

PREDICTING THE GOOGLE STOCK MARKET VALUES USING RNN ALGORITHM

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Technology degree in Information Technology

By

SAYED SAJID ALI (37120065)

ROHIT KUMAR TIWARI (37120060)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

APRIL - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Sayed Sajid Ali (37120065)**, **Rohit Kumar Tiwari (37120060)** who carried out the project entitled **"PREDICTING THE GOOGLE STOCK MARKET VALUES USING RNN ALGORITHM"** under my supervision from September 2020 to April 2021.

Internal Guide
Mr. P. Saravanan M.E.,

Head of the Department
Dr. R SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Sayed Sajid Ali (37120065)**, **Rohit Kumar Tiwari (37120060)** hereby declare that the Project Report entitled “**Google Stock Market Prediction**” done by me under the guidance of **Mr.P.Saravanan M.E.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr. R. Subhashini M.E., Ph.D. , Head** of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr.P. Saravanan M.E.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

Stock price prediction is the process by which future stock prices are forecast on the basis of past prices. Stock price prediction is useful for investors to increase the profits from stock trading. There are two traditional approaches to prediction: technical and fundamental analysis. Technical analysis tries to identify some patterns from historical data while fundamental analysis focuses on overall economy, the company's financial condition and its management. while fundamental analysis focuses on overall economy, the company's financial condition and its management. Stock price prediction is a very challenging task as it is highly volatile, non-linear and dynamic and is impacted by various factors like political conditions, economic situation, trend, seasonality, investor psychology etc. The recurrent neural network (RNN) is a type of artificial neural network with self-loop in its hidden layer(s), which enables RNN to use the previous state of the hidden neuron(s) to learn the current state given the new input. RNN is good at processing sequential data. Long short-term memory (LSTM) cell is a specially designed working unit that helps RNN better memorize the long-term context. Long Short Term Memory (LSTM) is among the most popular deep learning models used today. It is also being applied to time series prediction which is a particularly hard problem to solve due to the presence of long term trend, seasonal and cyclical fluctuations and random noise. The performance of LSTM is highly dependent on choice of several hyper-parameters which need to be chosen very carefully, in order to get good results. Being a relatively new model, there are no established guidelines for configuring LSTM.

TABLE OF CONTENTS

CHAPTER.NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 About Project	1
	1.2 RNN	3
	1.3 LSTM	3
	1.4 Proposed System	4
2	LITERATURE SURVEY	6
	2.1 Objective	6
	2.2 Literature Survey	6
3	MATERIALS AND METHODS	7
	3.1 Methodology	7
	3.1.1 Dataset Collection	7
	3.1.2 Data Pre-processing	7
	3.1.3 Prediction Using RNN Algorithm	7
	3.1.4 Memory Management Using LSTM	8
4	RESULT	10

	4.1 Result	10
	4.2 Output in Graphical format	10
5	CONCLUSION AND FUTURE WORK	11
	5.1 Conclusion	11
	5.2 Future Work	11
	REFERENCES	12
	APPENDIX	13
	A. SOURCE CODE	13
	B. SCREENSHOTS	16
	C. PUBLICATION WITH PLAGIARISM REPORT	19

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3	ARCHITECTURE DIAGRAM	9
4.1	OUTPUT GRAPH	10
B.1	SUB PLOT	16
B.2	OPEN MEAN GRAPH	17
B.3	CLOSE MEAN GRAPH	17
B.4	PERIOD GRAPH	17
B.5	FINAL OUTPUT GRAPH	18

CHAPTER 1

INTRODUCTION

1.1 ABOUT PROJECT:

The financial market is an attractive field of study; it offers a variety of opportunities to investors, market analysts, as well as researchers from various disciplines. The market participation perspectives may differ among individuals, for example, learning the market behaviors, deriving influential aspects, trading through stocks, predicting future market trend, recommending assets for portfolio management, etc., however, the lack of financial literacy and knowledge of fundamental economic principles can critically affect the investment returns. Therefore, an individual's understanding and approach towards a financial market can determine the type as well as extent of information required to study this domain. The economic market can be considered as a combination of financial investments, transactions, potential earning and/or losing, and several other actions that are performed at a massive level. It has extensively reached to a large number of fields and consequently, it also gets affected by numerous events; with this regard, the financial market can be interpreted as a model of complex systems. The recurrent neural network (RNN) is a type of artificial neural network with self-loop in its hidden layer(s), which enables RNN to use the previous state of the hidden neuron(s) to learn the current state given the new input. RNN is good at processing sequential data. Long short-term memory (LSTM) cell is a specially designed working unit that helps RNN better memorize the long-term context. Long Short Term Memory (LSTM) is among the most popular deep learning models used today. It is also being applied to time series prediction which is a particularly hard problem to solve due to the presence of long term trend, seasonal and cyclical fluctuations and random noise. The performance of LSTM is highly dependent on choice of several hyper-parameters which need to be chosen very carefully, in order to get good results. Stock price prediction is the process by which future stock prices are forecast on the basis of past prices. Stock price prediction is useful for investors to increase the profits from stock trading. There are two traditional approaches to

prediction: technical and fundamental analysis. Technical analysis tries to identify some patterns from historical data while fundamental analysis focuses on overall economy, the company's financial condition and its management. while fundamental analysis focuses on overall economy, the company's financial condition and its management. Stock price prediction is a very challenging task as it is highly volatile, non-linear and dynamic and is impacted by various factors like political conditions, economic situation, trend, seasonality, investor psychology etc. The recurrent neural network (RNN) is a type of artificial neural network with self-loop in its hidden layer(s), which enables RNN to use the previous state of the hidden neuron(s) to learn the current state given the new input. RNN is good at processing sequential data. Long short-term memory (LSTM) cell is a specially designed working unit that helps RNN better memorize the long-term context. Long Short Term Memory (LSTM) is among the most popular deep learning models used today. It is also being applied to time series prediction which is a particularly hard problem to solve due to the presence of long term trend, seasonal and cyclical fluctuations and random noise. The performance of LSTM is highly dependent on choice of several hyper-parameters which need to be chosen very carefully, in order to get good results. Being a relatively new model, there are no established guidelines for configuring LSTM. The proposed system aims at making prediction of Google Stock Prices using the training dataset from the last decades performances in the Wall Street, New York Stock Exchange. The system uses RNN and LSTM algorithm for our proposed system to work. At the end it will be predicting a graph which will compare google stock price from last decade which can be compared with the test data to calculate of this project. Test data contain the actual stock prices of google which can further used to compare new sets of data.

1.2 RNN

Recurrent Neural Network (RNN) are a type of Neural Network where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus, RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

RNN have a memory which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks. RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous output by giving each output as input to the next hidden layer. Hence these three layers can be joined together such that the weights and bias of all the hidden layers is the same, into a single recurrent layer.

1.3 LSTM

Long Short-Term Memory is an advanced version of recurrent neural network (RNN) architecture that was designed to model chronological sequences and their long-range dependencies more precisely than conventional RNNs. The major highlights include the interior design of a basic LSTM cell, the variations brought into the LSTM architecture, and few applications of LSTMs that are highly in demand. It also makes a comparison between LSTMs and GRUs. The article concludes with a list of disadvantages of the LSTM network and a brief introduction of the upcoming attention-based models that are swiftly replacing LSTMs in the real world. LSTM networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where RNNs fail. Talking about RNN, it is a network that works on

the present input by taking into consideration the previous output and storing in its memory for a short period of time (short-term memory). Out of its various applications, the most popular ones are in the fields of speech processing, non-Markovian control, and music composition. Nevertheless, there are drawbacks to RNNs. First, it fails to store information for a longer period of time. At times, a reference to certain information stored quite a long time ago is required to predict the current output. But RNNs are absolutely incapable of handling such “long-term dependencies”. Second, there is no finer control over which part of the context needs to be carried forward and how much of the past needs to be ‘forgotten’. Other issues with RNNs are exploding and vanishing gradients (explained later) which occur during the training process of a network through backtracking. Thus, Long Short-Term Memory (LSTM) was brought into the picture. It has been so designed that the vanishing gradient problem is almost completely removed, while the training model is left unaltered. Long time lags in certain problems are bridged using LSTMs where they also handle noise, distributed representations, and continuous values. With LSTMs, there is no need to keep a finite number of states from beforehand as required in the hidden Markov model (HMM). LSTMs provide us with a large range of parameters such as learning rates, and input and output biases. Hence, no need for fine adjustments. The complexity to update each weight is reduced to $O(1)$ with LSTMs, similar to that of Back Propagation Through Time (BPTT), which is an advantage.

1.4 PROPOSED SYSTEM

A data set is taken from keggle which is then incorporated into the RNN and LSTM algorithm then it will take data from train data set to get the final prediction. Recurrent neural networks (RNN) are the state-of-the-art algorithm for sequential data and are used by Apple's Siri and Google's voice search. It is the first algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data. Due to their internal memory, RNN's can remember important things about the input they received, which allows them to be very precise in predicting what's coming next. This is why they're the preferred

algorithm for sequential data like time series, speech, text, financial data, audio, video, weather and much more.

Machine learning is constantly evolving with new methods being developed every day. It is crucial that we update our knowledge constantly and the best way to do so is by building models for fun projects like stock price prediction. Although the LSTM model above is not good enough to be used in live trading, the foundations built by developing such a model can help us build better models that might one day be used in our trading system.

Neural networks are very adept at predicting time series data, and when coupled with sentiment data, can really make a practical model. Although the results here were impressive, I am still finding ways to improve it, and maybe actually develop a full trading strategy from it.

Recurrent neural networks can form a much deeper understanding of a sequence and its context compared to other algorithms. Long short-term memory networks are an extension for recurrent neural networks, which basically extends the memory. Therefore, it is well suited to learn from important experiences that have very long time lags in between. The units of an LSTM are used as building units for the layers of RNN, often called an LSTM network.

LSTMs enable RNNs to remember inputs over a long period of time. This is because LSTMs contain information in a memory, much like the memory of a computer. The LSTM can read, write and delete information from its memory. This memory can be seen as a gated cell, with gated meaning the cell decides whether or not to store or delete information (i.e., if it opens the gates or not), based on the importance it assigns to the information. The assigning of importance happens through weights, which are also learned by the algorithm.

CHAPTER 2

LITERATURE SURVEY

2.1 OBJECTIVE

The proposed system aims at making prediction of Google Stock Prices using the training dataset from the last decades performances in the Wall Street, New York Stock Exchange. The system uses RNN and LSTM algorithm for our proposed system to work. At the end it will be predicting a graph which will compare google stock price from last decade which can be compared with the test data to calculate of this project. Test data contain the actual stock prices of google which can further used to compare new sets of data.

2.2 LITERATURE SURVEY

- A. Thakkar, 2020. Fusion in stock market prediction is considered as an approach to integrate data. If you can actually do it accurately, fast and secretly you will have profits with little labour input, but Practically impossible to do at any retail level.
- Jaewoo Kang, 2019. A conventional neural network function approximator which takes stock chart as input for making global predictions. The model can be trained on data of relatively and liquid market, it will require large amount of data.
- Ashish Sharma, 2017. Employing traditional methods like fundamental and technical analysis. The expected return and corresponding level of risks prior to an investment, the desire of earning higher profits of investment can enforce to move towards risky assets.
- Poonam Somali, 2014. An intelligent stock prediction model using traditional techniques it uses neural networks and SVM leading machine learning techniques in stock market prediction area there is always some risk to investments due to its unpredictable behavior.

CHAPTER-3

METHODOLOGY

3.1.1 DATASET COLLECTION

We have taken the training dataset for stock prediction, for this dataset we will use required attributes to train our model. We are taking the dataset from keggel. The various column in the dataset depicts the particular date on which stock price has been recorded, description or the nature of market, days on which the highest and the lowest stock values has occurred.

3.1.2 DATA PREPROCESSING

Data preprocessing is a process that converts raw data into required format. Data preprocessing can be done by various methods like data cleaning, data reduction and data integration. Here the dataset is collected from different resources which have different attributes and formats. Some data can be duplicate and they contain some attributes which are not useful as well as containing NULL values, and so we convert the raw data into its required format.

3.1.3 PREDICTION USING RNN ALGORITHM

Recurrent Neural Network are a type of Neural Network where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus, RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

RNN have a memory which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task

on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks. RNN converts the independent activations into dependent activations by providing the same weights and biases to all the layers, thus reducing the complexity of increasing parameters and memorizing each previous outputs by giving each output as input to the next hidden layer. Hence these three layers can be joined together such that the weights and bias of all the hidden layers is the same, into a single recurrent layer. Long Short-Term Memory is an advanced version of recurrent neural network (RNN) architecture that was designed to model chronological sequences and their long-range dependencies more precisely than conventional RNNs. The major highlights include the interior design of a basic LSTM cell, the variations brought into the LSTM architecture, and few applications of LSTMs that are highly in demand. It also makes a comparison between LSTMs and GRUs. The article concludes with a list of disadvantages of the LSTM network and a brief introduction of the upcoming attention-based models that are swiftly replacing LSTMs in the real world. LSTM networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where RNNs fail. Talking about RNN, it is a network that works on the present input by taking into consideration the previous output and storing in its memory for a short period of time (short-term memory). Out of its various applications, the most popular ones are in the fields of speech processing, non-Markovian control, and music composition.

3.1.4 MEMORY MANAGEMENT USING 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. In a NN that only contains one hidden layer the number of nodes in the input layer always depend on the dimension of the data, the nodes of the input layer connect to the hidden layer via links called 'synapses. The relation between every two nodes from, has a coefficient called weight, which is the decision maker for signals. The process of learning is naturally a continues adjustment of weights, after completing the process of learning, the Artificial NN will have optimal weights for each synapse.

The hidden layer nodes apply a sigmoid or tangent hyperbolic (tanh) function on the sum of weights coming from the input layer which is called the activation function, this transformation will generate values, with a minimized error rate between the train and test data using the SoftMax function. After completing this process, the model will be trained. The classes of NN that predict future value base on passed sequence of observations is called Recurrent Neural Network (RNN) this type of NN make use of earlier stages to learn of data and forecast futures trends. 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. RNN can't store long time memory, so the use of the Long Short-Term Memory (LSTM) based on "memory line" proved to be very useful in forecasting cases with long time data. In a LSTM the memorization of earlier stages can be performed trough gates with along memory line incorporated. The ability of memorizing sequence of data makes the LSTM a special kind of RNNs.

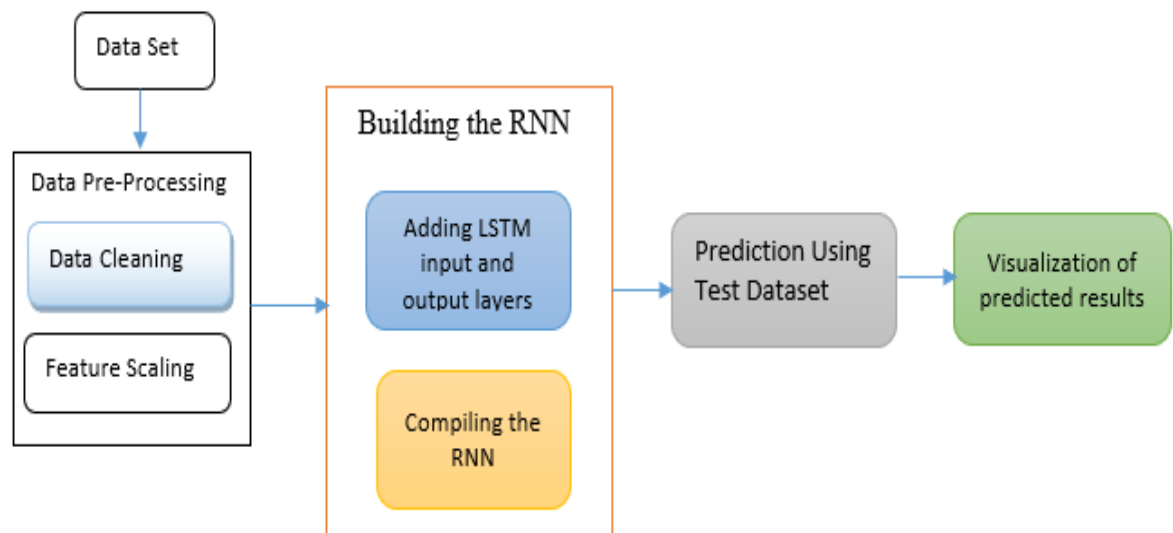


fig-3. Architecture Diagram

CHAPTER 4

RESULT AND DISCUSSION

After all these steps, we can use matplotlib to visualize the result of our predicted stock price and the actual stock price. While the exact price points from our predicted price weren't always close to the actual price, our model did still indicate overall trends such as going up or down. This project teaches us the LSTMs can be somewhat effective in times series forecasting.

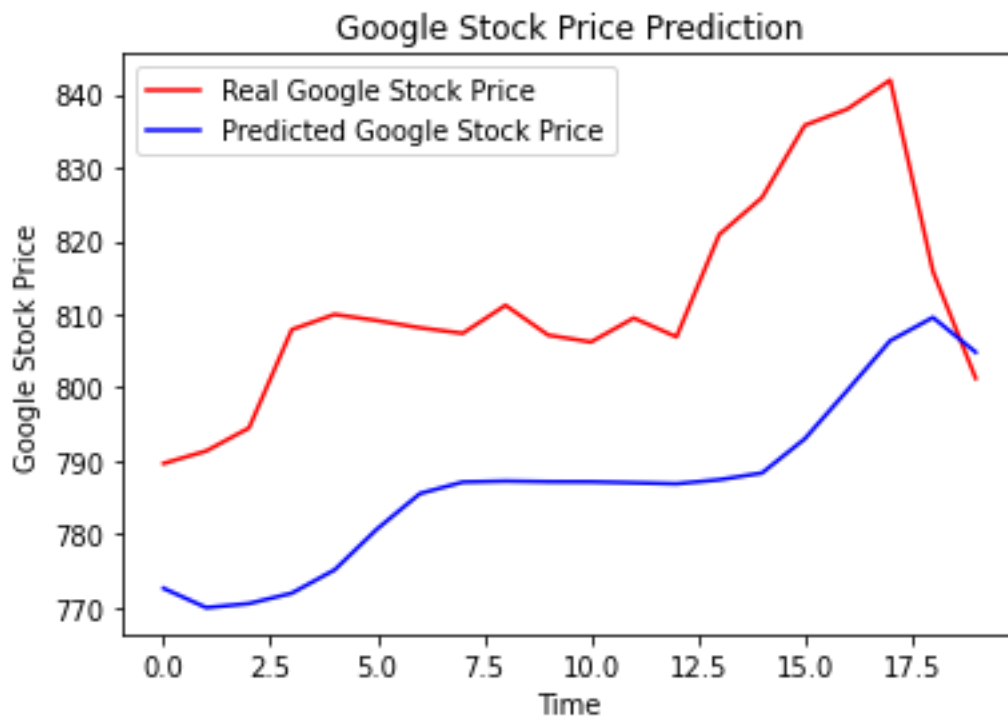


Fig-4.1 Output Graph

CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 CONCLUSION:

Machine learning is constantly evolving with new methods being developed every day. It is crucial that we update our knowledge constantly and the best way to do so is by building models for fun projects like stock price prediction. Although the LSTM model above is not good enough to be used in live trading, the foundations built by developing such a model can help us build better models that might one day be used in our trading system. Neural networks are very adept at predicting time series data, and when coupled with sentiment data, can really make a practical model. Although the results here were impressive, I am still finding ways to improve it, and maybe actually develop a full trading strategy from it.

5.2 FUTURE WORK:

In future a web application can be created to predict the stock prices of different firms like Netflix, Amazon, Reliance etc., we can easily predict the stock prices and invest efficiently which in turn will give us more profit. We can note the daily analysis of stock values of various multinational companies such as Netflix, Amazon, Reliance, Paytm, Facebook, Twitter etc., produced by the implemented algorithm. Even the users can see the reports or the analysis and they can invest accordingly.

REFERENCES

- [1] Neural Networks Applications in Finance : A Pratical Introduction – A. Thakkar Erika W. Gilbert and Mary M. Pashley,2016.
- [2] Stock Prediction - A Neural Network Approach - • Jaewoo Kang,KTH,2019
- [3] Using Neural Networks for Predicting Stock Market Prices – Ashish Sharma, 2017.
- [4] Poonam Sonamali - An intelligent stock prediction model using traditional techniques it uses neural networks, 2014.

APPENDIX

A. SOURCE CODE

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import datetime
from google.colab import files
uploaded = files.upload()
dataset = pd.read_csv('Google_Stock_Price_Train.csv', index_col="Date", parse_dates=True)

dataset.head()

dataset.isna().any()

dataset.info()

dataset['Open'].plot(figsize=(16,6))
# convert column "a" of a DataFrame
dataset["Close"] = dataset["Close"].str.replace(',', '').astype(float)

# convert column "a" of a DataFrame
dataset["Close"] = dataset["Close"].str.replace(',', '').astype(float)

# 7 day rolling mean
dataset.rolling(7).mean().head(20)
dataset['Open'].plot(figsize=(16,6))
dataset.rolling(window=30).mean()['Close'].plot()
dataset['Close: 30 Day Mean'] = dataset['Close'].rolling(window=30).mean()
dataset[['Close', 'Close: 30 Day Mean']].plot(figsize=(16,6))

# Optional specify a minimum number of periods
dataset['Close'].expanding(min_periods=1).mean().plot(figsize=(16,6))
training_set=dataset['Open']
training_set=pd.DataFrame(training_set)

# Feature Scaling
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
```

```

# Creating a data structure with 60 timesteps and 1 output
X_train = []
y_train = []
for i in range(60, 1258):
    X_train.append(training_set_scaled[i-60:i, 0])
    y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)

# Reshaping
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
# Part 2 - Building the RNN

# Importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout

# Initialising the RNN
regressor = Sequential()

# Adding the first LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True, input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))

# Adding a second LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a third LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))

# Adding a fourth LSTM layer and some Dropout regularisation
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))

# Adding the output layer
regressor.add(Dense(units = 1))

# Compiling the RNN

```

```

regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')

# Fitting the RNN to the Training set
regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)

# Part 3 - Making the predictions and visualising the results

# Getting the real stock price of 2017
dataset_test = pd.read_csv('Google_Stock_Price_Test.csv', index_col="Date", parse_dates=True)

from google.colab import files
uploaded = files.upload()
real_stock_price = dataset_test.iloc[:, 1:2].values
dataset_test.head()
dataset_test.info()
dataset_test["Volume"] = dataset_test["Volume"].str.replace(',', '').astype(float)
test_set=dataset_test['Open']
test_set=pd.DataFrame(test_set)
test_set.info()

# Getting the predicted stock price of 2019
dataset_total = pd.concat((dataset['Open'], dataset_test['Open']), axis = 0)
inputs = dataset_total[len(dataset_total) - len(dataset_test) - 60:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)
X_test = []
for i in range(60, 80):
    X_test.append(inputs[i-60:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
predicted_stock_price = regressor.predict(X_test)
predicted_stock_price = sc.inverse_transform(predicted_stock_price)
predicted_stock_price=pd.DataFrame(predicted_stock_price)
predicted_stock_price.info()

# Visualising the results
plt.plot(real_stock_price, color = 'red', label = 'Real Google Stock Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Google Stock Price')

```

```
plt.title('Google Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
plt.show()
```

B. SCREENSHOTS

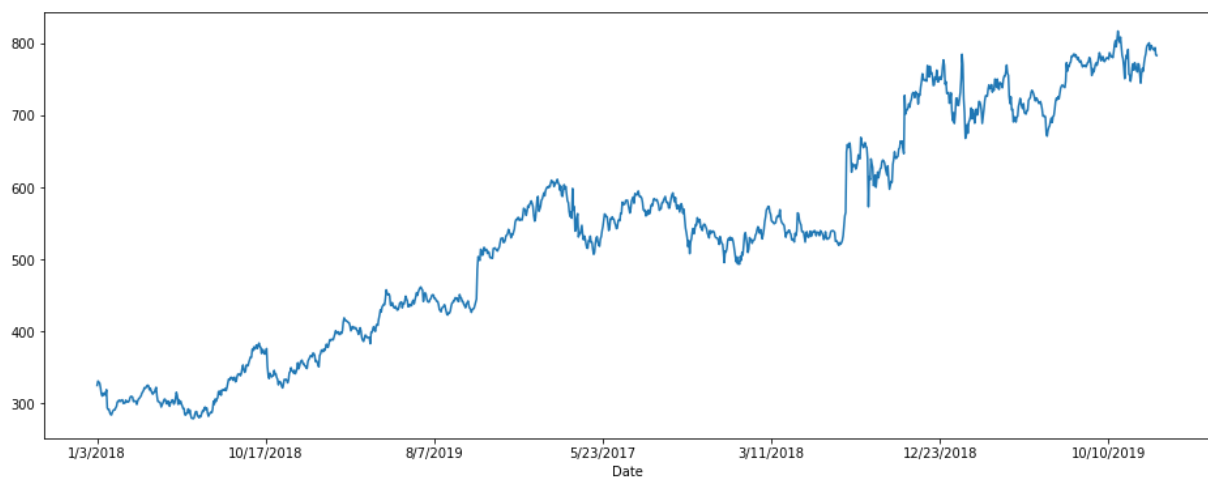


Fig. B.1 Sub Plot

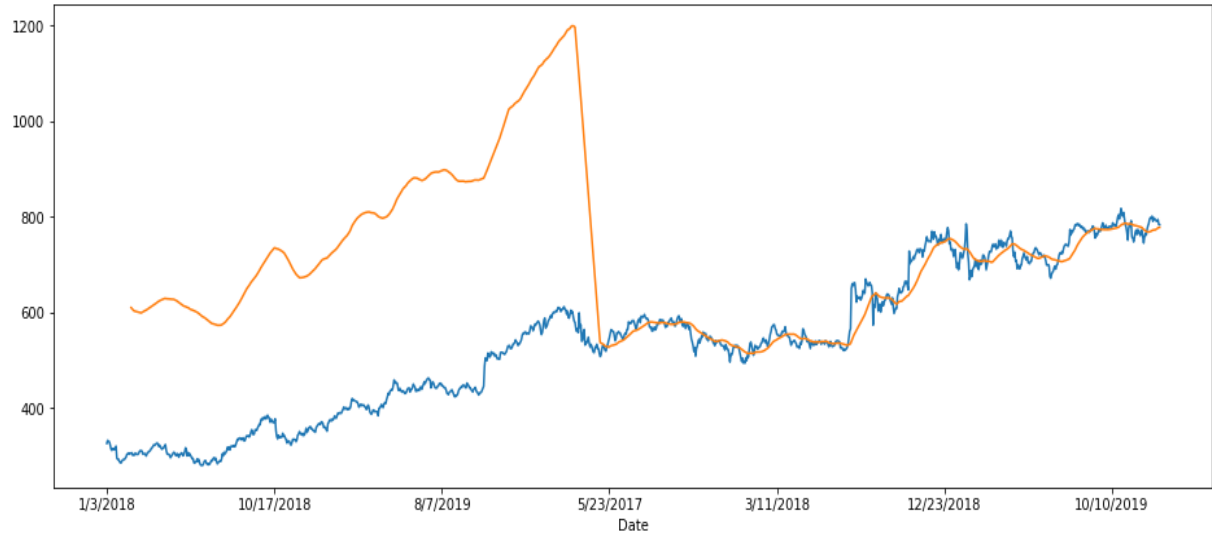


Fig B.2 Open Mean Graph

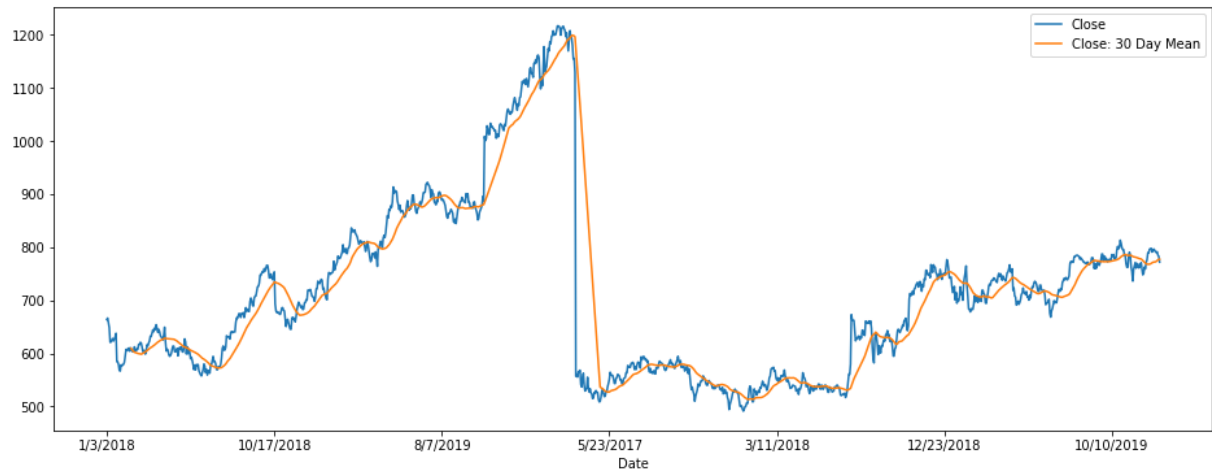


Fig . B.3 Close Mean Graph

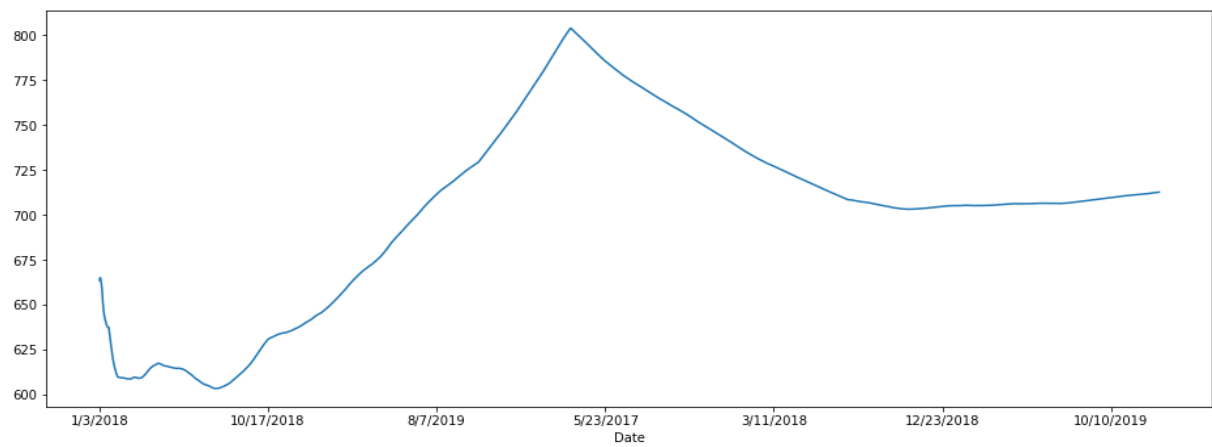


Fig . B.4 Period Graph

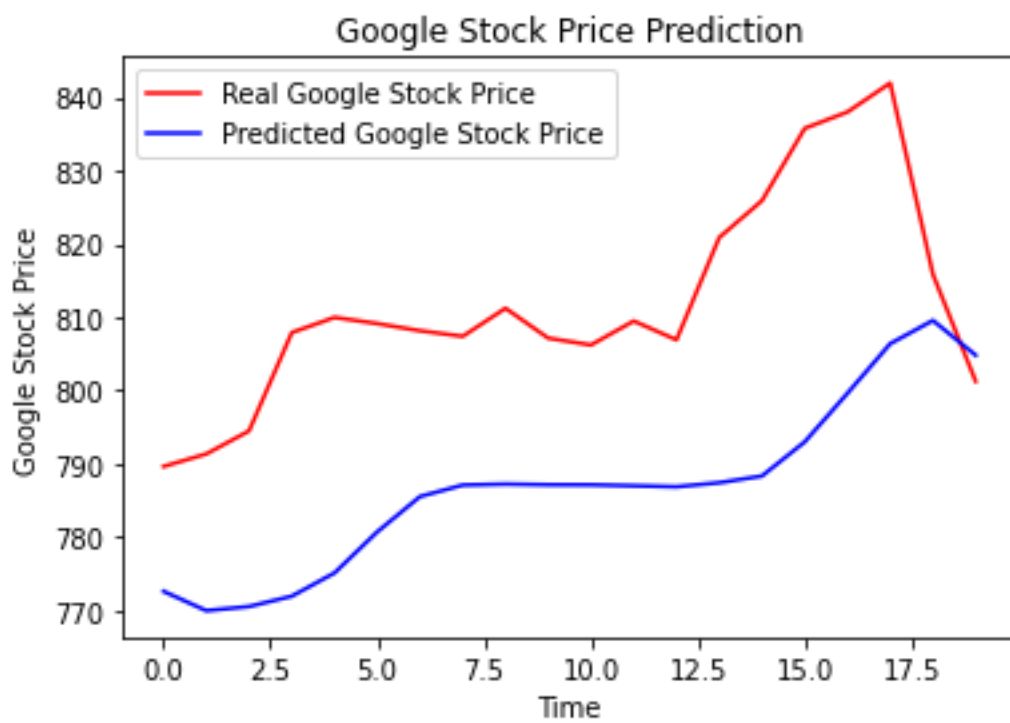
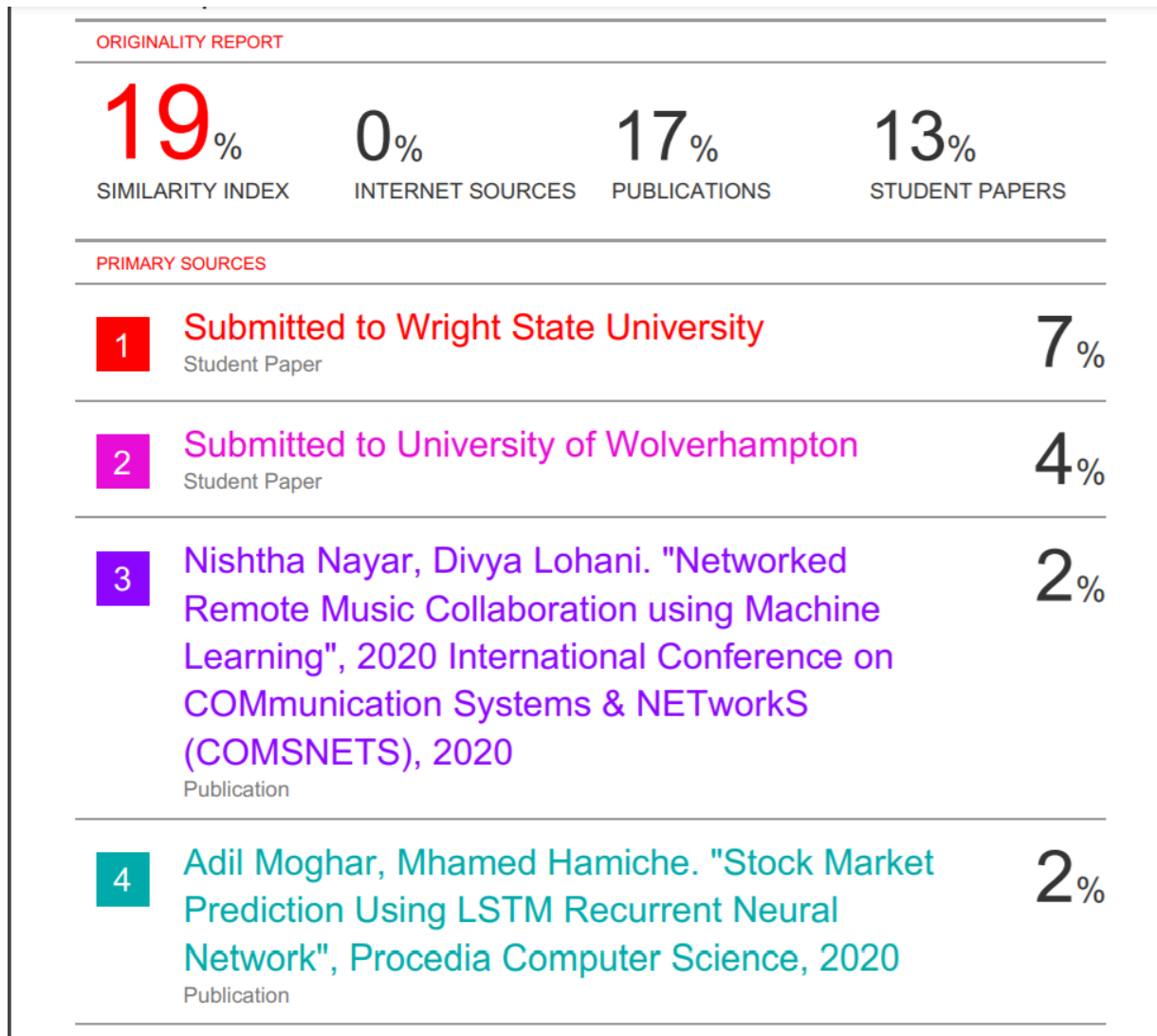


Fig. B.5 – Final Output Graph

C. PLAGIARISM REPORT



5	"Deep Learning Technique for Detecting NSCLC", International Journal of Recent Technology and Engineering, 2019	2%
---	---	----

Publication

6	Venkata Vara Prasad D, Lokeswari Y Venkataramana, P. Senthil Kumar,	1%
---	---	----

Prasannamedha G, Soumya K., Poornema A.J..
"Water quality analysis in a lake using deep learning methodology: prediction and validation", International Journal of Environmental Analytical Chemistry, 2020

Publication

Exclude quotes	On
----------------	----

Exclude matches	Off
-----------------	-----

Exclude bibliography	On
----------------------	----
