A PROJECT REPORT

On

“STOCK PRICE PREDICTION”

Submitted in partial fulfillment of the requirements to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA

For the award of the degree

BACHELOR of TECHNOLOGY

In

COMPUTER SCIENCE &ENGINEERING

Submitted by

CH.RAMESH BABU	(18JK1A0507)
L.ROHITH	(18JK1A0541)
M.V.S.S.AYYAPPA REDDY	(18JK1A0543)

Under the guidance of

P.Vamsi Lala B.Tech,M.Tech,P.Hd.

Assistant Professor Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GUNTUR ENGINEERING COLLEGE :: YANAMADALA

(Affiliated to JNTUniversity :: KAKINADA)

2018-2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GUNTUR ENGINEERING COLLEGE :: YANAMADALA

(Affiliated to JNTUniversity :: KAKINADA)

2018-2022

CERTIFICATE

This is to certify that this project work titled “**STOCK PRICE PREDICTION**” is the bonafide work of **CH.RAMESH BABU (18JK1A0507)**,**Mr. L.ROHITH (18JK1A0541), Mr. M.V.S.S.AYYAPPA REDDY**
(18JK1A0543) who carried out the work under supervision and submitted in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in Computer Science & Engineering from **JNTUK- Kakinada** during the academic year 2021-2022.

HEAD OFTHEDEPARTMENT PROJECT GUIDE

EXTERNAL EXAMINER

DECLARATION

We hereby inform that this main project entitled "**STOCK PRICE PREDICTION**" has been Carried out and submitted in partial fulfillment for the award to the degree of **Bachelor of Technology in Computer Science and Engineering** to **Jawaharlal Nehru Technological University Kakinada** under the guidance of **P.VAMSI LALA B.Tech , M.Tech , P.Hd, Assist. Professor**, Dept. of Computer Science and Engineering. The work embodied in this project work is original and has not been submitted in part or full for any degree of this or any degree of any otheruniversity.

CH.RAMESH BABU	(18JK1A0507)
L.ROHITH	(18JK1A0541)
M.V.S.S.AYYAPPA REDDY	(18JK1A0543)

ACKNOWLEDGEMENT

We would like to express our profound gratitude towards Mr.P.VAMSI LALA **B.Tech, M.Tech, P.Hd**. **Assistant Professor,** Department of COMPUTER SCIENCE AND ENGINEERING**,** who played a supervisory role to utmost perfection, enabled us to seek through our B.Tech main project and for guiding as an internal guide methodically andmeticulously.

We are highly indebted to **Dr. A Balaji M. Tech, Ph.D.**, **Head of the Department**, Computer Science and Engineering for providing us all the necessary support.

We are very much thankful to the **college management for** their continuous support and facilities provided.

We render our deep sense of gratitude to **Dr. M.RAVINDRA KRISHNA, Principal,** for permitting us to carry out our main project works. We would like to express our sincere thanks to Computer Science and Engineering staff for lending us their time to help us and complete the work successfully.

We would also like to thank our staff, parents and friends for their enduring encouragement and assistance whenever required.

CH.RAMESH BABU	(18JK1A0507)
L.ROHITH	(18JK1A0541)
M.V.S.S.AYYAPPA REDDY	(18JK1A0543)

CONTENTS

GUNTUR ENGINEERING COLLEGE
(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada)
NH-05, Yanamadala, GUNTUR-522019, AP, INDIA.
https://gecg.in/ https://gecg.in/cse.php

Institute Vision and Institute Mission

Institute Vision:

To be a recognized premier institution in engineering education, research, and application of knowledge to benefit society.

Institute Mission:

- Impact technical knowledge at all levels to transform the engineers capable to address the challenges.
- Inculcate competencies for all-round development to meet industrial, technical, and leadership skills.
- Provide learner-centric ambience with collaborations
- Promote creativity, knowledge development, ethical and values and inter-personal skills

A. Balaji
Signature of the HOD

GUNTUR ENGINEERING COLLEGE

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada)
NH-05, Yanamadala, GUNTUR-522019, AP, INDIA.
https://gecg.in/ https://gecg.in/cse.php

Department of Computer Science and Engineering

Vision

- To be a centre of excellence in computer engineering education, empowering Graduates as highly skilled professionals.

Mission

- DM1: Provide a learning Platform with emphasis on technical excellence.
- DM2: Organize trainings to meet industrial and emerging technologies.
- DM3: Install the drive for higher learning and research
- DM4: Inculcate leadership and entrepreneur skills with industry collaborations

A. Balaji
Signature of the HOD

Program Specific Outcomes (PSOs)

PSOI: Analyse: Identify the data; Design suitable algorithm by using Latest software for real Time applications.

PSO2: Computing paradigms: Understanding the evolutionary changes in computing process knowledge of context aware applicability of paradigms and meet the challenges of the future.

A. Balaji
Signature of the HOD

GUNTUR ENGINEERING COLLEGE
(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada)
NH-05, Yanamadala, GUNTUR-522019, AP, INDIA.
https://gecg.in/ https://gecg.in/cse.php

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEOI: Analyse, design, develop and test the software products and provide creative, Sustainable solution.

PEO2: Use modern tools, techniques to analyse and provide solutions to the industrial problems.

PEO3: Demonstrate the managerial and leadership skills to become successful professionals.

A.Balaji
Signature of the HOD

PO-1 Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-2 Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3 Design/development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

PO-4 Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-5 Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6 The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-7 Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9 Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10 Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11 Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12 Life-Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

A.Balaji

Signature of the HOD

GUNTUR ENGINEERING COLLEGE
(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada)
NH-05, Yanamadala, GUNTUR-522019, AP, INDIA.
https://gecg.in/ https://gecg.in/cse.php

Project Course Outcomes

CO425.1: Review the literature and develop solutions for problem statement

CO425.2: Implement hardware and/or software requirements for identified problems

CO425.3: Design the code for the problem statement

CO425.4: Test and analyze the modules of planned project.

CO2425.5: Write technical report and deliver presentation.

CO425.6: Apply engineering and management principles to achieve project goal.

Course Outcomes – Program Outcome correlation

	PO 1	PO2	PO3	PO4	PO5	PO 6	PO 7	PO8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
C425.1	3	2		2		2	2						3		
C425.2	3	2		2		2			1	2	1		2	2	3
C425.3		3			2							2	1		3
C425.4	2	2	3	2	2		1	1	2	2			3		2
C425.5			3		2		2		1		2	2	3	2	
C425.6						3				2		2		3	

3:High **2:Medium** **1: Low**

CO-PO Mapping with Reasons:

1. **CO425.1** is mapped with PO1, PO2 and PO4, PO6, PO7 as basic knowledge of Engineering and problem Analysis activities are highly essential to conduct examinations on existing systems which have been using in industries as a part of and to define the problem of proposedsystem.
2. **CO425.2** is mapped with PO1, PO2, PO4 and PO6, PO9, PO10, PO11 as for identification, gathering analysis and classification of requirements for the proposed system, basic knowledge of engineering and Analysis steps along with complex problem analysis through the efforts of team work in order to meet the specific needs of thecustomer.
3. **CO425.3** is mapped with PO2, PO5 and PO12 as to conduct the literature review and to examine the relevant systems to understand and identify the merits and demeritsof each too enhance and develop the proposed as per theneed.
4. **CO425.4** is mapped with PO1, PO2, PO3, PO4, PO5 and PO7, PO8, PO9, PO10 because modularization and design of the project is needed after requirements elicitation. For modularization and design of the project, Basic knowledge of Engineering, Analysis capabilities, Design skills and communication is needed between team members as different modules are designed individually beforeintegration.
5. **CO425.5** is mapped with PO3, PO5, PO7, PO9, PO11 and PO12 as to construct the project latest technologies are needed. The development of project is done individually and in groups with well-defined communication by using the engineering and management principles.
6. **CO425.6** is mapped with PO6, PO10 and PO12 because during and after completion of the project, documentation is needed along with proper methods of presentation through understanding and application of engineering and management principles, which in turn needs well defined communication between the team members with all the ethical values. Even the project development team defines the future enhancements as a part of the project development after identifying the scope of theproject.

CO-PSOs Mapping with Reasons:

1. **CO425.1** is mapped with **PSO1** as examining of existing systems and identification of the problem is a part of Application Development activity and identification of evolutionary changes in latesttechnologies.

2. **CO425.2** is mapped with **PSO1, PSO2** and **PSO3**as identifying and classifying the requirements is a part of Application development and evolutionary computing changes and also follows ethicalprinciples.

3. **CO425.3** is mapped with **PSO1, PSO3** as review of literature is a part of application development activity by recognizing the computing technologies and their evolutionary changes.

4. **CO425.4** is mapped with **PSO1, PSO3** because modularization and logical design is also a part of Application development and follows computing changes using Deep learning technology.

5. **CO425.5** is mapped with **PSO1, PSO2** as Testing, Development and Integration of project activities are part of Application development and follows ethicalprinciples.

6. **CO425.6** is mapped with **PSO2** as for project documentation and presentation; the project team members apply the professional and leadership qualities.

ABSTRACT

In this project we attempt to implement machine learning approach to predict stock prices. Machine learning is effectively implemented in forecasting stock prices. The objective is to predict the stock prices in order to make more informed and accurate investment decisions. We propose a stock price prediction system that integrates mathematical functions, machine learning, and other external factors for the purpose of achieving better stock prediction accuracy and issuing profitable trades.

There are two types of stocks. You may know of intraday trading by the commonly used term "day trading." Interday traders hold securities positions from at least one day to the next and often for several days to weeks or months. LSTMs are very powerful in sequence prediction problems because they’re able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price. While predicting the actual price of a stock is an uphill climb, we can build a model that will predict whether the price will go up or down.

Keywords: LSTM, CNN, ML, DL, Trade Open, Trade Close, Trade Low, Trade High

LIST OF SYMBOLS

Xt	Input at current state
X(t-1)	Input at Previous state
Ct	Current Cell State
C(t-1)	Previous Cell State
ht	Current hidden/output State
h(t-1)	Previous hidden/output State
σ	Sigmoid Function
tanh	Hyperbolic tangent function

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

LSTM	Long Short-Term Memory
ATS	Automated Trading System
GRU	Gated Recurrent Unit
ML	Machine Learning
SVM	Support Vector Machine
EMH	Efficient Market hypothesis
AI	Artificial Intelligence
NN	Neural Networks
ARMA	Autoregressive Moving Average
DRL	Deep Reinforcement Learning
LMS	Least Mean Square
UML	Unified modelling Language
MSE	Mean Squared Error
RMSE	Root Mean Squared Error

CHAPTER 1
INTRODUCTION

The financial market is a dynamic and composite system where people can buy and sell currencies, stocks, equities and derivatives over virtual platforms supported by brokers. The stock market allows investors to own shares of public companies through trading either by exchange or over the counter markets. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening new business or the need of high salary career. Stock markets are affected by many factors causing the uncertainty and high volatility in the market. Although humans can take orders and submit them to the market, automated trading systems (ATS) that are operated by the implementation of computer programs can perform better and with higher momentum in submitting orders than any human. However, to evaluate and control the performance of ATSs, the implementation of risk strategies and safety measures applied based on human judgements are required. Many factors are incorporated and considered when developing an ATS, for instance, trading strategy to be adopted, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of the future stock value, and specific news related to the stock being analysed.

Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period of time to predict the result in the next time unit. Many time-series prediction algorithms have shown their effectiveness in practice. The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type - Long-short Term Memory (LSTM) and Gated Recurrent Unit (GRU). Stock market is a typical area that presents time-series data and many researchers study on it and proposed various models. In this project, LSTM model is used to predict the stock price.

From gradually the very past years some forecasting models are developed for this kind of purpose and they had been applied to money market prediction. Generally, this classification is done by:

1. Time series analysis
2. Fundamental analysis
3. Technical analysis

Time Series Analysis

The definition of forecasting can be like this the valuation of some upcoming result or results by analysing the past data. It extents different areas like industry and business, economics and finance, environmental science. Forecasting problems can be classified as follows:

- Long term forecasting (estimation beyond 2 years)
- Medium-term forecasting (estimation for 1 to 2 years)
- Short term forecasting (estimation for weeks or months, days, minutes, few seconds)

The analysis [1] of time consist of several forecasting problems. The designation of a time series is a linear classification of observations for a selected variable. The variable of the stockprice in our case. Which can weather multivariate or univariate? Only particular stock isincluded in the univariate data while more than one company for various instances of time isadded in multivariate. For investigating trends, patterns and cycle or periods the analysis oftime series advantages in the present data. In spending money wisely an early data of the bullishor bearish in the case of the stock market. Also, for categorizing the best-performing companiesthe analysis of patterns plays its role for a specific period. This makes forecasting as well astime series analysis an important research area.

Fundamental analysis

Fundamental Analysts are concerned with the business that reasons the stock itself. They assessa company's historical performance as well as the reliability of its accounts. Different performance shares are created that aid the fundamental forecaster with calculating the validity of a stock, such as the P/E ratio. Warren Buffett is probably the foremost renowned of all Fundamental Analysts.

What fundamental analysis within the stock market is making an attempt to reach, is organizingthe true value of a stock, that then will be matched with the worth it is being listed on stock markets and so finding out whether or not the stock on the market is undervalued or not. Find out the correct value will be completed by numerous strategies with primarily a similar principle. The principle is that an organization is price all of its future profits. Those future profits has to be discounted to their current value. This principle goes on the theory that a business is all about profits and nothing else. Differing to technical analysis, the fundamental analysis is assumed as further as a long approach.
Fundamental analysis is created on conviction that hominoid society desires capital to make progress and if the company works well, than it should be rewarded with an additional capital and outcome in a surge in stock price. Fundamental analysis is usually used by the fund managers as it is the maximum sensible, objective and prepared from openly existing data likefinancial statement analysis.

One more meaning of fundamental analysis is on the far side bottom-up business analysis, it discusses the top-down analysis since initial analysing the world economy, followed by country analysis and also sector analysis, and last the company level analysis.

Technical analysis

Chartists or the technical analysts are not involved with any other of the fundamentals of the company. The long run price of a stock based generally exclusively on the trends of the past value (a form of time series analysis) that is set by them. The head and shoulders or cup and saucer are various numerous patterns that are employed. Also the techniques, patterns are used just like the oscillators, exponential moving average (EMA), support and momentum and volume indicators. Candlestick patterns, believed to have been initial developed by Japanese rice merchants, are nowadays widely used by technical analysts. For the short-term approaches, the technical analysis is used compare to long-run ones. So, in commodities and forex markets it is more predominant wherever traders target short-term price movements. There are basic rules are used in this analysis, first all significant about a company is already priced into the stock, another being that the value changes in trends and finally that history (of prices) tendsto repeat itself that is especially due to the market science.

1.1 MOTIVATION FOR WORK

Businesses primarily run over customer's satisfaction, customer reviews about their products. Shifts in sentiment on social media have been shown to correlate with shifts in stock markets. Identifying customer grievances thereby resolving them leads to customer satisfaction as well as trustworthiness of an organization. Hence there is a necessity of an un biased automated system to classify customer reviews regarding any problem. In today's environment where we're justifiably suffering from data overload (although this does not mean better or deeper insights), companies might have mountains of customer feedback collected; but for mere humans, it's still impossible to analyse it manually without any sort of error or bias. Oftentimes, companies with the best intentions find themselves in an insights vacuum. You know you need insights to inform your decision making and you know that you're lacking them, but don't know how best to get them. Sentiment analysis provides some answers into what the most important issues are, from the perspective of customers, at least. Because sentiment analysis can be automated, decisions can be made based on a significant amount of data rather than plain intuition.

1.2 PROBLEM STATEMENT

Time Series forecasting & modelling plays an important role in data analysis. Time series analysis is a specialized branch of statistics used extensively in fields such as Econometrics & Operation Research. Time Series is being widely used in analytics & data science. Stock prices are volatile in nature and price depends on various factors. The main aim of this project is to predict stock prices using Long short term memory (LSTM).

1.3 Objectives

A stock market prediction is described as an action of attempting to classify the future value of the company stock or other financial investment traded on the stock exchange. The forthcomingprice of a stock of the successful estimation is called the Yield significant profit. This helps you to invest wisely for making good profits.

1.4 Applications

- Business
- Companies
- Insurance company
- Government Agency
- This application is helpful for stock investors, sellers, buyers, brokers.

1.5 Organization of Report

Chapter 2 contains a literature survey that provides a summary of individual paper.

Chapter 3 provides an overview of existing work for stock price prediction that has been doneusing LSTM, CNN and Hybrid Approach of LSTM+CNN.

Chapter 4 presents Implementation and its results, tools and technology used to achieve this and dataset detail.

Chapter 5 contains a conclusion about stock price prediction and future work about what you are wanted to do in future.

CHAPTER 2
LITERATURE SURVEY

2.1 INTRODUCTION

"What other people think" has always been an important piece of information for most of us during the decision-making process. The Internet and the Web have now (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet. The interest that individual users show in online opinions about products and services, and the potential influence such opinions wield, is something that is driving force for this area of interest. And there are many challenges involved in this process which needs to be walked all over in order to attain proper outcomes out of them. In this survey we analysed basic methodology that usually happens in this process and measures that are to be taken to overcome the challenges being faced.

2.2 EXISTING METHODS

2.2.1 Stock Market Prediction Using Machine Learning

The research work done by V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India. In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python. In this paper we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute frequencies.

2.2.2 Forecasting the Stock Market Index Using Artificial Intelligence Techniques

The research work done by Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering. The weak form of Efficient Market hypothesis (EMH) states that it is impossible to forecast the future price of an asset based on the information contained in the historical prices of an asset. This means that the market behaves as a random walk and as a result makes forecasting impossible. Furthermore, financial forecasting is a difficult task due to the intrinsic complexity of the financial system. The objective of this work was to use artificial intelligence (AI) techniques to model and predict the future price of a stock market index. Three artificial intelligence techniques, namely, neural networks (NN), support vector machines and neuro-fuzzy systems are implemented in forecasting the future price of a stock market index based on its historical price information. Artificial intelligence techniques have the ability to take into consideration financial system complexities and they are used as financial time series forecasting tools.

Two techniques are used to benchmark the AI techniques, namely, Autoregressive Moving Average (ARMA) which is linear modelling technique and random walk (RW) technique. The experimentation was performed on data obtained from the Johannesburg Stock Exchange. The data used was a series of past closing prices of the All Share Index. The results showed that the three techniques have the ability to predict the future price of the Index with an acceptable accuracy. All three artificial intelligence techniques outperformed the linear model. However, the random walk method out performed all the other techniques. These techniques show an ability to predict the future price however, because of the transaction costs of trading in the market, it is not possible to show that the three techniques can disprove the weak form of market efficiency. The results show that the ranking of performances support vector machines, neuro-fuzzy systems, multilayer perceptron neural networks is dependent on the accuracy measure used.

2.2.3 Indian stock market prediction using artificial neural networks ontick data

The research work done by Dharmaraja Selvamuthu, Vineet Kumar and Abhishek Mishra Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India. A stock market is a platform for trading of a company's stocks and derivatives at an agreed price. Supply and demand of shares drive the stock market. In any country stock market is one of the most emerging sectors. Nowadays, many people are indirectly or directly related to this sector. Therefore, it becomes essential to know about market trends. Thus, with the development of the stock market, people are interested in forecasting stock price. But, due to dynamic nature and liable to quick changes in stock price, prediction of the stock price becomes a challenging task. Stock m Prior work has proposed effective methods to learn event representations that can capture syntactic and semantic information over text corpus, demonstrating their effectiveness for downstream tasks such as script event prediction. On the other hand, events extracted from raw texts lacks of common-sense knowledge, such as the intents and emotions of the event participants, which are useful for distinguishing event pairs when there are only subtle differences in their surface realizations. To address this issue, this paper proposes to leverage external common-sense knowledge about the intent and sentiment of the event.

Experiments on three event-related tasks, i.e., event similarity, script event prediction and stock market prediction, show that our model obtains much better event embeddings for the tasks, achieving 78% improvements on hard similarity task, yielding more precise inferences on subsequent events under given contexts, and better accuracies in predicting the volatilities of the stock market1. Markets are mostly a non-parametric, non-linear, noisy and deterministic chaotic system (Ahangar et al. 2010). As the technology is increasing, stock traders are moving towards to use Intelligent Trading Systems rather than fundamental analysis for predicting prices of stocks, which helps them to take immediate investment decisions. One of the main aims of a trader is to predict the stock price such that he can sell it before its value decline, or buy the stock before the price rises. The efficient market hypothesis states that it is not possible to predict stock prices and that stock behaves in the random walk. It seems to be very difficult to replace the professionalism of an experienced trader for predicting the stock price. But because of the availability of a remarkable amount of data and technological advancements we can now formulate an appropriate algorithm for prediction whose results can increase the profits for traders or investment firms. Thus, the accuracy of an algorithm is directly proportional to gains made by using the algorithm.

2.2.4 The Stock Market and Investment

The research work done by Manh Ha Duong Boriss Siliverstovs. Investigating the relation between equity prices and aggregate investment in major European countries including France, Germany, Italy, the Netherlands and the United Kingdom. Increasing integration of European financial markets is likely to result in even stronger correlation between equity prices in different European countries. This process can also lead to convergence in economic development across European countries if developments in stock markets influence real economic components, such as investment and consumption. Indeed, our vector autoregressive models suggest that the positive correlation between changes equity prices and investment is, in general, significant. Hence, monetary authorities should monitor reactions of share prices to monetary policy and their effects on the business cycle.

2.2.5 Automated Stock Price Prediction Using Machine Learning

The research work done by Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut. Traditionally and in order to predict market movement, investors used to analyse the stock prices and stock indicators in addition to the news related to these stocks. Hence, the importance of news on the stock price movement. Most of the previous work in this industry focused on either classifying the released market news as (positive, negative, neutral) and demonstrating their effect on the stock price or focused on the historical price movement and predicted their future movement. In this work, we propose an automated trading system that integrates mathematical functions, machine learning, and other external factors such as news' sentiments for the purpose of achieving better stock prediction accuracy and issuing profitable trades. Particularly, we aim to determine the price or the trend of a certain stock for the coming end-of-day considering the first several trading hours of the day. To achieve this goal, we trained traditional machine learning algorithms and created/trained multiple deep learning models taking into consideration the importance of the relevant news. Various experiments were conducted, the highest accuracy (82.91%) of which was achieved using SVM for Apple Inc. (AAPL) stock.

2.2.6 Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model

The research work done by Hyeong Kyu Choi, B.A Student Dept. of Business Administration Korea University Seoul, Korea. Predicting the price correlation of two assets for future time periods is important in portfolio optimization. We apply LSTM recurrent neural networks (RNN) in predicting the stock price correlation coefficient of two individual stocks. RNN's are competent in understanding temporal dependencies. The use of LSTM cells further enhances its long-term predictive properties. To encompass both linearity and nonlinearity in the model, we adopt the ARIMA model as well. The ARIMA model filters linear tendencies in the data and passes on the residual value to the LSTM model. The ARIMA-LSTM hybrid model is tested against other traditional predictive financial models such as the full historical model, constant correlation model, single-index model and the multi-group model. In our empirical study, the predictive ability of the ARIMA-LSTM model turned out superior to all other financial models by a significant scale. Our work implies that it is worth considering the ARIMALSTM model to forecast correlation coefficient for portfolio optimization.

2.2.7 Event Representation Learning Enhanced with External Common-sense Knowledge

The research work done by Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, Junwen Duan Research Center for Social Computing and Information Retrieval Harbin Institute of Technology, China. Prior work has proposed effective methods to learn event representations that can capture syntactic and semantic information over text corpus, demonstrating their effectiveness for downstream tasks such as script event prediction. On the other hand, events extracted from raw texts lacks of common-sense knowledge, such as the intents and emotions of the event participants, which are useful for distinguishing event pairs when there are only subtle differences in their surface realizations. To address this issue, this paper proposes to leverage external common-sense knowledge about the intent and sentiment of the event. Experiments on three event-related tasks, i.e., event similarity, script event prediction and stock market prediction, show that our model obtains much better event embeddings for the tasks, achieving 78% improvements on hard similarity task, yielding more precise inferences on subsequent events under given contexts, and better accuracies in predicting the volatilities of the stock market.

2.2.8 Forecasting directional movements of stock prices for intraday trading using LSTM and random forests

The research work done by Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari SahooDepartment of Computer Science & Information Systems, BITS Pilani K.K.Birla Goa campus, India bDivision of Mathematical Sciences, Nanyang Technological University, Singapore cDepartment of Mathematics, BITS Pilani K.K.Birla Goa campus, India. We employ both random forests and LSTM networks (more precisely CuDNNLSTM) as training methodologies to analyse their effectiveness in forecasting out-of-sample directional movements of constituent stocks of the S&P 500 from January 1993 till December 2018 for intraday trading. We introduce a multi-feature setting consisting not only of the returns with respect to the closing prices, but also with respect to the opening prices and intraday returns. As trading strategy, we use Krauss et al. (2017) and Fischer & Krauss (2018) as benchmark and, on each trading day, buy the 10 stocks with the highest probability and sell short the 10 stocks with the lowest probability to outperform the market in terms of intraday returns – all with equal monetary weight. Our empirical results show that the multi-feature setting provides a daily return, prior to transaction costs, of 0.64% using LSTM networks, and 0.54% using random forests. Hence, we outperform the single-feature setting in Fischer & Krauss (2018) and Krauss et al. (2017) consisting only of the daily returns with respect to the closing prices, having corresponding daily returns of 0 .41% and of 0 .39% with respect to LSTM and random forests, respectively. 1 Keywords: Random forest, LSTM, Forecasting, Statistical Arbitrage, Machine learning, Intraday trading.

2.2.9 A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance

The research work done by Xiao-Yang Liu1 Hongyang Yang,Qian Chen4,Runjia ZhangLiuqing Yang Bowen Xiao Christina Dan Wang Electrical Engineering, 2Department of Statistics, 3Computer Science, Columbia University, 3AI4Finance LLC., USA, Ion Media Networks, USA, Department of Computing, Imperial College, 6New York University (Shanghai). As deep reinforcement learning (DRL) has been recognized as an effective approach in quantitative finance, getting hands-on experiences is attractive to beginners. However, to train a practical DRL trading agent that decides where to trade, at what price, and what quantity involves error-prone and arduous development and debugging. In this paper, we introduce a DRL library FinRL that facilitates beginners to expose themselves to quantitative finance and to develop their own stock trading strategies. Along with easily-reproducible tutorials, FinRL library allows users to streamline their own developments and to compare with existing schemes easily.

Within FinRL, virtual environments are configured with stock market datasets, trading agents are trained with neural networks, and extensive back testing is analysed via trading performance. Moreover, it incorporates important trading constraints such as transaction cost, market liquidity and the investor's degree of risk-aversion. FinRL is featured with completeness, hands-on tutorial and reproducibility that favors beginners: (i) at multiple levels of time granularity, FinRL simulates trading environments across various stock markets, including NASDAQ-100, DJIA, S&P 500, HSI, SSE 50, and CSI 300; (ii) organized in a layered architecture with modular structure, FinRL provides fine-tuned state-of-the-art DRL algorithms (DQN, DDPG, PPO, SAC, A2C, TD3, etc.), commonly used reward functions and standard evaluation baselines to alleviate the debugging workloads and promote the reproducibility, and (iii) being highly extendable, FinRL reserves a complete set of user-import interfaces. Furthermore, we incorporated three application demonstrations, namely single stock trading, multiple stock trading, and portfolio allocation. The FinRL library will be available on GitHub at link https://github.com/AI4Finance-LLC/FinRL-Library.

2.2.10 An innovative neural network approach for stock market prediction

The research work done by Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin. To develop an innovative neural network approach to achieve better stock market predictions. Data were obtained from the live stock market for real-time and off-line analysis and results of visualizations and analytics to demonstrate Internet of Multimedia of Things for stock analysis. To study the influence of market characteristics on stock prices, traditional neural network algorithms may incorrectly predict the stock market, since the initial weight of the random selection problem can be easily prone to incorrect predictions.

Based on the development of word vector in deep learning, we demonstrate the concept of "stock vector." The input is no longer a single index or single stock index, but multi-stock high-dimensional historical data. We propose the deep long short-term memory neural network (LSTM) with embedded layer and the long short-term memory neural network with automatic encoder to predict the stock market. In these two models, we use the embedded layer and the automatic encoder, respectively, to vectorize the data, in a bid to forecast the stock via long short-term memory neural network. The experimental results show that the deep LSTM with embedded layer is better. Specifically, the accuracy of two models is 57.2 and 56.9%, respectively, for the Shanghai A-shares composite index. Furthermore, they are 52.4 and 52.5%, respectively, for individual stocks. We demonstrate research contributions in IMMT for neural network-based financial analysis. 2.2.11 An Intelligent Technique for Stock Market Prediction

2.2.11 An Intelligent Technique for Stock Market Prediction

The research work done by M. Mekayel Anik · M. Shamsul Arefin (B) Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh. A stock market is a loose network of economic transactions between buyers and sellers based on stocks also known as shares. In stock markets, stocks represent the ownership claims on businesses. These may include securities listed on a stock exchange as well as those only traded privately. A stock exchange is a place where brokers can buy and/or sell stocks, bonds, and other securities. Stock market is a very vulnerable place for investment due to its volatile nature. In the near past, we faced huge financial problems due to huge drop in price of shares in stock markets worldwide. This phenomenon brought a heavy toll on the international as well as on our national financial structure. Many people lost their last savings of money on the stock market. In 2010–2011 financial year, Bangladeshi stock market faced massive collapse [1]. This phenomenon can be brought under control especially by strict monitoring and instance stock market analysis. If we can analyse stock market correctly in time, it can become a field of large profit and may become comparatively less vulnerable for the investors.

Stock market is all about prediction and rapid decision making about investment, which cannot be done without thorough analysis of the market. If we can predict the stock market by analysing historical data properly, we can avoid the consequences of serious market collapse and to be able to take necessary steps to make market immune to such situations.

CHAPTER 3
METHODOLOGY

3.1 Overview of Existing Work

Stock Price Prediction by Machine Learning present to estimate the stock future value and machine learning technique like LSTM for existing work. This machine-learning algorithm isto perform the best predicting result of the stock future price. LSTM is capable to catching themodifications in the behaviour of the stock price for the indicated period in this proposed system.

Propose [3] a machine learning-based normalization for stock price prediction. The dataset utilized for analysis was selected from NSE-TATA Finance. It consists of approximately 9 lakh records of the required Stock price and other relevant data. The data reflected the stock price at some time intervals for every day of the year. It contains various data like date, symbol, openprice, close price, low price, high price and volume. Here, the data for only one company was considered. All the data was available in a file of CSV format which was first read and transformed into a data frame using the Pandas library in Python. The normalization of the datawas performed through the sklearn library in Python and the data were divided into training and testing sets. The experiment set was kept as 20% of the available dataset. This paper focuses on two architecture Regression-based Model and LSTM. The Regression-based Model is employed for predicting unbroken values through some given autonomous values Regressionuses a given linear function for predicting continuous values of the most important amongst them and made the predictions using these. LSTM architecture is able to identify the changes in trends which show evident from the result. LSTM is identified as the best model for the proposed methodology. This shows that the proposed system is capable of identifying some interrelation within the data. In the stock market, there may not always follow the same cycle or may not always be in a regular pattern for the changes that are occurred. The period of the existence will differ and the existence of the trend is based on the companies and the sectors. For investors, this type of analysis of trends and cycles will obtain more profit. We must use networks like LSTM as they rely on the current information to analyse various information.

3.2 PROPOSED SYSTEMS

The prediction methods can be roughly divided into two categories, statistical methods and artificial intelligence methods. Statistical methods include logistic regression model, ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, convolutional neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc. They used Long short-term memory network (LSTM).

Long short-term memory network:

Long short-term memory network (LSTM) is a particular form of recurrent neural network (RNN).

Working of LSTM:

LSTM is a special network structure with three “gate” structures. Three gates are placed in an LSTM unit, called input gate, forgetting gate and output gate. While information enters the LSTM’s network, it can be selected by rules. Only the information conforms to the algorithm will be left, and the information that does not conform will be forgotten through the forgetting gate.

The experimental data in this paper are the actual historical data downloaded from the Internet. Three data sets were used in the experiments. It is needed to find an optimization algorithm that requires less resources and has faster convergence speed.

- Used Long Short-term Memory (LSTM) with embedded layer and the LSTM neural network with automatic encoder.
- LSTM is used instead of RNN to avoid exploding and vanishing gradients.
- In this project python is used to train the model, MATLAB is used to reduce dimensions of the input. MySQL is used as a dataset to store and retrieve data.
- The historical stock data table contains the information of opening price, the highest price, lowest price, closing price, transaction date, volume and so on.
- The accuracy of this LSTM model used in this project is 57%.

LMS filter:

The LMS filter is a kind of adaptive filter that is used for solving linear problems. The idea of the filter is to minimize a system (finding the filter coefficients) by minimizing the least mean square of the error signal.

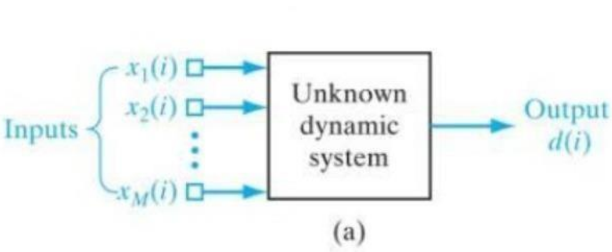


Fig. 1: LMS Inputs and Outputs

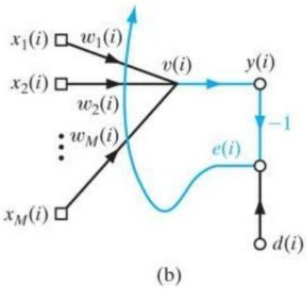
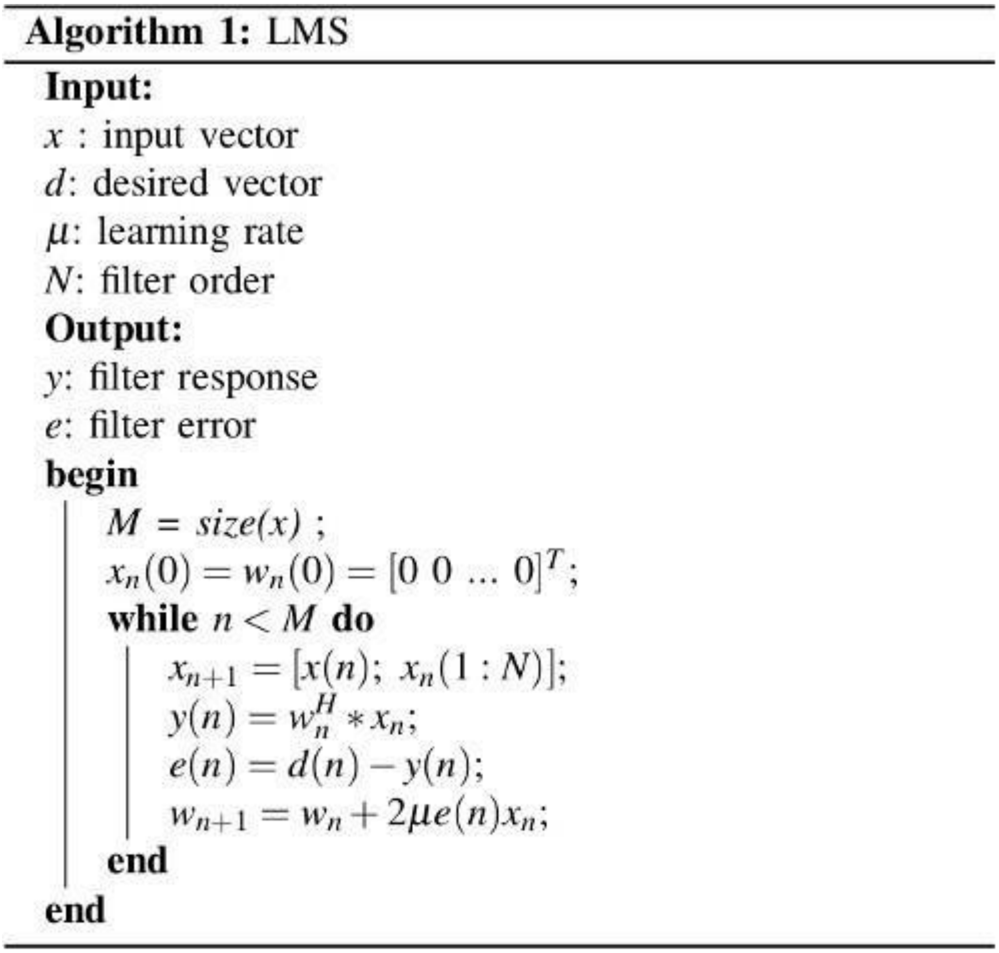


Fig 2: LMS updating weights



```
Algorithm 1: LMS
Input:
$x$ : input vector
$d$: desired vector
$\mu$: learning rate
$N$: filter order
Output:
$y$: filter response
$e$: filter error
begin
    $M = size(x)$ ;
    $x_n(0) = w_n(0) = [0\ 0\ \dots\ 0]^T$;
    while $n < M$ do
        $x_{n+1} = [x(n);\ x_n(1:N)]$;
        $y(n) = w_n^H * x_n$;
        $e(n) = d(n) - y(n)$;
        $w_{n+1} = w_n + 2\mu e(n) x_n$;
    end
end
```

In general, we don't know exactly if the problem can be solved very well with linear approach, so we usually test a **linear** and a **non-linear** algorithm. Since the internet always shows non-linear approaches, we will use LMS to prove that stock market prediction **can** be done with linear algorithms with a **good precision**.

But this filter **mimetizes** a system, that is, if we apply this filter in our data, we will have the **filter coefficients** trained, and when we input a new vector, our filter coefficients will output a response that the original system would (in the best case). So we just have to do a *tricky* modification for using this filter to predict data.

The system:

First, we will delay our input vector by *l* positions, where *l* would be the quantity of days we want to predict, this *l* new positions will be filled by **zeros**.

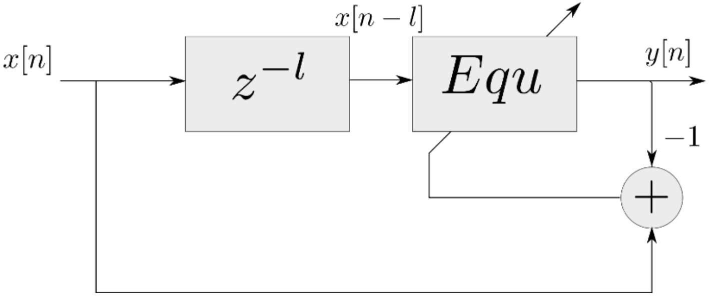
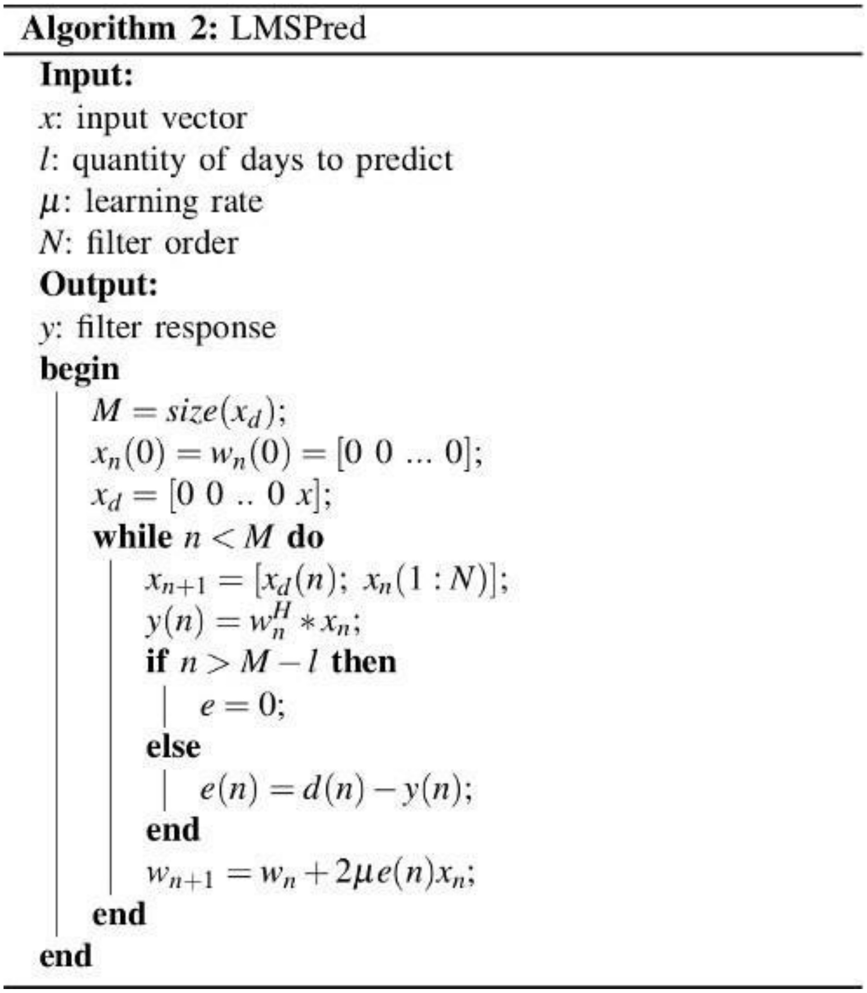


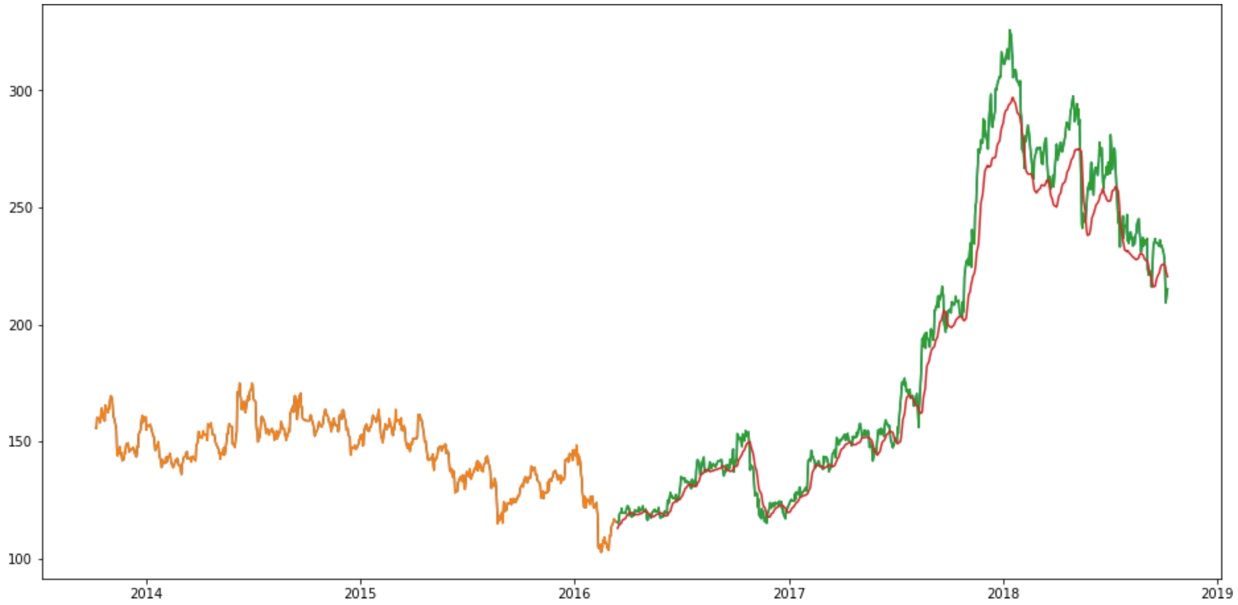
Fig. 3: LMS updating weights

When we apply the LMS filter, we will train the filter to the first 178 data. After that, we will set the error as zero, so the system will start to output the answers as the original system to the last *l* values. We will call the *tricky* modification as the **LMSPred algorithm**.



```
Algorithm 2: LMSPred
  Input:
  $x$: input vector
  $l$: quantity of days to predict
  $\mu$: learning rate
  $N$: filter order
  Output:
  $y$: filter response
  begin
      $M = size(x_d)$;
      $x_n(0) = w_n(0) = [0\ 0\ \ldots\ 0]$;
      $x_d = [0\ 0\ ..\ 0\ x]$;
      while $n < M$ do
          $x_{n+1} = [x_d(n);\ x_n(1:N)]$;
          $y(n) = w_n^H * x_n$;
          if $n > M - l$ then
              $e = 0$;
          else
              $e(n) = d(n) - y(n)$;
          end
          $w_{n+1} = w_n + 2\mu e(n) x_n$;
      end
  end
```

Results



One example of stock market prediction result

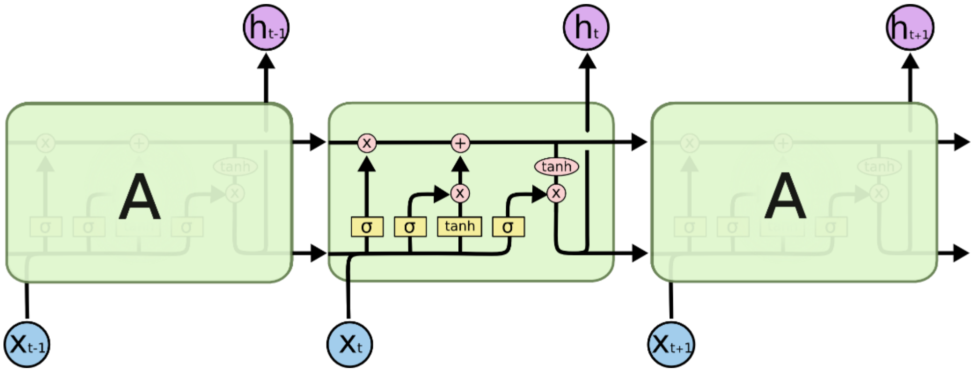


Fig. 4: LSTM Architecture

Forget Gate:

A forget gate is responsible for removing information from the cell state.

- The information that is no longer required for the LSTM to understand things or the information that is of less importance is removed via multiplication of a filter.
- This is required for optimizing the performance of the LSTM network.
- This gate takes in two inputs; h_t-1 and x_t. h_t-1 is the hidden state from the previous cell or the output of the previous cell and x_t is the input at that particular time step.

Input Gate:

1. Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from hi-1 and x_t.
2. Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
3. Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation.

The functioning of an output gate can again be broken down to three steps:

- Creating a vector after applying tanh function to the cell state, thereby scaling the values to the range -1 to +1.
- Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to be output from the vector created above. This filter again employs a sigmoid function.
- Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as a output and also to the hidden state of the next cell.

LSTM Model Building

Input : DataSet

```
df=pd.read_csv(io.BytesIO(uploaded['NSE-TATA.csv']))
df.head()

df["Date"]=pd.to_datetime(df.Date,format="%Y-%m-%d")
df.index=df['Date']

plt.figure(figsize=(16,8))
plt.plot(df["Close"],label='Close Price history')

from keras.models import Sequential
from keras.layers import LSTM,Dropout,Dense

data=df.sort_index(ascending=True,axis=0)
new_dataset=pd.DataFrame(index=range(0,len(df)),columns=['Date','Close'])

for i in range(0,len(data)):
new_dataset["Date"][i]=data['Date'][i]
new_dataset["Close"][i]=data["Close"][i]


new_dataset.index=new_dataset.Date
new_dataset.drop("Date",axis=1,inplace=True)

final_dataset=new_dataset.values

train_data=final_dataset[0:600,:]
valid_data=final_dataset[600:,:]

scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(final_dataset)
```

```
x_train_data,y_train_data=[],[]

for i in range(60,len(train_data)):
x_train_data.append(scaled_data[i-60:i,0])
y_train_data.append(scaled_data[i,0])

x_train_data,y_train_data=np.array(x_train_data),np.array(y_train_data)

x_train_data=np.reshape(x_train_data,(x_train_data.shape[0],x_train_data.shape[1],1))

lstm_model=Sequential()
lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train_data.shape[1],1)))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(1))

lstm_model.compile(loss='mean_squared_error',optimizer='adam')
lstm_model.fit(x_train_data,y_train_data,epochs=1,batch_size=1,verbose=2)

inputs_data=new_dataset[len(new_dataset)-len(valid_data)-60:].values
inputs_data=inputs_data.reshape(-1,1)
inputs_data=scaler.transform(inputs_data)


X_test=[]
for i in range(60,inputs_data.shape[0]):
X_test.append(inputs_data[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))
closing_price=lstm_model.predict(X_test)
closing_price=scaler.inverse_transform(closing_price)

lstm_model.save("saved_lstm_model.h5")
```

Hardware Requirements:

- RAM: 4 GB
- Storage: 500 GB
- CPU: 2 GHz or faster
- Architecture: 32-bit or 64-bit

Software Requirements:

- Python 3.5 in Google Colab is used for data pre-processing, model training and prediction.
- Operating System: windows 7 and above or Linux based OS or MAC OS

Functional requirements

Functional requirements describe what the software should do (the functions). Think about the core operations.

Because the "functions" are established before development, functional requirements should be written in the future tense. In developing the software for Stock Price Prediction, some of the functional requirements could include:

- The software shall accept the tw_spydata_raw.csv dataset as input.
- The software should shall do pre-processing (like verifying for missing data values) on input for model training.
- The software shall use LSTM ARCHITECTURE as main component of the software.
- It processes the given input data by producing the most possible outcomes of a CLOSING STOCK PRICE

Notice that each requirement is directly related to what we expect the software to do. They represent some of the core functions.

Non-Functional requirements

Product properties

- Usability: It defines the user interface of the software in terms of simplicity of understanding the user interface of stock prediction software, for any kind of stock trader and other stakeholders in stock market.
- Efficiency: maintaining the possible highest accuracy in the closing stock prices in shortest time with available data.

Performance: It is a quality attribute of the stock prediction software that describes the responsiveness to various user interactions with it.

3.2.1 SYSTEM ARCHITECTURE

1) Preprocessing of data

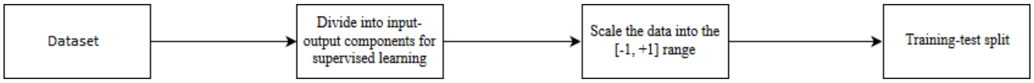

Dataset
Divide into input-output components for supervised learning
Scale the data into the [-1, +1] range
Training-test split

Fig. 5: Pre-processing of data

2) Overall Architecture

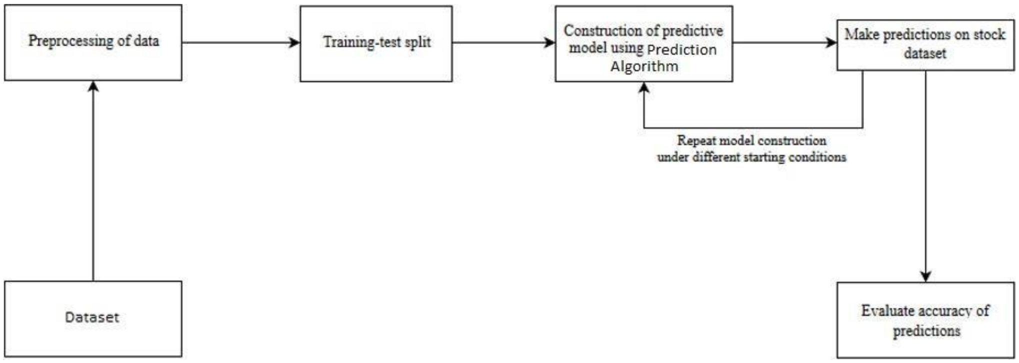

Preprocessing of data
Training-test split
Construction of predictive model using Prediction Algorithm
Make predictions on stock dataset
Repeat model construction under different starting conditions
Dataset
Evaluate accuracy of predictions

Fig. 6: Overall Architecture

CHAPTER 4
DESIGN

4.1 Structure Chart

A structure chart (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name.

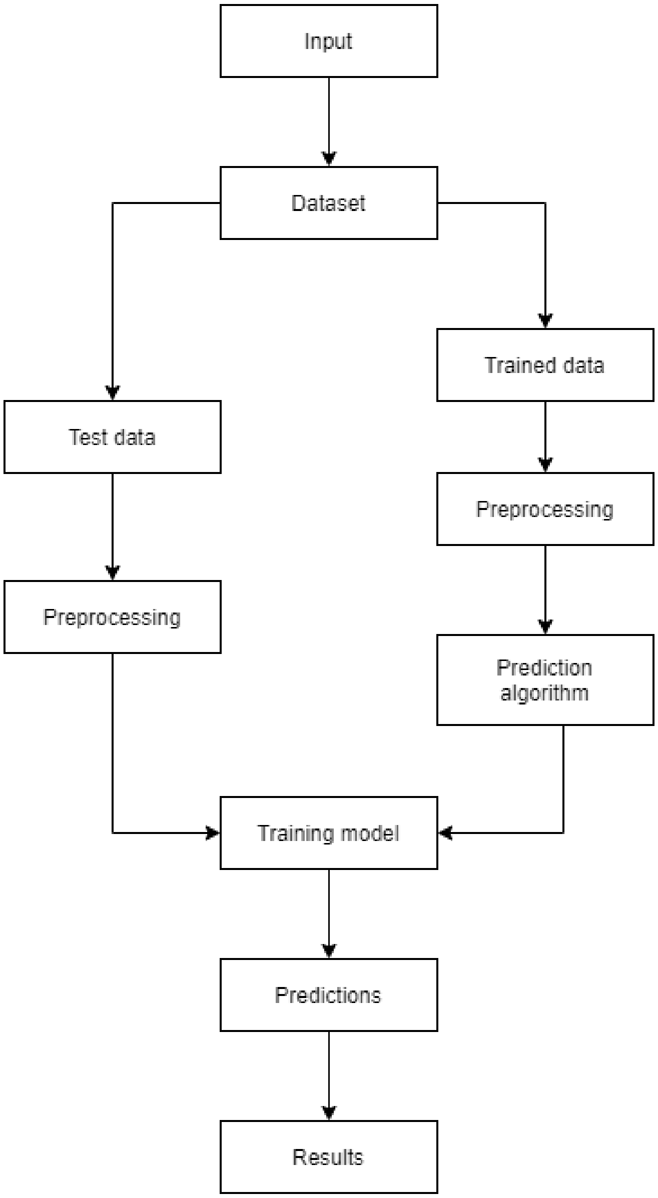


Fig. 7: Training and prediction

4.2 UML Diagrams

A UML diagram is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence. UML diagram contains graphical elements (symbols) - UML nodes connected with edges (also known as paths or flows) - that represent elements in the UML model of the designed system. The UML model of the system might also contain other documentation such as use cases written as templated texts.

The kind of the diagram is defined by the primary graphical symbols shown on the diagram. For example, a diagram where the primary symbols in the contents area are classes is class diagram. A diagram which shows use cases and actors is use case diagram. A sequence diagram shows sequence of message exchanges between lifelines.

UML specification does not preclude mixing of different kinds of diagrams, e.g. to combine structural and behavioral elements to show a state machine nested inside a use case. Consequently, the boundaries between the various kinds of diagrams are not strictly enforced. At the same time, some UML Tools do restrict set of available graphical elements which could be used when working on specific type of diagram.

UML specification defines two major kinds of UML diagram: structure diagrams and behavior diagrams.

Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

4.2.1 Use Case Diagram

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems.
- Goals that your system or application helps those entities (known as actors) achieve.
- The scope of your system.

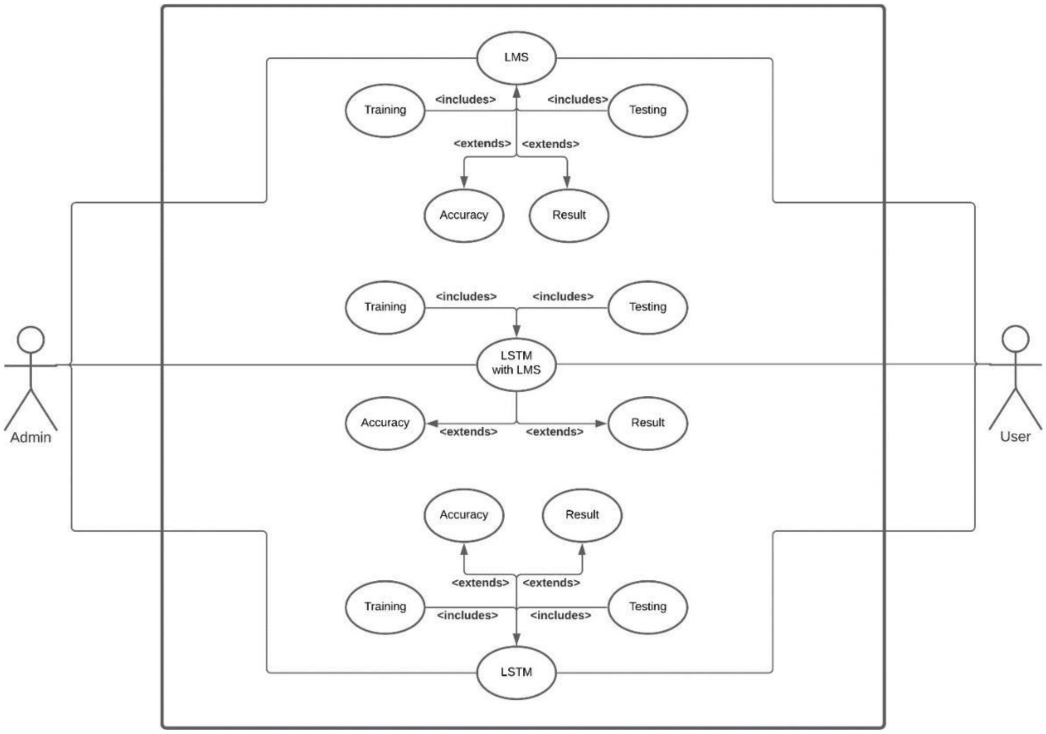


Fig. 8: Using LMS, LSTM and LSTM with LMS in the system

4.2.2 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

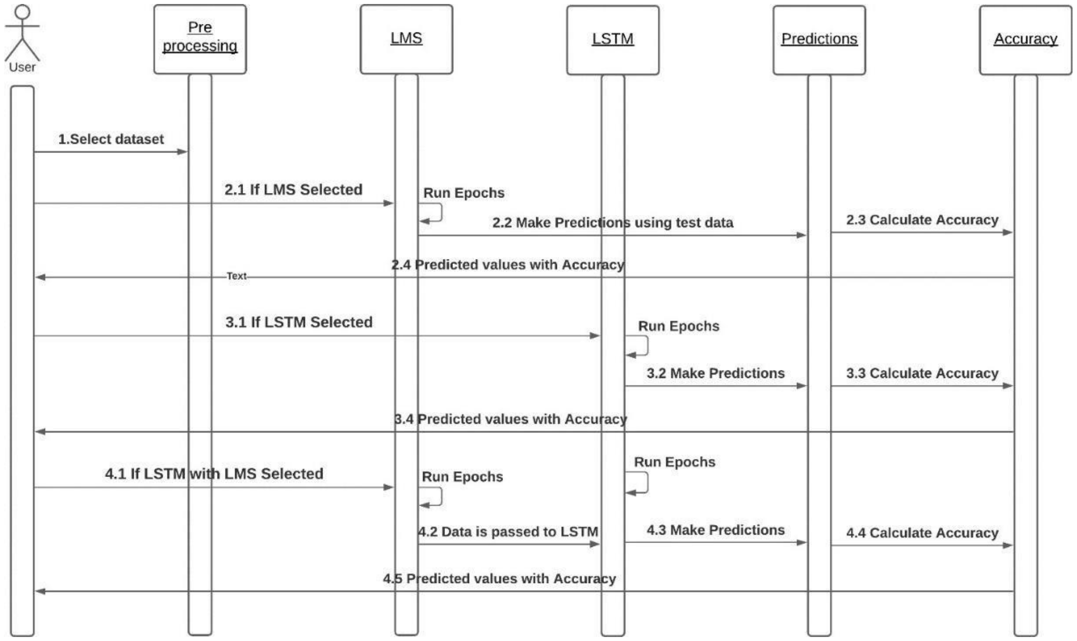


Fig. 9: Execution based on model selection

4.2.3 Activity Diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

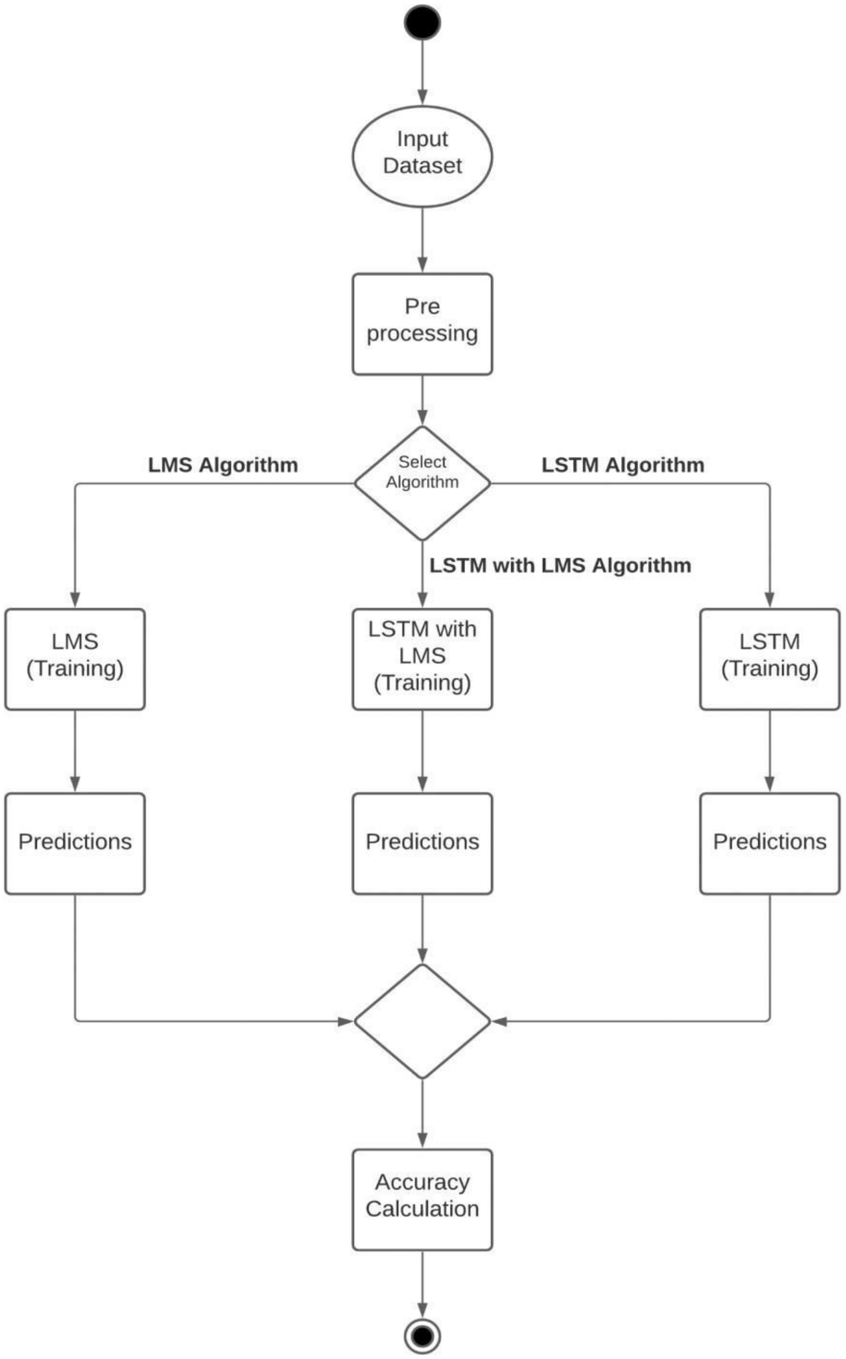


Fig. 10: Execution based on algorithm selection

4.2.4 Collaboration Diagram

Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases.

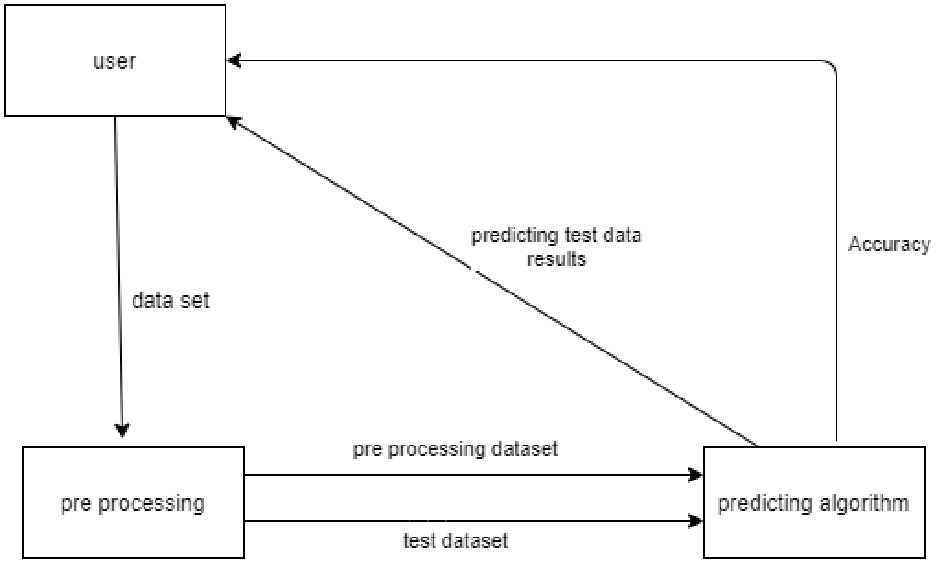


Fig. 11: Data transfer between modules

4.2.5 Flow Chart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

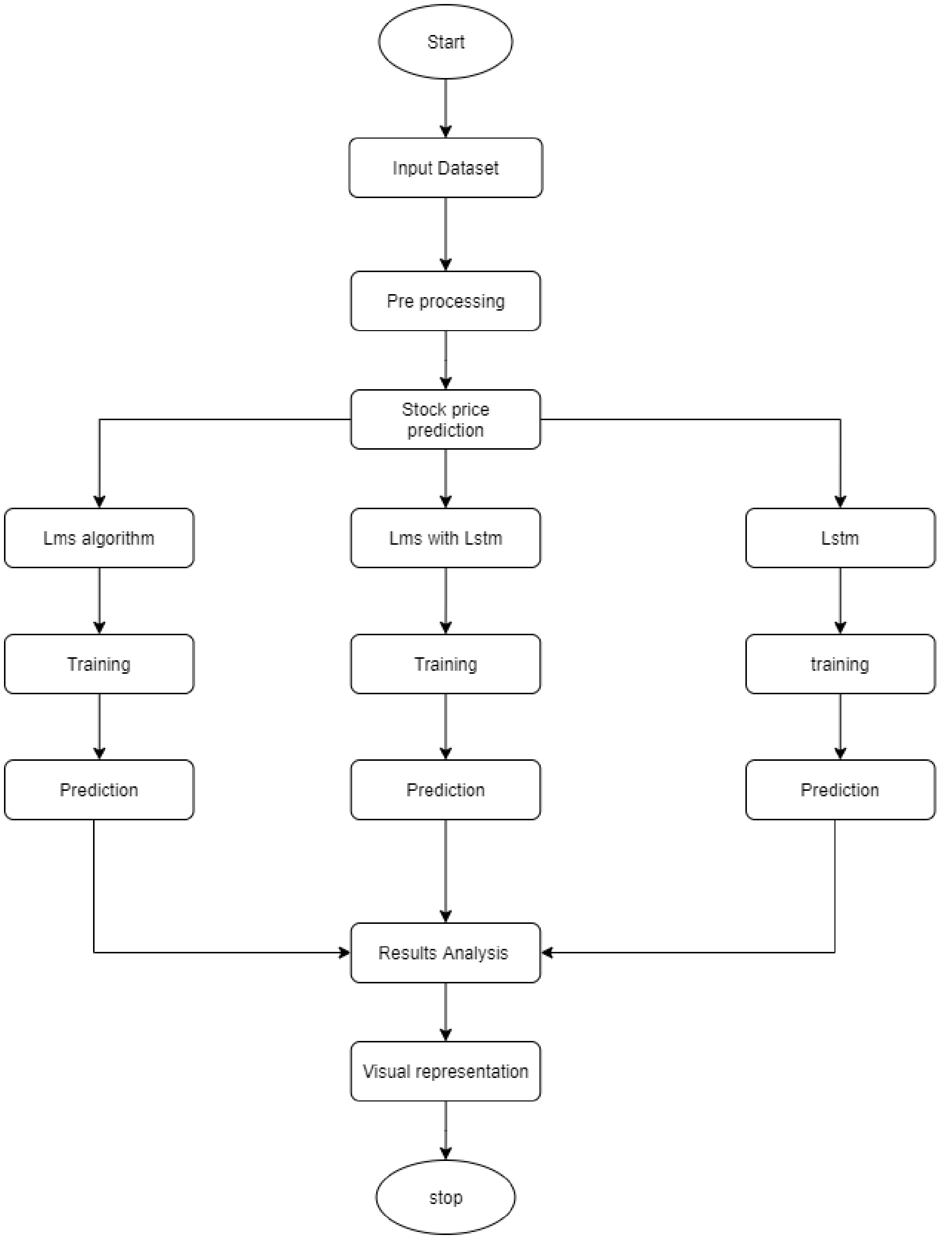


Fig. 12: Flow of execution

4.2.6 Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

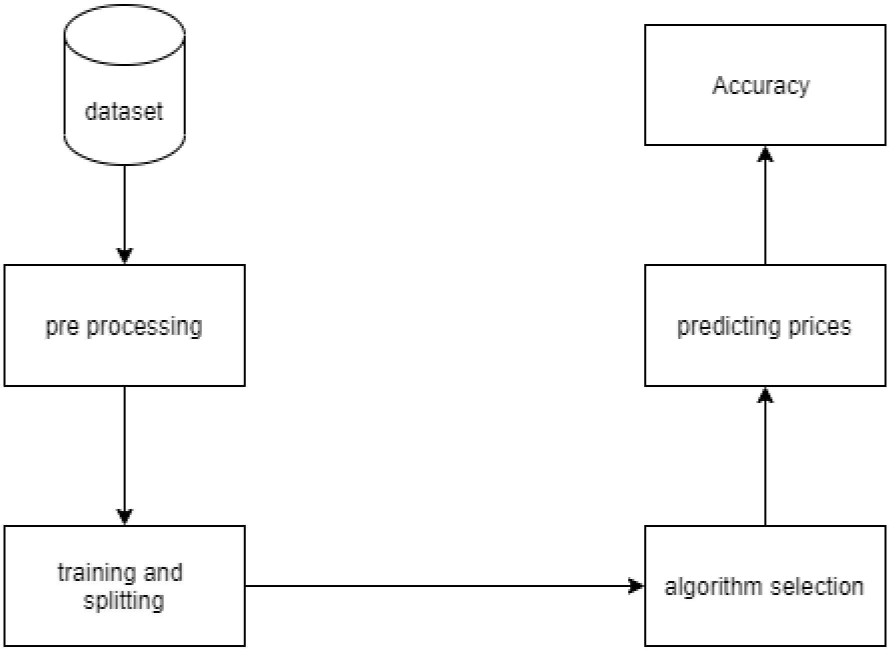


Fig. 13: Components present in the system

4.3 Tool & Technologies

4.3.1 PYTHON

The language of select for this project was Python. This was a straightforward call for many reasons.

1. Python [19] as a language has a vast community behind it. Any problems which may be faced is simply resolved with visit to Stack Overflow. Python is the foremost standard language on the positioning that makes it is very straight answer to any question.
2. Python [19] is an abundance of powerful tools ready for scientific computing Packages. The packages like NumPy, Pandas and SciPy area unit freely available and well documented. These Packages will intensely scale back, and variation the code necessary to write a given program. This makes repetition fast.
3. Python is a language as [19] forgiving and permits for the program that appear as if pseudo code. This can be helpful once pseudo code give in tutorial papers should be required and verified. Using python this step is sometimes fairly trivial.

However, Python is [19] not without its errors. The python is dynamically written language and packages are area unit infamous for Duck writing. This may be frustrating once a packagetechnique returns one thing that, for instance, looks like an array instead of being an actual array. Plus the standard Python documentation did not clearly state the return type of a method, this can't lead without a lot of trials and error testing otherwise happen in a powerfully writtenlanguage. This is a problem that produces learning to use a replacement Python package or library more difficult than it otherwise may be.

4.3.2 NUMPY

Numpy is python package which provide scientific and higher level mathematical abstractions wrapped in python. It is [20] the core library for scientific computing, that contains a providetools for integrating C, strong n-dimensional array object, C++ etc. It is also useful in randomnumber capability, linear algebra etc.

Numpy's array type augments the Python language with an efficient data structure used for numerical work, e.g., manipulating matrices. Numpy additionally provides basic numerical routines, like tools for locating Eigenvectors

4.3.3 SCIKIT LEARN

Scikit-learn [21] could be a free machine learning library for Python. It features numerous classification, clustering and regression algorithms like random forests, k-neighbours, support vector machine, and it furthermore supports Python scientific and numerical libraries like SciPy and NumPy.

In Python Scikit-learn is specifically written, with the core algorithms written in Cython to get the performance. Support vector machines are enforced by a Cython wrapper around LIBSVM .i.e., linear support vector machines and logistic regression by a similar wrapper around LIBLINEAR.

4.3.4 TENSORFLOW

In the TensorFlow [22]has an open source software library for numerical computation using data flow graphs. Inside the graph nodes represent mathematical formulae, the edges of graphrepresent the multidimensional knowledge arrays (tensors) communicated between them. Theversatile architecture permits to deploy the computation to at least one or many GPUs or CPUsin a desktop, mobile device, servers with a single API. TensorFlow was firstly developing by engineers and researchers acting on the Google Brain Team at intervals Google's Machine Intelligence analysis organization for the needs of conducting deep neural networks research and machine learning, but, the system is generally enough to be appropriate in a wide range of alternate domains as well.

Google Brain's second-generation system is TensorFlow. Whereas the reference implementation runs on single devices, TensorFlow can run on multiple GPUs and CPUs. TensorFlow is offered on Windows, macOS, 64-bit Linux and mobile computing platforms together with iOS and Android.

4.3.5 KERAS

Keras is [23] a high-level neural networks API, it is written in Python and also capable of running on top of the Theano, CNTK, or. TensorFlow. It was developed with attention on enabling quick experimentation. having the ability to travel from plan to result with the smallest amount doable delay is key to doing great research.Keras permits for straightforward and quick prototyping (through user-friendliness, modularity, and extensibility). Supports each recurrent networks and convolutional networks, also as combinations of the 2. Runs seamlessly on GPU and CPU. The library contains numerous implementations of generally used neural network building blocks like optimizers, activation functions, layers, objectives and a number of tools to create operating with text and image data easier. The code is hosted on GitHub, and community support forums embody the GitHub issues page, a Gitter channeland a Slack channel.

4.3.6 JUPYTER DASH

Dash is an open-source Python framework used for building analytical web applications. It is a powerful library that simplifies the development of data-driven applications. It’s especially useful for Python data scientists who aren’t very familiar with web development. Users can create amazing dashboards in their browser using dash.

Built on top of Plotly.js, React, and Flask, Dash ties modern UI elements like dropdowns, sliders and graphs directly to your analytical python code.

Dash apps consist of a Flask server that communicates with front-end React components using JSON packets over HTTP requests.

Dash applications are written purely in python, so NO HTML or JavaScript is necessary.

4.3.7 MATHPLOTLIB

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.[3] SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter. Since then it has an active development community[4] and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012[5] and was further joined by Thomas Caswell.[6][7] Matplotlib is a NumFOCUS fiscally sponsored project.[8]

Matplotlib 2.0.x supports Python versions 2.7 through 3.10. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6.[9] Matplotlib has pledged not to support Python 2 past 2020 by signing the Python 3 Statement.[10]

4.3.8 GOOGLE COLABS

The Google Co-Labs is an open-source web application that enables to making and sharing documents that contain visualizations, narrative text, live code and equations. Uses include: data , data visualization, data transformation, statistical modelling, machine learning, numerical simulation, data cleaning and much more [24].

CHAPTER 5
EXPERIMENT ANALYSIS

5.1 system configuration

This project can run on commodity hardware. We ran entire project on an Intel I5 processor with 8 GB Ram, 2 GB Nvidia Graphic Processor, It also has 2 cores which runs at 1.7 GHz, 2.1 GHz respectively. First part of the is training phase which takes 10-15 mins of time and the second part is testing part which only takes few seconds to make predictions and calculate accuracy.

5.1.1 Hardware Requirements:

- RAM: 4 GB
- Storage: 500 GB
- CPU: 2 GHz or faster
- Architecture: 32-bit or 64-bit

5.1.2 Software requirements

- Python 3.5 in Google Colab is used for data pre-processing, model training and prediction.
- Operating System: windows 7 and above or Linux based OS or MAC OS.

5.2 Sample code

1. Imports:

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline

from matplotlib.pylab import rcParams
rcParams['figure.figsize']=20,10
from keras.models import Sequential
from keras.layers import LSTM,Dropout,Dense
from sklearn.preprocessing import MinMaxScaler
```

1. Read the dataset:

```
df=pd.read_csv("NSE-TATA.csv")
df.head()
```

2. Analyze the closing prices from dataframe:

```
df["Date"]=pd.to_datetime(df.Date,format="%Y-%m-%d")
df.index=df['Date']

plt.figure(figsize=(16,8))
plt.plot(df["Close"],label='Close Price history')
```

3. Sort the dataset on date time and filter “Date” and “Close” columns:

```
data=df.sort_index(ascending=True,axis=0)
new_dataset=pd.DataFrame(index=range(0,len(df)),columns=['Date','Close'])

for i in range(0,len(data)):
    new_dataset["Date"][i]=data['Date'][i]
    new_dataset["Close"][i]=data["Close"][i]
```

4. Normalize the new filtered dataset:

```
scaler=MinMaxScaler(feature_range=(0,1))
final_dataset=new_dataset.values

train_data=final_dataset[0:600,:]
valid_data=final_dataset[600:,:]

new_dataset.index=new_dataset.Date
new_dataset.drop("Date",axis=1,inplace=True)
scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(final_dataset)

x_train_data,y_train_data=[],[]

for i in range(60,len(train_data)):
    x_train_data.append(scaled_data[i-60:i,0])
    y_train_data.append(scaled_data[i,0])

x_train_data,y_train_data=np.array(x_train_data),np.array(y_train_data)

x_train_data=np.reshape(x_train_data,(x_train_data.shape[0],x_train_data.shape[1],1))
```

5. Build and train the LSTM model:

```
lstm_model=Sequential()
lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train_data.shape[1],1)))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(1))

inputs_data=new_dataset[len(new_dataset)-len(valid_data)-60:].values
inputs_data=inputs_data.reshape(-1,1)
inputs_data=scaler.transform(inputs_data)

lstm_model.compile(loss='mean_squared_error',optimizer='adam')
lstm_model.fit(x_train_data,y_train_data,epochs=1,batch_size=1,verbose=2)
```

6. Take a sample of a dataset to make stock price predictions using the LSTM model:

```
X_test=[]
for i in range(60,inputs_data.shape[0]):
   X_test.append(inputs_data[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_closing_price=lstm_model.predict(X_test)
predicted_closing_price=scaler.inverse_transform(predicted_closing_price)
```

7. Save the LSTM model:

```
lstm_model.save("saved_model.h5")
```

9. Visualize the predicted stock costs with actual stock costs:

```
train_data=new_dataset[:600]
valid_data=new_dataset[600:]
valid_data['Predictions']=predicted_closing_price
plt.plot(train_data["Close"])
plt.plot(valid_data[['Close',"Predictions"]])
```

Importing Jupiter Dash

```
pip install jupyter-dash -q
```

Main Code:

```
from dash.dash import Dash
import dash
from dash import dcc
from dash import html
import pandas as pd
import plotly.graph_objs as go
from dash.dependencies import Input, Output
from keras.models import load_model
from sklearn.preprocessing import MinMaxScaler
```

```
import numpy as np
import io
import plotly.express as px
from jupyter_dash import JupyterDash

app = JupyterDash(_name_)

scaler=MinMaxScaler(feature_range=(0,1))



from google.colab import files

uploaded = files.upload()

df_nse = pd.read_csv(io.BytesIO(uploaded['NSE-TATA.csv']))

df_nse["Date"]=pd.to_datetime(df_nse.Date,format="%Y-%m-%d")
df_nse.index=df_nse['Date']


data=df_nse.sort_index(ascending=True,axis=0)
new_data=pd.DataFrame(index=range(0,len(df_nse)),columns=['Date','Close'])

for i in range(0,len(data)):
    new_data["Date"][i]=data['Date'][i]
    new_data["Close"][i]=data["Close"][i]

new_data.index=new_data.Date
new_data.drop("Date",axis=1,inplace=True)

dataset=new_data.values

train=dataset[0:600,:]
valid=dataset[600:,:]

scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(dataset)


x_train,y_train=[],[]

for i in range(60,len(train)):
    x_train.append(scaled_data[i-60:i,0])
    y_train.append(scaled_data[i,0])
```

```
x_train,y_train=np.array(x_train),np.array(y_train)

x_train=np.reshape(x_train,(x_train.shape[0],x_train.shape[1],1))

lstm_model=load_model("saved_lstm_model.h5")



inputs=new_data[len(new_data)-len(valid)-60:].values
inputs=inputs.reshape(-1,1)
inputs=scaler.transform(inputs)




X_test=[]
for i in range(60,inputs.shape[0]):
    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))
closing_price=lstm_model.predict(X_test)
closing_price=scaler.inverse_transform(closing_price)

train=new_data[:600]
valid=new_data[600:]
valid['Predictions']=closing_price



df= pd.read_csv(io.BytesIO(uploaded['stock_data.csv']))
```

App Desiging:

```
app.layout = html.Div([

    html.H1("Stock Price Analysis Dashboard", style={"textAlign": "center"}),

    dcc.Tabs(id="tabs", children=[

        dcc.Tab(label='NSE-TATAGLOBAL Stock Data',children=[
                            html.Div([
                                html.H2("Actual closing price",style={"textAlign": "center"}),


                                dcc.Graph(
                                    id="Actual Data",
                                    figure={
                                        "data":[
                                                                            go.Scatter(
                                                                                        x=train.index,
                                                                                        y=valid["Close"],
                                                                                        mode='markers'

                                                                            )

                                                                ],
                                                                "layout":go.Layout(
                                                                            title='scatter plot',
                                                                            xaxis={'title':'Date'},
                                                                            yaxis={'title':'Closing Rate'}
                                                                )
                                                    }

                                        ),
                                        html.H2("LSTM Predicted closing price",style={"textAlign": "center"}),
                                        dcc.Graph(
                                                    id="Predicted Data",
                                                    figure={
                                                                "data":[
                                                                            go.Scatter(
                                                                                        x=valid.index,
                                                                                        y=valid["Predictions"],
                                                                                        mode='markers'
                                                                            )

                                                                ],
                                                                "layout":go.Layout(
                                                                            title='scatter plot',
                                                                            xaxis={'title':'Date'},
                                                                            yaxis={'title':'Closing Rate'}
                                                                )
                                                    }

                                        )
                            ])


]),

dcc.Tab(label='Facebook Stock Data', children=[
    html.Div([
        html.H1("Stocks High vs Lows",
                style={'textAlign': 'center'}),
                dcc.Dropdown(id='my-dropdown',
                            options=[{'label': 'Tesla', 'value': 'TSLA'},
                                      {'label': 'Apple','value': 'AAPL'},
                                      {'label': 'Facebook', 'value': 'FB'},
                                      {'label': 'Microsoft','value': 'MSFT'}],
                            multi=True,value=['FB'],
                            style={"display": "block", "margin-left": "auto",
                                   "margin-right": "auto", "width": "60%"}),

                dcc.Graph(id='highlow'),
                html.H1("Stocks Market Volume", style={'textAlign': 'center'}),

                dcc.Dropdown(id='my-dropdown2',
                            options=[{'label': 'Tesla', 'value': 'TSLA'},
                                      {'label': 'Apple','value': 'AAPL'},
                                      {'label': 'Facebook', 'value': 'FB'},
                                      {'label': 'Microsoft','value': 'MSFT'}],
                            multi=True,value=['FB'],
                            style={"display": "block", "margin-left": "auto",
                                   "margin-right": "auto", "width": "60%"}),
                dcc.Graph(id='volume')
            ], className="container"),
        ])


    ])
])



@app.callback(Output('highlow','figure'),
              [Input('my-dropdown','value')])
def update_graph(selected_dropdown):
    dropdown = {"TSLA": "Tesla","AAPL": "Apple","FB": "Facebook","MSFT": "Microsoft",}
    trace1 = []
    trace2 = []
    for stock in selected_dropdown:
        trace1.append(
          go.Scatter(x=df[df["Stock"] == stock]["Date"],
                     y=df[df["Stock"] == stock]["High"],
                     mode='lines', opacity=0.7,
                     name=f'High {dropdown[stock]}',textposition='bottom center'))
        trace2.append(
          go.Scatter(x=df[df["Stock"] == stock]["Date"],
                     y=df[df["Stock"] == stock]["Low"],
                     mode='lines', opacity=0.6,
                     name=f'Low {dropdown[stock]}',textposition='bottom center'))
```

```
                 traces = [trace1, trace2]
    data = [val for sublist in traces for val in sublist]
    figure = {'data': data,
              'layout': go.Layout(colorway=["#5E0DAC", '#FF4F00', '#375CB1',
                                            '#FF7400', '#FFF400', '#FF0056'],
            height=600,



            title=f"High and Low Prices for {', '.join(str(dropdown[i]) for i in selected_dropdown)} Over Time",
            xaxis={"title":"Date",
                   'rangeselector': {'buttons': list([{'count': 1, 'label': '1M',
                                                       'step': 'month',
                                                       'stepmode': 'backward'},

                                                      {'count': 6, 'label': '6M',
                                                       'step': 'month',
                                                       'stepmode': 'backward'},
                                                      {'step': 'all'}])},
                   'rangeslider': {'visible': True}, 'type': 'date'},
             yaxis={"title":"Price (USD)"})}
    return figure



@app.callback(Output('volume', 'figure'),
              [Input('my-dropdown2', 'value')])
def update_graph(selected_dropdown_value):
    dropdown = {"TSLA": "Tesla","AAPL": "Apple","FB": "Facebook","MSFT": "Microsoft",}
    trace1 = []
    for stock in selected_dropdown_value:
        trace1.append(
          go.Scatter(x=df[df["Stock"] == stock]["Date"],
                     y=df[df["Stock"] == stock]["Volume"],
                     mode='lines', opacity=0.7,
                     name=f'Volume {dropdown[stock]}', textposition='bottom center'))
    traces = [trace1]
    data = [val for sublist in traces for val in sublist]
    figure = {'data': data,
              'layout': go.Layout(colorway=["#5E0DAC", '#FF4F00', '#375CB1',
                                            '#FF7400', '#FFF400', '#FF0056'],
            height=600,
            title=f"Market Volume for {', '.join(str(dropdown[i]) for i in selected_dropdown_value)} Over Time",
            xaxis={"title":"Date",
                   'rangeselector': {'buttons': list([{'count': 1, 'label': '1M',
                                                       'step': 'month',
                                                       'stepmode': 'backward'},
                                                        {'count': 6, 'label': '6M',
                                                         'step': 'month',
                                                         'stepmode': 'backward'},
                                                        {'step': 'all'}])},
                    'rangeslider': {'visible': True}, 'type': 'date'},
            yaxis={"title":"Transactions Volume"})}
    return figure
```

Running the App:

```
app.run_server()
```

Building LSTM Model and along with prereading data

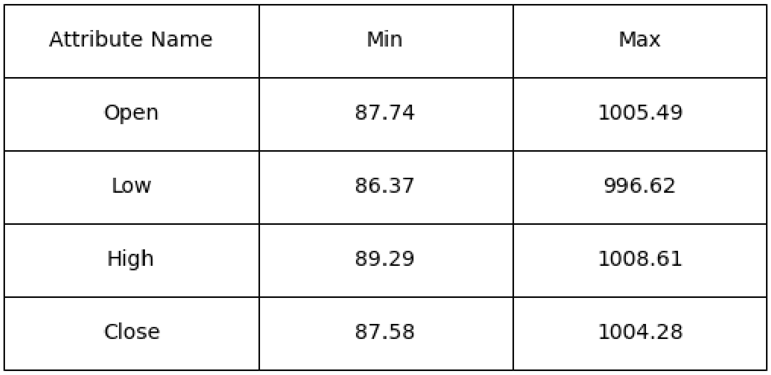
```
import pandas as pd
import numpy as np
import io
%matplotlib inline
import matplotlib.pyplot as plt



from google.colab import files
uploaded = files.upload()

from matplotlib.pylab import rcParams
rcParams['figure.figsize']=20,10

from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))

df=pd.read_csv(io.BytesIO(uploaded['NSE-TATA.csv']))
df.head()

df["Date"]=pd.to_datetime(df.Date,format="%Y-%m-%d")
df.index=df['Date']

plt.figure(figsize=(16,8))
plt.plot(df["Close"],label='Close Price history')
```

```
from keras.models import Sequential
from keras.layers import LSTM,Dropout,Dense

data=df.sort_index(ascending=True,axis=0)
new_dataset=pd.DataFrame(index=range(0,len(df)),columns=['Date','Close'])

for i in range(0,len(data)):
    new_dataset["Date"][i]=data['Date'][i]
    new_dataset["Close"][i]=data["Close"][i]



new_dataset.index=new_dataset.Date
new_dataset.drop("Date",axis=1,inplace=True)



final_dataset=new_dataset.values

train_data=final_dataset[0:600,:]
valid_data=final_dataset[600:,:]


scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(final_dataset)

x_train_data,y_train_data=[],[]




for i in range(60,len(train_data)):
    x_train_data.append(scaled_data[i-60:i,0])
    y_train_data.append(scaled_data[i,0])

x_train_data,y_train_data=np.array(x_train_data),np.array(y_train_data)

x_train_data=np.reshape(x_train_data,(x_train_data.shape[0],x_train_data.shape[1],1))

lstm_model=Sequential()
lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train_data.shape[1],1)))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(1))
```

```
    lstm_model.compile(loss='mean_squared_error',optimizer='adam')
    lstm_model.fit(x_train_data,y_train_data,epochs=1,batch_size=1,verbose=2)

    inputs_data=new_dataset[len(new_dataset)-len(valid_data)-60:].values
    inputs_data=inputs_data.reshape(-1,1)
    inputs_data=scaler.transform(inputs_data)





    X_test=[]
    for i in range(60,inputs_data.shape[0]):
        X_test.append(inputs_data[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))
    closing_price=lstm_model.predict(X_test)
    closing_price=scaler.inverse_transform(closing_price)

    lstm_model.save("saved_lstm_model.h5")

train_data=new_dataset[:600]
valid_data=new_dataset[600:]
valid_data['Predictions']=closing_price
plt.plot(train_data["Close"])
plt.plot(valid_data[['Close',"Predictions"]])
plt.show()
```

5.3 Sample Input

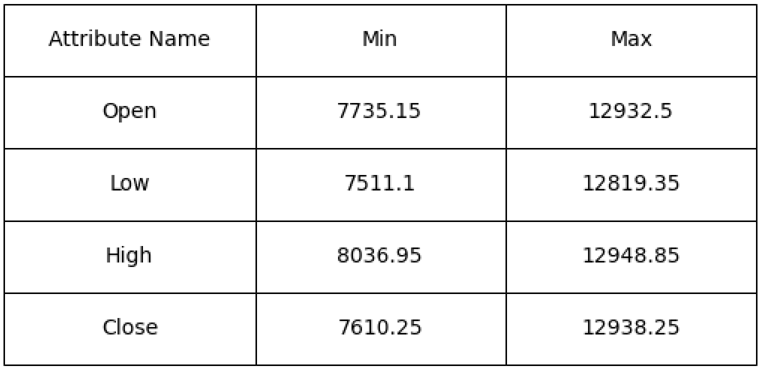
Google



Attribute Name	Min	Max
Open	87.74	1005.49
Low	86.37	996.62
High	89.29	1008.61
Close	87.58	1004.28

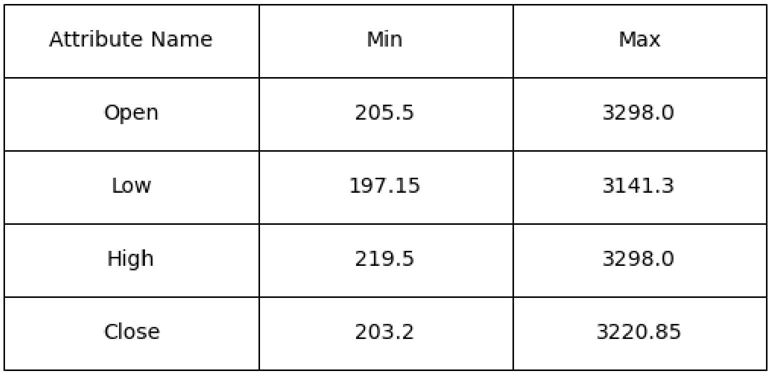
Table 1: Min and Max of columns in Google Dataset

Nifty50



Attribute Name	Min	Max
Open	7735.15	12932.5
Low	7511.1	12819.35
High	8036.95	12948.85
Close	7610.25	12938.25

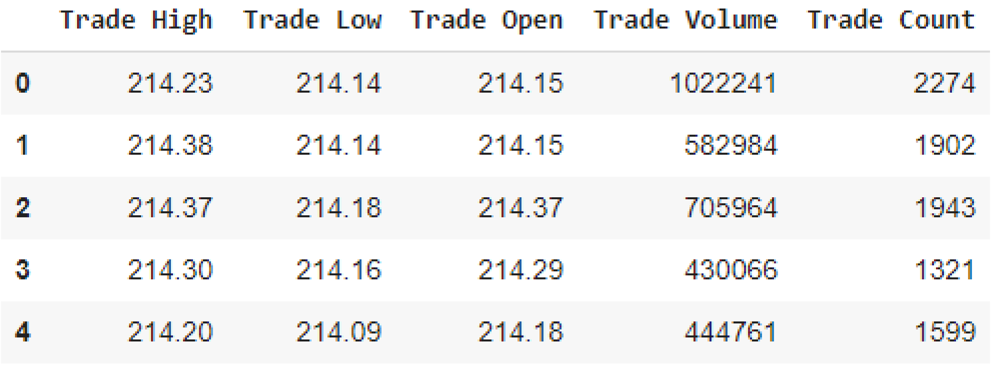
Table 2: Min and Max of columns in Nifty50DatasetRelianc



Attribute Name	Min	Max
Open	205.5	3298.0
Low	197.15	3141.3
High	219.5	3298.0
Close	203.2	3220.85

Table 3: Min and Max of columns in Reliance

DatasetSample Input:

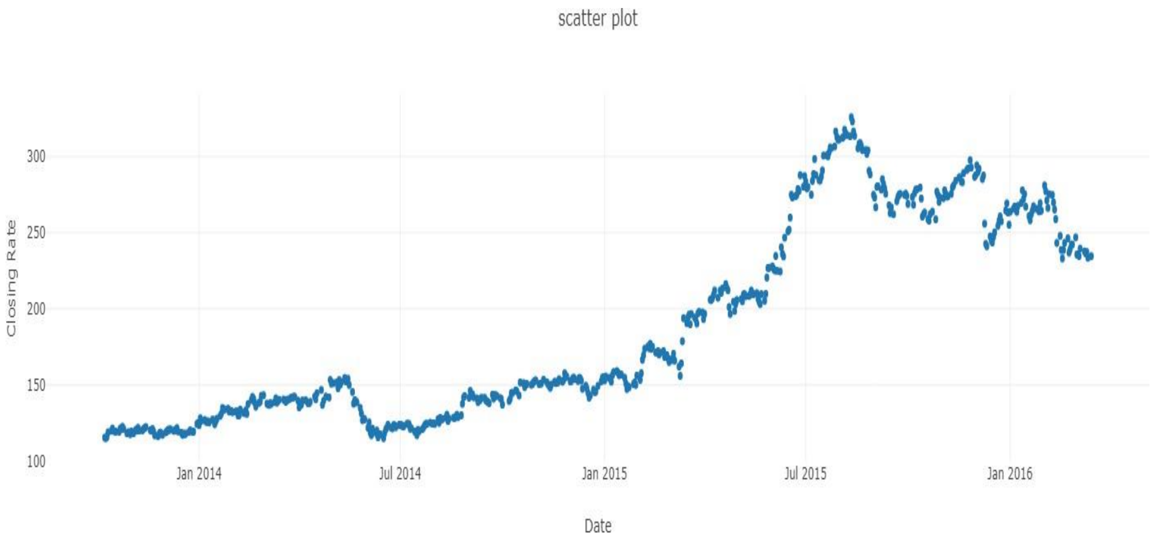


	Trade High	Trade Low	Trade Open	Trade Volume	Trade Count
0	214.23	214.14	214.15	1022241	2274
1	214.38	214.14	214.15	582984	1902
2	214.37	214.18	214.37	705964	1943
3	214.30	214.16	214.29	430066	1321
4	214.20	214.09	214.18	444761	1599

Table 4: Sample input

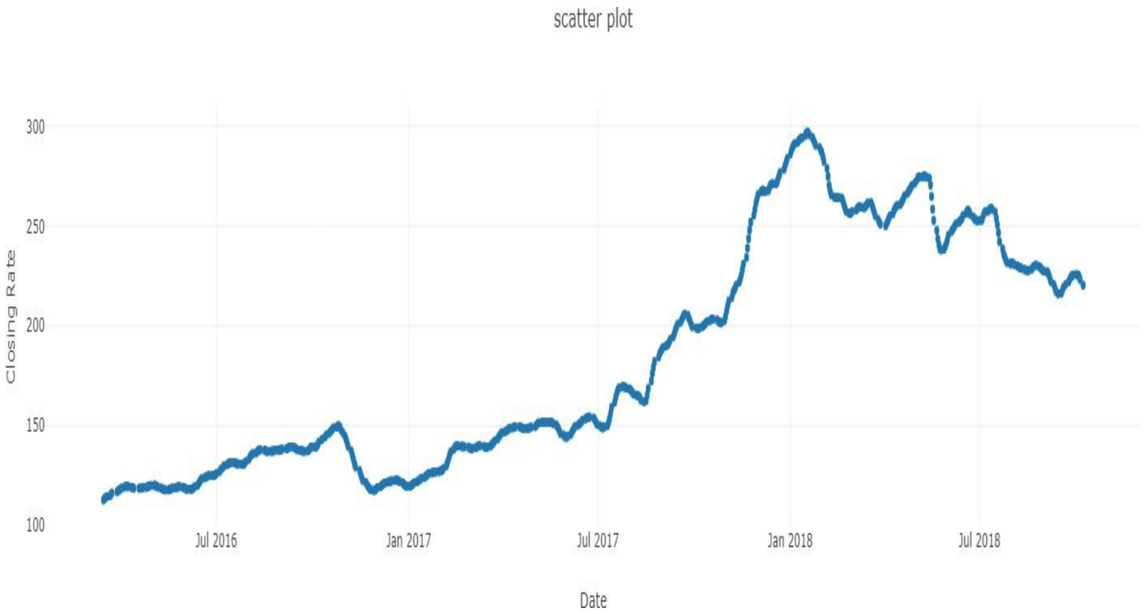
NSE-TATA

Input Data Sample Plot

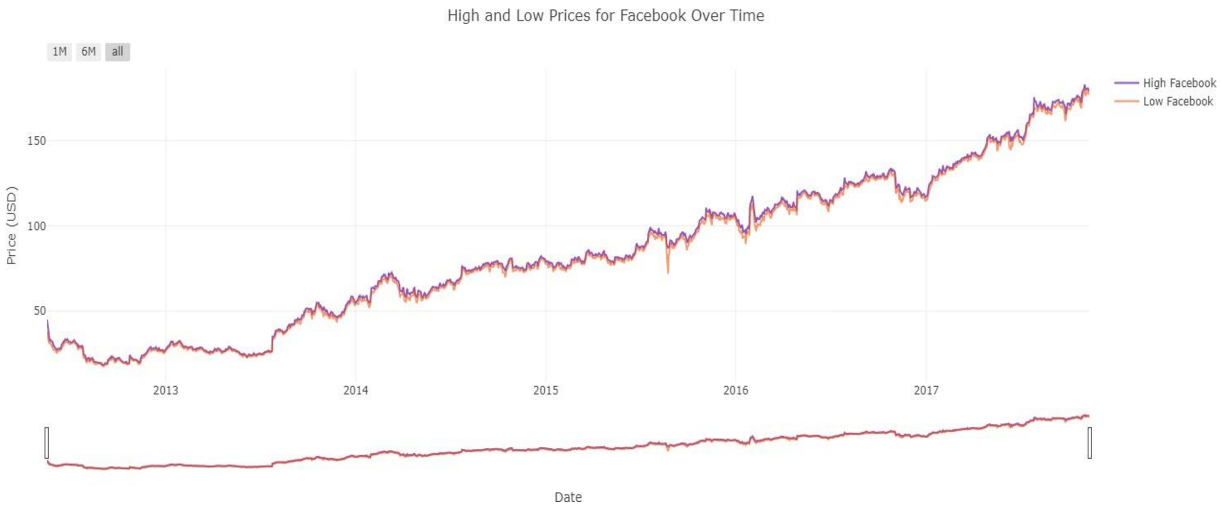


LSTM Predited Closing Price

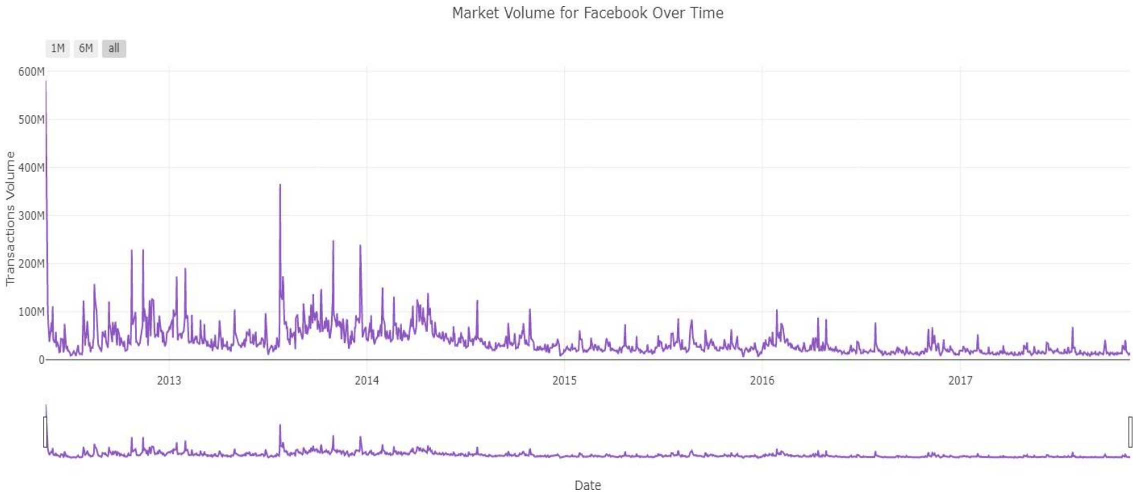
Sample Output



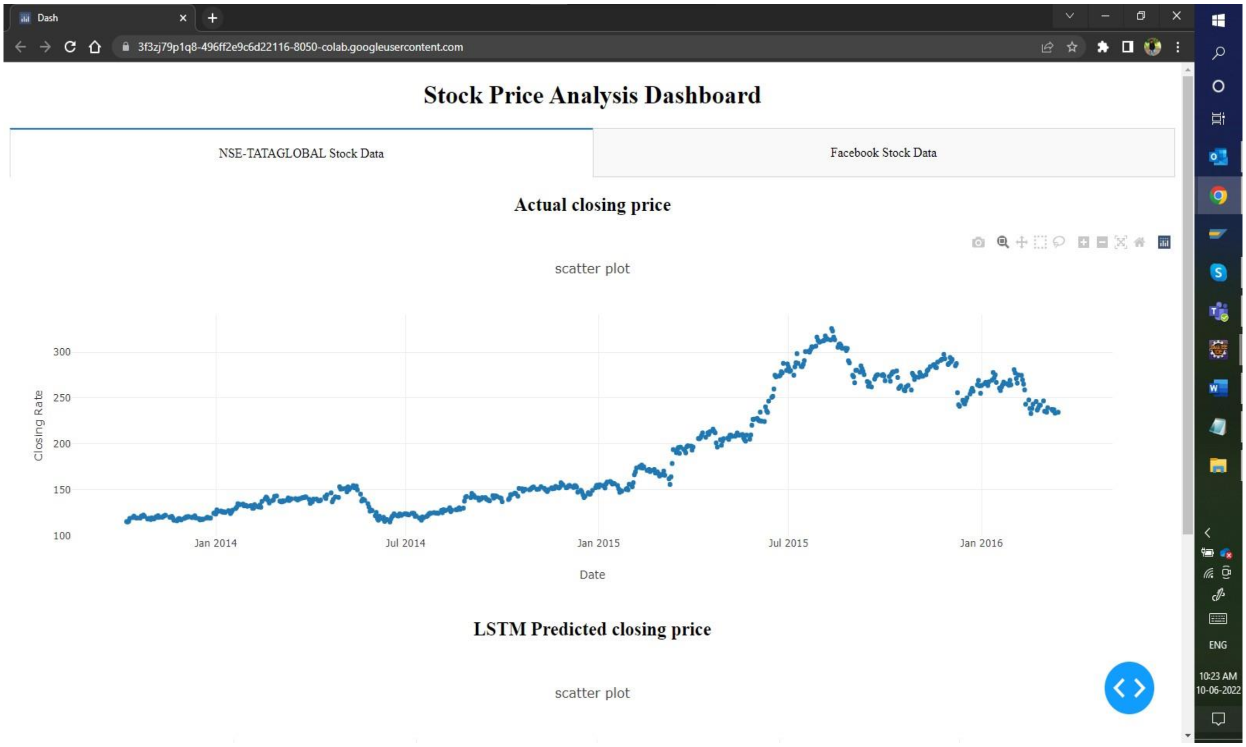
FaceBook Over Time Analysis



FaceBook Volume Analysis



5.4 Website Page



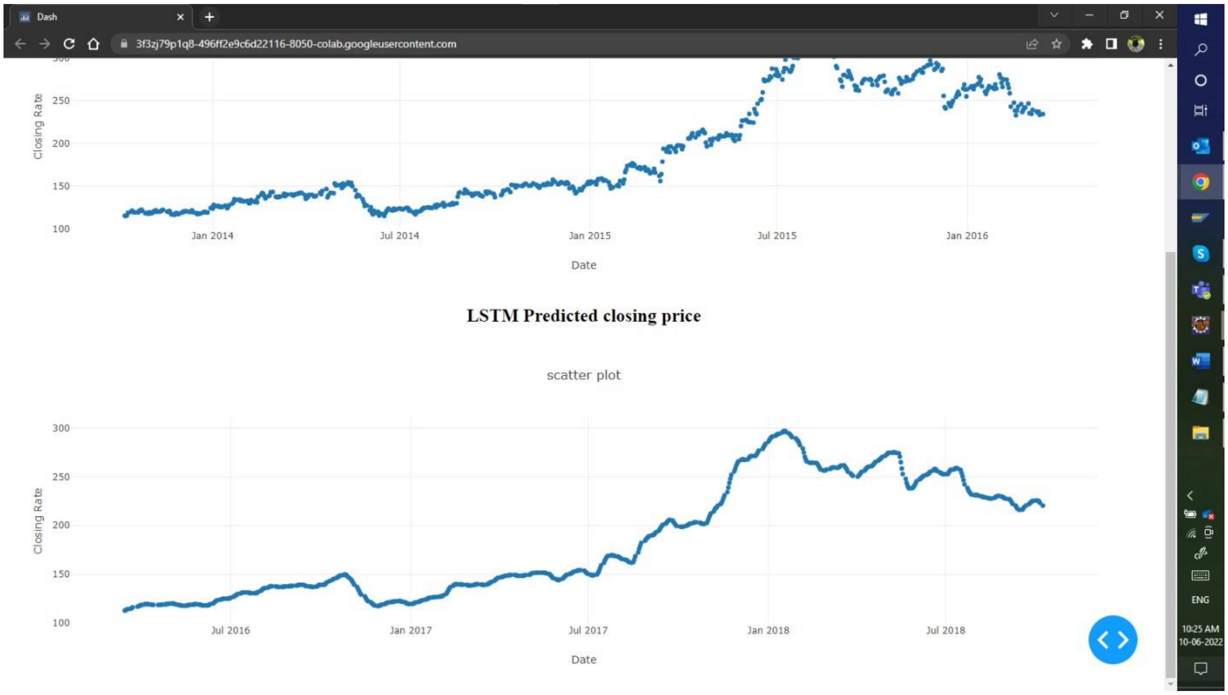
3f3zj79p1q8-496ff2e9c6d22116-8050-colab.googleusercontent.com

Stock Price Analysis Dashboard

NSE-TATAGLOBAL Stock Data | Facebook Stock Data

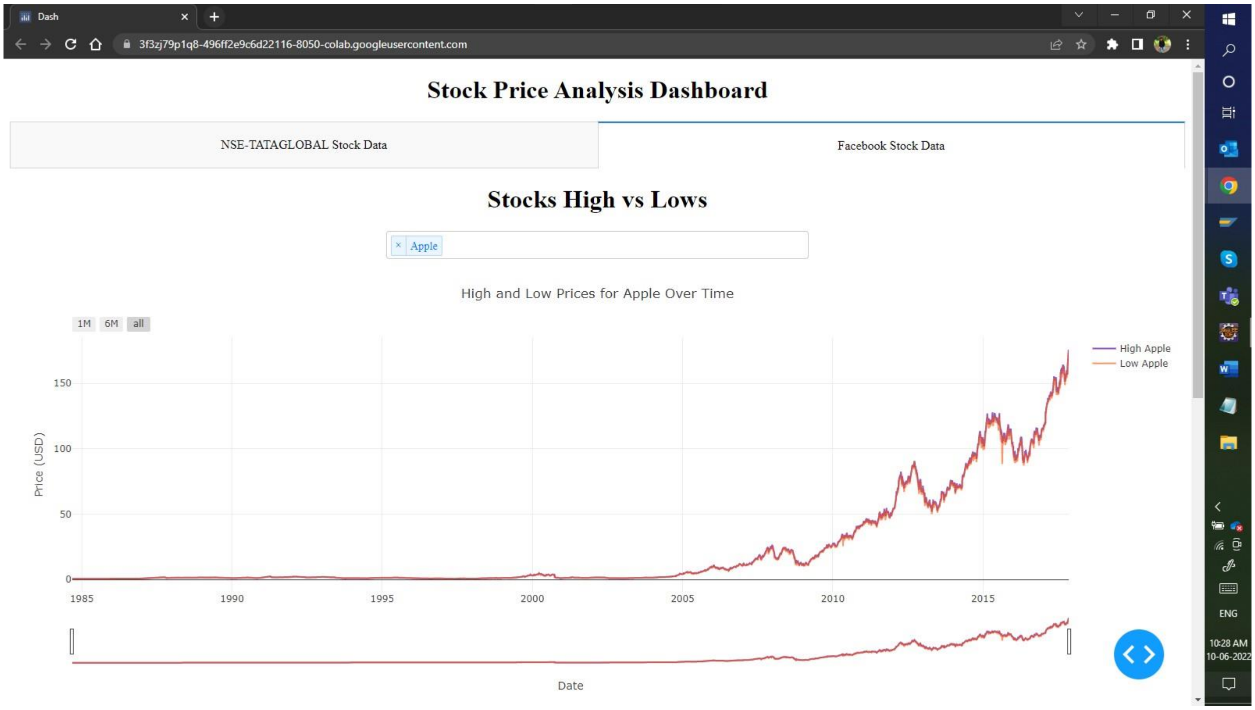
Actual closing price

LSTM Predicted closing price



3f3zj79p1q8-496ff2e9c6d22116-8050-colab.googleusercontent.com

LSTM Predicted closing price



Dash
3f3zj79p1q8-496ff2e9c6d22116-8050-colab.googleusercontent.com
10:28 AM
10-06-2022
ENG

Stock Price Analysis Dashboard

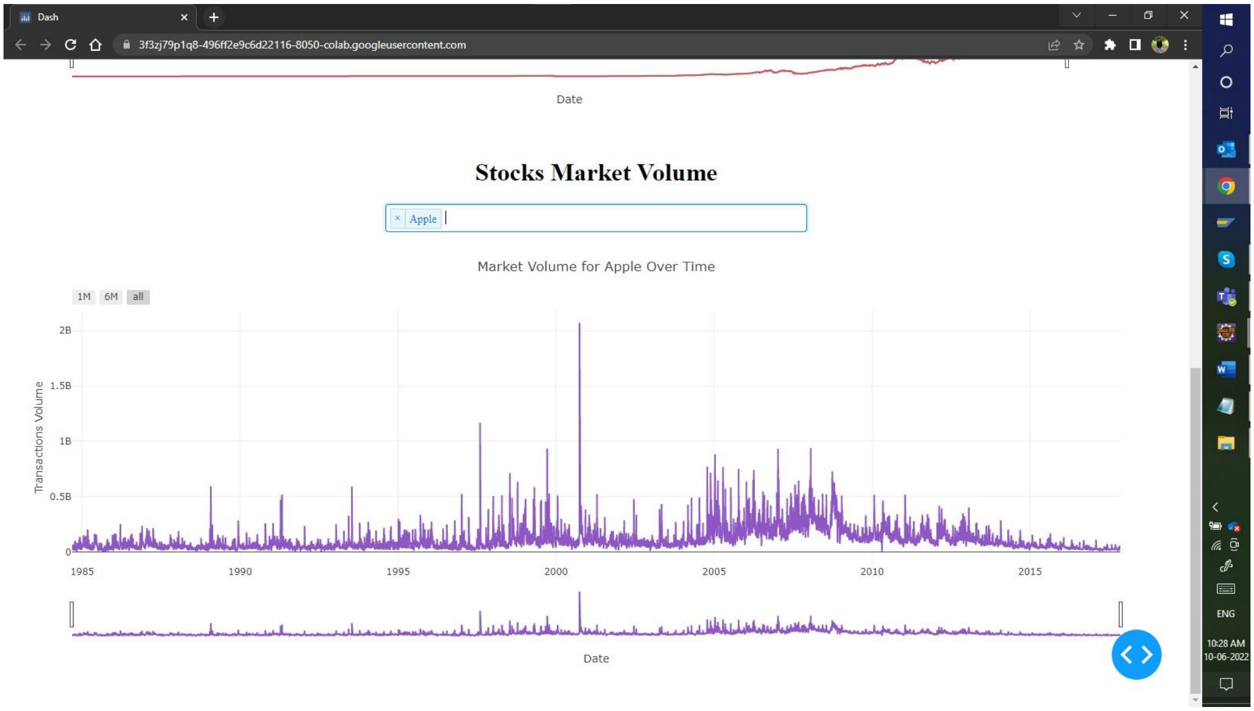
NSE-TATAGLOBAL Stock Data

Facebook Stock Data

Stocks High vs Lows

× Apple

High and Low Prices for Apple Over Time



Dash
3f3zj79p1q8-496ff2e9c6d22116-8050-colab.googleusercontent.com
Date
10:28 AM
10-06-2022
ENG

Stocks Market Volume

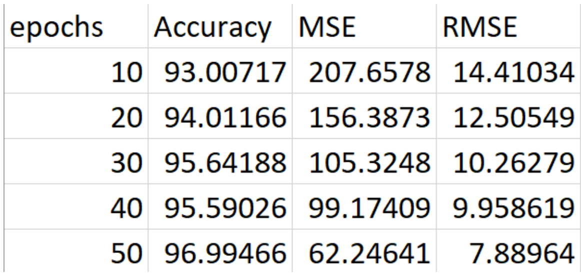
× Apple

Market Volume for Apple Over Time

5.5 Performance Measure

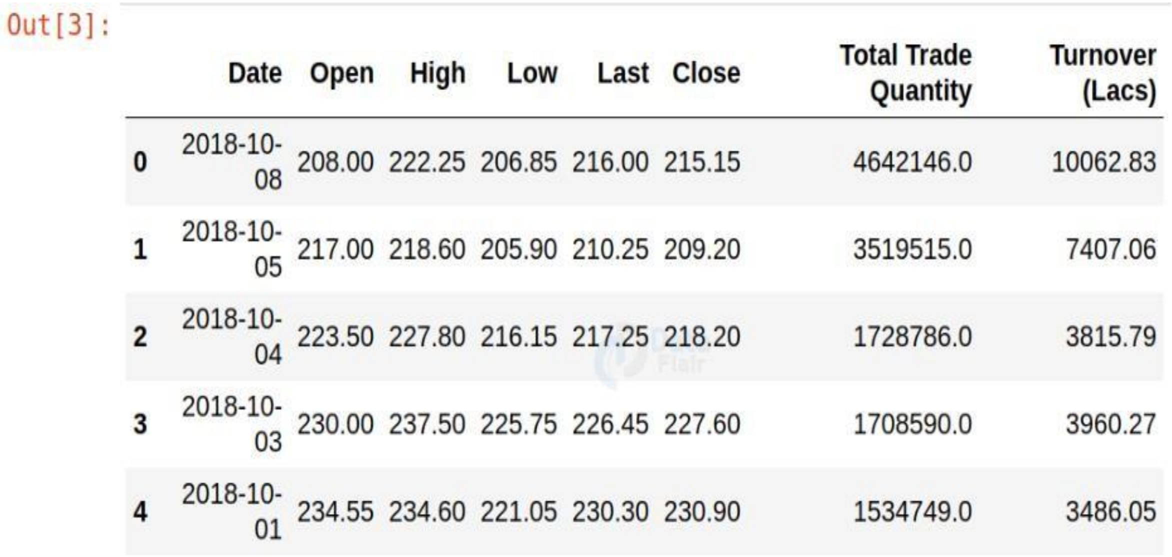
5.5.1 LSTM

5.5.1.1 NSE-TATA



epochs	Accuracy	MSE	RMSE
10	93.00717	207.6578	14.41034
20	94.01166	156.3873	12.50549
30	95.64188	105.3248	10.26279
40	95.59026	99.17409	9.958619
50	96.99466	62.24641	7.88964

Table 5: Epochs for NSE-TATA Dataset using LSTM

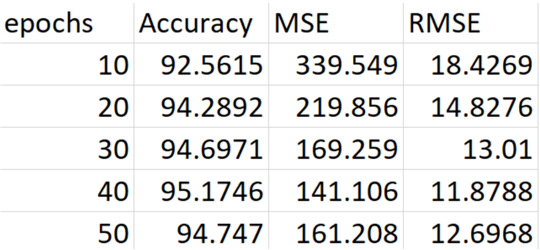


Out[3]:

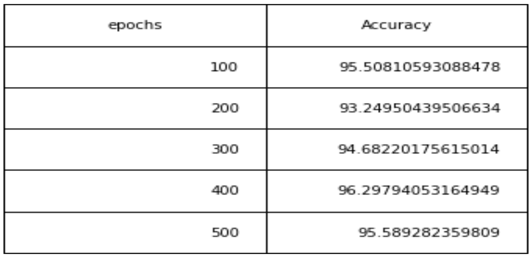
	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

5.5.2 LSTM with LMS

5.5.2.1 Google



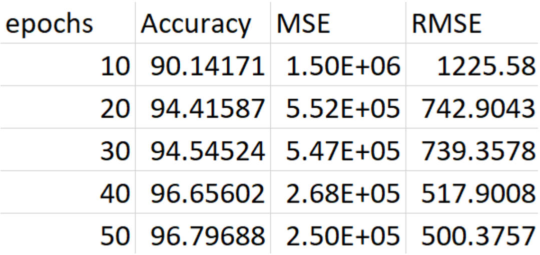
epochs	Accuracy	MSE	RMSE
10	92.5615	339.549	18.4269
20	94.2892	219.856	14.8276
30	94.6971	169.259	13.01
40	95.1746	141.106	11.8788
50	94.747	161.208	12.6968



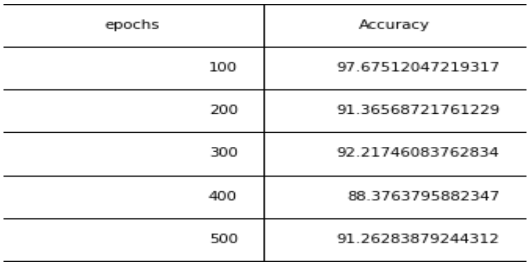
epochs	Accuracy
100	95.50810593088478
200	93.24950439506634
300	94.68220175615014
400	96.29794053164949
500	95.589282359809

Table 8: Epochs for Google Dataset using LSTM with LMS

5.5.2.2 Nifty50



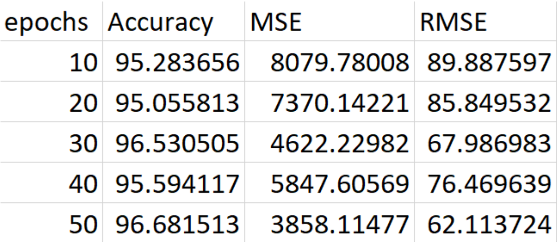
epochs	Accuracy	MSE	RMSE
10	90.14171	1.50E+06	1225.58
20	94.41587	5.52E+05	742.9043
30	94.54524	5.47E+05	739.3578
40	96.65602	2.68E+05	517.9008
50	96.79688	2.50E+05	500.3757



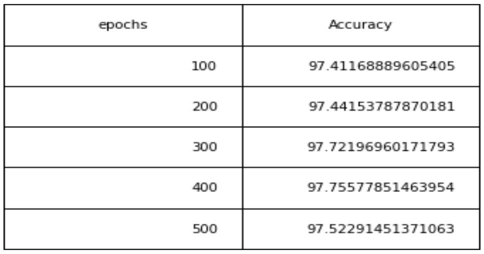
epochs	Accuracy
100	97.67512047219317
200	91.36568721761229
300	92.21746083762834
400	88.3763795882347
500	91.26283879244312

Table 9: Epochs for Nifty50 Dataset using LSTM with LMS

5.5.2.3 Reliance



epochs	Accuracy	MSE	RMSE
10	95.283656	8079.78008	89.887597
20	95.055813	7370.14221	85.849532
30	96.530505	4622.22982	67.986983
40	95.594117	5847.60569	76.469639
50	96.681513	3858.11477	62.113724



epochs	Accuracy
100	97.41168889605405
200	97.44153787870181
300	97.72196960171793
400	97.75577851463954
500	97.52291451371063

Table 10: Epochs for Reliance Dataset using LSTM with LMS

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this project, we are predicting closing stock price of any given organization, we developed a web application for predicting close stock price using LMS and LSTM algorithms for prediction. We have applied datasets belonging to Google, Nifty50, TCS, Infosys and Reliance Stocks and achieved above 95% accuracy for these datasets.

6.2 Future work

- We want to extend this application for predicting cryptocurrency trading.
- We want to add sentiment analysis for better analysis.

REFERENCES

Datasets: NSE-TATA, facebook, apple

[1] Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, Giridhar Maji, Narayan C. Debnath, Soumya Sen

[2] S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017.

[3] Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM” in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015.

[4] Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, “An innovative neural network approach for stock market prediction”, 2018

[5] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.

[6] Pranav Bhat Electronics and Telecommunication Department, Maharashtra Institute of Technology, Pune. Savitribai Phule Pune University - A Machine Learning Model for Stock Market Prediction.

[7] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning.

[8] V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India - Stock Market Prediction Using Machine Learning.

[9] Asset Durmagambetov currently works at the mathematics, CNTFI. Asset does research in Theory of Computation and Computing in Mathematics, Natural Science, Engineering and Medicine. Their current project is 'The Riemann Hypothesis-Millennium Prize Problems' - stock market predictions.

[10] Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.

[11] Manh Ha Duong Boriss Siliverstovs June 2006 - The Stock Market and Investment.

[12] Dharmaraja Selvamuthu , Vineet Kumar and Abhishek Mishra Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India - Indian stock market prediction using artificial neural networks on tick data.

[13] Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering - Forecasting the Stock Market Index Using Artificial Intelligence Techniques.

[14] Xiao-Yang Liu1 Hongyang Yang,Qian Chen4,Runjia ZhangLiuqing Yang Bowen Xiao Christina Dan Wang Electrical Engineering, 2Department of Statistics, 3Computer Science, Columbia University, 3AI4Finance LLC., USA, Ion Media Networks, USA, Department of Computing, Imperial College, 6New York University (Shanghai) - A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance.

[15] Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari SahooDepartment of Computer Science & Information Systems, BITS Pilani K.K.Birla Goa campus, India bDivision of Mathematical Sciences, Nanyang Technological University, Singapore cDepartment of Mathematics, BITS Pilani K.K.Birla Goa campus, India - Forecasting directional movements of stock prices for intraday trading using LSTM and random forests.

[16] Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, Junwen Duan Research Center for Social Computing and Information Retrieval Harbin Institute of Technology, China - Event Representation Learning Enhanced with External Commonsense Knowledge.

[17] Huicheng Liu Department of Electrical and Computer Engineering Queen’s University, Canada - Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network.

[18] Hyeong Kyu Choi, B.A Student Dept. of Business Administration Korea University Seoul, Korea = Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model.

[19] M. Nabipour Faculty of Mechanical Engineering, Tarbiat Modares University, 14115-143 Tehran, Iran; Mojtaba.nabipour@modares.ac.ir - Deep Learning for Stock Market prediction.

[20] Lavanya Ra SRM Institute of Science and Technology | SRM · Department of Computer Science - Stock Market Prediction.

[21] M. Mekayel Anik · M. Shamsul Arefin (B) Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh - An Intelligent Technique for Stock Market Prediction