STOCK MARKET PREDICTION

Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree Of

BACHELOR OF ENGINEERINGIN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

SYED AHMED RAFAE	(1604-17-733-020)
SYED OSMAN HUSSAIN	(1604-17-733-030)
MOHD SHAKEEB	(1604-17-733-036)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT MUFFAKHAM JAH
COLLEGE OF ENGINEERING & TECHNOLOGY
(Affiliated to Osmania University)
Mount Pleasant, 8-2-249, Road No. 3, Banjara Hills, Hyderabad-34

2021

Date: 04 / 05/ 2021

CERTIFICATE

This is to certify that the project dissertation titled "**STOCK MARKET PREDICTION**" being submitted by

1. SYED AHMED RAFAE (1604-17-733-020)
2. SYED OSMAN HUSSAIN (1604-17-733-030)
3. MOHD SHAKEEB (1604-17-733-036)

in Partial Fulfillment of the requirements for the award of the degree of BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2018-19 is the bonafide work carried out by them. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signatures:

Internal Project Guide
(Mr. Mohd Abdul Raheem)
(Assistant Professor)

Head CSED
(Dr. A.A. Moiz Qyser)

External Examiner

DECLARATION

This is to certify that the work reported in the major project entitled "**STOCK MARKET PREDICTION**" is a record of the bonafide work done by us in the Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Osmania University. The results embodied in this report are based on the project work done entirely by us and not copied from any other source.

1. SYED AHMED RAFAE (1604-17-733-020)
2. SYED OSMAN HUSSAIN (1604-17-733-030)
3. MOHD SHAKEEB (1604-17-733-036)

ACKNOWLEDGEMENT

Our hearts are filled with gratitude to the Almighty for empowering us with courage, wisdom and strength to complete this project successfully. We give him all the glory, honor and praise.

We thank our Parents for having sacrificed a lot in their lives to impart the best education to us and make us promising professionals for tomorrow.

We would like to express our sincere gratitude and indebtness to our project supervisor **Mr. Mohd Abdul Raheem** for his valuable suggestions and interest throughout the course of this project.

We are happy to express our profound sense of gratitude and indebtedness to **Prof. Dr. Ahmed Abdul Moiz Qyser,** Head of the Computer Science and Engineering Department, for his valuable and intellectual suggestions apart from educate guidance constant encouragement right throughout our work and making successful.

With a great sense of pleasure and privilege, we extend our gratitude to **Prof. Dr. Syed Shabbeer Ahmed,** Associated Head of Computer Science and Engineering Department, project in-charge, offered valuable suggestions was a pre-requisite to carry out support in every step.

We are pleased to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to make this project work a total success.

SYED AHMED RAFAE

SYED OSMAN HUSSAIN

MOHD SHAKEEB

ABSTRACT

The prediction of a stock market direction may serve as an early recommendation system for short-term investors and as an early financial distress warning system for long-term shareholders. Forecasting accuracy is the most important factor in selecting any forecasting methods. Research efforts in improving the accuracy of forecasting models are increasing since the last decade. The appropriate stock selections those are suitable for investment is a very difficult task. The key factor for each investor is to earn maximum profits on their investments.

Because of dependency on various factors, the stock prices are dynamic, highly noisy, and nonlinear time series data. The stock market prediction has always caught the attention of many analysts and researchers. Predicting stock prices is a challenging problem in itself because of the number of variables which are involved. In the short term, the market behaves like a voting machine but in the longer term, it acts like a weighing machine and hence there is scope for predicting the market movements for a longer timeframe. Application of machine learning techniques and other algorithms for stock price analysis and forecasting is an area that shows great promise. In this project, we first provide a concise review of stock markets, We then focus on some of the research achievements in stock analysis and prediction. We discuss technical, fundamental, short- and long-term approaches used for stock.

This project aims to shed light on the process of web scraping, emphasizing its importance in the new ’Big Data’ era with an illustrative application of such methods in financial markets. The work essentially focuses on different scraping methodologies that can be used to obtain large quantities of heterogeneous data in real-time. Automatization of data extraction systems is one of the main objectives pursued in this work, immediately followed by the development of a framework for predictive modeling. applying neural networks and deep learning methods to the data obtained through web scraping.

Anyone with nary an idea on the stock market will have a tool that may assist them in the quicks and that is day trading. The economy of our country is not a toy for huge corporations to fiddle with. It is the livelihood of many and retribution against the corporations is due with the first step being the mob short selling of GameStop by Redditors. These methods are applied on 5 years of data retrieved from Yahoo Finance. The results will be used to analyze and predict price

CONTENTS

1. INTRODUCTION

1.1 OBJECTIVE

The ultimate goal of our application is to serve retail investors as a third-party investment tool that uses machine learning to help them navigate in the fast-changing stock market. The project aims to introduce and democratize the latest machine learning technologies that is Neural Networks for retail investors. No prediction is 100% accurate. Therefore, the upper bound and lower bound of the stock prices will be displayed to illustrate the trading range the investors should be looking at. This application serves as a supplementary quantitative tool for investors to see the market at a different perspective with the help of technology.

The objective of the proposed work is to study and improve the supervised learning algorithms to predict the stock price. The technical objectives will be implemented in Python. The system must be able to access a list of historical prices. It must calculate the estimated price of stock based on the historical data. It must also provide an instantaneous visualization of the market index

Figure 1: Stock Market Analysis

1.2 WHAT IS THE PROBLEM?

Investors are familiar with the saying, “buy low, sell high” but this does not provide enough context to make proper investment decisions. Before an investor invests in any stock, he needs to be aware how the stock market behaves. Investing in a good stock but at a bad time can have disastrous results, while investment in a mediocre stock at the right time can bear profits. Financial investors of today are facing this problem of trading as they do not properly understand as to which stocks to buy or which stocks to sell in order to get optimum profits. Predicting long term value of the stock is relatively easy than predicting on day-to-day basis as the stocks fluctuate rapidly every hour based on world events.

1.3 WHY THIS IS A PROJECT RELATED TO THIS CLASS?

The solution to this problem demands the use of tools and technologies related to the field of data mining, pattern recognition, machine learning and data prediction. The application will predict the stock prices for the next trading day. The requirements and the functionality of this application correlates it to the class.

1.4 WHY THIS APPROACH IS BETTER?

The Proposed approach makes use Long Short-Term Memory (LSTM).

The major advantage of using this method is LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications

1.5 TYPES OF NEURAL NETWORKS?

Neural networks can be classified into different types which are used for different purposes. The below listed are common types of Neural Networks

1.5.1 Multi-layer perceptron (MLP)

A multi-layer perceptron (MLP) is a perceptron however there is added complexity through the advent of layers. There are three types of layers in an MLP:

Input Layer, Hidden Layer and Output Layer.

Input Layer: The input layer is what it sounds like, the data you are inputting into the neural network.

Hidden Layer: The hidden layers are composed of most of the neurons in the neural network and is the heart of manipulating the data to get a desired output

Output Layer: The output layer is the final product from manipulating the data in the neural network and can represent different things

1.5.2 Convolution Neural Networks (CNN)

A convolutional neural network still uses the same principles that MLPs use, however this neural network implements convolutional layers. It is important to note that convolutional neural networks are usually used for images and video.

1.5.3 Recurrent Neural Networks (RNN)

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior.

RNNs can use their internal state (memory) to process variable length sequences of inputs.

Data that depends on past instances of itself to predict the future are examples of temporal data. Things like stock market data, time-series data, brain-wave data

1.6 PROBLEM STATEMENT

Financial analysts investing in stock market usually are not aware of the stock market behavior. They are facing the problem of trading as they do not properly understand which stocks to buy or which stocks to sell in order to get more profits. In today’s world, all the information pertaining to stock market is available. Analyzing all this information individually or manually is tremendously difficult. As such, automation of the process is required. This is where Data mining techniques help. Understanding that analysis of numerical time series gives close results, intelligent investors use machine learning techniques in predicting the stock market behavior. This will allow financial analysts to foresee the behavior of the stock that they are interested in and thus act accordingly. The input to our system will be historical data from Yahoo Finance. Appropriate data would be applied to find the stock price trends. Hence the prediction model will notify the up or down of the stock price movement for the next trading day and investors can act upon it so as to maximize their chances of gaining a profit. The entire system would be implemented in Python programming language using open source libraries. Hence it will effectively be a zero-cost system.

2. LITERATURE SURVEY

2.1 STOCK PRICE PREDICTIONS

The art of forecasting the stock prices has been a difficult task for many of the researchers and analysts. In fact, investors are highly interested in the research area of stock price prediction. For a good and successful investment, many investors are keen in knowing the future situation of the stock market. Good and effective prediction systems for stock market help traders, investors, and analyst by providing supportive information like the future direction of the stock market. In this work, we present a recurrent neural network (RNN) and Long Short-Term Memory (LSTM) approach to predict stock market indices. The initial focus of our literature survey was to explore generic online learning algorithms and see if they could be adapted to our use case i.e., working on real-time stock price data. These included Online AUC Maximization, Online Transfer Learning, and Online Feature Selection. However, as we were unable to find any potential adaptation of these for stock price prediction, we then decided to look at the existing systems, analyze the major drawbacks of the same, and see if we could improve upon them. We zeroed in on the correlation between stock data (in the form of dynamic, long-term temporal dependencies between stock prices) as the key issue that we wished to solve. A brief search of generic solutions to the above problem led us to RNN's and LSTM. We decided to use an LSTM neural network to perform stock predictions. We concluded our literature survey by looking at how LSTM can be used to predict the close prices of tech giants like Apple, Microsoft, IBM, Facebook.

2.2 NEURAL NETWORKS

A neural network attempts to learn a function that maps the input features to the output predictions, serving as a universal function approximator. It consists of a network of neurons, each of which represents a weighted sum of inputs. Outputs from neurons are fit into activation functions which introduce non-linearity to the system, and then passed to some other neurons. In a typical dense feedforward neural network, the network consists of layers of neurons stacked together, with neurons between individual layers fully connected.

Optimization of neural networks is usually done through backpropagation with gradient descent, which essentially propagates the error from the output layer back to the input layer, while computing the gradient of the error against each parameter in the process.

2.3 RECURRENT NEURAL NETWORKS

Recurrent neural network is a type of neural network where connections between neurons allow temporal, sequential information to be stored and processed in the network. One typical architecture is formed by feeding the output of the current unit back to the input with a time delay so that the network can use the information in processing the next input. Various techniques have been developed over the years to train such type of network. One of the popular approaches is backpropagation through time (BPTT), whose central idea is to unroll the recurrent network into a feedforward network, where each layer represents a timestep.

The term “recurrent neural network” is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled.

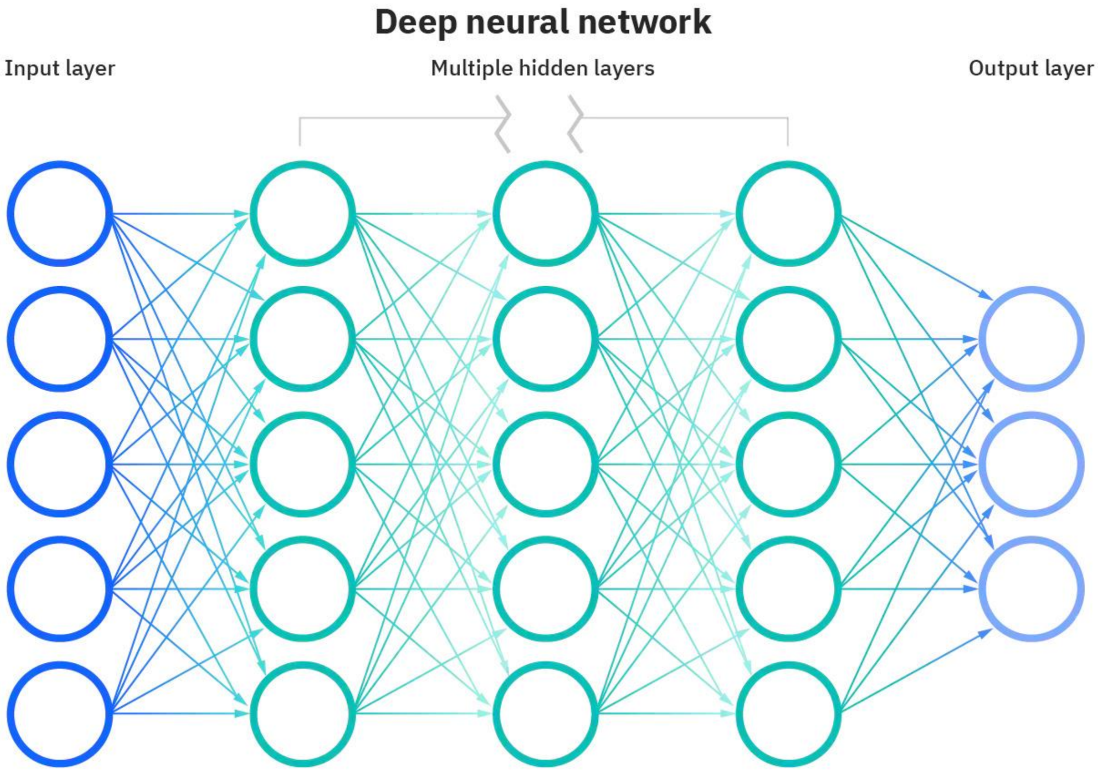


Figure 2: Deep Neural Network

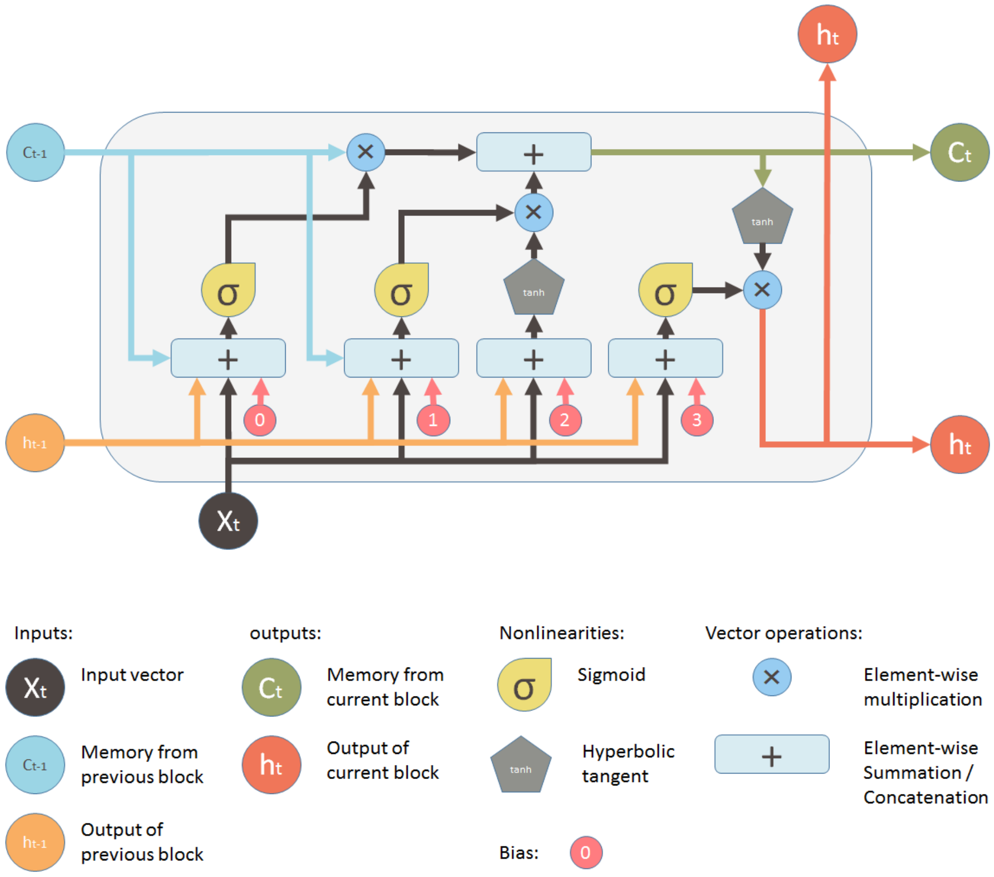
2.4 LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies.

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

All recurrent neural networks have the form of a chain of repeating modules of neural networks. In standard RNNs, this repeating module will have a very simple structure.

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.



ht
Ct-1
Ct
σ
σ
tanh
σ
tanh
0
1
2
3
ht-1
ht
Xt
Inputs:
Xt Input vector
Ct-1 Memory from previous block
ht-1 Output of previous block
outputs:
Ct Memory from current block
ht Output of current block
Nonlinearities:
σ Sigmoid
tanh Hyperbolic tangent
Vector operations:
× Element-wise multiplication
+ Element-wise Summation / Concatenation
Bias: 0

Figure 3: Architecture of LSTM

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

An LSTM has four “gates”: Forget, Remember, Learn and Use (or output)

It also has three inputs: long-term memory, short-term memory, and E. (E is some training example/new data)

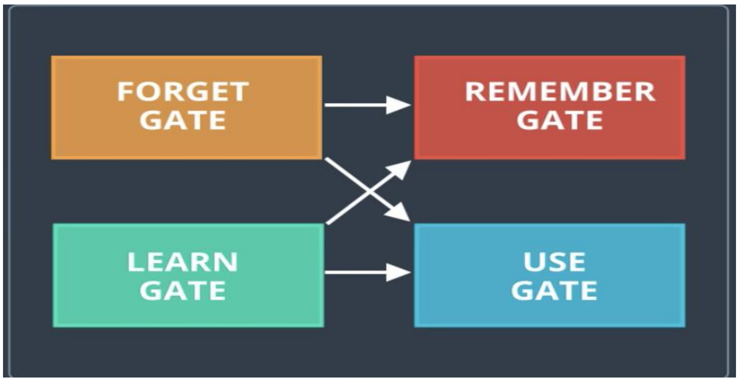


Figure 4: Information flow in LSTM cell

Learn Gate

This gate combines existing Short-term memory (STM) and some input “E, multiplies by a matrix (W) and adds b. Then squishes this all into a tanh function.

$$N_t = \tanh(W_n[STM_{t-1}, E_t] + b_n)$$

This combination gives us “N”.

Then it ignores some of the short-term memory, by multiplying the combined result by an “ignore factor”.

The ignore factor (I) is calculated by combining STM and E, with a new set of W(weights) and b(biases)

$$i_t = \sigma(W_i[STM_{t-1}, E_t] + b_i)$$

Once we have N and I, we multiply them together, and that’s the result of

the learn gate.

We have “learned” our new information (E).

Forget Gate

Forget gate is the gate you use to dump out all the unnecessary long-term information. Kind of like when you study for a big exam, and the next day you forget everything. That's the power of the forget gate.

Basically, the long-term memory (LTM) gets multiplied by a **forget factor (f).** This factor will make some of the long-term information be "forgotten"

The forget factor is this:

$$f_t = \sigma(W_f[STM_{t-1}, E_t] + b_f)$$

It is computed by taking the short-term memory, and input (E), multiplying them by some weights and biases and squishing them into a sigmoid function.

This function (f) gets multiplied by LTM — and boom, we're left with LTM that we *need.*

Remember Gate

This gate takes the information from the forget gate and adds it to the information from the learn gate, to compute the new long term memory.

Remember gate = Learn gate output + Forget gate output

Use Gate

Use gate takes the LTM from the forget gate, and STM + E from the learn gate and uses them to come up with a new short-term memory or an output (same thing).

For example, if we were trying to classify images, the output would be the network classification.

It takes the output of the learn gate, and applies a sigmoid function, so the equation looks like this:

$$V_t = \sigma(W_v[STM_{t-1}, E_t] + b_v)$$

3. REQUIREMENTS ANALYSIS AND FEASIBILITY STUDY

3.1 FEASIBILITY STUDY

3.1.1 Technical Feasibility

Feasibility Study Simply put, stock market cannot be accurately predicted. The future, like any complex problem, has far too many variables to be predicted. The stock market is a place where buyers and sellers converge. When there are more buyers than sellers, the price increases. When there are more sellers than buyers, the price decreases. So, there is a factor which causes people to buy and sell. It has more to do with emotion than logic. Because emotion is unpredictable, stock market movements will be unpredictable. Although, we've tried incorporating factor with the most weight in predicting stocks it is invariably futile to be accurate but, it is not so hopeless when predicting trends instead of exact numbers and that's what this project aims to do with the help of LSTM.

3.1.2 Economical Feasibility

The product that is made here is made by using completely open source information and tools and there was no need for a budget or money and so it is accurate to say that this project is economically feasible

3.1.3 Functional Feasibility

There are some fundamental financial indicators by which a company's stock value can be estimated. Some of the indicators and factors are: Price-to-Earning (P/E) Ratio, Price-to-Earning Growth (PEG) Ratio, Price-to-Sales (P/S) Ratio, Price/Cash Flow (P/CF) Ratio, Price-to-Book Value (P/BV) Ratio and Debt-to-Equity Ratio. Some of the parameters are available and accessible on the web but all of them aren't. So we are confined to use the variables that are available to us. The proposed system will not always produce accurate results since it does not account for the human behaviours. Factors like change in company's leadership, internal matters, strikes, protests, natural disasters, and change in the authority cannot be taken into account for relating it to the change in Stock market by the machine. The objective of the system is to give an approximate idea of where the stock market might be headed. It does not give a long term forecasting of a stock value. There are way too many reasons to acknowledge for the long term output of a current stock. Many things and parameters may affect it on the way due to which long term forecasting is just not feasible

3.2 REQUIREMENTS ANALYSIS

After the extensive analysis of the problems in the system, we are familiarized with the requirement that the current system needs. The requirement that the system needs is categorized into the functional and non-functional requirements. These requirements are listed below:

3.2.1 Functional Requirements

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require are as follows:

- The system should be able to generate an approximate share price.
- The system should collect accurate data from the NEPSE website in consistent manner.

3.2.2 Non-Functional Requirements

Non-functional requirement is a description of features, characteristics and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. The non-functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. Based on these the non-functional requirements are as follows:

- The system should provide better accuracy.
- The system should have simple interface for users to use.
- To perform efficiently in short amount of time.

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

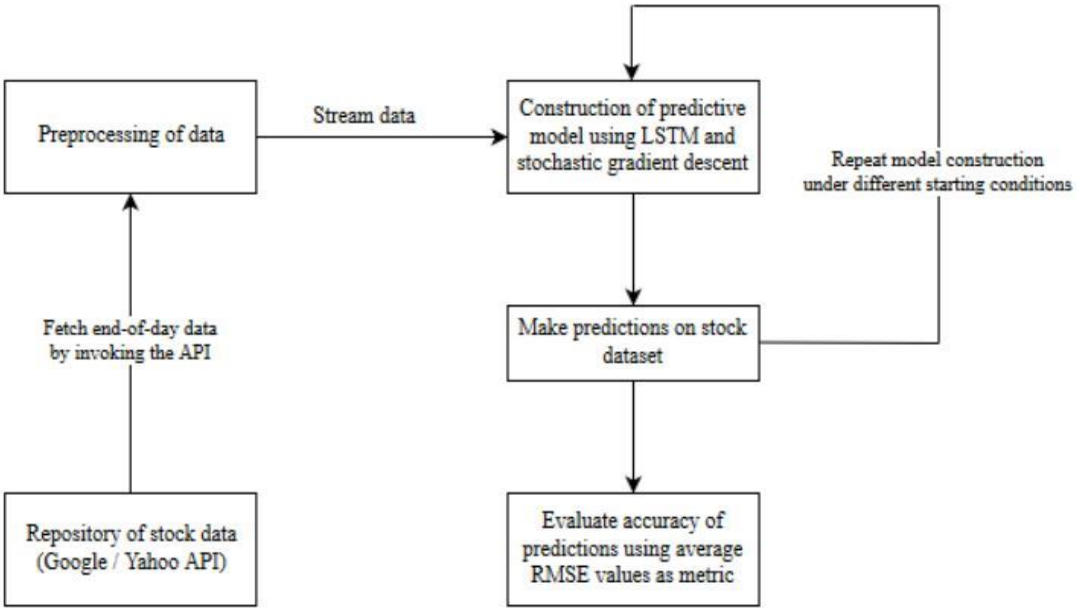

Preprocessing of data
Stream data
Construction of predictive model using LSTM and stochastic gradient descent
Repeat model construction under different starting conditions
Fetch end-of-day data by invoking the API
Make predictions on stock dataset
Repository of stock data (Google / Yahoo API)
Evaluate accuracy of predictions using average RMSE values as metric

Figure 5: System Architecture

4.2 FLOW DIAGRAM

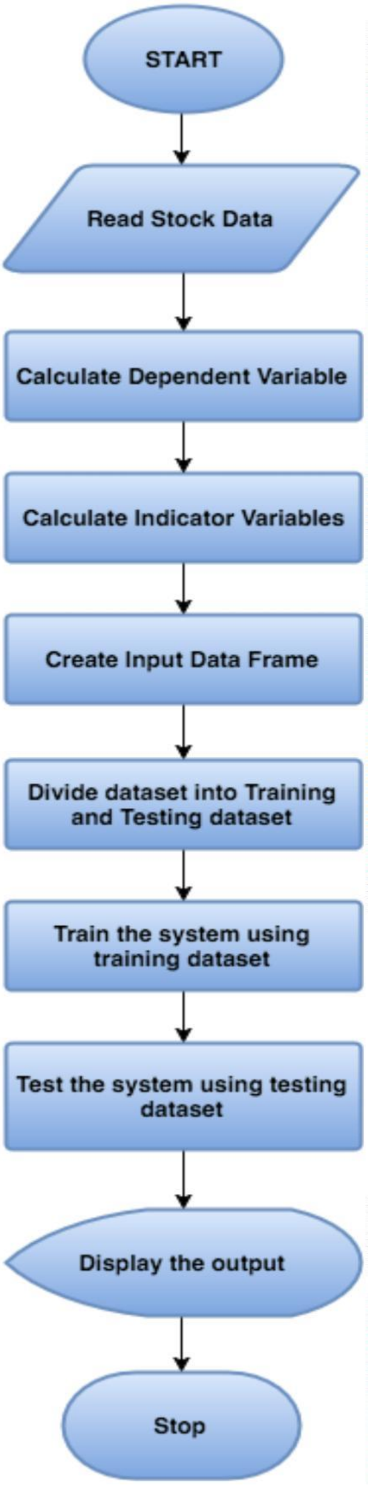

START
Read Stock Data
Calculate Dependent Variable
Calculate Indicator Variables
Create Input Data Frame
Divide dataset into Training and Testing dataset
Train the system using training dataset
Test the system using testing dataset
Display the output
Stop

Figure 6: Flow Diagram

4.3 USE CASE DIAGRAMS

4.3.1 Use Case Diagram for App Admin

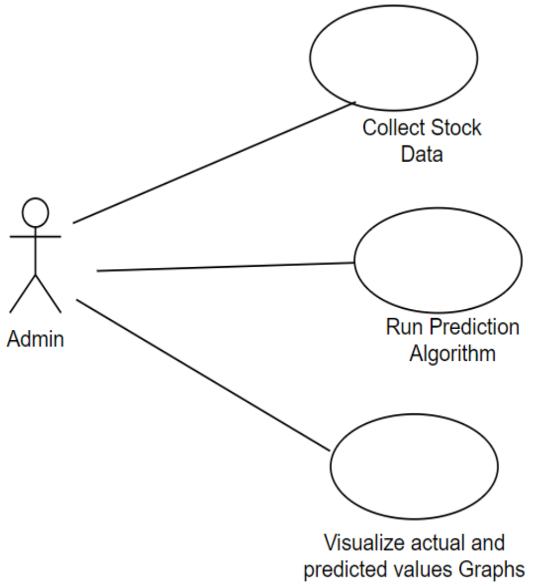


Figure 7: Use case diagram from App Admin

Use Case Index

Use case ID	Use case name	Primary actor	Scope	Complexity	Priority
1	Collect stock data	Admin	In	High	1
2	Run Prediction Algorithm	Admin	In	High	1
3	Visualize Actual and Predicted values Graphs	Automated UI Appln	In	High	1

Table 1: Use Case Index

Use case description:

Use case ID: 1

Use case name: Collect stock data

Description: Every required stock data of NSE will be available in Yahoo Finance. Automated User Interface Application Backend will be able to collect the data for system.

Use case ID: 2

Use case name: Run Prediction Algorithm

Description: Prediction result will be handled and generated by Automated User Interface Application Backend. The system will be built, through which the result of prediction and system performance will be analyzed.

Use case ID: 3

Use case name: Visualize Actual and Predicted Values Graphs

Description: The Actual and Predicted values of close prices is visualized on the dashboard.

4.3.2 Use Case Diagram for User

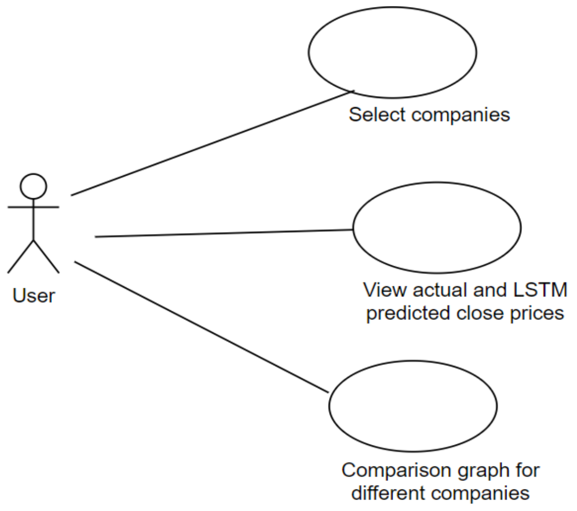

Select companies
User
View actual and LSTM predicted close prices
Comparison graph for different companies

Figure 8: Use case diagram from App Admin

Use case ID	Use case name	Primary actor	Scope	Complexity	Priority
1	Select Companies	User	In	High	1
2	View Actual and LSTM Predicted Prices	User	In	High	1
3	Comparison Graphs for Different Companies	Automated UI Appln	In	High	1

Table 2: Use Case Index

Use case description:

Use case ID: 1

Use case name: Select Companies

Description: The User can Select Multiple Companies embedded in the Application to view the Close Prices of particular dates.

Use case ID: 2

Use case name: View Actual and LSTM Predicted Prices

Description: The Actual and LSTM Predicted values of close prices is visualized on the dashboard in the form of Graphs.

Use case ID: 3

Use case name: Comparison Graphs for different Companies

Description: The User can compare close prices for different companies listed on the menu.

4.4 ENTITY RELATIONSHIP DIAGRAMS

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between instances of those entity types.

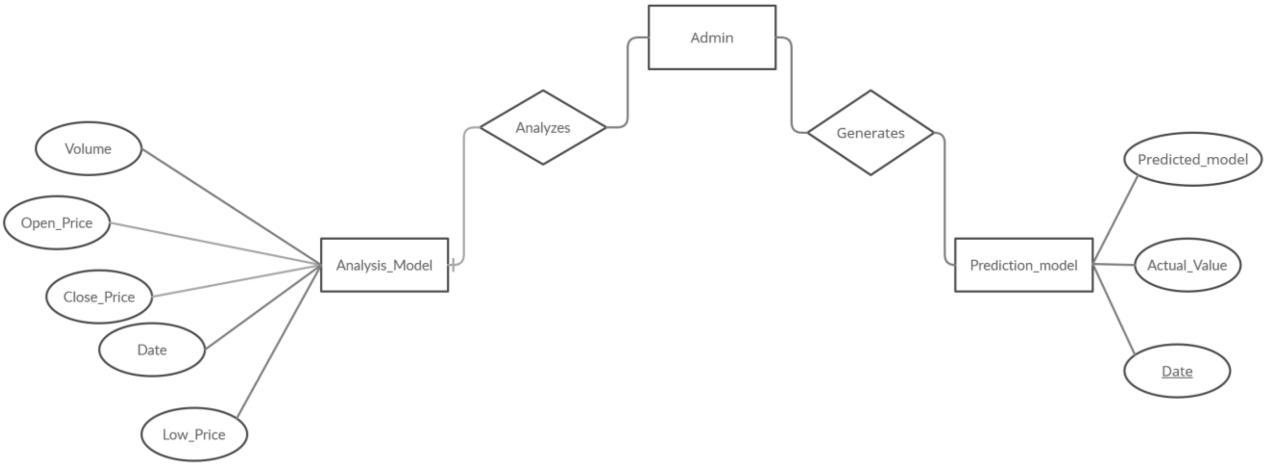


Figure 9: E-R Diagram(1-1)

5. METHODOLOGY - DESIGN

5.1 HOW TO COLLECT INPUT?

Input data is taken from Yahoo Finance using following steps:

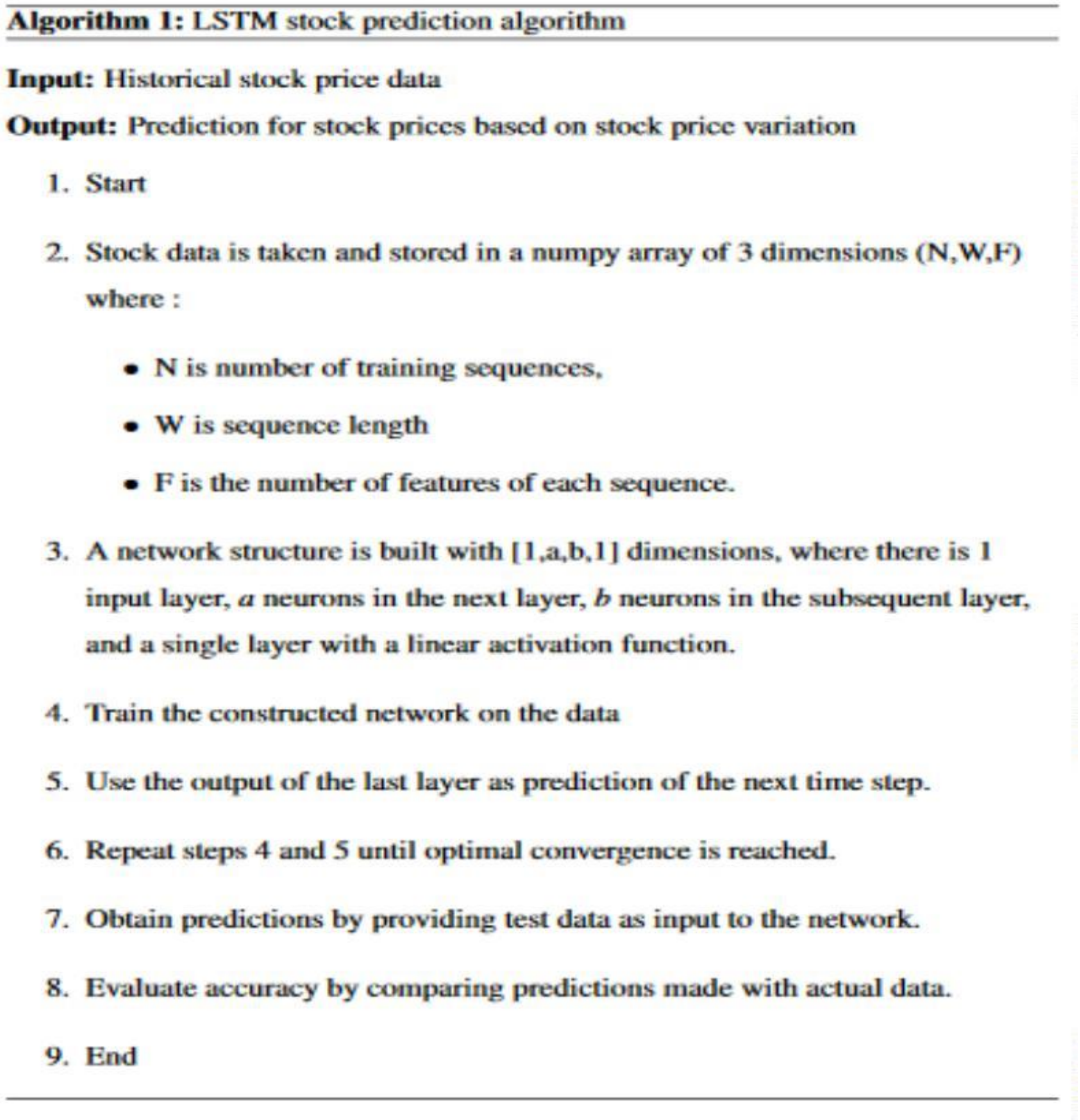
1. For our project, we are considering Tech Giants namely Apple, Microsoft, Facebook and IBM
2. Use stock's ticker symbol from step a to get data from Yahoo Finance.
3. System will take last 5 years' stock data of the company using 'pandas' library in python.
4. Further we divide the data into two parts, Training data and Testing data, where 70% of the data will be used for training and 30% of the data will be used for testing.

5.2 HOW TO SOLVE PROBLEM?

To solve this problem, we have the below steps"

Step 1: Collect stock data from Yahoo Finance platform.
Step 2: Split the data into Training and Testing datasets
Step 3: Train the LSTM model using Training dataset
Step 4: Test the LSTM model using Testing dataset.
Step 5: Plot the graph of Actual prices and LSTM Predicted prices and display on a dashboard.

5.2.1 ALGORITHM DESIGN



Algorithm 1: LSTM stock prediction algorithm

Input: Historical stock price data

Output: Prediction for stock prices based on stock price variation

1. Start
2. Stock data is taken and stored in a numpy array of 3 dimensions (N,W,F) where :
   - N is number of training sequences,
   - W is sequence length
   - F is the number of features of each sequence.
3. A network structure is built with [1,a,b,1] dimensions, where there is 1 input layer, *a* neurons in the next layer, *b* neurons in the subsequent layer, and a single layer with a linear activation function.
4. Train the constructed network on the data
5. Use the output of the last layer as prediction of the next time step.
6. Repeat steps 4 and 5 until optimal convergence is reached.
7. Obtain predictions by providing test data as input to the network.
8. Evaluate accuracy by comparing predictions made with actual data.
9. End

5.2.2 LANGUAGE USED

Programming Language ‘Python’ is used because of its wide acceptance and is an interpreted high-level general-purpose programming language.

Features of Python:

- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems

5.2.2 LIBRARIES USED

- NUMPY
- PANDAS
- PANDA DATA READER
- KERAS
- DASH
- MATPLOTLIB
- SCIKIT LEARN

5.2.3 TOOLS USED

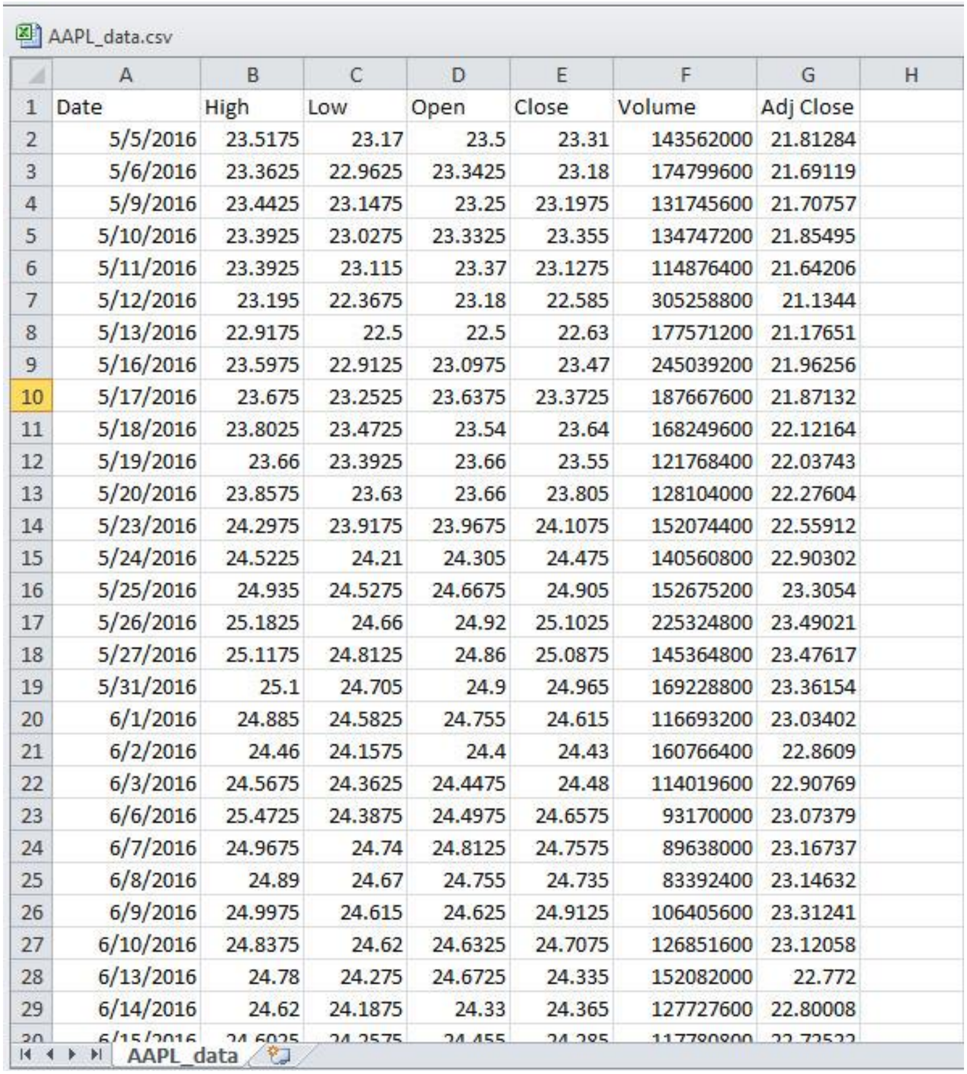
5.2.3.1 VISUAL STUDIO CODE

Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux, MacOS.

Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Program is Written, Compiled and Executed in VS Code.

5.2.4 DATASETS USED



AAPL_data.csv

	A	B	C	D	E	F	G	H
1	Date	High	Low	Open	Close	Volume	Adj Close	
2	5/5/2016	23.5175	23.17	23.5	23.31	143562000	21.81284	
3	5/6/2016	23.3625	22.9625	23.3425	23.18	174799600	21.69119	
4	5/9/2016	23.4425	23.1475	23.25	23.1975	131745600	21.70757	
5	5/10/2016	23.3925	23.0275	23.3325	23.355	134747200	21.85495	
6	5/11/2016	23.3925	23.115	23.37	23.1275	114876400	21.64206	
7	5/12/2016	23.195	22.3675	23.18	22.585	305258800	21.1344	
8	5/13/2016	22.9175	22.5	22.5	22.63	177571200	21.17651	
9	5/16/2016	23.5975	22.9125	23.0975	23.47	245039200	21.96256	
10	5/17/2016	23.675	23.2525	23.6375	23.3725	187667600	21.87132	
11	5/18/2016	23.8025	23.4725	23.54	23.64	168249600	22.12164	
12	5/19/2016	23.66	23.3925	23.66	23.55	121768400	22.03743	
13	5/20/2016	23.8575	23.63	23.66	23.805	128104000	22.27604	
14	5/23/2016	24.2975	23.9175	23.9675	24.1075	152074400	22.55912	
15	5/24/2016	24.5225	24.21	24.305	24.475	140560800	22.90302	
16	5/25/2016	24.935	24.5275	24.6675	24.905	152675200	23.3054	
17	5/26/2016	25.1825	24.66	24.92	25.1025	225324800	23.49021	
18	5/27/2016	25.1175	24.8125	24.86	25.0875	145364800	23.47617	
19	5/31/2016	25.1	24.705	24.9	24.965	169228800	23.36154	
20	6/1/2016	24.885	24.5825	24.755	24.615	116693200	23.03402	
21	6/2/2016	24.46	24.1575	24.4	24.43	160766400	22.8609	
22	6/3/2016	24.5675	24.3625	24.4475	24.48	114019600	22.90769	
23	6/6/2016	25.4725	24.3875	24.4975	24.6575	93170000	23.07379	
24	6/7/2016	24.9675	24.74	24.8125	24.7575	89638000	23.16737	
25	6/8/2016	24.89	24.67	24.755	24.735	83392400	23.14632	
26	6/9/2016	24.9975	24.615	24.625	24.9125	106405600	23.31241	
27	6/10/2016	24.8375	24.62	24.6325	24.7075	126851600	23.12058	
28	6/13/2016	24.78	24.275	24.6725	24.335	152082000	22.772	
29	6/14/2016	24.62	24.1875	24.33	24.365	127727600	22.80008	
30	6/15/2016	24.6025	24.2575	24.455	24.285	117780800	22.72522	

Table 3: Stock Market dataset

The above is a sample dataset which is used for analysis of stock market data. It is in '.csv' format. It contains fields like Date, High Price, Low Price, Open Price, Close Price, Volume.

Date: The date on which stock price is noted.

High Price: The highest recorded stock price for that particular date.

Low Price: The lowest recorded stock price for that particular date.

Close Price: closing price is the last price anyone paid for a share of that stock during the business hours of the exchange where the stock trades.

Open Price: The opening price is the price from the first transaction of a business day.

Volume: Trading volume is a measure of how much of a given financial asset has traded in a period of time

The datasets used are:

AAPL.csv

MSFT.csv

TSLA.csv

FB.csv

- Used datasets of Apple, Microsoft, Tesla and Facebook.
- Used Historical data of 5 years (2016-2021)
- 80 percent data used to train the Neural Network model and 20 percent to test the Neural Network model.
- Used Yahoo finance to get the accurate data with the help of pandas data-reader library of python

5.3 HOW TO GENERATE OUTPUT?

Steps to generate the Output

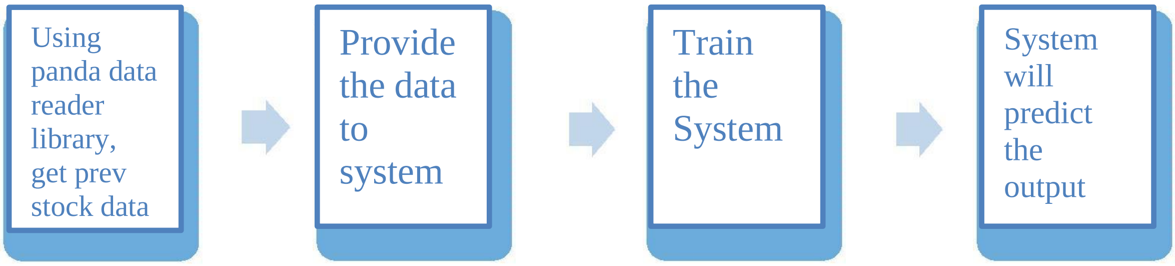

Using panda data reader library, get prev stock data
Provide the data to system
Train the System
System will predict the output

Figure 10: Steps to generate output

Perform the above below to generate output

Step 1: Using pandas data reader library, get the last 4 years' data

Step 2: Provide the data to the system.

Step 3: Train the system

Step 4: System will predict output

6. METHODOLOGY – IMPLEMENTATION

6.1 RESEARCH IMPLEMENTATION

6.1.1 Stock Price Data Collection

Data is collected from Yahoo Finance. It offers up to 20 years of daily stock price information on S&P500 stocks. A Python script is written to retrieve stock prices of different stocks automatically. The retrieved stock prices are stored as .csv files in a local folder during development and testing. In deployment, the downloaded stock price data will be transformed into a Graph which is displayed on screen.

6.1.2 Data Preprocessing

Python scripts written to transform the raw stock prices (.csv files) into feature vectors, for training, predicting and testing respectively. The scripts take the input options and the raw stock prices as inputs and produce the correct features by building the lookback arrays and the moving averages. It concatenates the features into the final feature vectors, which will be passed to the model for training or testing. The 3 scripts share common operations in building a dataset except the output size and the range of dates to build from, so common functions are written to centralize the logic instead of repeating the same index-calculation-intensive work across functions. NumPy and Pandas are used to build the datasets.

“Numpy” is a library that provides effective n-dimensional array data structures as well as functions for array manipulations. It is frequently used for machine learning tasks because it is much for performant than Python lists, as NumPy arrays are implemented as densely packed lists, instead of a dynamic array where the elements are not stored contiguously.

“Pandas” is a popular framework for pre-processing time series data. It has various utilities for reading raw input files such as .csv and transforming time series data to the correct format. Pandas uses NumPy as the underlying data structure, so it is very convenient to interoperate between the two

6.1.3 Model

A model base class is used as a common interface for all machine learning models. All models then have their own model class, specifying model-specific details like methods to build the model, train the model, use the model and save the model. To decouple model configurations from software code to provide flexibility and robustness and save engineering effort as mentioned in 2.2.2, each model is defined by a JSON object, which specifies the model's architecture and hyperparameters with model options and the model inputs with input options. A corresponding model can then be created by passing the object to the model class constructor. The model options specify which machine learning model to use, and the hyperparameters for the model like the number of hidden layers, the number of hidden units, activation functions used, as well as optimization algorithms and loss functions. Some example model options are in Appendix A. Apart from model configurations, the input can also vary, as there are many possible features that could be added to or removed from the feature vectors. The input options specify the features input that a model should expect, like the number of previous stock prices as features and different moving averages. The input options are related to a model in terms of the input format. All neural networks built in Keras requires the input tensor shape for layer shape inference during model building, a Python function is written to calculate the input shape for a given input option.

6.1.4 Training

In training, a randomized initial model is first generated from the model options definition. A training set is generated by the build training dataset script, which generates the training set features from the input options and the raw stock price data. Then, the data is fed into the model for training.

6.1.5 Saving Trained Model

All trained models are saved for predicting stock prices in the future. Keras models are saved in h5 format, and scikit-learn models are saved with a Python library named pickle. A dedicated saving format is designed (Appendix C), such that same models (same hash for same model options and input options) for different stocks are saved in the same directory with no collision

6.1.6 Predicting Stock Price

When predicting stock price, the saved model will first be loaded. Then, a feature vector specified by the input options is built with the build predict dataset script, which is the same as the build training dataset except it returns a flatten 1D feature vector. The feature vector is inputted into the model to predict stock price. For 10-day predict, the predictions are directly outputted. For 1-day predict, the predicted stock price is appended to the raw dataset as if it Page 44 of 124 happened before, then a new feature vector is generated for predicting the stock price for the day after, the process is repeated to predict the stock prices for all next 10 days

7. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components. Sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Test Cases:

A test case is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

7.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Test Case: Stock Close Price Prediction

Test Objective: To check whether the predicted stock prices are accurate and to compare with actual stock prices

Test Description: Stock data collected from Yahoo finance and separated as two different datasets namely ‘Training dataset’ and ‘Testing dataset’ and the LSTM algorithm predicted the close prices which gave accurate results.

Test Results: All the test cases mentioned above passed successfully. No discrepancy was encountered leading to the analysis.

8. ANALYSIS AND OUTPUT

8.1 ANALYSIS

The factors that are taken into account for change in the closing price of a particular company are: General Index, Price difference, highest value, lowest value, share volume and closing price. We performed analysis on obtained data to establish relation between our output parameters and the selected factors.

Our Predicted model is not 100% accurate but with the fluctuations it is nearer to accurate model which can be very useful for prediction of future closing prices for companies.

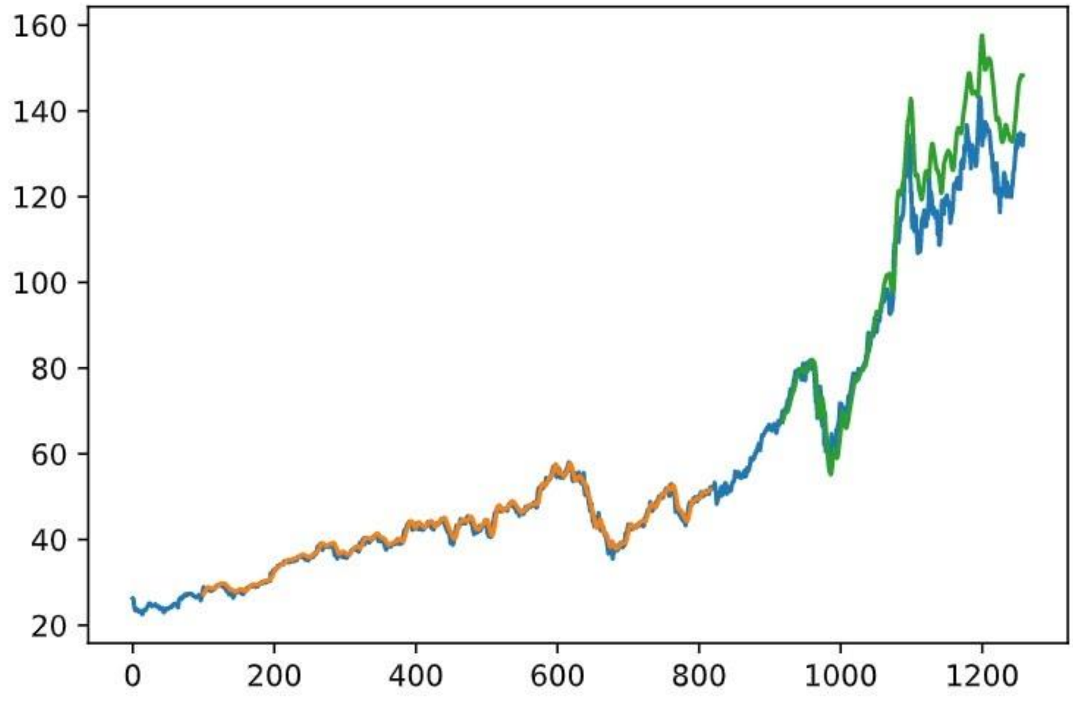
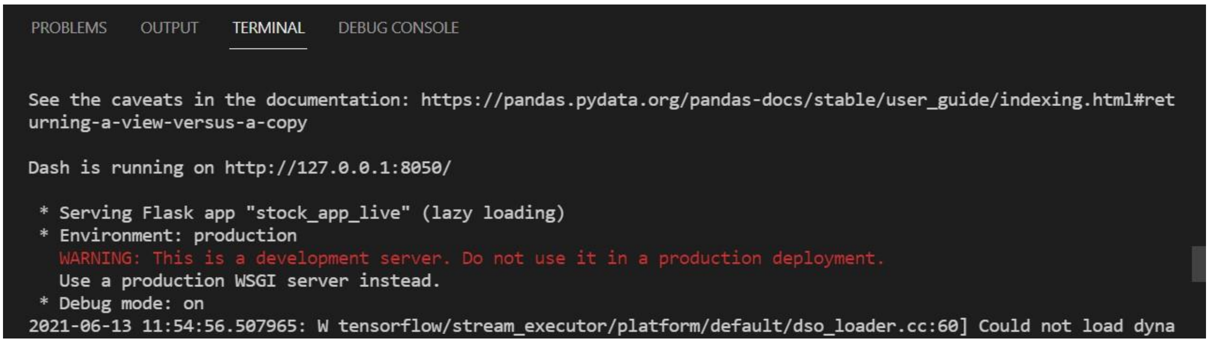


Figure 11: Stock Market Analysis

8.2 OUTPUT

After collection of data the future share price is predicted using Artificial Neural Network model LSTM. The value is then compared the next day with the actual value. The results and deviations of three random companies namely APPLE, FACEBOOK, TESLA and MICROSOFT are illustrated in graphical form below.



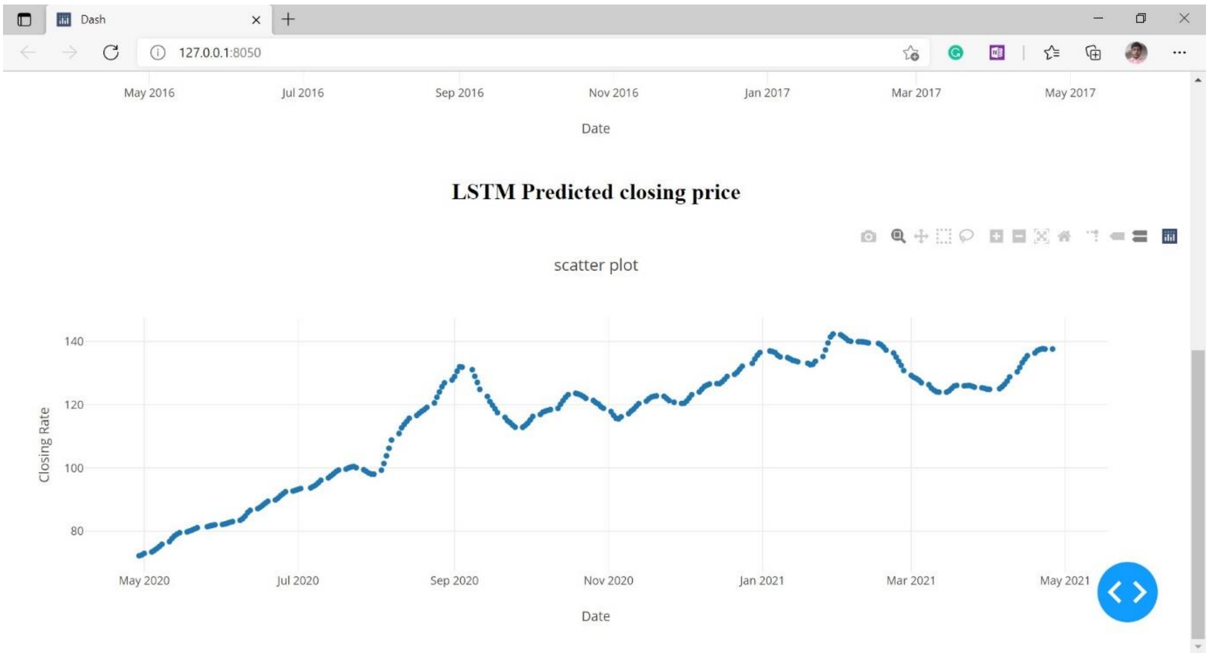
```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy

Dash is running on http://127.0.0.1:8050/

 * Serving Flask app "stock_app_live" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
2021-06-13 11:54:56.507965: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dyna
```

Figure 12: Execution Terminal

Following the link opens a website with 5 tabs:

1. Apple Stock Data
2. Tesla Stock Data
3. Microsoft Stock Data
4. Facebook Stock Data
5. Comparison of Stock Data

1. Apple Stock Data with Prediction



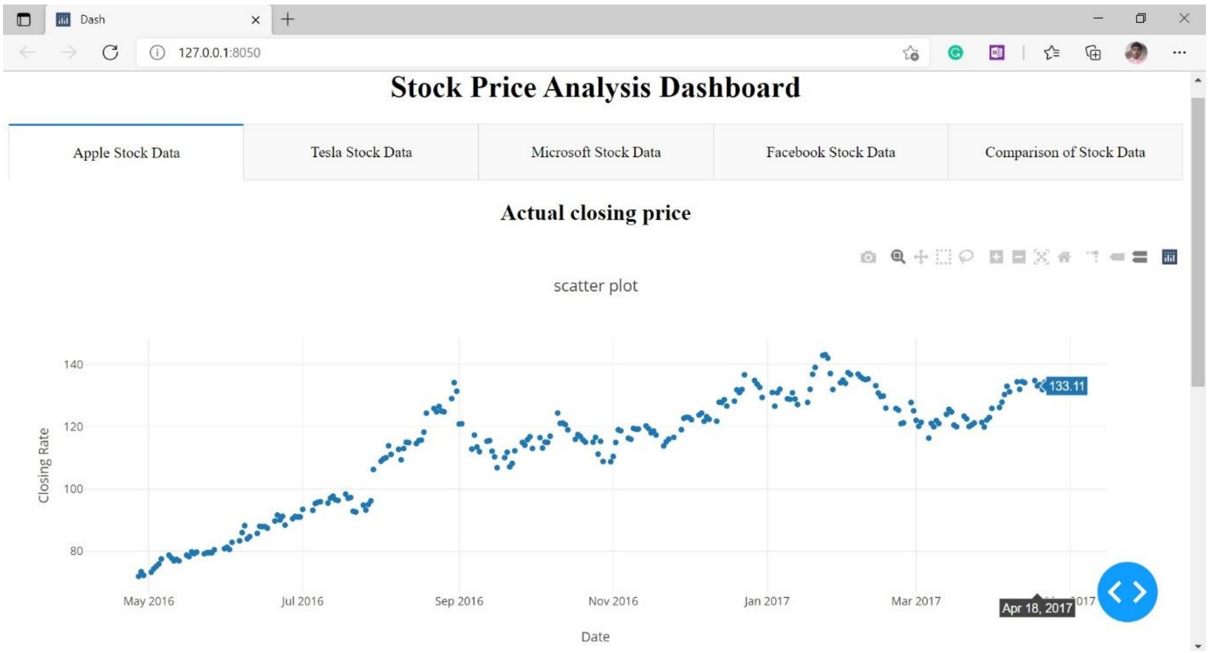
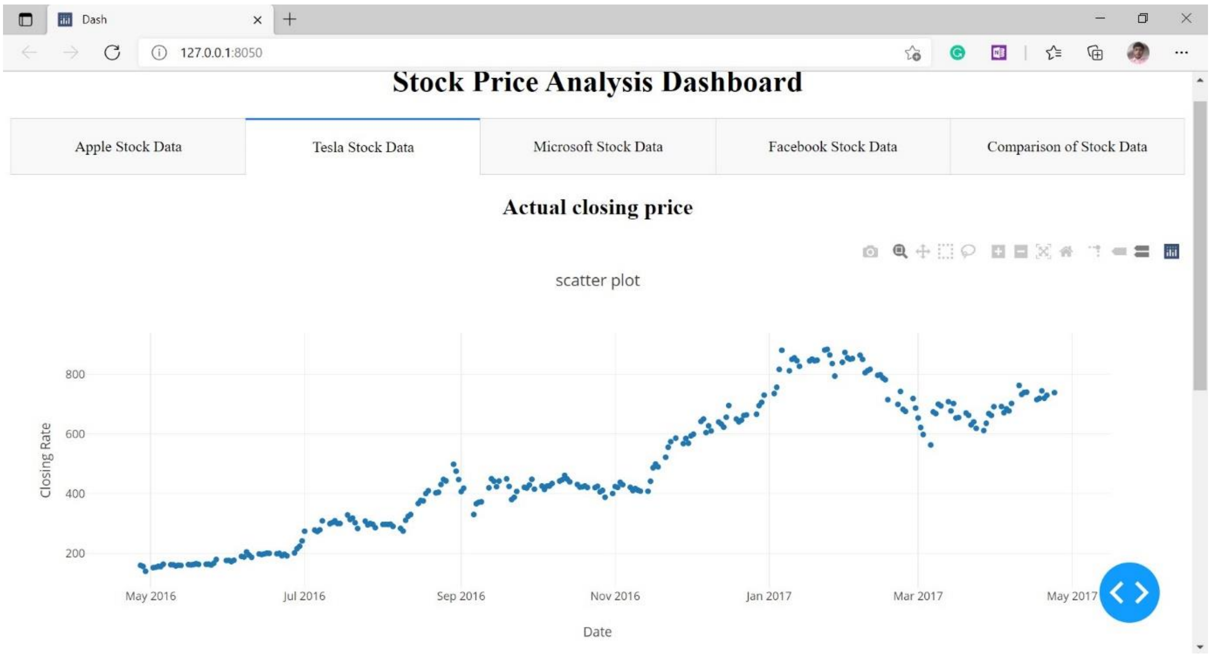


Figure 13: Actual vs LSTM predicted stock data of Apple

2. Tesla Stock Data with prediction



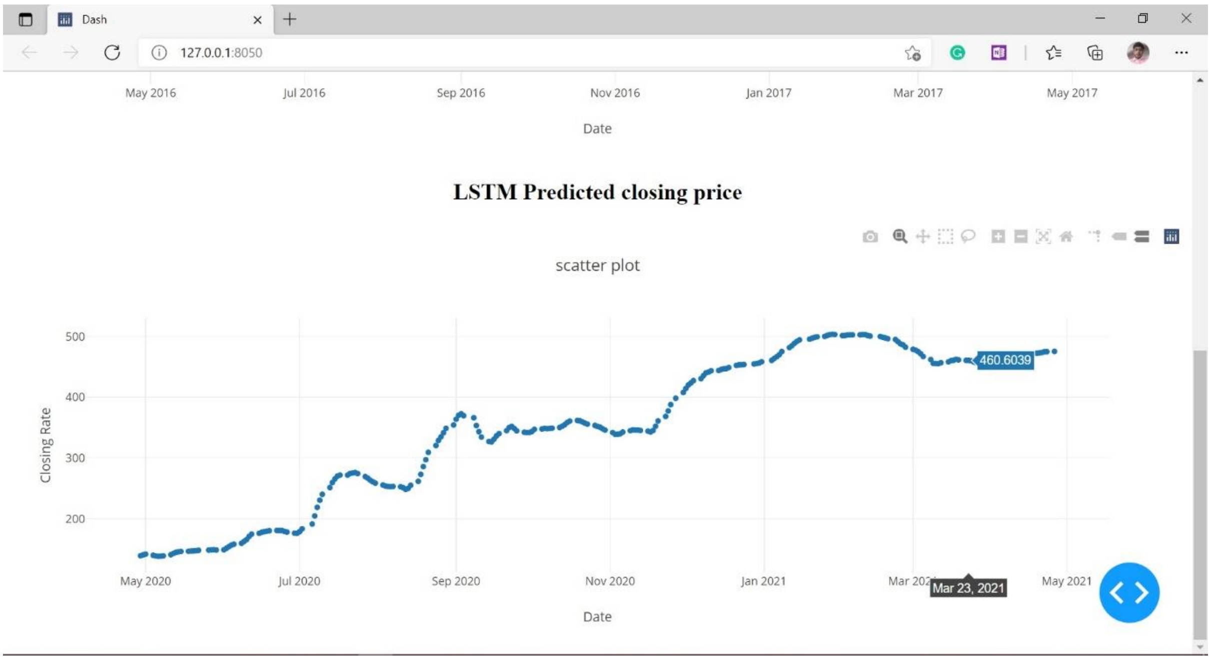
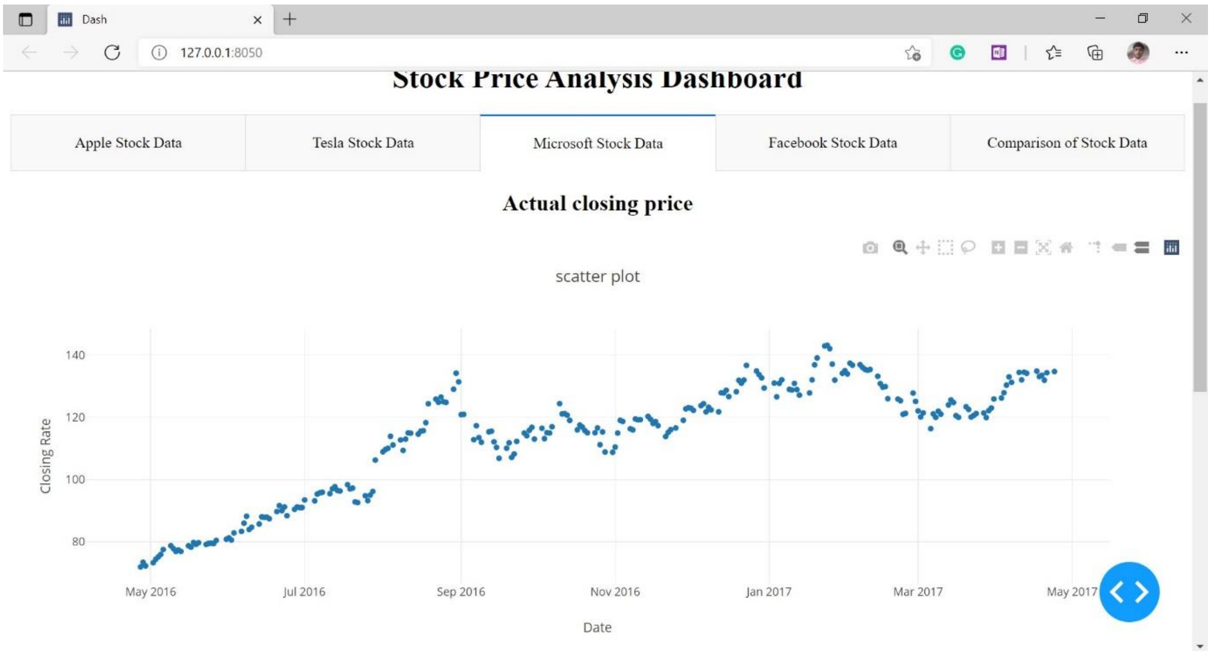


Figure 14: Actual vs LSTM predicted stock data of Tesla

3. Microsoft Stock Data



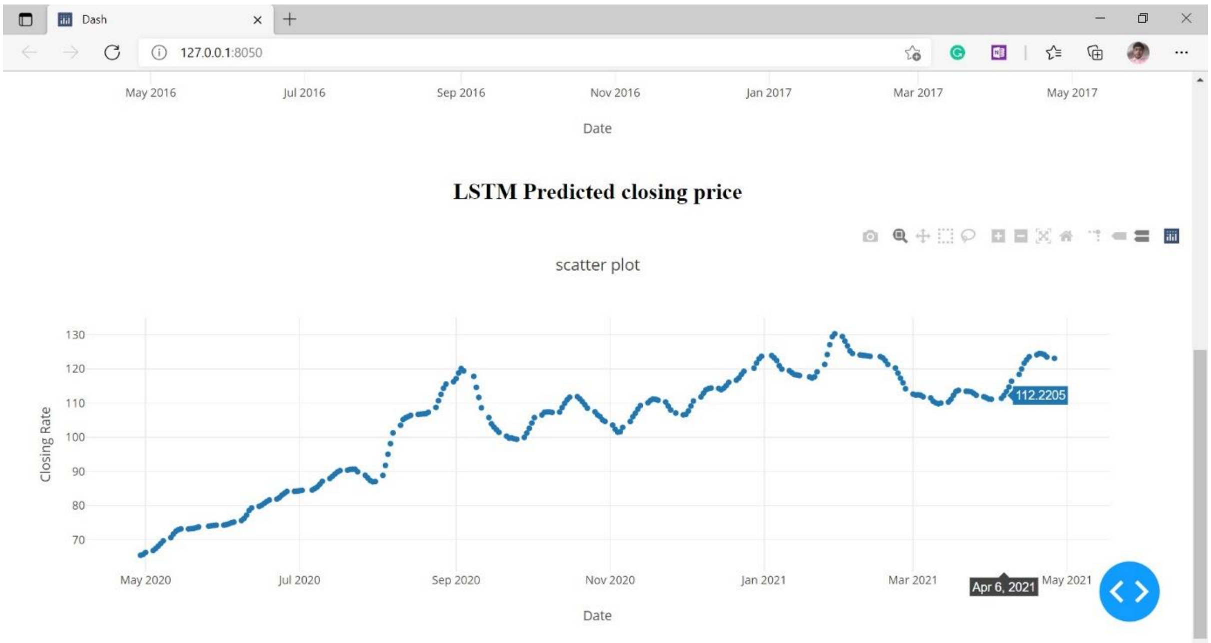
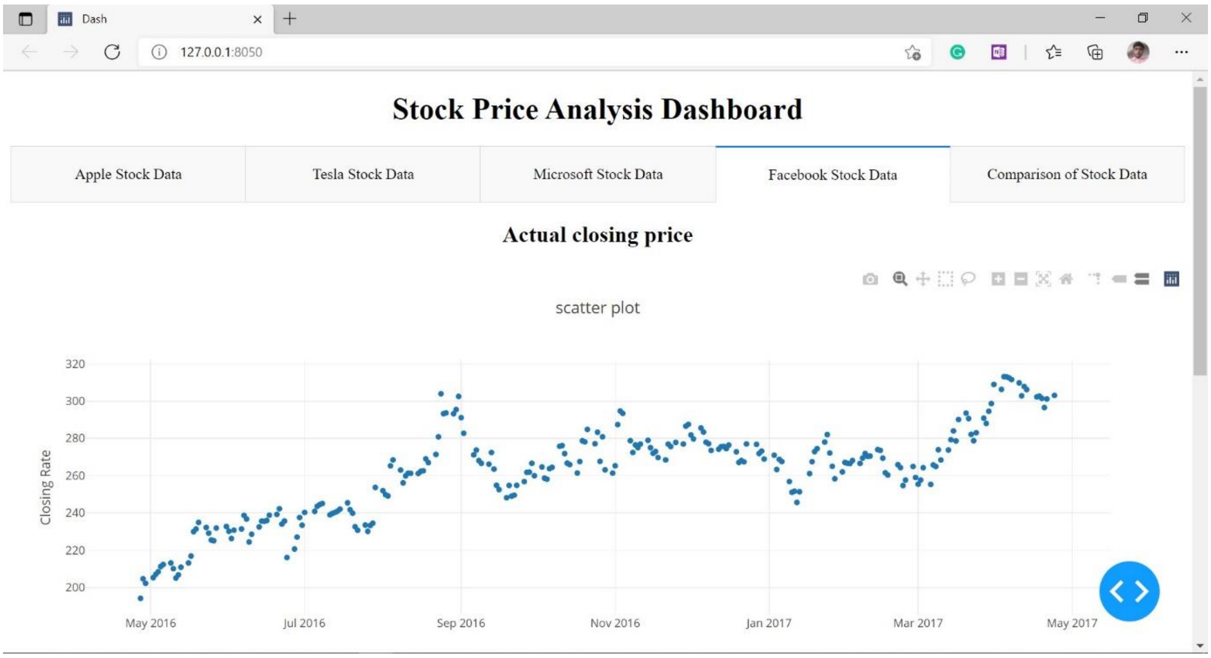


Figure 15: Actual vs LSTM predicted stock data of Microsoft

4. <u>Facebook Stock Data with Prediction</u>



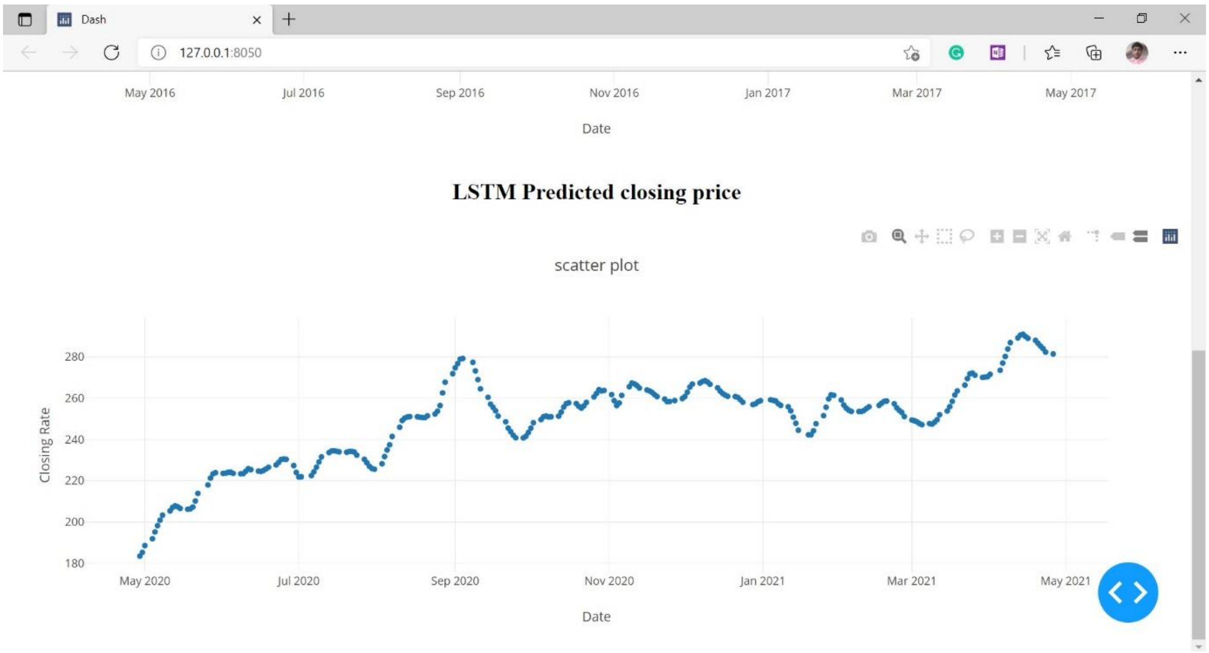
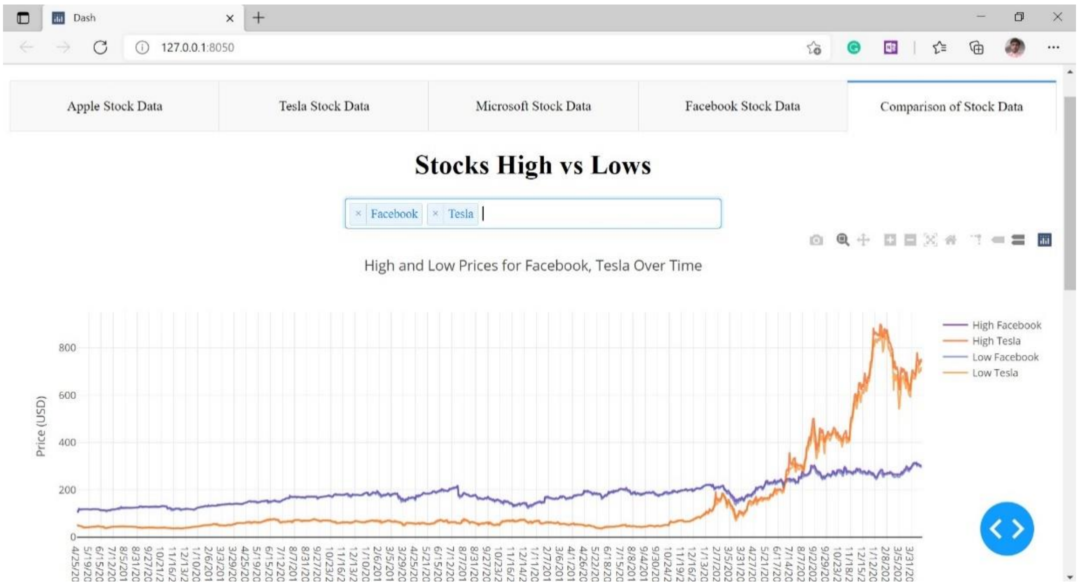


Figure 16: Actual vs LSTM predicted stock data of Facebook

5. <u>Comparison of stocks with each other:</u>



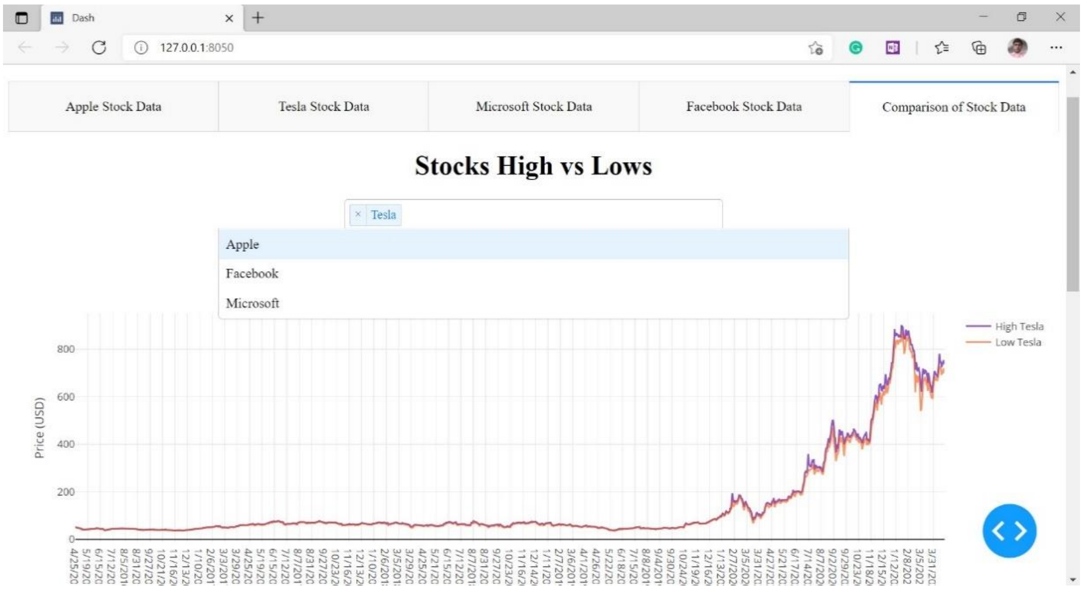


Figure 17: Stocks Comparision

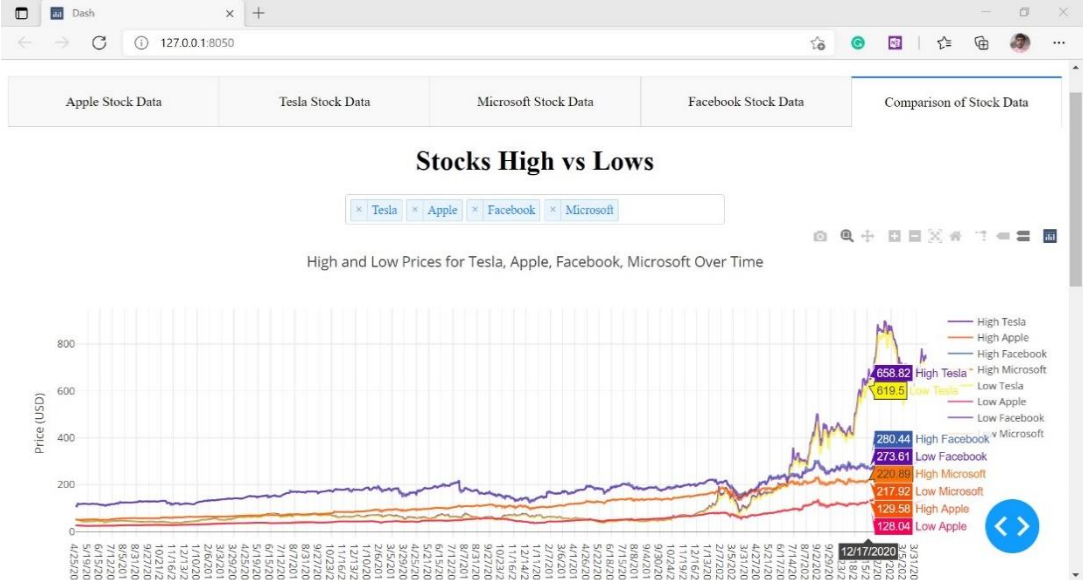


Figure 18: Stocks’ High vs Low

9. CONCLUSION AND FUTURE WORKS

9.1 CONCLUSION

We implement the application of Artificial Neural Network to the task of stock market prediction and ANN model and salient feature. Our initial analysis show significant correlation between different input parameter. The result obtained in both the cases was fairly accurate. As is indent from fig 8.2.1, 8.2.2 the prediction is fairly accurate unless there is huge and sudden variation in the actual data. On other hand, this also proves the hypotheses that stock market are actually unpredictable. After the phase of prediction and analysis, the result will be displayed to users in the form web pages

9.2 FUTURE WORKS

Future scope of improvement

- Potential improvement can be made to our data collection and analysis method.
- Future research can be done with possible improvement such as more refined data and more accurate algorithm.
- Implementation of discussion forums and economic news portal including other sector apart from hydropower and going in national level.

10. REFERENCES

- https://www.researchgate.net/publication/331345883_Stock_Market_Prediction_Using_Machine_Learning

- https://www.researchgate.net/publication/340874376_Stock_market_analysis_using_candlestick_regression_and_market_trend_prediction_CKRM

- M. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques," 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2016, pp. 322-327.

- K. A. Althelaya, E. M. El-Alfy and S. Mohammed, "Evaluation of bidirectional LSTM for short-and long-term stock market prediction," 2018 9th International Conference on Information and Communication Systems (ICICS), 2018, pp. 151-156, doi: 10.1109/IACS.2018.8355458

- Forecasting directional movements of stock prices for intraday trading using LSTM and random forests by Pushpendu Ghosh , Ariel Neufeld , Jajati Keshari Sahoo

- S. Liu, G. Liao and Y. Ding, "Stock transaction prediction modeling and analysis based on LSTM," 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2018, pp. 2787-2790, doi: 10.1109/ICIEA.2018.8398183.

- Alex Sherstinsky, Fundamentals of Recurrent Neural Network (RNN) and Long Short-TermMemory (LSTM) network, https://doi.org/10.1016/j.physd.2019.132306.

- C. R. Madhuri, M. Chinta and V.V.N.V.P. Kumar, "Stock Market Prediction for Time-series Forecasting using Prophet upon ARIMA," 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020, pp. 1-5, doi: 10.1109/ICSSS49621.2020.9202042.C. R. Madhuri, M. Chinta and V.V.N.V.P. Kumar, "Stock Market Prediction for Time-series Forecasting using Prophet upon ARIMA,"