of more sophisticated deep learning models that can capture even more complex patterns in financial data. Another area of interest is the integration of external data sources, such as news articles and social media feeds, to improve the accuracy of predictions. Additionally, the development of more interpretable models and techniques for feature selection could increase the trust and confidence of investors and financial institutions in the predictions generated by these models. Furthermore, the effectiveness of these models could be evaluated in real-world scenarios to better understand their practical applications and limitations. Overall, the future scope of research in this field is vast, and there is significant potential for innovation and advancement in the development and application of Machine Learning and Deep Learning Techniques for Stock Price Prediction.

In contrast to other project works, the current work also has some limitations of its own. Following are some key limitations:

1. Stock market predictions heavily rely on the quality and accessibility of the training data in order to be accurate and reliable. Financial data can be erratic, and inaccurate or

missing data can have a negative effect on the model's performance. Besides, there are many external factors that influence stock market data like political situations, market sentiment etc. Additionally, there isn't a lot of attention paid to the pre-processing of the data in this work.

2. Data from the stock market frequently displays non-stationary behaviour, which means that patterns and relationships evolve over time. Since machine learning models are based on the assumption of stationarity, they might find it difficult to adjust to shifting market conditions and identify developing patterns. They also inherently involve risk and uncertainty.
3. As for the different datasets, best predictions for each have been given by different models. Thus, a common model can be constructed which can give the best prediction irrespective of the datasets.
4. Comparatively lower hardware resources and software optimization can lead to slower processing in case of epochs in LSTM.

Bibliography

[1] M. Vijh, D. Chandola, V. A. Tikkiwal and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," pp. 1-8, 2019.

[2] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," pp. 1-6, 2020.

[3] M. Roondiwala, H. Patel and S. Varma, "Predicting Stock Prices Using LSTM," *International Journal of Science and Research (IJSR),* pp. 1-3, 2015.

[4] P. Yu and X. Yan, "Stock Price Prediction based on Deep Neural Networks," pp. 1-20, 2019.

[5] K. Khare, O. Darekar, P. Gupta and D. V. Attar, "Short Term Stock Price Prediction Using Deep Learning," 2017.

[6] Y. Lin, H. Guo and J. Hu, "An SVM-based Approach for Stock Market Trend Prediction," pp. 1-7, 2013.

[7] Z. Liu, Z. Dang and J. Yu, "Stock Price Prediction Model based on RBF-SVM Algorithm," pp. 1-4, 2020.

[8] J. Heo and J. Y. Yang, "Stock Price Prediction Based on Financial Statements Using SVM," pp. 1-10, 2016.

[9] V. Gururaj, S. V. R and A. K, "Stock Market Prediction using Linear Regression and Support Vector Machines," pp. 1-4, 2019.

[10] B. Panwar, G. Dhuriya, P. Johri, S. S. Yadav and N. Gaur, "Stock Market Prediction Using Linear Regression and SVM," pp. 1-3, 2021.

[11] D. Bhuriya, G. Kaushal, A. Sharma and U. Singh, "Stock Market Prediction Using A Linear Regression," pp. 1-4, 2017.

[12] "Linear Regression in Machine Learning," [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/linear-regression-in-machine-learning/.

[13] A. Raj, "Unlocking the True Power of Support Vector Regression: Using Support Vector Machine for Regression Problems," [Online]. Available: https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0. [Accessed 2020].

[14] J. Patterson and A. Gibson, Deep Learning: A Practitioner's Approach, O'Reilly, 2017.

[15] "What are Recurrent Neural Networks, IBM," [Online]. Available: https://www.ibm.com/topics/recurrent-neural-networks#:~:text=A%20recurrent%20neural%20network%20(RNN,data%20or%20time%20series%20data..

[16] C. Olah, "Understanding LSTM Networks," August 27, 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[17] "RMSE," [Online]. Available: https://c3.ai/glossary/data-science/root-mean-square-error-rmse.

[18] "MAPE," [Online]. Available: https://www.statisticshowto.com/mean-absolute-percentage-error-mape.