MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY, WEST BENGAL

STOCK PRICE PREDICTION USING MACHINE LEARNING

MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF COMPUTER APPLICATION

BY

Name: Soumallya Dey **Roll No: 12071020005**

Name: Agni Rakshit **Roll No: 12071020045**

UNDER THE GUIDANCE OF

PROF. (DR.) FALGUNI CHAKRABORTY

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER APPLICATIONS, BCREC

DR. B. C. ROY ENGINEERING COLLEGE

[Approved by AICTE & Affiliated to MAKAUT]

Fuljhore, Jemua Road, Durgapur – 713206

West Bengal, India

2020 - 2022

DR. B.C. ROY ENGINEERING COLLEGE

(Approved by AICTE & Affiliated to MAKAUT)

Campus: Jemua Road, Fuljhore, Durgapur – 713206 (W.B.), India Tel: (0343) 250-1353/4106/4121/2449, Fax: (0343) 250-4059/3424 E-mail: info@bcrec.ac.in • Website: bcrec.ac.in

To Whom It May Be Concern

This is to certify that, **AGNI RAKSHIT**, Roll No: **12071020045**, **SOUMALLYA DEY** Roll No: **12071020005** students of 3rd year, final semester in the Department of Computer Applications[2020 – 2022], Dr. B.C. Roy Engineering College, Durgapur – 713206, worked on the major project entitled.

" STOCK PRICE PREDICTION "

They have worked sincerely, dedicatedly and enthusiastically under my supervision. To the best of my knowledge, this is an original and technically accurate work. I hereby recommend that the report prepared by them may be accepted as partial fulfilment of the requirement for the degree of **Master of Computer Application (MCA) from Maulana Abul Kalam Azad University of Technology, West Bengal.**

Prof.(Dr.) Falguni Chakraborty
Assistant Professor,
Department of Computer Applications
Dr. B. C. Roy Engineering College, Durgapur

Forwarded by:

Prof. (Dr.) Pabitra Kumar Dey
Head of the Department
Department of Computer Applications
Dr. B. C. Roy Engineering College, Durgapur

ACKNOWLEDGEMENT

The experience that I have gathered during the project has been unique and overwhelming. For this I am pleased to express my deepest sense gratitude and regards to my respected project guide Prof.(Dr.) Falguni Chakraborty. For providingus with resources whenever we needed, encouraging us to work freely and confidently and allowing us to pursue the project in Machine Learning
In spite of his busy schedules, his frequent valuable advice and master suggestionat every stage lead to a systematic approach in completing this project without which this project work could not take its shape.

I wish to express my gratitude to Dr. B.C. Roy Engineering College, Durgapur for allowing me to carry out this project.

We are very much thankful to all who have provided guidance andencouragement to us.

_____

Name: SOUMALLYA DEY

(Roll No. - 12071020005)

_____

Name: AGNI RAKSHIT

(Roll No.-12071020045)

Date: ……

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

CONTENTS

INTRODUCTION

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 companiesthrough 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 smallinitial 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 submitthem 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 manyreal-world applications such as weather forecasting and financial market prediction. It usesthe 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 commonalgorithms 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

COLLEGE PROFILE

Dr B.C. Roy Engineering College is located at the heart of Durgapur. One of thegrowing industrial hubs in the country. It offers courses that confirm to the largestrequirements and syllabus structured by MAKAUT formally known as WBUT. It has affiliation & approved of most reputed Indian names:

1. Approved by AICTE, New Delhi
2. NBA accredited for three programmes
3. Affiliated to Maulana Abul Kalam Azad University of technology(MAKAUT), formally known as WBUT.
4. Approved by Department of Higher Education, Govt. Of West Bengal.
5. ISO: 9001-2008 certified institution.

The vision of the transform the institution into global centre of learning through synergic of creativity, innovation & discipline.

The mission of the organisation is summarized as follows:

1. Create ideal ambience for learning & growth.
2. Help student inherit value-systems, be creative & agile thinkers.
3. Establish discipline; value added education & training & placement.
4. Develop a symbiotic relationship between the institution, society & community, for mutual betterment.

OBJECTIVES

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.

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).

SYSTEM ANALYSIS

DEFINITION

System Analysis is the process of studying a procedure in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

A systems analyst researches problem, plans solutions, recommends software and systems, and coordinates development to meet business or other requirements. The main goal of this system analyst is to collect different data from different site, process these data and generate progress as well as daily report.

System analyst operates in a dynamic environment where change is a way of life. The environment may be a business firm, a business application, or a computer system. to construct a system the following key elements must be considered: - Input: Input is what data the system receives to produce a certain output.

Output: What goes out from the system after being processed is known as Output.

Processing: The process involved to transform input into output is known as Processing.

Control: In order to get the desired results it is essential to monitor and control the input, Processing and the output of the system. This job is done by the control. Feedback: The Output is checked with the desired standards of the output set and the necessary steps are taken for achieving the output as per the standards, this process is called as Feedback. It helps to achieve a much better control in the system.

Boundaries: The boundaries are nothing but the limit of the system. Setting up boundaries helps for better concentration of the actives carried in the system.

Environment: The things outside the boundary of the system are known as environment. Change in the environment affects the working of the system.

Interfaces: The interconnections and the interactions between the sub-systems are known as the Interfaces. They may be inputs and outputs of the systems.

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 on tick 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 Zhang Liuqing 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.

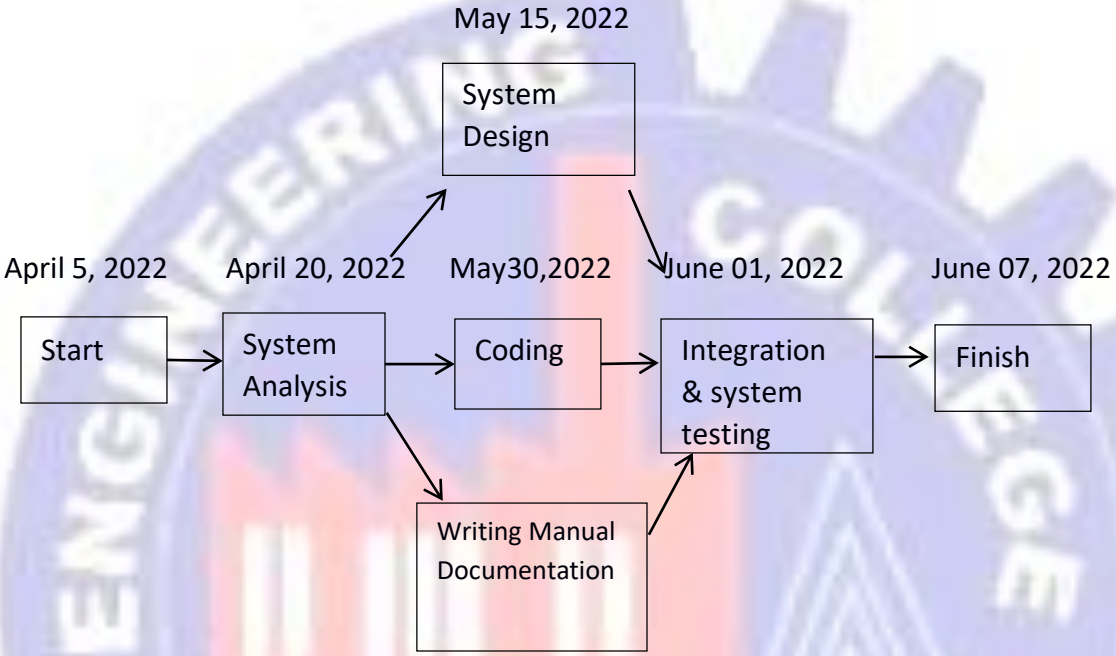
2.3 Pert Chart

A PERT chart is a network of boxes and arrows. There are different variation pertcharts. Some uses the boxes to represent the activities, and some uses the arrowsto do so. Each box thus represents the activity. The arrows are used to represent the dependencies of the activities on one another activity on the head of the arrowcannot start until the activity at the tail of the arrow is finished. Just s in the nodesin work breakdown structure the boxes in work in a PERT chart can be decoratedwith starting and ending dates.

For big projects a PERT chart may contain hundreds of nodes and span many pages. The availability of computer support is essential for managing large PERTnetwork. Many such tools are available for most computers. Any delay in any activity in this path will clearly want to

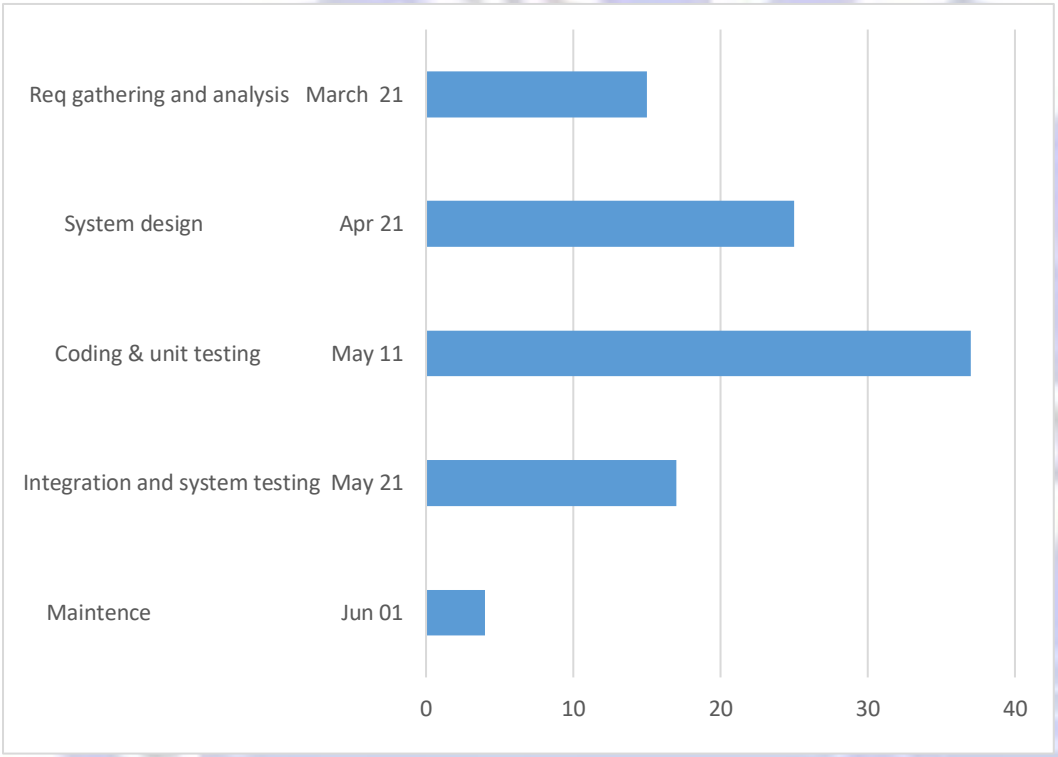
monitor these activities. On the critical path much more closely than other activity.

As far my project is concerned the total project is done by three months (approx.)from APRIL, 2022 - JUNE, 2022.



2.4 Gantt chart

Gantt chart was invented by a mechanical engineer named Henry Gantt in 1910. A Gantt chart is simply a type of bar chart that visually represents a project plan over time. It shows start and end dates for tasks, displays milestones, and allows for dependencies between tasks. With all the features of Henry Gantt's project management system, it's no wonder that even now, more than 100 years later, the Gantt chart is still the preferred tool for managing projects of all sizes and types.



Hardware Requirements:

•RAM: 4 GB

•Storage: 500 GB

•CPU: 2 GHz or faster

•Architecture: 32-bit or 64-bit

Software Requirements:

•Python 3.9 in PyCharm 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.

System Design & Development

3.1PROPOSED 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).

3.2 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%.

3.3 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.

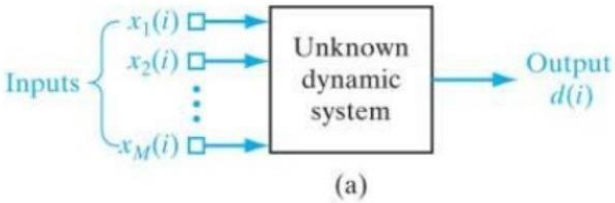


Fig. 1: LMS Inputs and Outputs

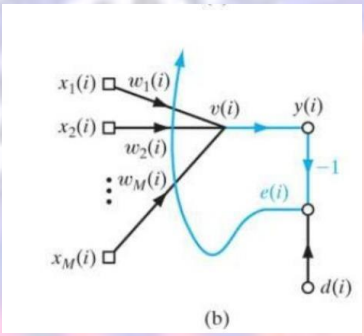


Fig 2: LMS updating weights

Algorithm 1: LMS

Input:
x : input vector
d: desired vector
μ: learning rate
N: filter order
Output:
y: filter response
e: filter error
begin
  $M = size(x)$;
  $x_n(0) = w_n(0) = [0\ 0\ \ldots\ 0]^T$;
  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.

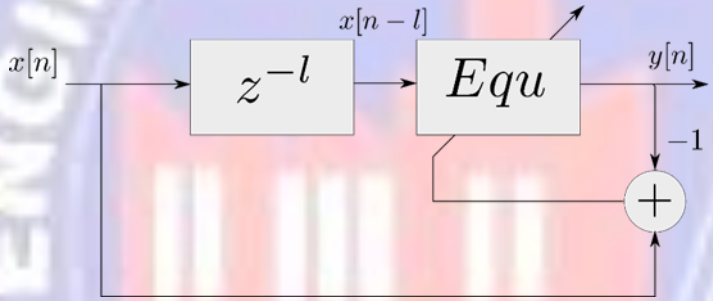


Fig. 3: LMS updating weights

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

μ: 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

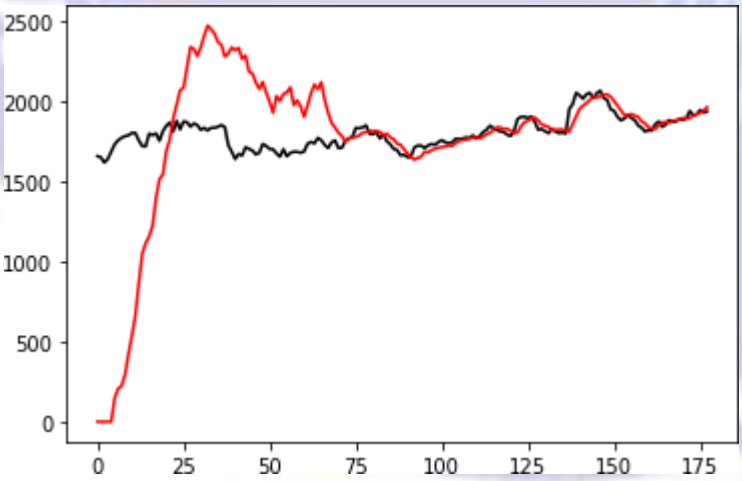
$e(n) = d(n) - y(n)$;

end

$w_{n+1} = w_n + 2\mu e(n) x_n$;

end

end



LSTM ARCHITECTURE

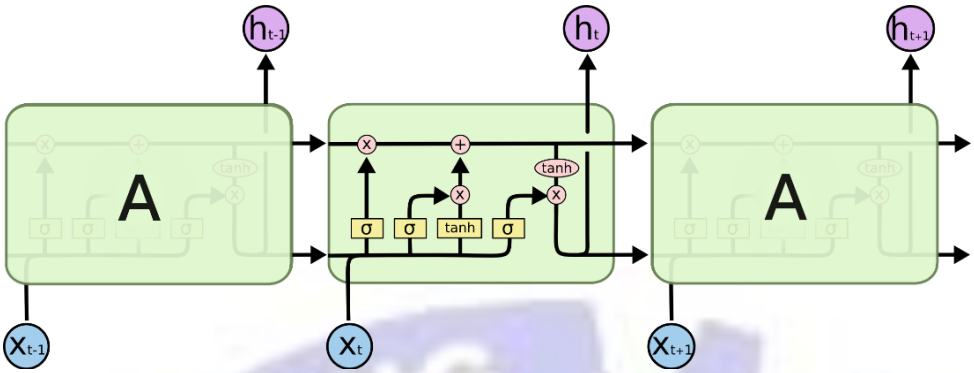
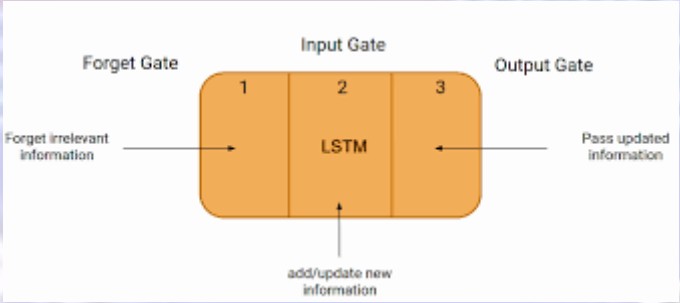


Fig. 4: LSTM Architecture

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.

OUTPUT GATE

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
•	Inputs: dataset
•	Outputs: forecasted data
•
•	# Split dataset into 75% training and 25% testing data
•	size = length(dataset) * 0.75
•	train = dataset [0 to size]
•	test = dataset [size to length(dataset)]
•
•	# Procedure to fit the LSTM model
•	Procedure LSTMAlgorithm (train, test, train_size, epochs)
•	X = train
•	y = test
•	model = Sequential ()
•	model.add(LSTM(50), stateful=True)
•	model.compile(optimizer='adam', loss='mse')
•	model.fit(X, y, epochs=epochs, validation_split=0.2)
•	return model
•
•	# Procedure to make predictions
•	Procedure getPredictonsFromModel (model, X)
•	predictions = model.predict(X)
•	return predictions
•
•	epochs = 100
•	neurons = 50
•	predictions = empty
# Fit the LSTM model
model = LSTMAlgorithm (train, epoch, neurons)
# Make predictions
pred = model.predict(train)

# Validate the model n = len(dataset)

error = 0
for i in range(n): error += (abs(real[i] - pred[i])/real[i]) * 100 accuracy = 100 - error/n
```

3.1.1SYSTEM ARCHITECTURE

1)Preprocessing of data

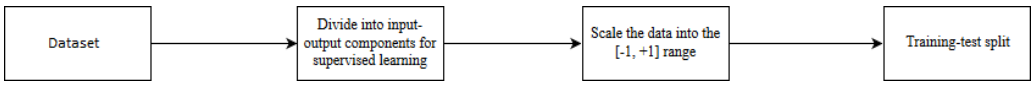


Fig. 5: Pre-processing of data

2)Overall Architecture

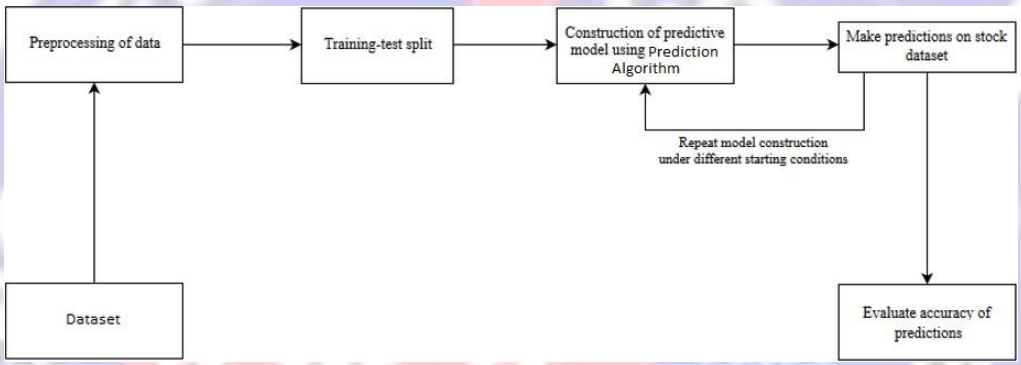


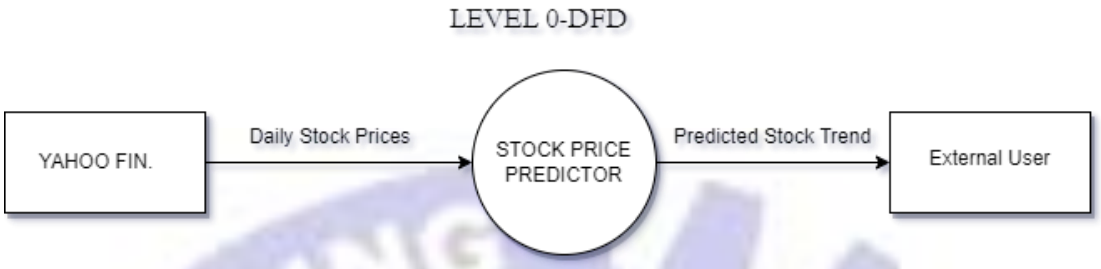
Fig. 6: Overall Architecture

3.4Data Flow Diagram (DFD)

Data flow diagram is graphical representation of flow of data in an information system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled.

3.4.1 DFD 0 Level

The 0 Level DFD shows flow of data of application. DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analysed or modelled.



3.4.2 DFD 1 Level

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. This DFD describes main functions carried out by the system, as we break down the high-level process of the Context Diagram into its sub-processes

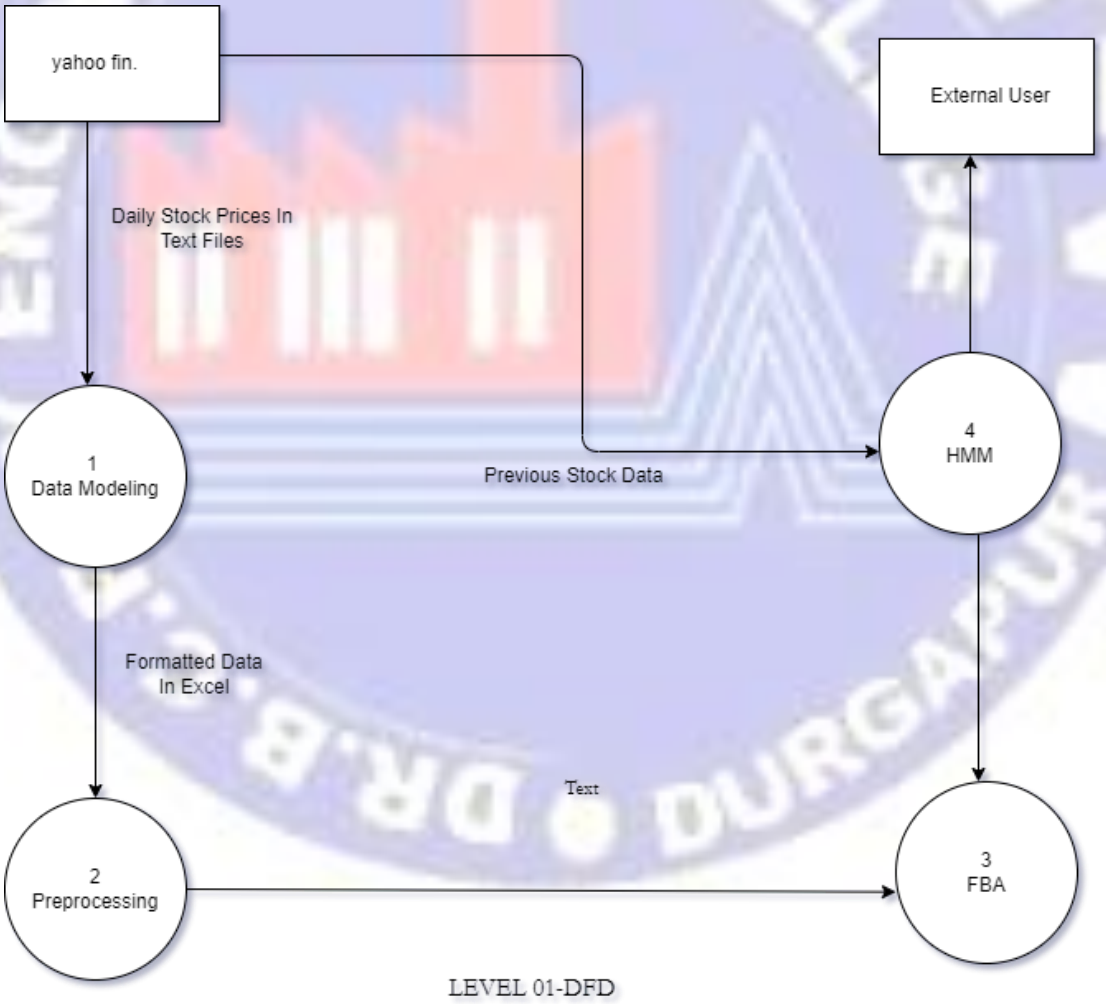


Fig. 7: Level 01 DFD

3.5 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.

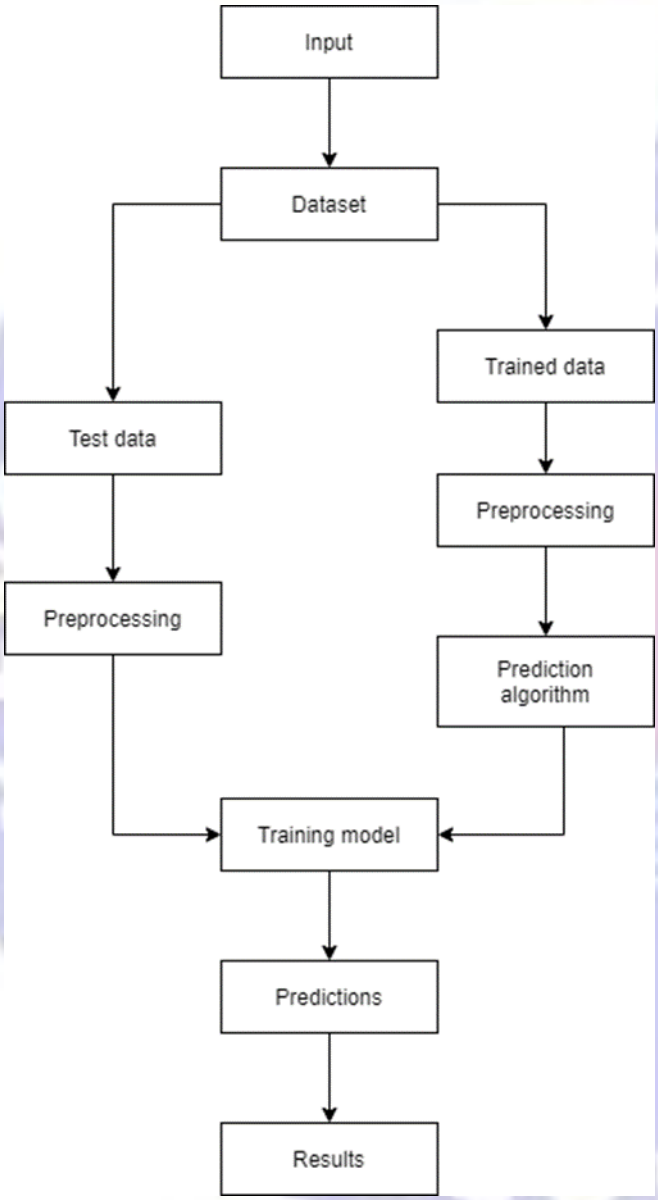


Fig. 8: Training and prediction

3.6 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.

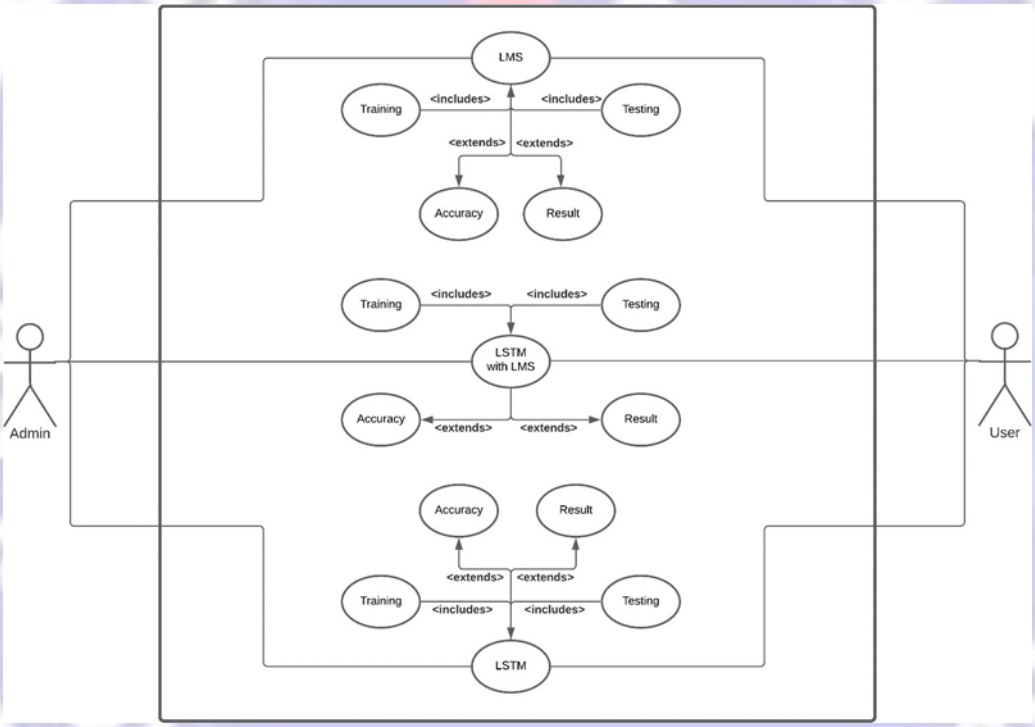


Fig. 9: Using LMS, LSTM and LSTM with LMS in the system

3.7 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.

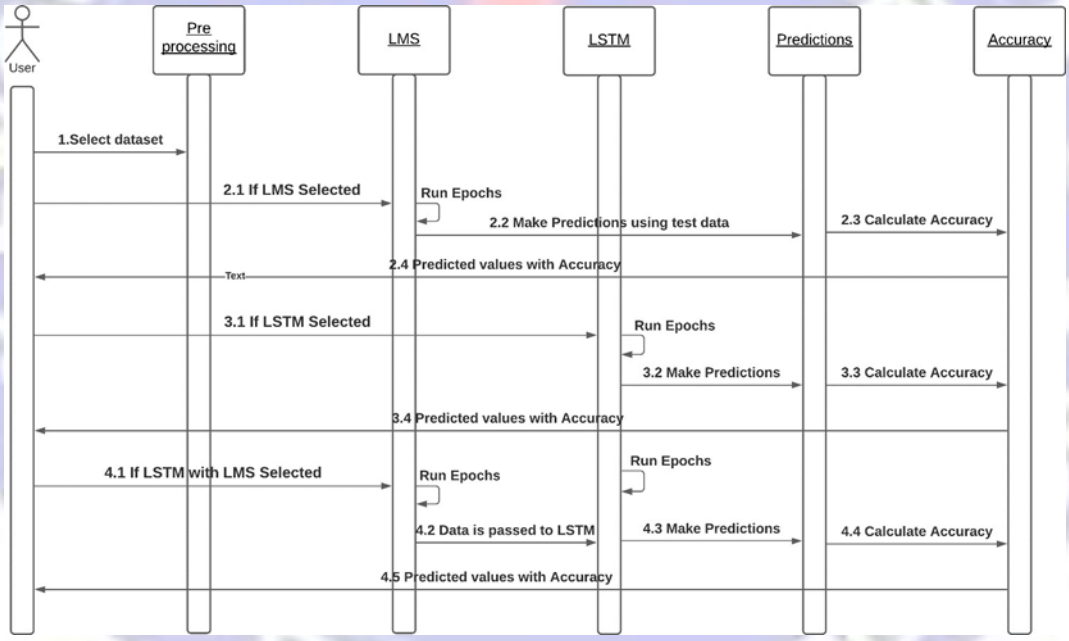


Fig. 10: Execution based on model selection

3.8Activity 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.

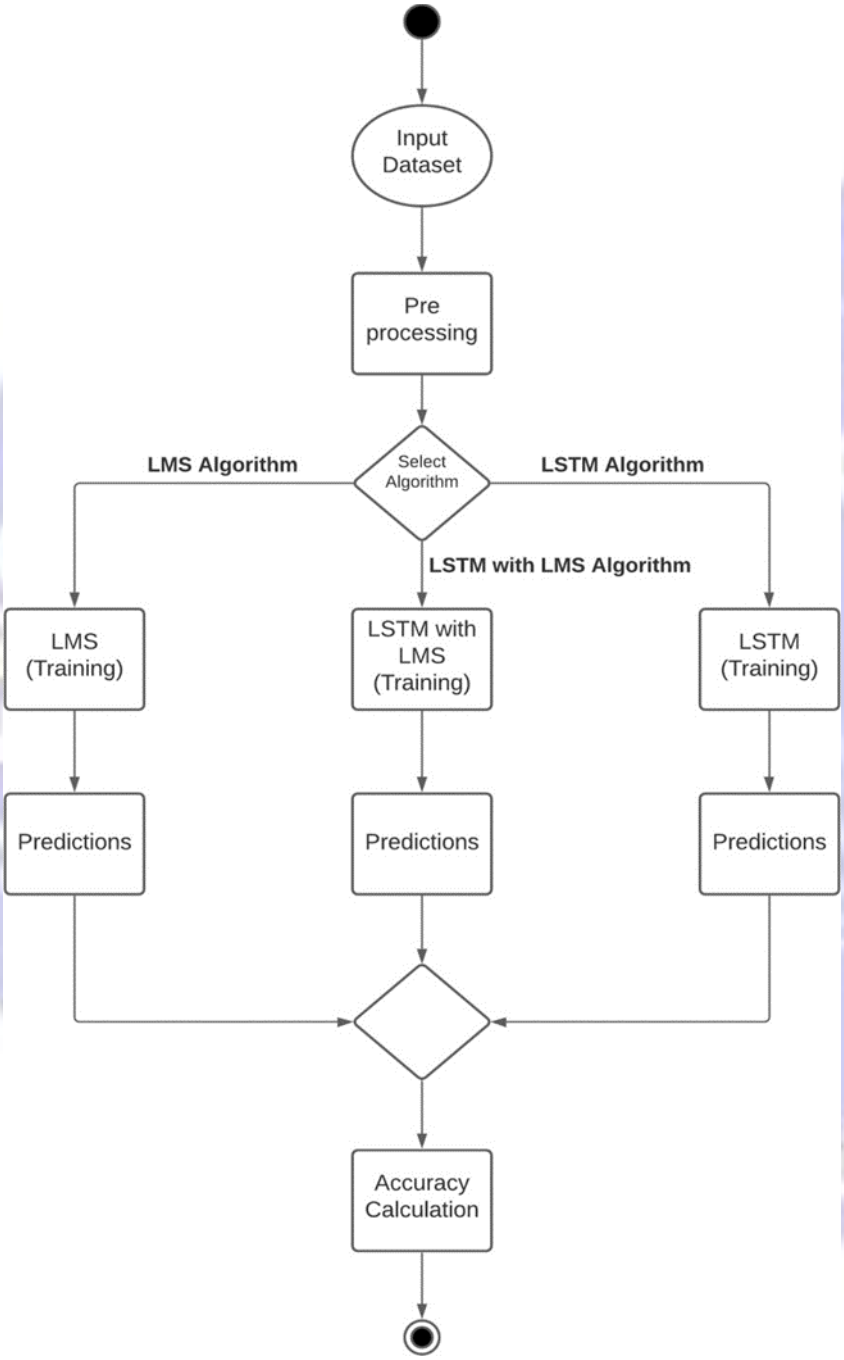


Fig. 11: Execution based on algorithm selection

3.9 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.

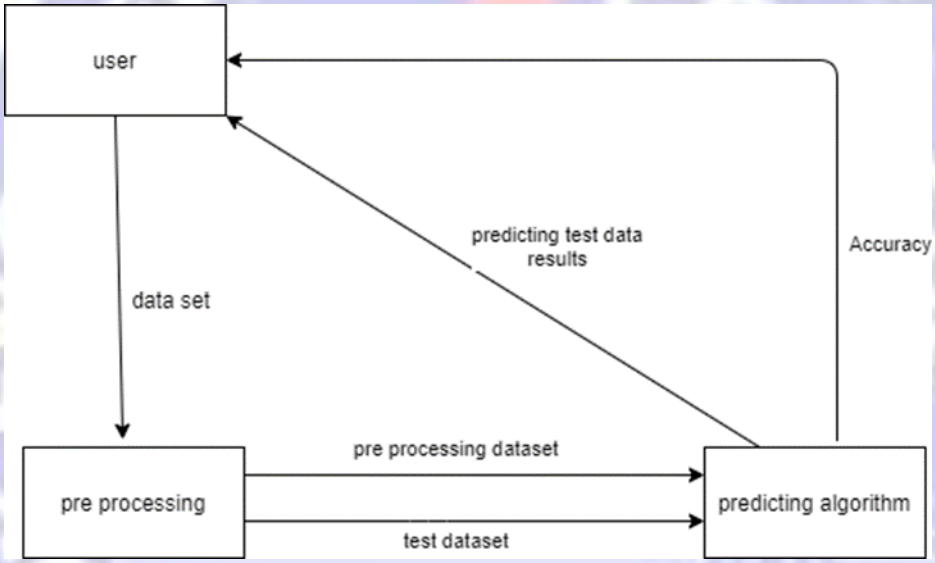


Fig. 12: Data transfer between modules

3.10 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.

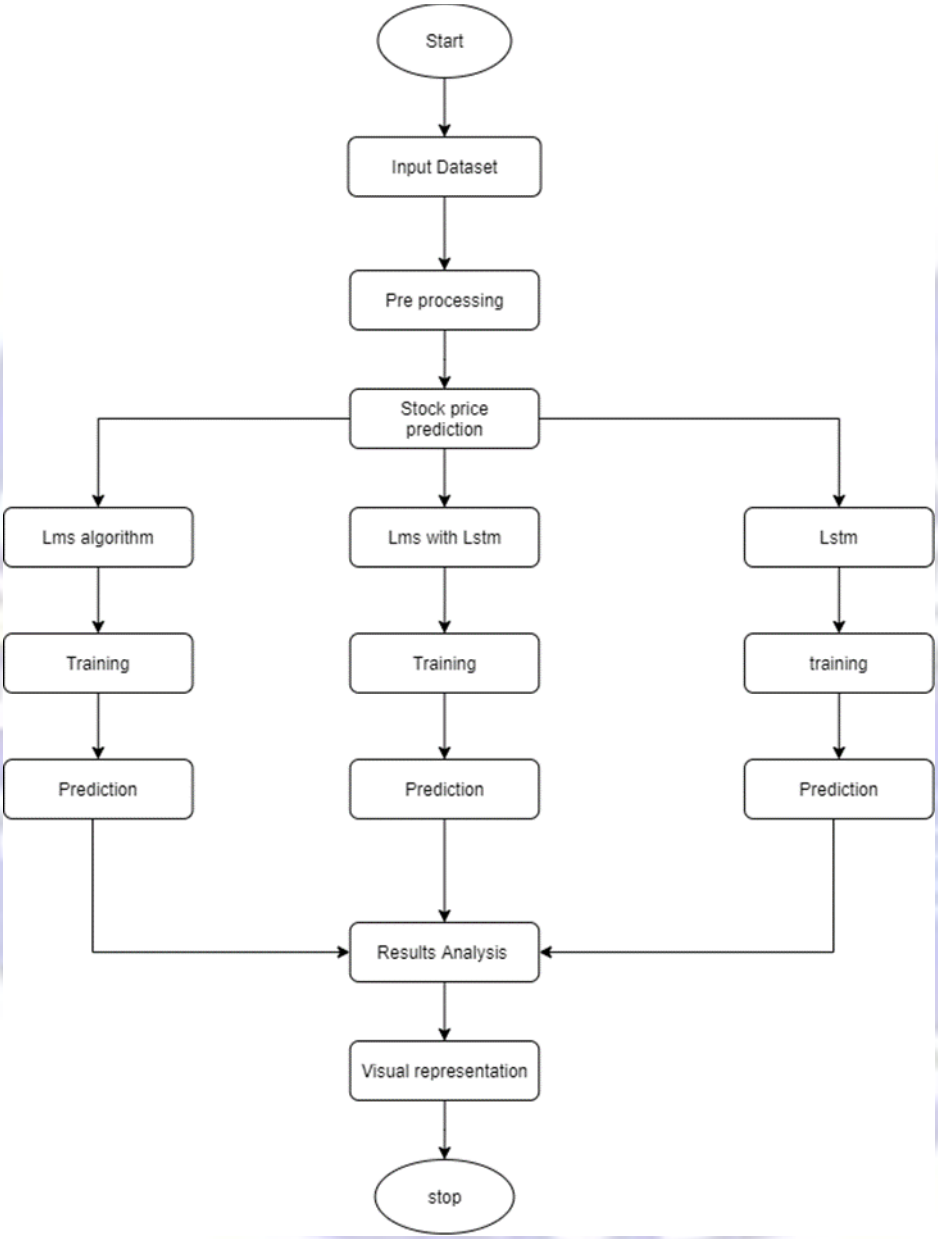


Fig. 13: Flow of execution

3.11 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.

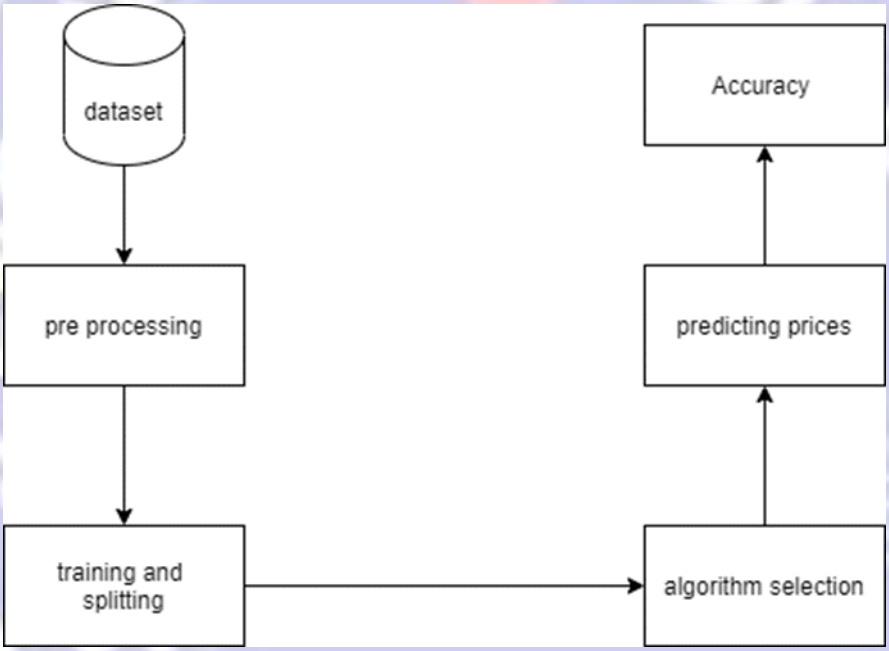


Fig. 14: Components present in the system

System Testing and Implementation

4.TESTING

4.1INTRODUCTION

The following describes the level of team interaction necessary to have a successful product. The Test Team will work closely with the Development Team to achieve a high quality design and user interface specification based on customer requirements. The Test Team is responsible for visualizing test cases and raising quality issues and concerns during meeting to address issues early enough in the development cycle. The Test Team will work closely with The Development Team to determine whether or not the application meets standards for completeness. If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the

area. Since the application interacts with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

4.2Types of Testing

Unit Testing:-

Unit Testing is done at the source or code level language-specific programming errors such as bad syntax, logical errors or to test particular functions or code modules. The unit testing cases shall be designed to test the validity of the programs correctness.

White Box Testing:-

In white box testing, the UI is by passed. Inputs and outputs are tested directly at the code level and the result are compared against specifications. This forms of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply. Each function of the binary tree repository is executed independently, therefore a program flow for each function has been derived from the code.

Black Box Testing:-

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

System Testing:-

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress and configuration sensitivity. But in our case well focus only on function validation and performance. And in both cases we will use the black box method of testing.

4.3 TESTING PLAN

Type of test	**Will the test be performed**	**Comments/explanation**	**Software component**
Requirements testing	Yes	Needs to be done to cope up with changing environment	Fluctuation in the share market.
Unit	Yes	Maximum number of defects are found. Each component of code was tested or analysed accordingly not only to ensure the best quality of the developed software but also to make sure that code behaves in the same way as it was intended to. Unit testing was performed as and when the component was developed.	• Package Import code • CSV upload code • Training code • Validation code • Prediction of the day code
Integration	Yes	All the well-developed sub-system are integrated together and tested called as integration testing	

Performance	Yes	Performance is the major criteria for evaluating any type of the system. It holds importance and is tested likewise.	Performance of different Algorithms is measured in combination. Algorithms are: • LSTM Algorithms • LSM Algorithms
Stress	No	-	-
Compliance	No	-	-
Security	NO	-	-

Table 1: Testing Plan

TEST TEAM DETAILS		
Role	**Name**	**Responsibility**
Testing In charge	Agni Rakshit Soumallya Dey	To perform requirements, unit, integration, performance and load testing.

Table 2: Test Team details

Activity	**Start Date**	**Completion Date**	**Hours**	**Comments**
Develop input	15-04-2022	20-05-2022	6	Nominal and trivial issues tested using the standard test cases designed for the system.
Test Region Setup	23-05-2022	08-06-2022	10	Test region is no defined to check all the features individually as well as in combination

Table 3: Testing Schedule

Test Environment
Software Items
• Windows 11 • Internet connection • Python 3.10 • PyCharm
Hardware Items
• Personal Computer/Laptop • Network Interface card • Wireless connection or connecting cable

Table 4 : Test Environment

4.4 COMPONENT TESTING

S.No	Components that require testing	Type of testing required	Technique for writing test case
1	CSV to XLs converter code	Unit testing	White Box Testing
2	Model code	Unit testing	White Box Testing
3	Graph code	Unit testing	White Box Testing
4	sgenerator code	Unit testing	White Box Testing
5	Destination	System testing	Black Box Testing
6	Source	System testing	Black Box Testing

Table 5: Component decomposition and identification of tests required

4.5 Test cases

Test ID	T1
Input	Enter the starting and the last date to update the data
Expected Output	Data fetched from yahoo finance
Status	Pass

Test ID	T2
Input	Predict the stock rate for the very next day
Expected Output	We get low, high, opening and closing for the next day
Status	Pass

Test ID	T3
Input	Predict the stock rate for any day after a week from the given set of data
Expected Output	Enter a date within a week
Status	Pass

Test ID	T4
Input	Check the precision of output by entering a date whose values are already known
Expected Output	Outputs are Almost precise
Status	Pass

4.6ERROR AND EXCEPTION HANDLING

Test Case Id	Test Case	Debugging Technique
T1	Fetching data from yahoo finance	Check the posmdownloader code and check for errors

4.7 LIMITATION OF THE SOLUTION

- The precision of the output is not even near to the actual value.
- System sometimes hand due to loss of connection to Internet.

Conclusion & Future Work

5. CONCLUSION AND FUTURE WORK

5.1Conclusion

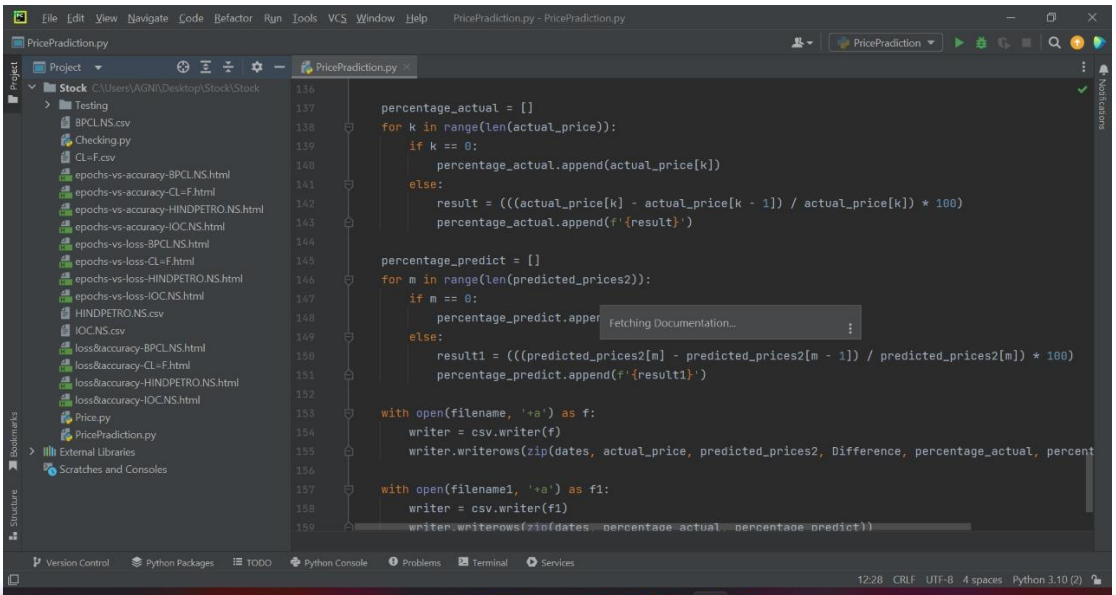
In this project, we are predicting closing stock price of any given organization, we developed a web application for predicting close stock price using LMS and LSTM algorithms for prediction. We have applied datasets belonging to ioc, bpcl, cl=f, hindpetro Stocks and achieved above 95% accuracy for these datasets.

5.2Future work

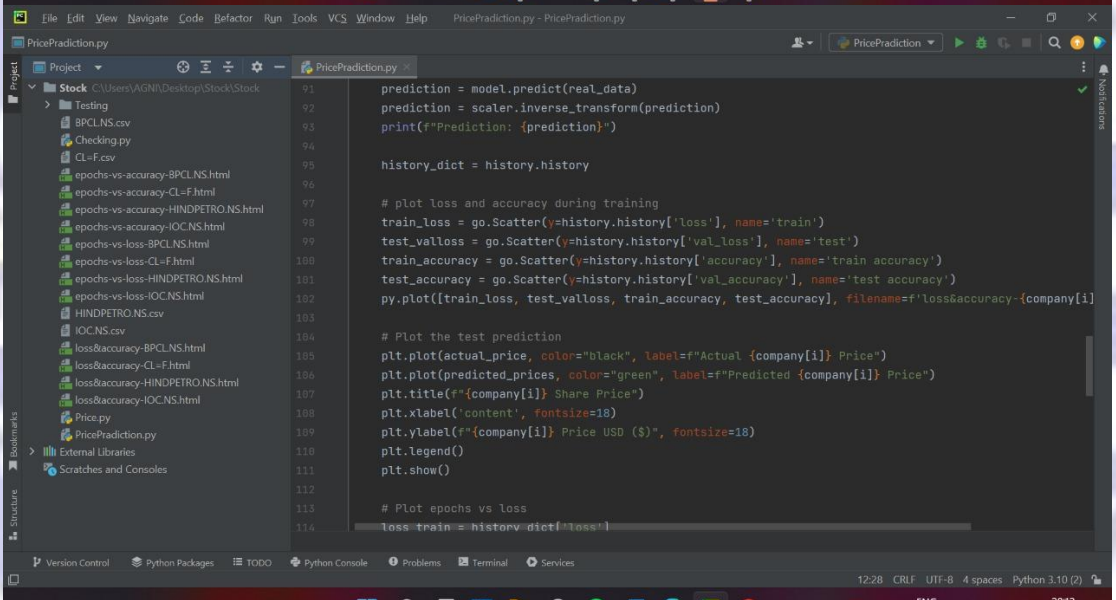
- We want to extend this application for predicting cryptocurrency trading.
- We want to add sentiment analysis for better analysis.

Appendix

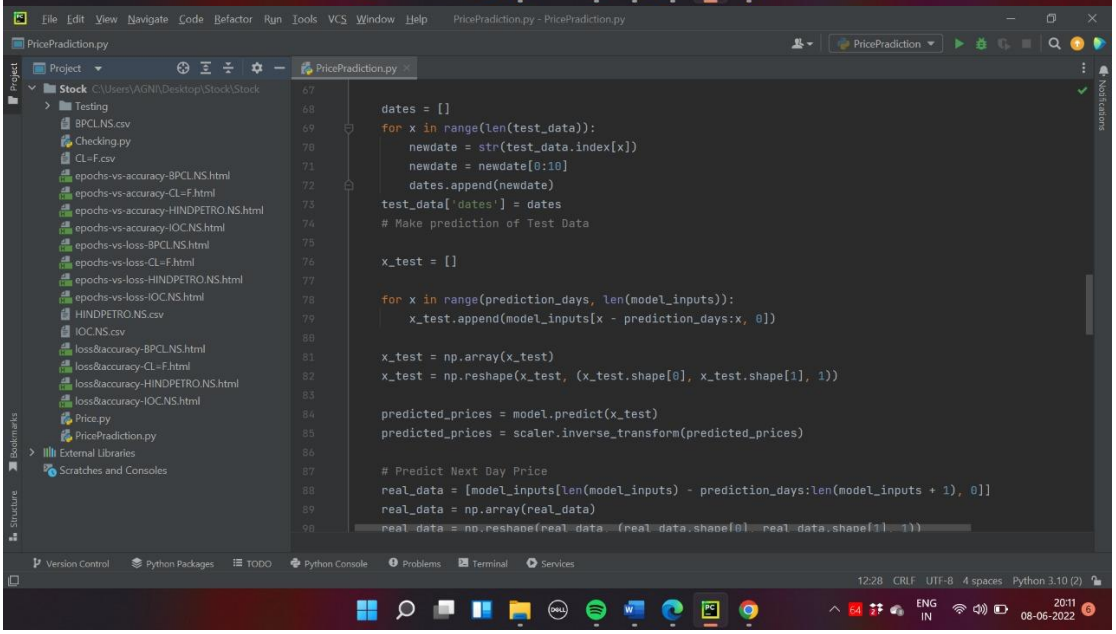
A.1 SAMPLE CODE



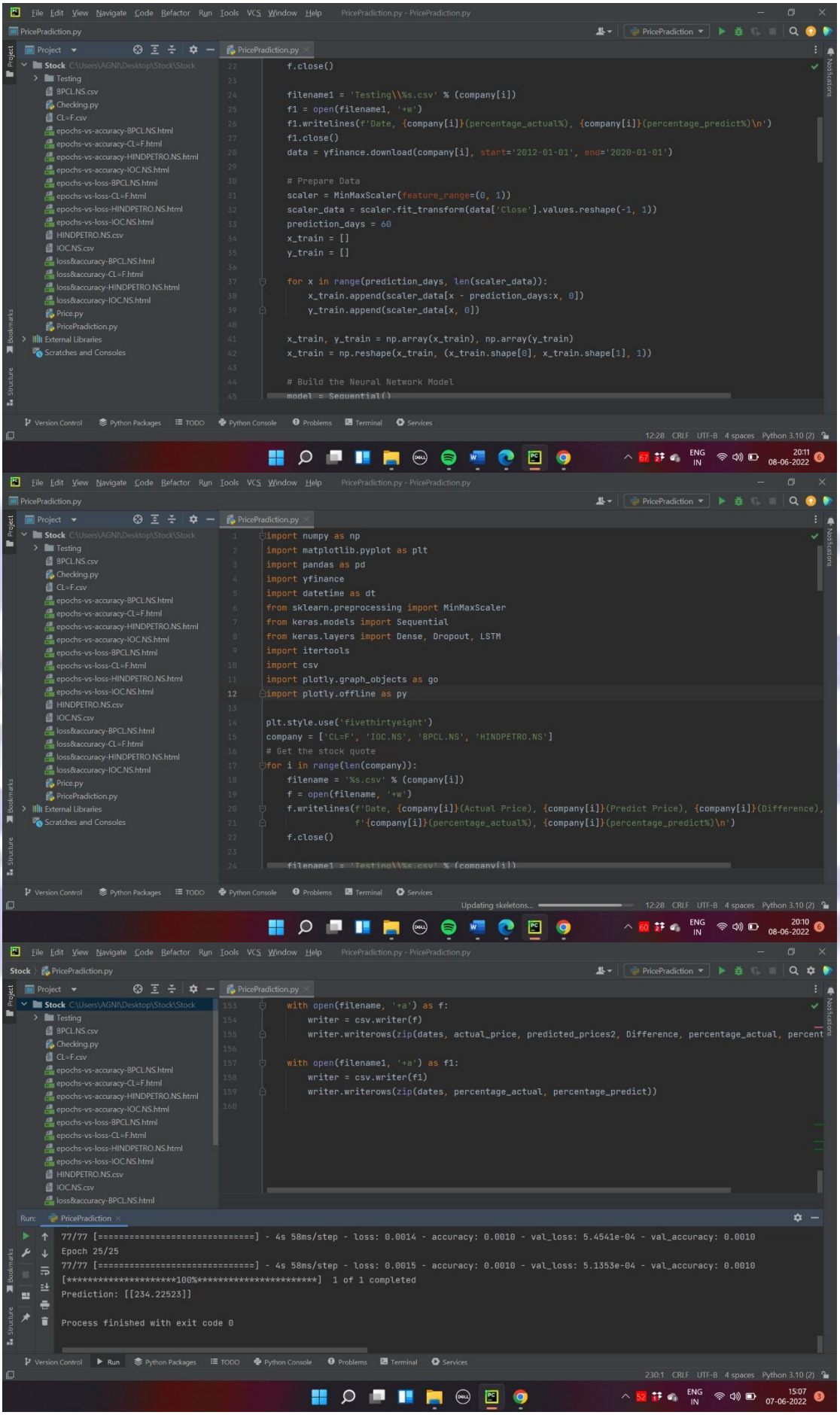
```
    percentage_actual = []
    for k in range(len(actual_price)):
        if k == 0:
            percentage_actual.append(actual_price[k])
        else:
            result = (((actual_price[k] - actual_price[k - 1]) / actual_price[k]) * 100)
            percentage_actual.append(f'{result}')

    percentage_predict = []
    for m in range(len(predicted_prices2)):
        if m == 0:
            percentage_predict.apper
        else:
            result1 = (((predicted_prices2[m] - predicted_prices2[m - 1]) / predicted_prices2[m]) * 100)
            percentage_predict.append(f'{result1}')

    with open(filename, '+a') as f:
        writer = csv.writer(f)
        writer.writerows(zip(dates, actual_price, predicted_prices2, Difference, percentage_actual, percent

    with open(filename1, '+a') as f1:
        writer = csv.writer(f1)
        writer.writerows(zip(dates, percentage_actual, percentage_predict))
```



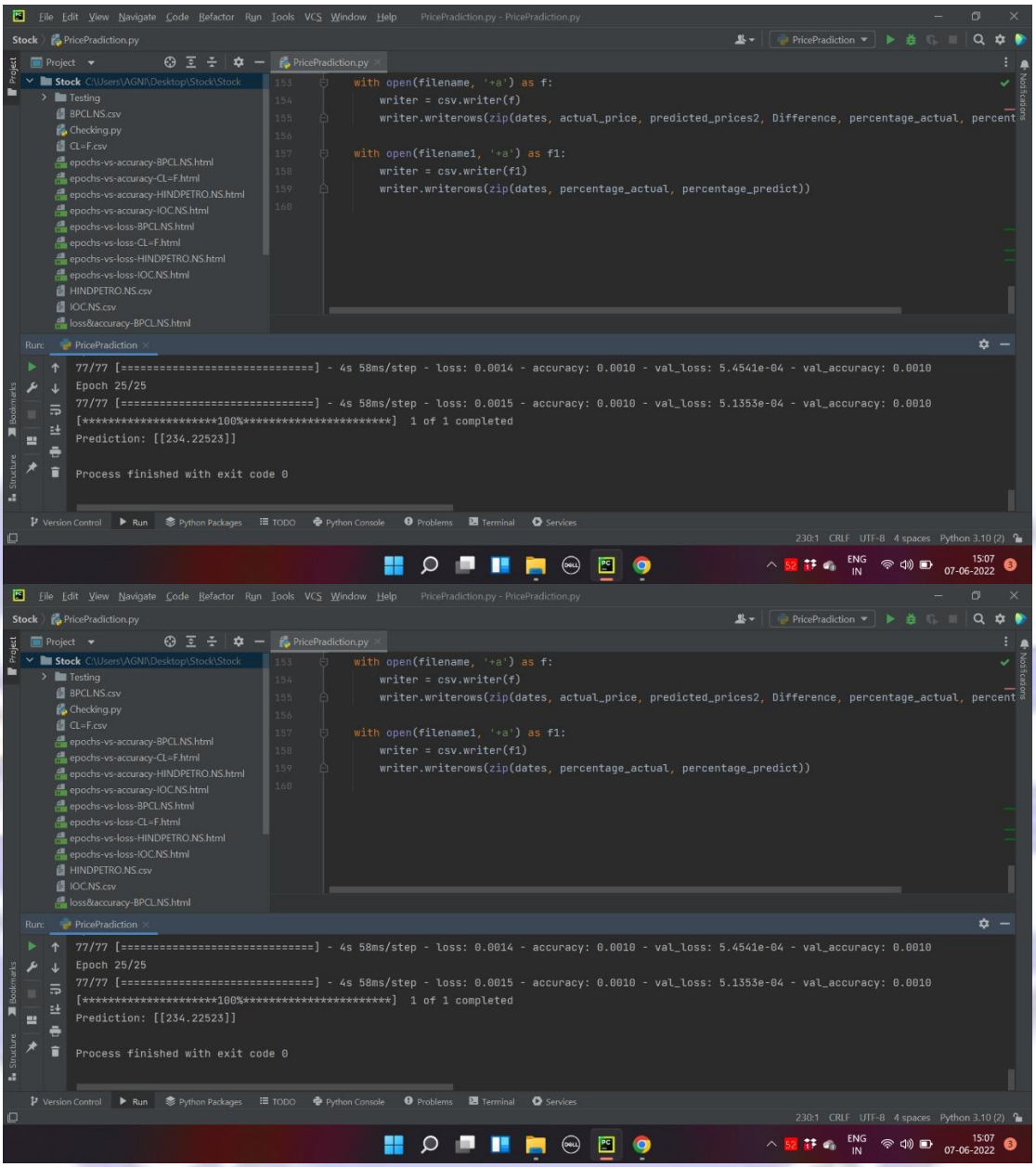
```
    prediction = model.predict(real_data)
    prediction = scaler.inverse_transform(prediction)
    print(f"Prediction: {prediction}")

    history_dict = history.history

    # plot loss and accuracy during training
    train_loss = go.Scatter(y=history.history['loss'], name='train')
    test_valloss = go.Scatter(y=history.history['val_loss'], name='test')
    train_accuracy = go.Scatter(y=history.history['accuracy'], name='train accuracy')
    test_accuracy = go.Scatter(y=history.history['val_accuracy'], name='test accuracy')
    py.plot([train_loss, test_valloss, train_accuracy, test_accuracy], filename=f'loss&accuracy-{company[i]

    # Plot the test prediction
    plt.plot(actual_price, color="black", label=f"Actual {company[i]} Price")
    plt.plot(predicted_prices, color="green", label=f"Predicted {company[i]} Price")
    plt.title(f"{company[i]} Share Price")
    plt.xlabel('content', fontsize=18)
    plt.ylabel(f"{company[i]} Price USD ($)", fontsize=18)
    plt.legend()
    plt.show()

    # Plot epochs vs loss
    loss_train = history_dict['loss']
```



```
    dates = []
    for x in range(len(test_data)):
        newdate = str(test_data.index[x])
        newdate = newdate[0:10]
        dates.append(newdate)
    test_data['dates'] = dates
    # Make prediction of Test Data

    x_test = []

    for x in range(prediction_days, len(model_inputs)):
        x_test.append(model_inputs[x - prediction_days:x, 0])

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

    predicted_prices = model.predict(x_test)
    predicted_prices = scaler.inverse_transform(predicted_prices)

    # Predict Next Day Price
    real_data = [model_inputs[len(model_inputs) - prediction_days:len(model_inputs + 1), 0]]
    real_data = np.array(real_data)
    real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))
```


f.close()
filename1 = 'Testing\\%s.csv' % (company[i])
f1 = open(filename1, '+w')
f1.writelines(f'Date, {company[i]}(percentage_actual%), {company[i]}(percentage_predict%)\n')
f1.close()
data = yfinance.download(company[i], start='2012-01-01', end='2020-01-01')
# Prepare Data
scaler = MinMaxScaler(feature_range=(0, 1))
scaler_data = scaler.fit_transform(data['Close'].values.reshape(-1, 1))
prediction_days = 60
x_train = []
y_train = []
for x in range(prediction_days, len(scaler_data)):
x_train.append(scaler_data[x - prediction_days:x, 0])
y_train.append(scaler_data[x, 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))
# Build the Neural Network Model
model = Sequential()
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import yfinance
import datetime as dt
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM
import itertools
import csv
import plotly.graph_objects as go
import plotly.offline as py
plt.style.use('fivethirtyeight')
company = ['CL=F', 'IOC.NS', 'BPCL.NS', 'HINDPETRO.NS']
# Get the stock quote
for i in range(len(company)):
filename = '%s.csv' % (company[i])
f = open(filename, '+w')
f.writelines(f'Date, {company[i]}(Actual Price), {company[i]}(Predict Price), {company[i]}(Difference),
f'{company[i]}(percentage_actual%), {company[i]}(percentage_predict%)\n')
f.close()
filename1 = 'Testing\\%s.csv' % (company[i])
with open(filename, '+a') as f:
writer = csv.writer(f)
writer.writerows(zip(dates, actual_price, predicted_prices2, Difference, percentage_actual, percent
with open(filename1, '+a') as f1:
writer = csv.writer(f1)
writer.writerows(zip(dates, percentage_actual, percentage_predict))
77/77 [==============================] - 4s 58ms/step - loss: 0.0014 - accuracy: 0.0010 - val_loss: 5.4541e-04 - val_accuracy: 0.0010
Epoch 25/25
77/77 [==============================] - 4s 58ms/step - loss: 0.0015 - accuracy: 0.0010 - val_loss: 5.1353e-04 - val_accuracy: 0.0010
[**********100%**********] 1 of 1 completed
Prediction: [[234.22523]]
Process finished with exit code 0

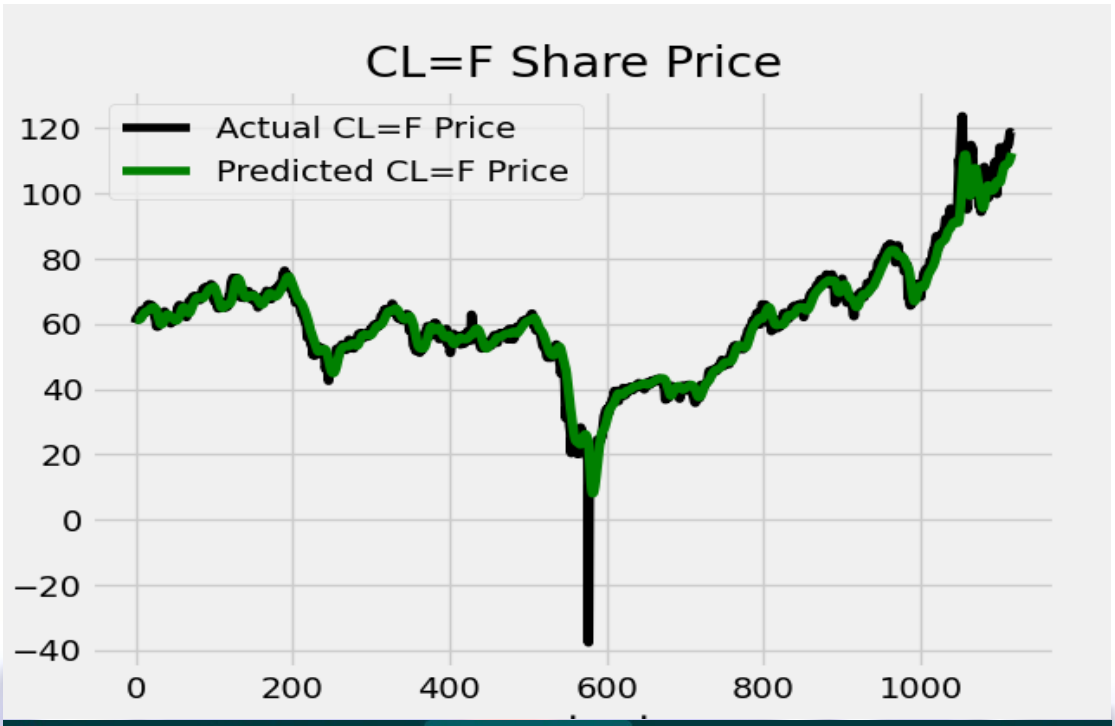
A.2 SCREEN SHOTS

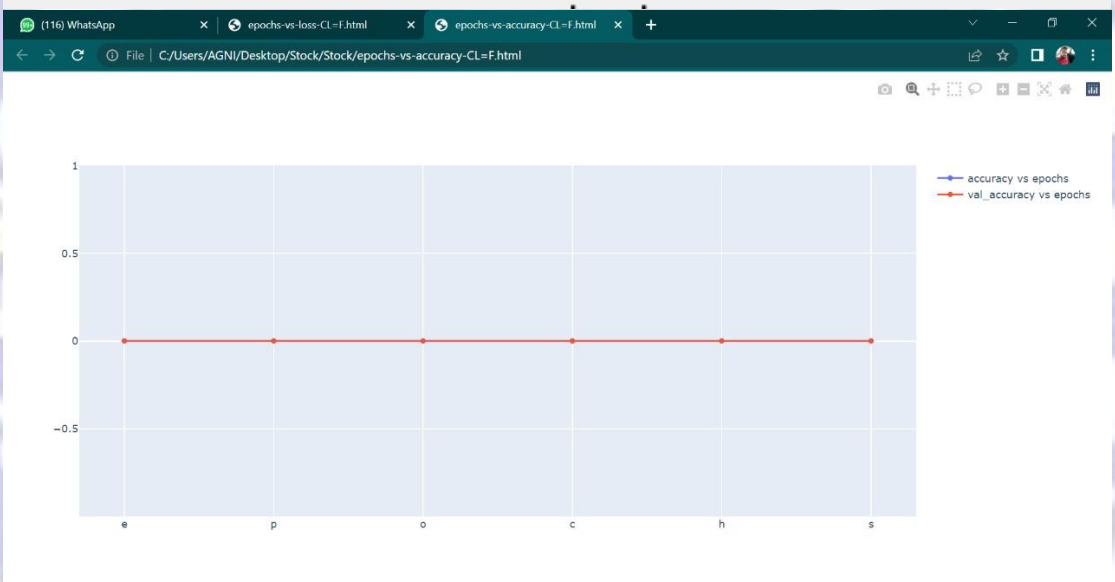


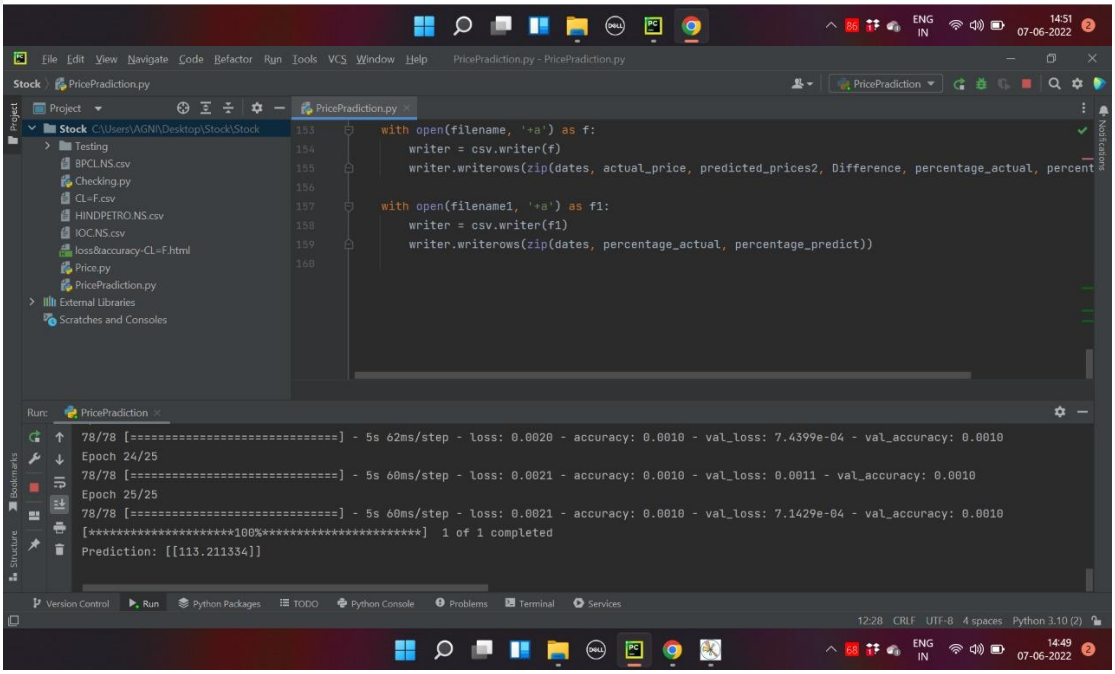
File Edit View Navigate Code Refactor Run Tools VCS Window Help
PricePradiction.py - PricePradiction.py
Stock
PricePradiction.py
PricePradiction
Project
Stock C:\Users\AGNI\Desktop\Stock\Stock
Testing
BPCL.NS.csv
Checking.py
CL=F.csv
epochs-vs-accuracy-BPCL.NS.html
epochs-vs-accuracy-CL=F.html
epochs-vs-accuracy-HINDPETRO.NS.html
epochs-vs-accuracy-IOC.NS.html
epochs-vs-loss-BPCL.NS.html
epochs-vs-loss-CL=F.html
epochs-vs-loss-HINDPETRO.NS.html
epochs-vs-loss-IOC.NS.html
HINDPETRO.NS.csv
IOC.NS.csv
loss&accuracy-BPCL.NS.html
with open(filename, '+a') as f:
writer = csv.writer(f)
writer.writerows(zip(dates, actual_price, predicted_prices2, Difference, percentage_actual, percent
with open(filename1, '+a') as f1:
writer = csv.writer(f1)
writer.writerows(zip(dates, percentage_actual, percentage_predict))
Run: PricePradiction
77/77 [==========] - 4s 58ms/step - loss: 0.0014 - accuracy: 0.0010 - val_loss: 5.4541e-04 - val_accuracy: 0.0010
Epoch 25/25
77/77 [==========] - 4s 58ms/step - loss: 0.0015 - accuracy: 0.0010 - val_loss: 5.1353e-04 - val_accuracy: 0.0010
[**********100%**********] 1 of 1 completed
Prediction: [[234.22523]]
Process finished with exit code 0
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
230:1 CRLF UTF-8 4 spaces Python 3.10 (2)
ENG IN
15:07 07-06-2022

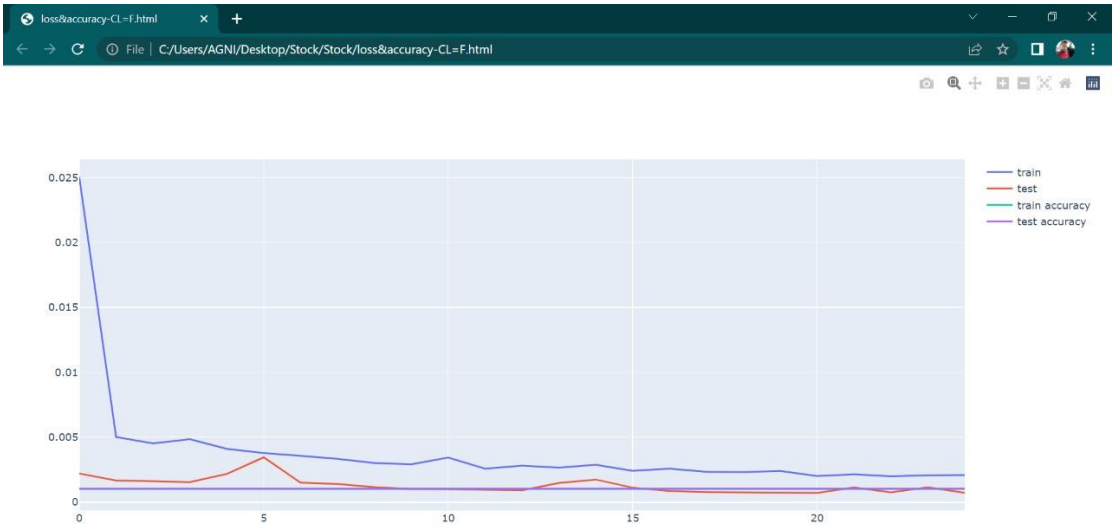

(116) WhatsApp
loss&accuracy-HINDPETRO.NS.h
epochs-vs-loss-HINDPETRO.NS.h
epochs-vs-accuracy-HINDPETRO
File | C:/Users/AGNI/Desktop/Stock/Stock/epochs-vs-accuracy-HINDPETRO.NS.html
accuracy vs epochs
val_accuracy vs epochs
1
0.5
0
−0.5
e
p
o
c
h
s

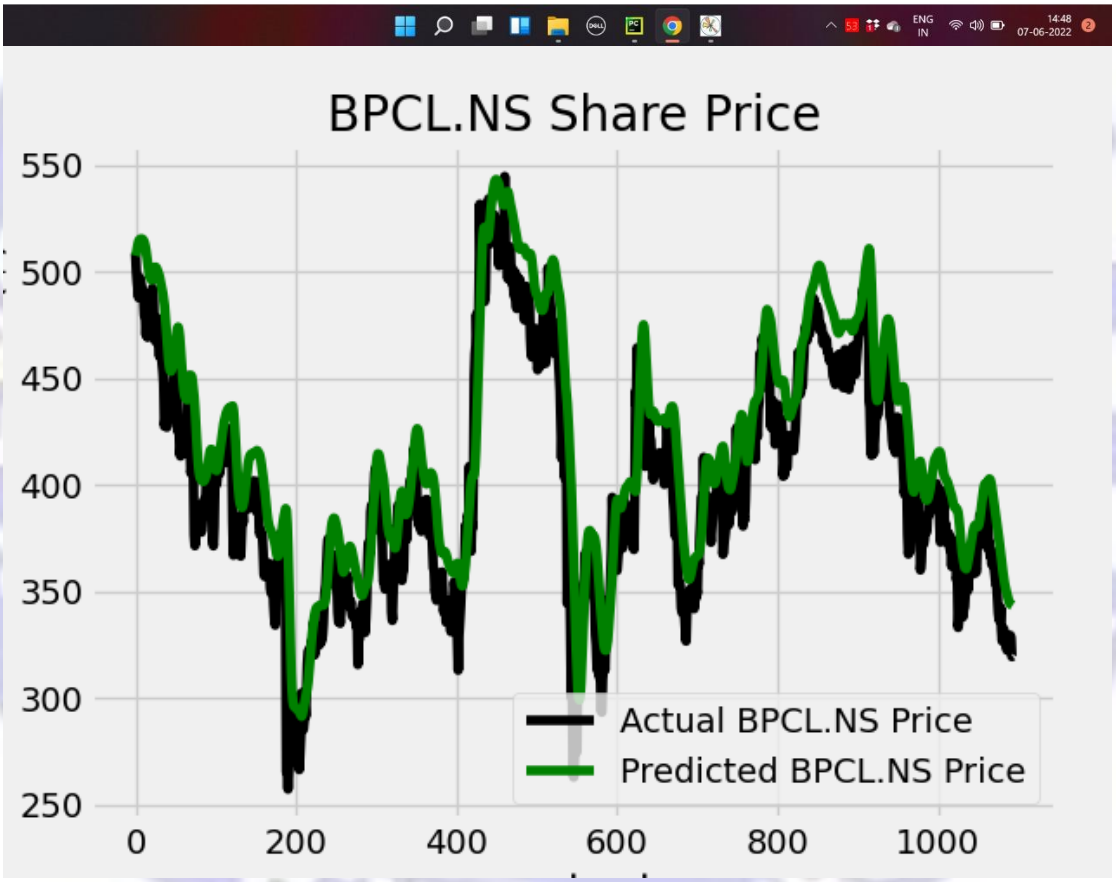

(116) WhatsApp
loss&accuracy-HINDPETRO.NS.h
epochs-vs-loss-HINDPETRO.NS.h
epochs-vs-accuracy-HINDPETRO
File | C:/Users/AGNI/Desktop/Stock/Stock/epochs-vs-loss-HINDPETRO.NS.html
loss vs epochs
val_loss vs epochs
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
e
p
o
c
h
s
ENG
IN
15:06
07-06-2022


CL=F Share Price
Actual CL=F Price
Predicted CL=F Price
120
100
80
60
40
20
0
−20
−40
0
200
400
600
800
1000

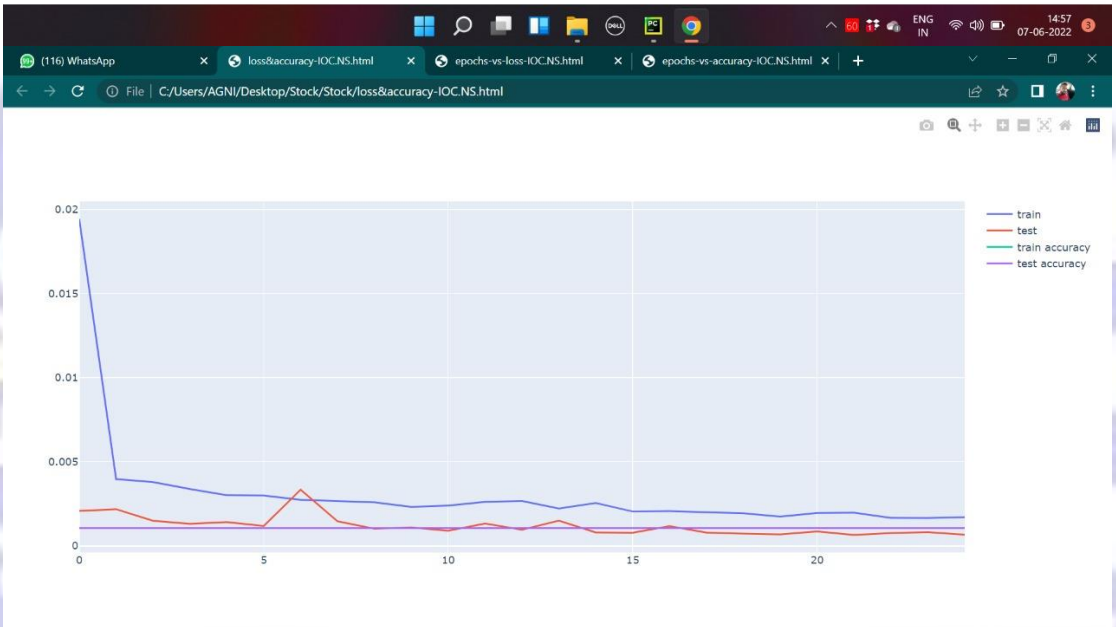

(116) WhatsApp
epochs-vs-loss-CL=F.html
epochs-vs-accuracy-CL=F.html
File | C:/Users/AGNI/Desktop/Stock/Stock/epochs-vs-accuracy-CL=F.html
accuracy vs epochs
val_accuracy vs epochs
1
0.5
0
−0.5
e
p
o
c
h
s

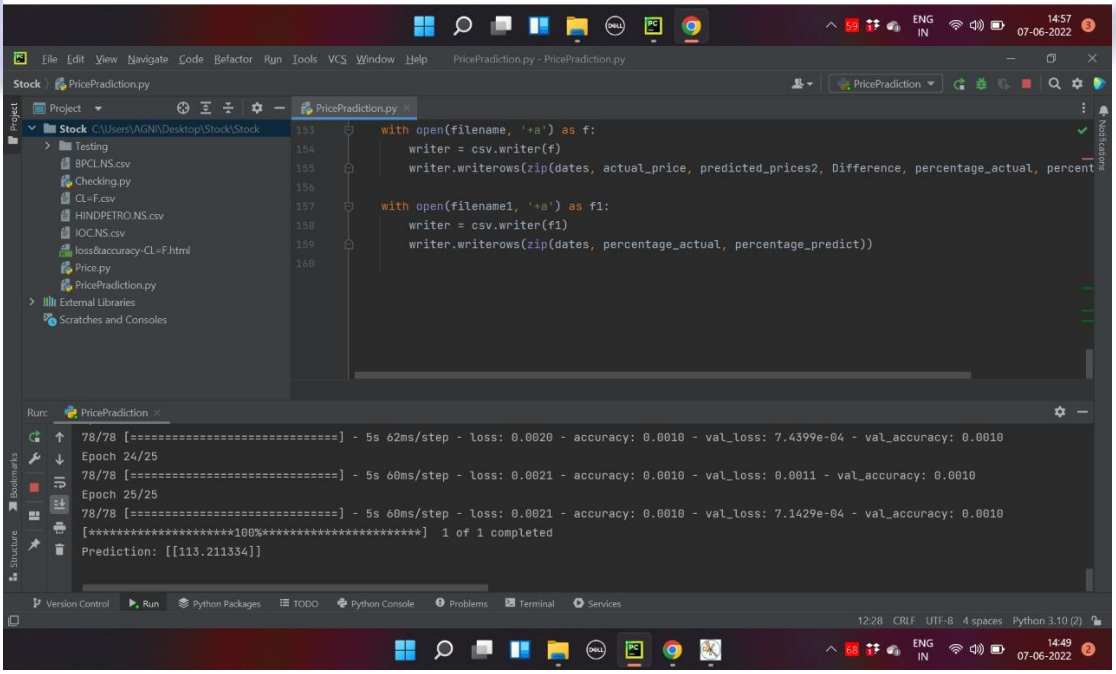

File Edit View Navigate Code Refactor Run Tools VCS Window Help
PricePradiction.py - PricePradiction.py
Stock
PricePradiction.py
Project
Stock C:\Users\AGNI\Desktop\Stock\Stock
Testing
BPCL.NS.csv
Checking.py
CL=F.csv
HINDPETRO.NS.csv
IOC.NS.csv
loss&accuracy-CL=F.html
Price.py
PricePradiction.py
External Libraries
Scratches and Consoles
with open(filename, '+a') as f:
writer = csv.writer(f)
writer.writerows(zip(dates, actual_price, predicted_prices2, Difference, percentage_actual, percent
with open(filename1, '+a') as f1:
writer = csv.writer(f1)
writer.writerows(zip(dates, percentage_actual, percentage_predict))
Run: PricePradiction
78/78 [==============================] - 5s 62ms/step - loss: 0.0020 - accuracy: 0.0010 - val_loss: 7.4399e-04 - val_accuracy: 0.0010
Epoch 24/25
78/78 [==============================] - 5s 60ms/step - loss: 0.0021 - accuracy: 0.0010 - val_loss: 0.0011 - val_accuracy: 0.0010
Epoch 25/25
78/78 [==============================] - 5s 60ms/step - loss: 0.0021 - accuracy: 0.0010 - val_loss: 7.1429e-04 - val_accuracy: 0.0010
[**********100%**********] 1 of 1 completed
Prediction: [[113.211334]]
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
12:28 CRLF UTF-8 4 spaces Python 3.10 (2)

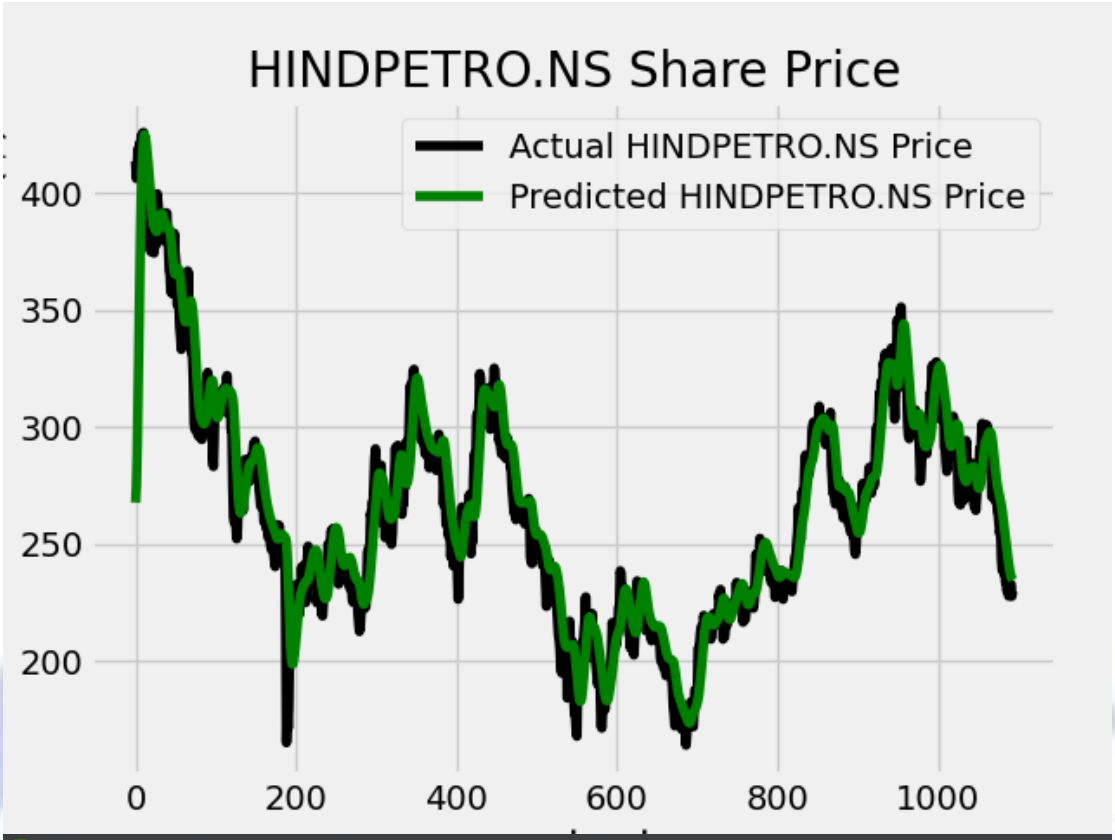

loss&accuracy-CL=F.html
File | C:/Users/AGNI/Desktop/Stock/Stock/loss&accuracy-CL=F.html
0.025
0.02
0.015
0.01
0.005
0
0
5
10
15
20
train
test
train accuracy
test accuracy

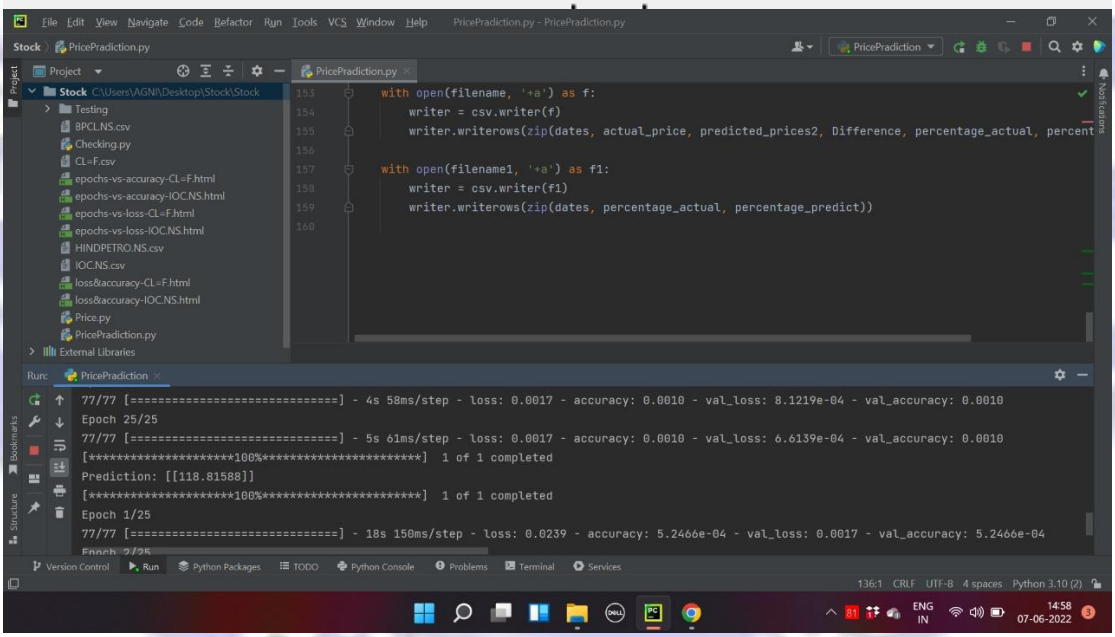

ENG
IN
14:48
07-06-2022
BPCL.NS Share Price
550
500
450
400
350
300
250
0
200
400
600
800
1000
Actual BPCL.NS Price
Predicted BPCL.NS Price


(116) WhatsApp
loss&accuracy-IOC.NS.html
epochs-vs-loss-IOC.NS.html
epochs-vs-accuracy-IOC.NS.html
File | C:/Users/AGNI/Desktop/Stock/Stock/epochs-vs-accuracy-IOC.NS.html
accuracy vs epochs
val_accuracy vs epochs
1
0.5
0
−0.5
e
p
o
c
h
s


(116) WhatsApp
loss&accuracy-IOC.NS.html
epochs-vs-loss-IOC.NS.html
epochs-vs-accuracy-IOC.NS.html
File | C:/Users/AGNI/Desktop/Stock/Stock/loss&accuracy-IOC.NS.html
train
test
train accuracy
test accuracy
0.02
0.015
0.01
0.005
0
0
5
10
15
20

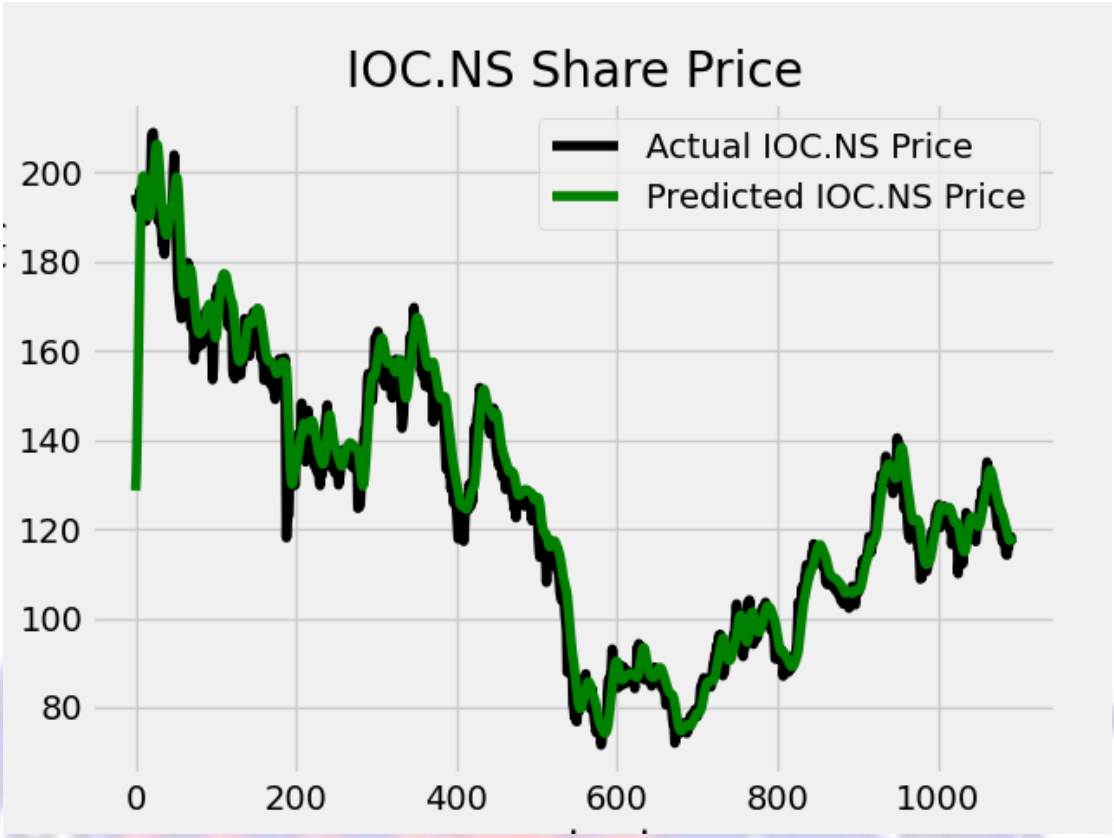

File Edit View Navigate Code Refactor Run Tools VCS Window Help
PricePradiction.py - PricePradiction.py
Stock
PricePradiction.py
Project
Stock C:\Users\AGNI\Desktop\Stock\Stock
Testing
BPCL.NS.csv
Checking.py
CL=F.csv
HINDPETRO.NS.csv
IOC.NS.csv
loss&accuracy-CL=F.html
Price.py
PricePradiction.py
External Libraries
Scratches and Consoles
with open(filename, '+a') as f:
writer = csv.writer(f)
writer.writerows(zip(dates, actual_price, predicted_prices2, Difference, percentage_actual, percent
with open(filename1, '+a') as f1:
writer = csv.writer(f1)
writer.writerows(zip(dates, percentage_actual, percentage_predict))
Run: PricePradiction
78/78 [==============================] - 5s 62ms/step - loss: 0.0020 - accuracy: 0.0010 - val_loss: 7.4399e-04 - val_accuracy: 0.0010
Epoch 24/25
78/78 [==============================] - 5s 60ms/step - loss: 0.0021 - accuracy: 0.0010 - val_loss: 0.0011 - val_accuracy: 0.0010
Epoch 25/25
78/78 [==============================] - 5s 60ms/step - loss: 0.0021 - accuracy: 0.0010 - val_loss: 7.1429e-04 - val_accuracy: 0.0010
[**********100%**********] 1 of 1 completed
Prediction: [[113.211334]]
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
12:28 CRLF UTF-8 4 spaces Python 3.10 (2)


HINDPETRO.NS Share Price
Actual HINDPETRO.NS Price
Predicted HINDPETRO.NS Price
400
350
300
250
200
0
200
400
600
800
1000


File Edit View Navigate Code Refactor Run Tools VCS Window Help
PricePradiction.py - PricePradiction.py
Stock
PricePradiction.py
PricePradiction
Project
Stock C:\Users\AGNI\Desktop\Stock\Stock
Testing
BPCL.NS.csv
Checking.py
CL=F.csv
epochs-vs-accuracy-CL=F.html
epochs-vs-accuracy-IOC.NS.html
epochs-vs-loss-CL=F.html
epochs-vs-loss-IOC.NS.html
HINDPETRO.NS.csv
IOC.NS.csv
loss&accuracy-CL=F.html
loss&accuracy-IOC.NS.html
Price.py
PricePradiction.py
External Libraries
with open(filename, '+a') as f:
writer = csv.writer(f)
writer.writerows(zip(dates, actual_price, predicted_prices2, Difference, percentage_actual, percent
with open(filename1, '+a') as f1:
writer = csv.writer(f1)
writer.writerows(zip(dates, percentage_actual, percentage_predict))
Run:
PricePradiction
77/77 [==========] - 4s 58ms/step - loss: 0.0017 - accuracy: 0.0010 - val_loss: 8.1219e-04 - val_accuracy: 0.0010
Epoch 25/25
77/77 [==========] - 5s 61ms/step - loss: 0.0017 - accuracy: 0.0010 - val_loss: 6.6139e-04 - val_accuracy: 0.0010
[**********100%**********] 1 of 1 completed
Prediction: [[118.81588]]
[**********100%**********] 1 of 1 completed
Epoch 1/25
77/77 [==========] - 18s 150ms/step - loss: 0.0239 - accuracy: 5.2466e-04 - val_loss: 0.0017 - val_accuracy: 5.2466e-04
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
136:1 CRLF UTF-8 4 spaces Python 3.10 (2)
ENG IN
14:58 07-06-2022


(116) WhatsApp
loss&accuracy-IOC.NS.html
epochs-vs-loss-IOC.NS.html
epochs-vs-accuracy-IOC.NS.html
File | C:/Users/AGNI/Desktop/Stock/Stock/epochs-vs-accuracy-IOC.NS.html
accuracy vs epochs
val_accuracy vs epochs
1
0.5
0
−0.5
e
p
o
c
h
s


(116) WhatsApp
loss&accuracy-IOC.NS.html
epochs-vs-loss-IOC.NS.html
epochs-vs-accuracy-IOC.NS.html
File | C:/Users/AGNI/Desktop/Stock/Stock/loss&accuracy-IOC.NS.html
train
test
train accuracy
test accuracy
0.02
0.015
0.01
0.005
0
0
5
10
15
20
ENG
IN
14:57
07-06-2022


IOC.NS Share Price
Actual IOC.NS Price
Predicted IOC.NS Price
200
180
160
140
120
100
80
0
200
400
600
800
1000

Bibliography

REFERENCES

[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.

[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.