STOCK PRICE FORECASTING AND RISK ANALYSIS USING LSTM ALGORITHM

A PROJECT REPORT

Submitted by

S.DHANRAJ (922520205021)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

V.S.B. ENGINEERING COLLEGE,KARUR-639111
(AN AUTONOMOUS INSTITUTION)

ANNA UNIVERSITY :: CHENNAI - 600 025

MAY 2023

ANNA UNIVERSITY :: CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**STOCK PRICE FORECASTING AND RISK ANALYSIS USING LSTM ALGORITHM**" is the bonafide work of "**S.DHANRAJ (922520205021)** who carried out under my supervision.

SIGNATURE	**SIGNATURE**
Mr. K. MANIVANNAN, M.Tech., (Ph.D.)	Mrs.S.SAHEBZATHI,ME.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	Assistant Professor
Department of Information Technology	Department of Information Technology
V.S.B. Engineering College	V.S.B. Engineering College
Karur-639111.	Karur-639111.

Submitted for Anna University project Viva-Voce held on ____________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost, we express our thanks to our parents for providing us a very nice environment for doing this project. We wish to express our sincere thanks to our founder and Chairman **Shri. V. S. BALSAMY B.Sc., L.L.B.,** for his endeavour in educating us in this premier institution.

We extend our gratitude to **Dr .V. Nirmal Kannan M.E., Ph.D.,** Principal and **Mr.T.S. Kirubhashankar M.E., (Ph.D.),** Vice Principal, **V.S.B. ENGINEERING COLLEGE,KARUR** for his high degree of encouragement and moral support during this project.

We express our indebtedness to **Mr.K.MANIVANNAN, M.Tech.,(Ph.D.).,** Head of the Department, Department of Information Technology for his guidance throughout the course of our project.

We are grateful to our project Supervisor **Mrs.S.SAHEBZATHI,ME.,** Assistant professor, Department of Information Technology for her valuable support.

Our sincere thanks to all the Faculty members of V.S.B. EngineeringCollege and our friends for their help in the successful completion of this projectwork.

Finally, we bow before god, the almighty who always had a better plan for us. We give our praise and glory to almighty god for successful completion of our project.

\

V.S.B. ENGINEERING COLLEGE
(Approved by AICTE, New Delhi, Affiliated to Anna University)
An ISO 9001:2015 Certified Institution
Accredited by NAAC, NBA Accredited Courses

DEPARTMENT OF INFORMATION TECHNOLOGY

Vision of the Institution:

We endeavour to impart futuristic technical education of the highest quality to the student community and to inculcate discipline in them to face the world with self-confidence and thus we prepare them for life as responsible citizens to uphold human values and to be of service at large. We strive to bring of the Institution as an Institution of academic excellence of International standard.

Mission of the Institution:

We transform persons into personalities by the state-of the art infrastructure, time consciousness, quick response and the best academic practices through assessment and advice.

Vision of the Department:

To provide professional computing training to make competent IT engineers and thus prepare them to work in the emerging trends of information technology field.

Mission of the Department:

1. Producing quality IT engineers with good technical knowledge by effective teaching.
2. Making competent engineers to adapt to the dynamic needs of industries.
3. Developing extensive learning skills in studies.
4. Inculcating strong ethical values and professionalism to serve society with responsibility.

Program Educational Objectives (PEOs)

PEO 1: To make graduates to be proficient in utilizing the fundamental knowledge of various streams in engineering and technology.

PEO 2: To enrich graduates with the core competencies necessary for applying knowledge of computers and technologies in the context of business enterprise.

PEO 3: To enable graduates think logically and to pursue lifelong learning to understand technical issues related to computing systems and to provide optimal solutions.

PEO 4: To enable graduates develop hardware and software system by understanding the importance of social, business and environmental needs in the social context.

Program Outcomes (POs)

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. 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.

PO3. Design/development of solutions: Design solutions for complex engineering problems & 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.

PO4. 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.

PO5. 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.

PO6. 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.

PO7. 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.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. 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.

PO11. 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.

PO12. 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.

Program Specific Outcome (PSOs)

PSO1: Apply the mathematical and the computing knowledge to identify and provide solutions for computing problems.

PSO2: Design and develop computer programs in the areas related to algorithms, networking, web design and data analytics of varying complexity.

PSO3: Apply standard Engineering practices and strategies in software project using open source environment to deliver a quality product for business success

TABLE OF CONTENTS

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.

We present an Artificial Neural Network (ANN) approach to predict stock market indices,particularly with respect to the forecast of their trend movements *up or down*. Exploiting different Neural Networks architectures, we provide numerical analysis of concrete financial time series. In particular, after a brief resume of the existing literature on the subject, we consider the Multi-layer Perceptron (MLP), the Convolutional Neural Networks (CNN), and the Long Short-Term Memory (LSTM) recurrent neural networks techniques. We focus on the importance of choosing the correct input features, along with their preprocessing, for the specific learning algorithm one wants to use. Eventually, we consider the S&P500 historical time series, predicting trend on the basis of data from the past days, and proposing a novel approach based on combination of wavelets and CNN, which outperforms the basic neural networks ones. We show, that neural networks are able to predict financial time series movements even trained only on plain time series data and propose more ways to improve results.

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

LIST OF FIGURES

LIST OF ABBREVIATION

S.NO.	ABBREVIATION	EXPANSION
1.	**ML**	Machine Learning
2.	**API**	Application Programming Interface
3	**ATS**	Automated Trading System
4	**CNN**	Convolutional Neural Network
5	**RNN**	Recurrent Neural Network
6	**LSTM**	Long Short Term Memory
7	**ER**	Entity Relationship
8	**MLP**	Multi Layer Perceptron
9	**ARMA**	Autoregressive Moving Average
10	**MSE**	Mean Square Error
11	**RMSE**	Root Mean Square Error
12	**MAE**	Mean Absolute Error
13	**GRU**	Gated Recurrent Unit
14	**DRL**	Deep Reinforcement Learning
15	**UML**	Unified Modeling Language

INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 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 analyzed. 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.

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1 Stock Market Prediction Using Machine Learning 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 capitalization and in the three different markets, employing prices with both daily and up-to-the-minute frequencies.

2.2 Forecasting the Stock Market Index Using Artificial Intelligence Techniques by Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg.

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, multi layer perceptron neural networks is dependent on the accuracy measure used.

2.3 Indian stock market prediction using artificial neural networks on tick data 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. 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.

2.4 The Stock Market and Investment 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 auto regressive 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.

2.5 Automated Stock Price Prediction Using Machine Learning 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.

2.6 Stock Price Correlation Coefficient Prediction with ARIMA LSTM Hybrid Model 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 non linearity 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 ARIMA LSTM model to forecast correlation coefficient for portfolio optimization.

2.7 Forecasting directional movements of stock prices for intraday trading using LSTM and random forests by Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari Sahoo Department of Computer Science & Information Systems, 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 outof-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 singlefeature 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.

2.8 An innovative neural network approach for stock market prediction 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.

SYSTEM ANALYSIS

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The approaches used to forecast future directions of share market prices are historically splitted into two main categories: those that rely on technical analysis, and those that rely on fundamental analysis. Technical analysis uses only historical data such as, e.g., past prices, volume of trading, volatility, etc., while fundamental analysis is based on external information like,e.g., interest rates, prices and returns of other assets,and other macro/micro-economic parameters. The *machine learning* approach to the latter problem has been declined in several forms, especially during recent years, also aiming to find an effective way to predict sudden financial crashes as, e.g., the one happened during the 2007-08 biennium. As an example, good results have been obtained using linear classifiers as the logistic regression one, which has been used to predict the Indian Stock market, or with respect to the S&P500 index. More complicated techniques, as large-margin classifier or Support Vector Machine (SVM), was the best choice for prediction before the rise of neural networks. The latter uses a *kernel trick* which allows to consider our input data as embedded into a higher-dimensional space,where it is linearly separable. Successful applications of SVM have been developed, e.g., with respect to the analysis the *Korea composite* stock prices market, and the NIKKEI 225 index, clearly showing how SVM outperforms logistic regression, linear discriminant analysis and even some neural network approaches. Another popular technique based on decision trees, i.e. the *random forest* one, was also applied to similar problems. where the authors studied the BM&F/BOVESPA stock market, resulting in a high accuracy obtained by using as input features different technical indicators such, e.g., simple and exponential moving averages, rate of change, stochastic oscillator, etc. Therefore mentioned references are all characterized by an accuracy in stock price movement forecasting that ranges between 70% and 90%. In what follows, we show that is possible to achieve the same accuracy by mean of neural networks, working with preprocessed open/close/high/low data, also working with high frequent, intra-day, data.

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 91%.

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.

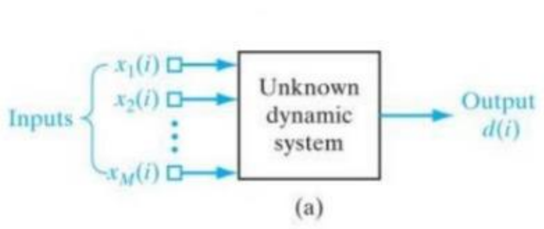


Fig. 1: LMS Inputs and Outputs

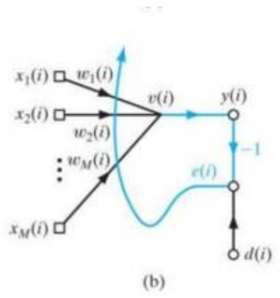


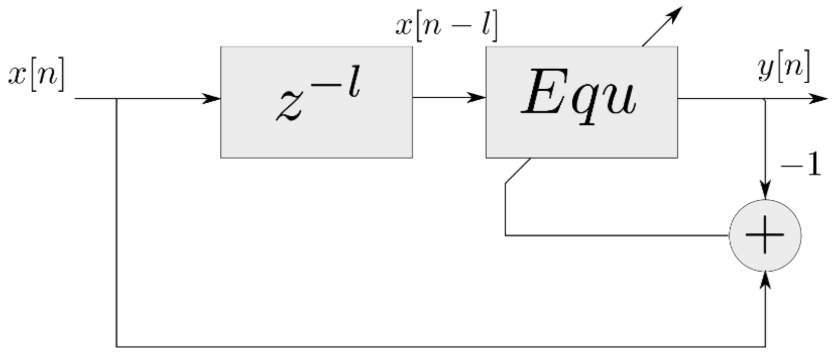
Fig 2: LMS updating weights

```
Algorithm 1: LMS
Input:
$x$ : input vector
$d$: desired vector
$\mu$: learning rate
$N$: filter order
Output:
$y$: filter response
$e$: filter error
begin
    $M = size(x)$ ;
    $x_n(0) = w_n(0) = [0\ 0\ \ldots\ 0]^T$;
    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.

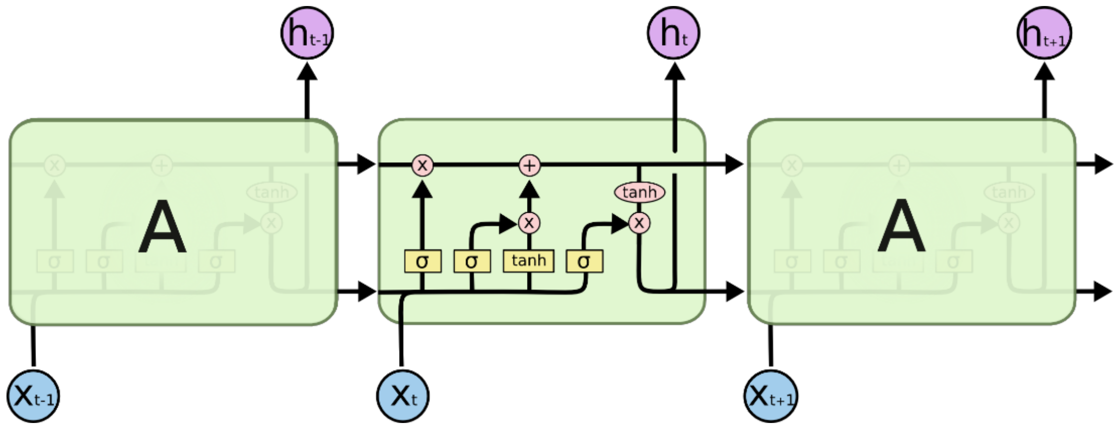


LMS updating weights

When we apply the LMS filter, we will train the filter to the first 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
    $M = size(x_d)$;
    $x_n(0) = w_n(0) = [0\ 0\ \dots\ 0]$;
    $x_d = [0\ 0\ ..\ 0\ x]$;
    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:



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.

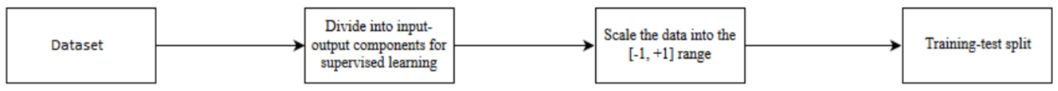
Output Gate:

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.

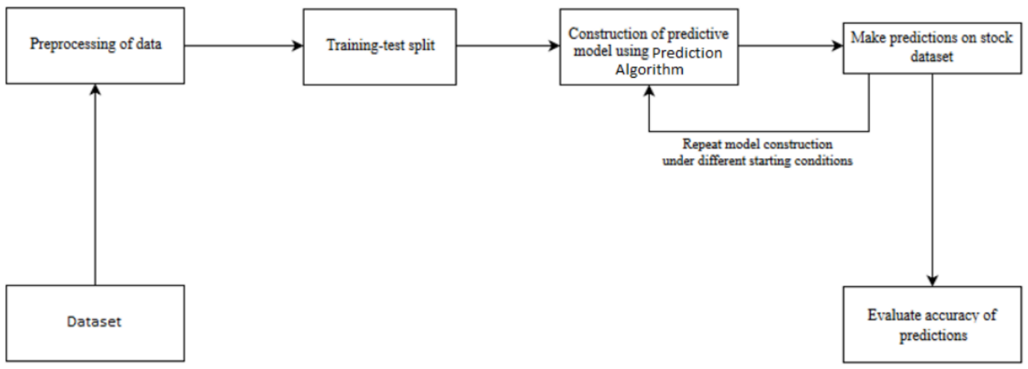
3.2.1 SYSTEM ARCHITECTURE

1)Preprocessing of data:


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

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

Overall Architecture

3.3 FEASIBILITY STUDIES

FEASIBILITY STUDIES

A feasibility study is a systematic and comprehensive analysis conducted to assess the practicality and viability of a proposed project. It involves evaluating various aspects of the project, such as technical, financial, operational, legal, and market considerations, to determine if the project is feasible and worth pursuing.The purpose of a feasibility study is to provide objective and evidence-based information to decision-makers, stakeholders, and investors. It helps them understand the potential benefits, risks, and challenges associated with the project before committing resources, time, and capital to its implementation.

3.3.1 ECONOMIC FEASIBILITY:

This project is completely feasible because it requires no extra financial investment and the cost of development is very minimal when compared to financial benefits of the application.

3.3.2 TECHNICAL FEASIBILITY:

An evaluation of the project's technical requirements, such as technology, equipment, infrastructure, and resources. This section examines whether the necessary technical capabilities exist or can be acquired within the project's constraints.This project is technically feasible because, all the technology needed for our project is readily available.

Operating System: Windows 11

Languages : Python with HTML

Documentation Tool : MS-Word 2021

3.3.3 OPERATIONAL FEASIBILITY:

The methods of processing and presentation are completely accepted by the clients since they can meet all user requirements. The proposed system will not cause any problem under any circumstances. This project is operationally feasible since the time requirements and the personal requirements are satisfied.

3.3.4 LEGAL FEASIBILITY:

During the social feasibility analysis, we look at how the project could change the community. This is done to gauge the level of public interest in the endeavour. Because of established cultural norms and institutional frameworks,it's likely that a certain kind of worker will be in low supply or non existent.

3.4 REQUIREMENT SPECIFICATION

3.4.1 HARDWARE REQUIREMENTS

Processor	: Intel i3 7th Gen Core 2.00GHZ
Hard disk	: 10 GB
RAM	: 4GB (minimum)
Keyboard	: 110 keys enhanced

3.4.2 SOFTWARE REQUIREMENTS

Operating system	: Windows7 (with service pack 1)/ 8/ 8.1 or10
Language	: Python with ML Libraries
Software	: Jupyter Lab

3.4.3 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 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.

3.4.4 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.

3.5 LANGUAGE SPECIFICATION– PYTHON

Among programmers, Python is a favourite because to its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively utilized by the programming community.

Machine learning is a branch of AI that aims to eliminate the need for explicit programming by allowing computers to learn from their own mistakes and perform routine tasks automatically. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make The desire for intelligent solutions to real-world problems has necessitated the need to develop AI further in order to automate tasks that are arduous to programme without AI. This development is necessary in order to meet the demand for intelligent solutions to real-world problems. Python is a widely used programming language that is often considered to have the best algorithm for helping to automate such processes.

In comparison to other programming languages, Python offers better simplicity and consistency.

In addition, the existence of an active Python community makes it simple for programmers to talkabout ongoing projects and offer suggestions on how to improve the functionalityof their programme.

ADVANTAGES OF USING PYTHON

Following are the advantages of using Python:

- Variety of Framework and libraries:

  A good programming environment requires libraries and frameworks. Python frameworks and libraries simplify programme development. Developers can speed up complex project coding with pre written code from a library. PyBrain,a modular machine learning toolkit in Python, provides easy-to-use algorithms.Python frameworks and libraries provide a structured and tested environment for the best coding solutions.

- Reliability

  Most software developers seek simplicity and consistency in Python. Python code is concise and readable, simplifying presentation. Compared to other programming languages, developers can write code quickly. Developers can get community feedback to improve their product or app. Python is simpler than other programming languages, therefore beginners may learn it quickly.Experienced developers may focus on innovation and solving real-world problems with machine learning because they can easily design stable and trustworthy solutions.

SYSTEM DESIGN

CHAPTER 4
SYSTEM DESIGN

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.

4.1 ARCHITECTURE DIAGRAM

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.

Start Activity: The initial step of the activity diagram.

Collect Stock Data: Collect historical stock market data from a reliable source.

Preprocess Data: Perform necessary preprocessing steps on the collected data, such as normalization or scaling.

Split Data into Training and Testing Sets: Divide the preprocessed data into two sets: one for training the LSTM model and the other for testing its performance.

Build LSTM Model: Construct the LSTM model architecture, including the number of layers, units, and input/output dimensions.

Train LSTM Model: Use the training set to train the LSTM model using the LSTM algorithm.

Evaluate LSTM Model: Assess the performance of the trained LSTM model using the testing set to measure accuracy or other evaluation metrics.

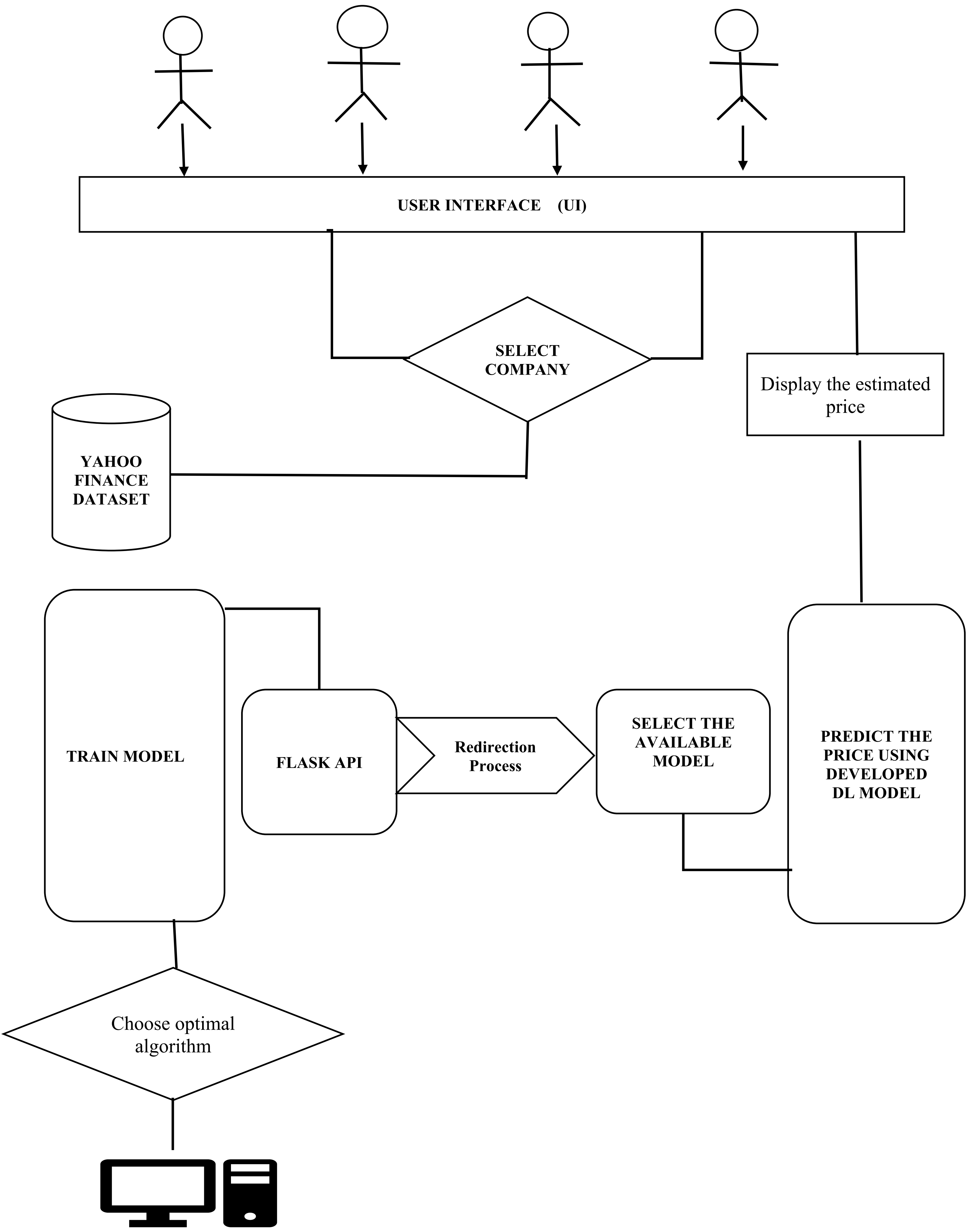
Make Stock Market Predictions: Apply the trained LSTM model to make predictions on future stock market data.

Display Predictions: Present the predictions in a suitable format, such as charts or tables.

Perform Post-processing: Perform any necessary post-processing steps on the predictions, such as transforming the data back to its original scale or format.

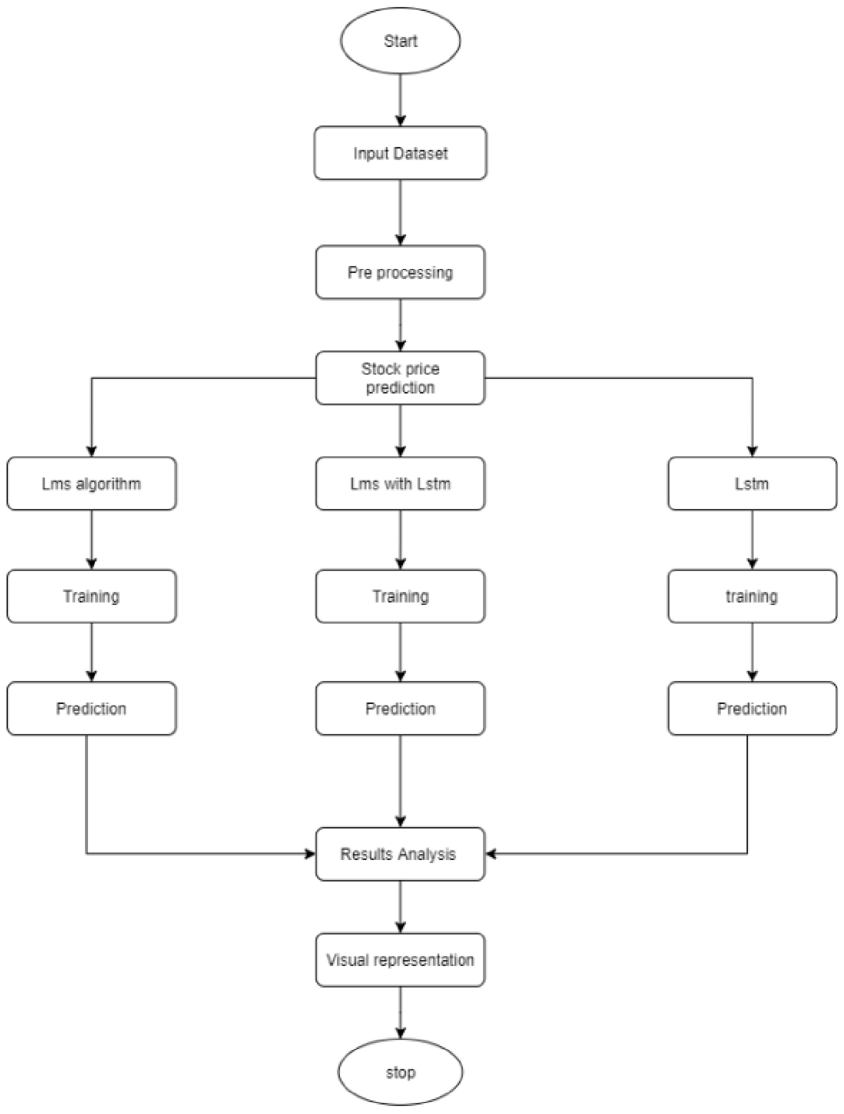
End Activity: The final step of the activity diagram.

Please note that this diagram provides a high-level overview and may not include all the specific details or steps involved in implementing the LSTM algorithm for stock market prediction.


USER INTERFACE (UI)
SELECT COMPANY
Display the estimated price
YAHOO FINANCE DATASET
TRAIN MODEL
FLASK API
Redirection Process
SELECT THE AVAILABLE MODEL
PREDICT THE PRICE USING DEVELOPED DL MODEL
Choose optimal algorithm

4.2 DATA FLOW DIAGRAM

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 step as boxes of various kinds, and their order by connecting the boxes with arrows.



4.3 USE CASE DIAGRAM

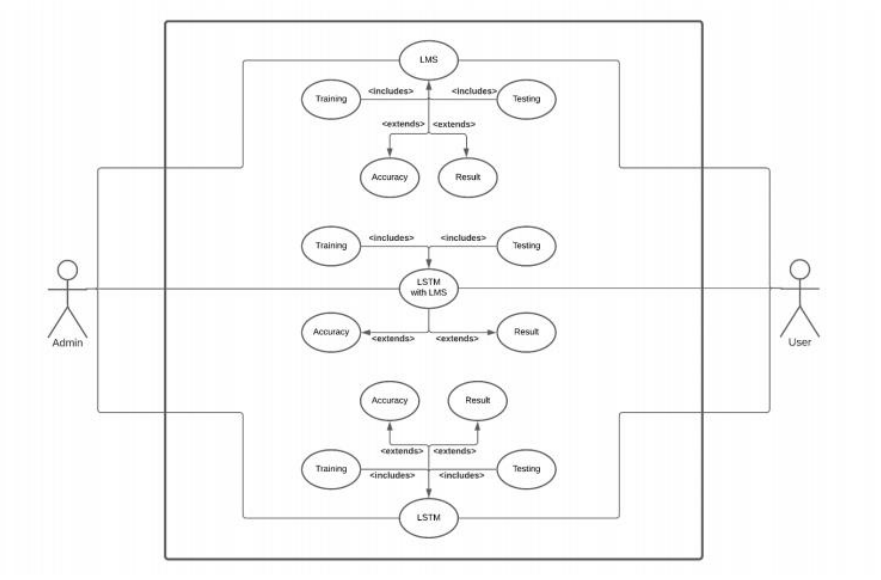
In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

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.

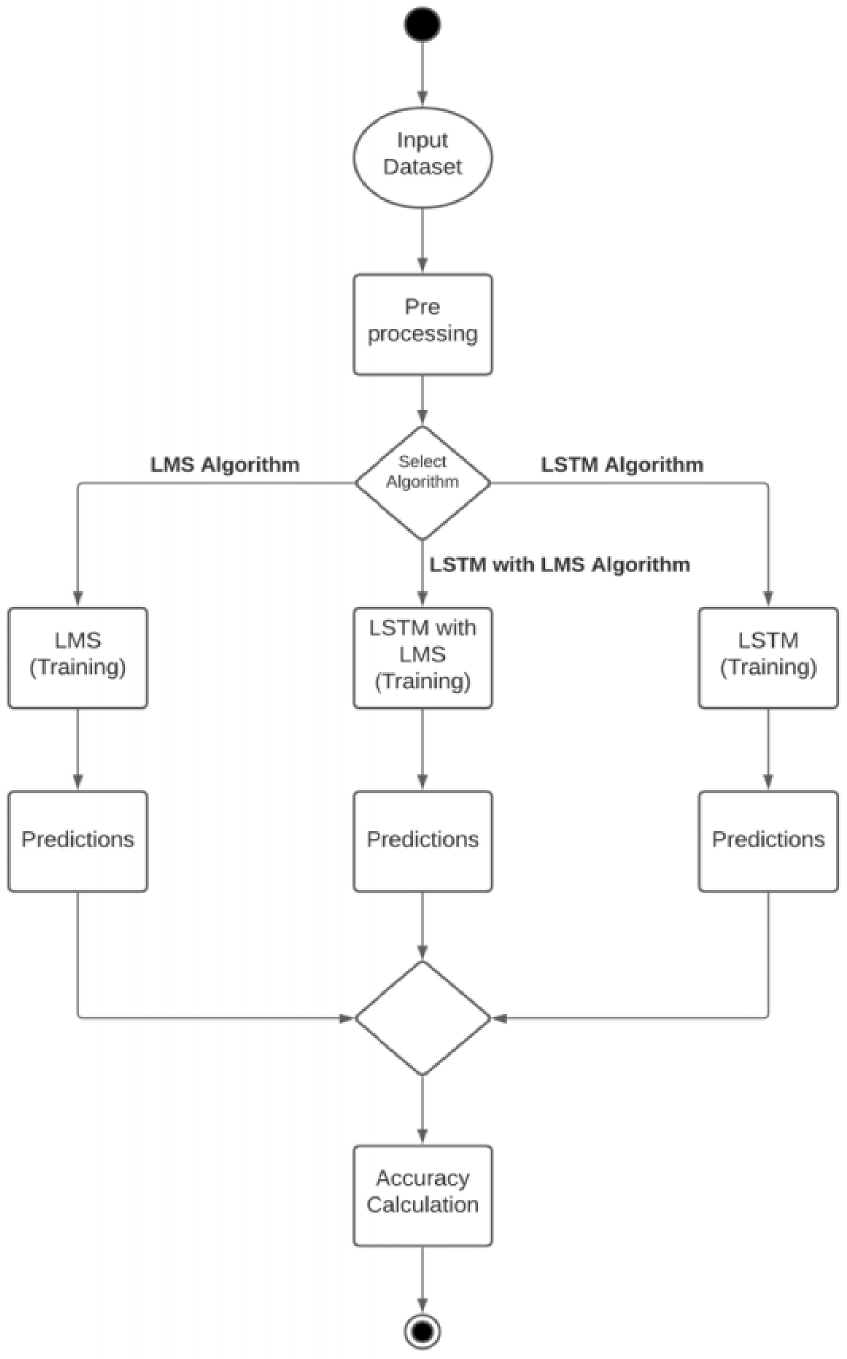
• The scope of your system.



LSTM WITH LMS SYSTEM

4.4 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.



4.5 SEQUENCE DIAGRAM

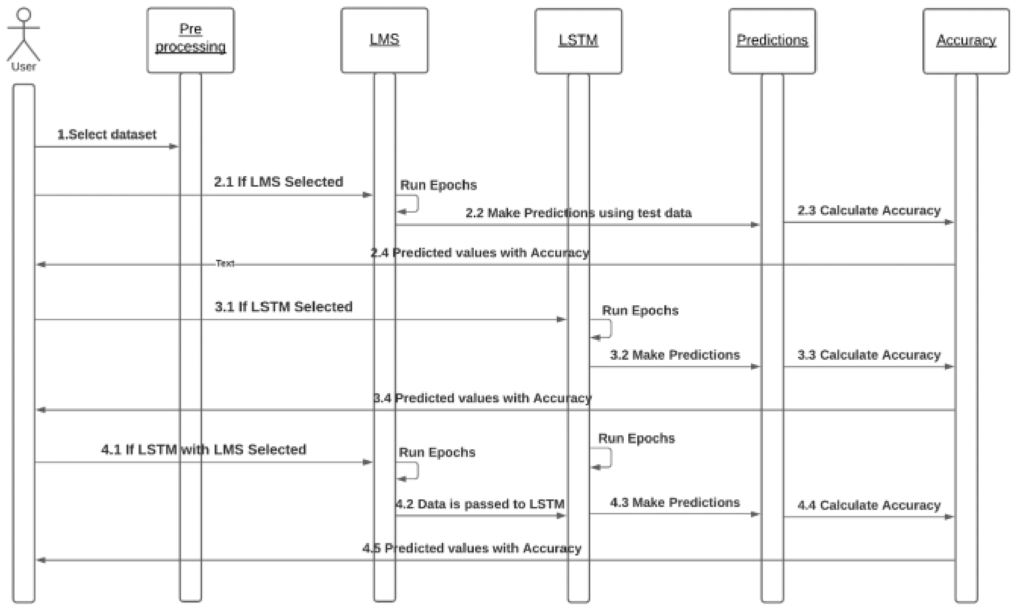
A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams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



4.6 CLASS DIAGRAM

Stock Data Class:

Represents the stock market data used for training and testing the LSTM model.

Attributes:

data: A list of floating-point values representing the stock market data.

Methods:

get_data(): Returns the stock market data.

set_data(data: List[float]): Sets the stock market data.

LSTM Model Class: Represents the LSTM model used for stock market prediction.

Attributes:

architecture: A string representing the architecture of the LSTM model.

training_parameters: A dictionary representing the parameters used for training the model.

trained_at: A DateTime object representing the timestamp when the model was trained.

accuracy: A float representing the accuracy of the trained model.

Methods:

get_architecture(): Returns the architecture of the LSTM model.

set_architecture(architecture: str): Sets the architecture of the LSTM model.

get_training_parameters(): Returns the training parameters of the LSTM model.

set_training_parameters(parameters: dict): Sets the training parameters of the LSTM model.

get_trained_at(): Returns the timestamp when the model was trained.

set_trained_at(time: DateTime): Sets the timestamp when the model was trained.

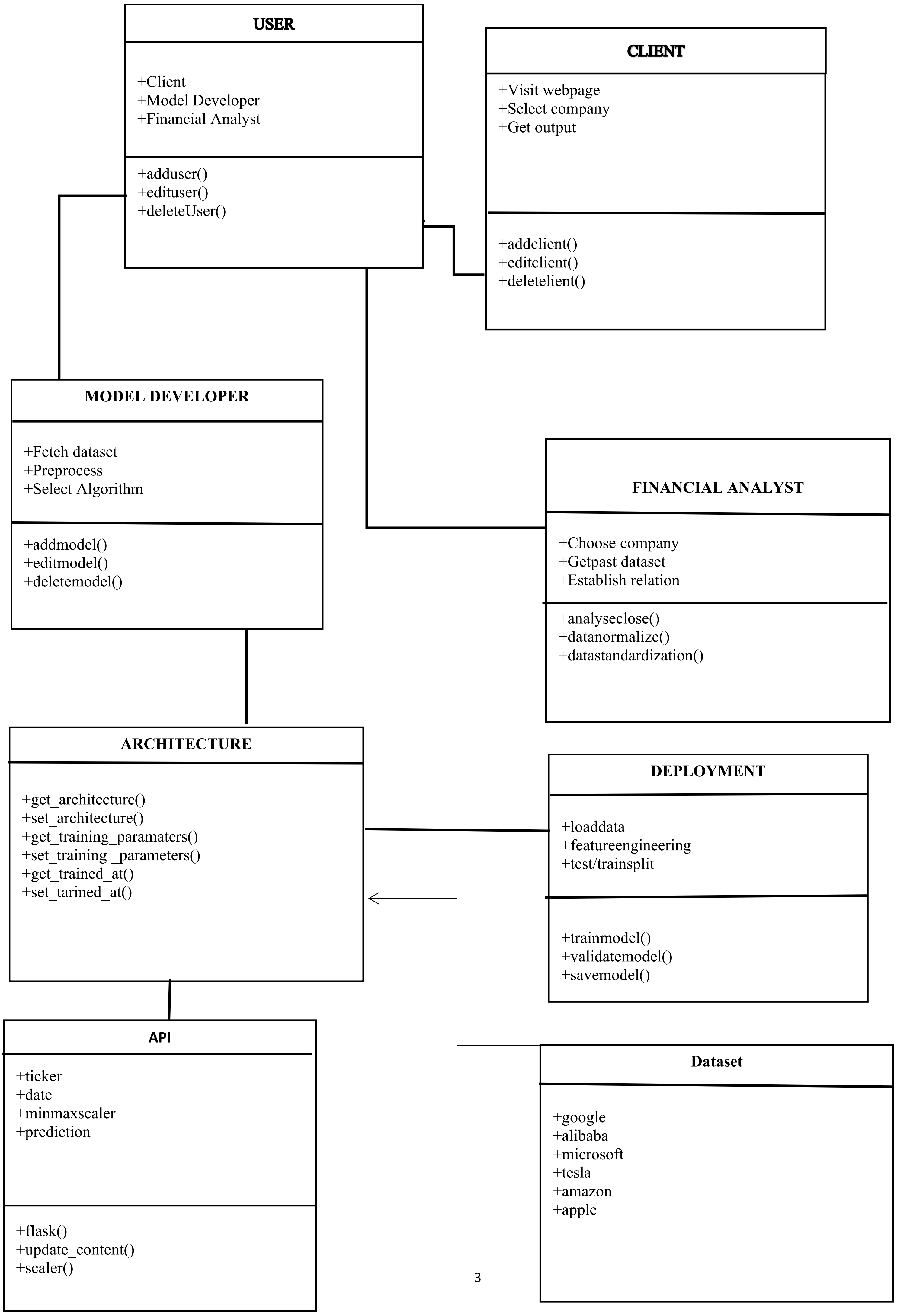
get_accuracy(): Returns the accuracy of the trained model.

set_accuracy(accuracy: float): Sets the accuracy of the trained model.

train_model(training_data: StockData, testing_data: StockData): Trains the LSTM model using the provided training and testing data.

make_predictions(data: StockData): Makes predictions using the trained LSTM model on the given data.

This class diagram provides a basic structure and may not include all the specific attributes or methods required for a comprehensive stock market prediction system using LSTM. Additionally, you may consider adding additional classes or refining the existing ones based on your specific requirements.


USER
+Client
+Model Developer
+Financial Analyst
+adduser()
+edituser()
+deleteUser()
CLIENT
+Visit webpage
+Select company
+Get output
+addclient()
+editclient()
+deletelient()
MODEL DEVELOPER
+Fetch dataset
+Preprocess
+Select Algorithm
+addmodel()
+editmodel()
+deletemodel()
FINANCIAL ANALYST
+Choose company
+Getpast dataset
+Establish relation
+analyseclose()
+datanormalize()
+datastandardization()
ARCHITECTURE
+get_architecture()
+set_architecture()
+get_training_paramaters()
+set_training _parameters()
+get_trained_at()
+set_tarined_at()
DEPLOYMENT
+loaddata
+featureengineering
+test/trainsplit
+trainmodel()
+validatemodel()
+savemodel()
API
+ticker
+date
+minmaxscaler
+prediction
+flask()
+update_content()
+scaler()
Dataset
+google
+alibaba
+microsoft
+tesla
+amazon
+apple

4.7 ER DIAGRAM

Dataset Entity:

Represents the stock market dataset used for training and testing the LSTM model. It contains information such as dataset ID, name, description, source, and timestamps for creation and update.

LSTM Model Entity:

Represents the LSTM model used for stock market prediction. It includes attributes like model ID, name, architecture, training parameters, training timestamp, and accuracy achieved by the model.

Stock Symbol Entity:

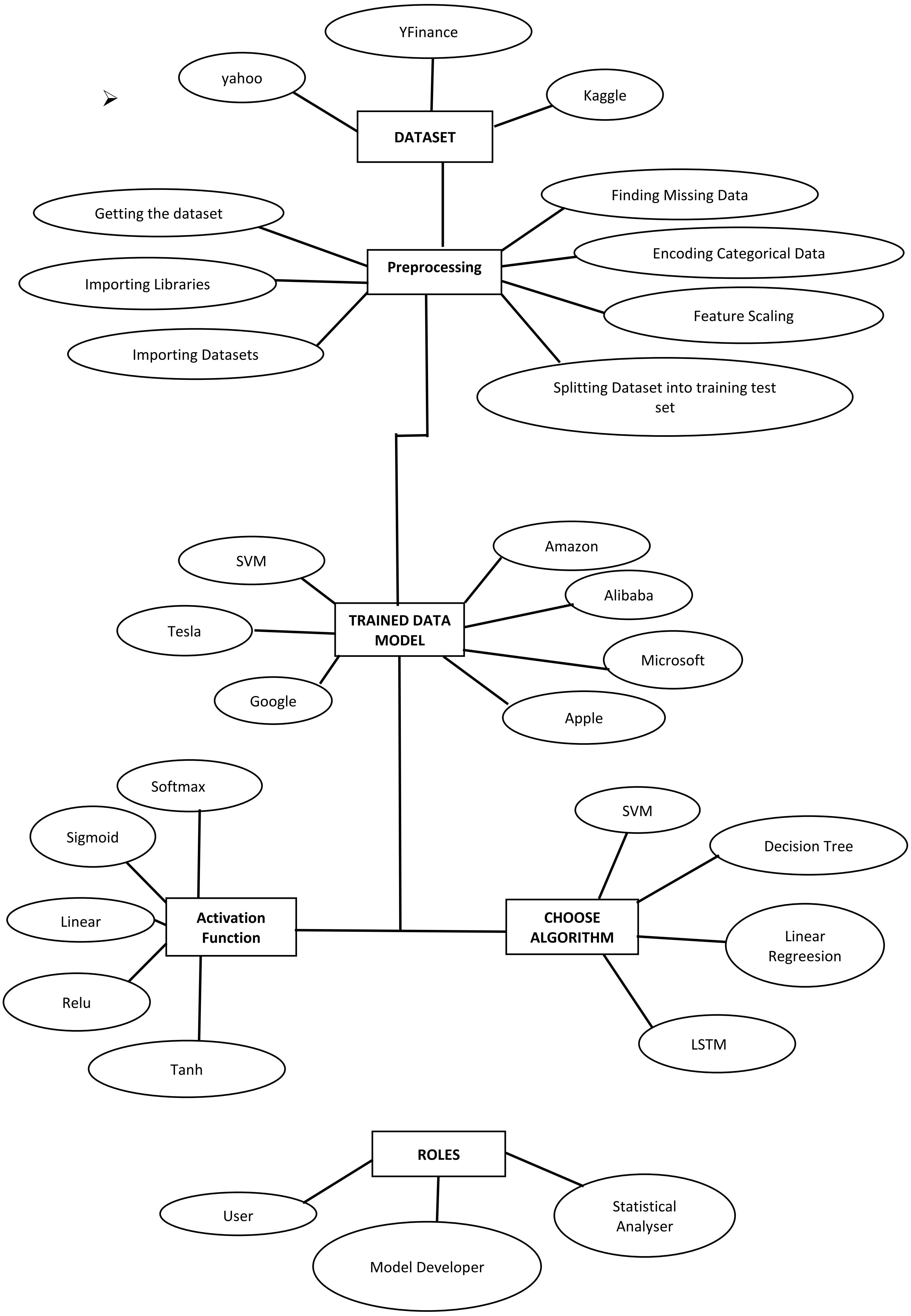
Represents the individual stock symbols or companies being predicted. It contains attributes like symbol ID, stock symbol, company name, and timestamps for creation and update.

Prediction Result Entity:

Represents the predicted results for each stock symbol using the LSTM model. It includes attributes like result ID, associated model ID, associated symbol ID, predicted date, and predicted price.

The relationships in the ER diagram are as follows:

- The Dataset entity has a one-to-many relationship with the LSTM Model entity, indicating that a dataset can be used to train multiple LSTM models.
- The LSTM Model entity has a one-to-many relationship with the Prediction Result entity, indicating that a model can produce multiple prediction results.
- The Stock Symbol entity has a one-to-many relationship with the Prediction Result entity, indicating that a stock symbol can have multiple prediction results.
- Please note that this ER diagram provides a basic structure and may not include all the specific attributes or relationships required for a comprehensive stock market prediction system using LSTM.


YFinance
yahoo
Kaggle
DATASET
Getting the dataset
Finding Missing Data
Encoding Categorical Data
Preprocessing
Importing Libraries
Feature Scaling
Importing Datasets
Splitting Dataset into training test set
Amazon
SVM
Alibaba
TRAINED DATA MODEL
Tesla
Microsoft
Google
Apple
Softmax
Sigmoid
SVM
Decision Tree
Linear
Activation Function
CHOOSE ALGORITHM
Linear Regreesion
Relu
LSTM
Tanh
ROLES
User
Statistical Analyser
Model Developer

MODULE DESCRIPTION

CHAPTER 5

MODULE DESCRIPTION

5.1 DATA ACQUISITION AND DATAPREPROSSING

Data Collection

Identify the data sources: Determine the relevant data sources for stock price prediction, such as financial databases, stock exchanges, news websites, or social media platforms.

Gather historical stock price data:

Retrieve historical stock price data for the target stocks or indices. This data typically includes information such as opening price, closing price, high and low prices, trading volume, and timestamps.Collect supplementary data: Gather additional data that can provide insights into stock price movements, such as company financial, economic indicators (e.g., GDP, inflation rate), news articles, analyst reports, and social media sentiment.

Data Preprocessing

Data cleaning:

Handle missing values, outliers, and inconsistencies in the collected data. This may involve techniques like interpolation, imputation, or removal of incomplete or erroneous data.

Feature scaling:

Normalize the data to ensure that all features have a similar scale. Common techniques include standardization (e.g., z-score normalization) or normalization (e.g., min-max scaling).

Feature engineering:

Create new features that capture meaningful patterns or relationships in the data. This can involve transforming existing features, creating lagged variables, or extracting technical indicators (e.g., moving averages, relative strength index) from the stock price data

5.2 MODEL IMPROVISATION

Feature Selection

Statistical analysis:

Perform statistical tests (e.g., correlation analysis, t-tests) to identify the features that exhibit a significant relationship with the target variable(stock price).

Domain expertise

Leverage expert knowledge in the finance domain to select features that are known to impact stock prices.

Dimensionality reduction:

Apply techniques such as principal component analysis (PCA) or feature importance ranking algorithms to reduce the dimensionality of the feature space while preserving the most relevant information.

Model Selection

Regression models:

Consider linear regression, polynomial regression, or ridge regression for modeling the relationship between features and stock prices.

Time series models:

Explore models specifically designed for time series data, such as auto regressive integrated moving average (ARIMA), auto regressive conditional heteroscedasticity (ARCH), or generalized auto regressive conditional heteroscedasticity (GARCH).

Machine learning models:

Evaluate algorithms like random forests, support vector machines (SVM),or gradient boosting machines (GBM) that can capture complex relationships in the data.

Deep learning models:

Utilize recurrent neural networks (RNNs), long short-term memory networks (LSTMs), or convolutional neural networks (CNNs) for modeling temporal dependencies and patterns in the stock price data.

5.3 MODEL TRAINING:

Splitting the data:

Divide the preprocessed data into training and testing sets. The training set is used to train the model, while the testing set is reserved for evaluating its performance.

Model parameter tuning:

Optimize the model's hyper-parameters using techniques like grid search or random search to find the combination that yields the best performance.

Training the model:

Feed the training data into the selected model, which learns the underlying patterns and relationships between features and stock prices through an iterative optimization process.

5.4 MODEL EVALUATION

Testing the model:

Use the reserved testing set to assess the model's performance. The model predicts stock prices for the testing data, and the predictions are compared against the actual prices.

Evaluation metrics:

Calculate various metrics, including mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and R-squared, to quantify the accuracy and reliability of the model's predictions.

Visualize results:

Plot the predicted stock prices alongside the actual prices to visually compare the model

SYSTEM TESTING

CHAPTER 6

SYSTEM TESTING

INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTING

UNIT TESTING:-

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

INTEGRATION TESTING: -

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

FUNCTIONAL TEST: -

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.Functional testing is centered on the following items:

- Valid Input: identified classes of valid input must be accepted.
- Invalid Input: identified classes of invalid input must be rejected.
- Functions: identified functions must be exercised.
- Output: identified classes of application outputs must be exercised.
- Systems/Procedures: interfacing systems or procedures must be invoked.
- Organization and preparation of functional tests is focused on requirements.

ACCEPTANCE TESTING

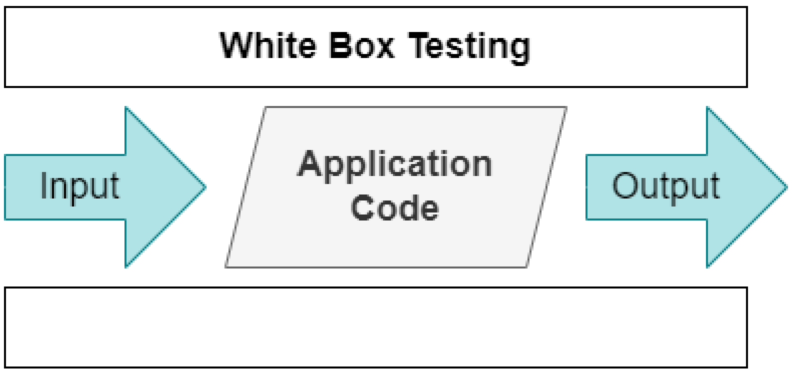
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.2 TESTING TECHNIQUES: -

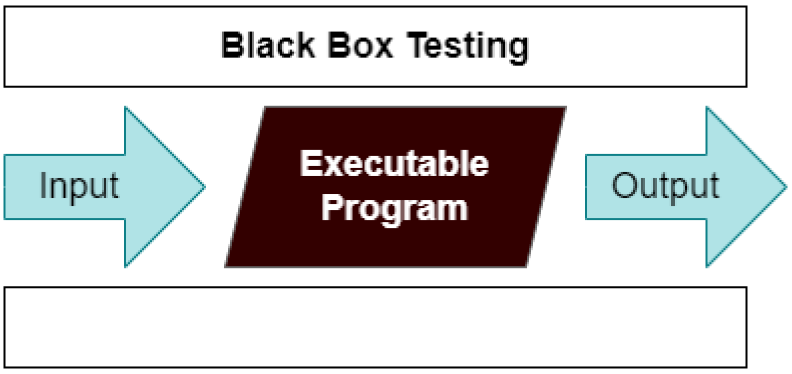
System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING: -



White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING: -



Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. User cannot ―seeǁ into it. The test provides inputs and responds to outputs without considering how the software works.

Test Cases

1. Test case for code coverage: Verify that the code for the system has been thoroughly tested, covering all branches,loops, and error handling scenarios.

2. Test case for data processing: Verify that the system is properly processing and normalizing input data, such as images and text descriptions, before passingthem to the AI model for analysis.

3. Test case for AI model performance: Verify that the AI model has been trained properly on a representative dataset of missing persons, and that it is producing accurate results.

CONCLUSION

CHAPTER 7

CONCLUSION

7.1 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 Yahoo Finance S&P500 for the Stocks price on Tesla,Amazon,Apple, Alibaba Microsoft and Google and achieved above 95% accuracy for these datasets.

In this work different artificial neural network approaches, namely MLP,CNN, and RNN, have been applied to the forecasting of stock market price movements. We compared results trained on a daily basis, for the S&P500 index, showing, in particular, that *convolutional neural networks* (CNN) can model financial time series better then all the other considered architectures.

7.2 FUTURE WORK

• We want to extend this application for predicting cryptocurrency trading.

• We want to add sentiment analysis for better analysis.

APPENDIX I

CHAPTER 8
APPENDIX I

CODING

BACKEND :

Model_development.py

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import tensorflow as tf

from tensorflow.keras.models import load_model

sns.set_style('whitegrid')

plt.style.use("fivethirtyeight")

from pandas_datareader.data import DataReader

import yfinance as yf

from pandas_datareader import data as pdr

yf.pdr_override()

# For time stamps

from datetime import datetime

en = datetime.now()

st = datetime(en.year - 10, en.month, en.day)

df = pdr.get_data_yahoo('TSLA', start=st, end=en)

data = df.filter(['Close'])

dataset = data.values

training_data_len = int(np.ceil( len(dataset) ))
```

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)
train_data = scaled_data[0:int(training_data_len), :]
x_train = []
y_train = []

for i in range(60, len(train_data)):
    x_train.append(train_data[i-60:i, 0])
    y_train.append(train_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))
from keras.models import Sequential
from keras.layers import Dense, LSTM,Bidirectional,Dropout,Flatten

# Build the LSTM model
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=
(x_train.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(50, return_sequences=False))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128))
model.add(Dense(1))
```

```
# model = Sequential()

# model.add(LSTM(16, return_sequences=True, activation='linear',
input_shape= (x_train.shape[1], 1)))

# model.add(LSTM(32, return_sequences=True, activation='linear'))

# model.add(LSTM(64, return_sequences=False, activation='linear',
input_shape= (x_train.shape[1], 1)))

# model.add(Flatten())

# model.add(Dense(128))

# model.add(Dense(1))



model.compile(optimizer='adam',
loss='mean_squared_error',metrics=[tf.keras.metrics.Precision(),tf.keras.
metrics.Recall()])

model.fit(x_train, y_train, batch_size=32, epochs=50)

model.save('./Models/model_TSLA.h5')

print("Model saved successfully")
```

<u>WEBPAGE INTEGRATION</u>

FRONT END:

```
<!DOCTYPE html>
<html>
<head>
  <title>Stock Prediction</title>
  <style>
    *
    {
    }
    body {
      /* background-color: #f2f2f2;
      margin: 0;
      padding: 0; */
    }
    body{
      background-image: url("{{ url_for('static', filename='image.jpg') }}");
            background-size: cover;
            background-repeat: no-repeat;
            /* Additional styling */
            margin: 0;
            padding: 0;
            height: 100vh;;
            background-size: cover;
            background-repeat: no-repeat;
```

```
        /* Additional styling */
        margin: 0;
        padding: 0;
        height: 100vh;
}
.container {
  font-family: "Helvetica Neue", Helvetica, Arial, sans-serif;
  max-width: 600px;
  margin: 0 auto;
  padding: 20px;
  background-color: #ffffff;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}
.discription
{
  font-family:"Tahoma";
  position:absolute;
  bottom: 0px;
  padding-bottom: 0%;
  border-radius: 20px;
  background-color: #ffffff;
  border:solid;
  border-color:rgb(34, 76, 194);
  color: red;
}
```

```
    h1 {
      color: #333;
      text-align: center;
    }
    select {
      display: block;
      width: 100%;
      padding: 10px;
      margin-bottom: 20px;
      font-size: 16px;
      border-radius: 5px;
      border: 1px solid #ccc;
    }
    .result {
      text-align: center;
      font-size: 18px;
      color: #333;
    }
    .timestamp {
      text-align: center;
      color: #999;
      font-size: 14px;
    }
  </style>
</head>
```

```
<body>

  <div class="container">

    <h1 style="color: crimson;">STOCK PRICE PREDICTION</h1>

    <select id="newContent">

      <option >Select the company</option>

      <option name="MICROSOFT" value="MSFT">Microsoft</option>

      <option name="TESLA" value="TSLA">Tesla</option>

      <option name="APPLE" value="AAPL">Apple</option>

      <option name="GOOGLE" value="GOOGL">Google</option>

      <option name="ALIBABA" value="BABA">Alibaba</option>

      <option name="AMAZON" value="AMZN">Amazon</option>

  </select><center>

  <input style="color: rgb(255, 255, 255); background-color: rgb(0, 47,
255);" onclick="updateContent()" type="submit" value="Forecast">

    <!-- </form> -->

    <p id="content">After submition please wait for a
moment</p></center>

    <div class="result" id="result"></div>

    <div class="timestamp" id="timestamp"></div>

  </div>

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script>

        function updateContent() {

            var newContent =
document.getElementById('newContent').value;

            $.ajax({
```

```
            url: '/update_content',

            type: 'POST',

            data: {new_content: newContent},

            success: function(response) {

                document.getElementById('content').innerHTML = "The
predicted price is <strong>"+response+"</strong>
Dollars<br><br>"+"**<strong>Disclaimer</strong>**<br> The price
Might not be accurate in all the cases.Invest on your own risk";

            }

        });

    }

    </script>

  <script>

   // Get the select element

   var select = document.getElementById('newContent');


   // Get the result element

   var result = document.getElementById('result');


   // Get the timestamp element

   var timestamp = document.getElementById('timestamp');


   // Event listener for the select element

   select.addEventListener('change', function() {

     result.innerHTML = 'You selected: ' + select.value;
```

```
      timestamp.innerHTML = 'Timestamp: ' + new
Date().toLocaleString();
    });
  </script>
<h3 style="background-color: #ffff;" class="discription">
  <marquee><span>Investment on stock market is subject to your own
  risk please analyze the market consistantly and make investment wisely.
  The proposed model will help you to forecast the price but it will not be
  accurate in all instance</span></h3>
</body>
</html>
```

APPLICATION PROGRAMMING INTERFACE:

```
import tensorflow as tf
import numpy as np
from flask import Flask,redirect,url_for,render_template,request
import uvicorn
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
from datetime import datetime as dt
import webbrowser
#Fetch historical stock data
app = Flask(__name__)
@app.route('/update_content', methods=['POST'])
def update_content():
```

```
new_content = request.form['new_content']
ticker = new_content
start_date = "2022-05-16"
end_date = "2023-05-21"
data = yf.download(ticker, start=start_date, end=end_date)
data = data[['Close']]
dataset = data.values.astype('float32')
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(dataset)
test_size = int(len(scaled_data))
test_data = scaled_data
inputs = data[len(data) - len(test_data):].values
inputs = inputs.reshape(-1, 1)
inputs = scaler.transform(inputs)
window_size = 60
X_test = []
for i in range(window_size, len(inputs)):
    X_test.append(inputs[i - window_size:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1]))
app=Flask(__name__)
match ticker:
    case "GOOGL":
        MODEL =
tf.keras.models.load_model(f"./Models/model_GOOGL.h5" )
```

```
        case "MSFT":
            MODEL =
  tf.keras.models.load_model(f"./Models/model_MSFT.h5" )
        case "TSLA":
            MODEL =
  tf.keras.models.load_model(f"./Models/model_TSLA.h5" )
        case "AAPL":
            MODEL =
  tf.keras.models.load_model(f"./Models/model_AAPL.h5" )
        case "BABA":
            MODEL =
  tf.keras.models.load_model(f"./Models/model_BABA.h5" )

    predictions = MODEL.predict(X_test)
    predictions = scaler.inverse_transform(predictions)
    predictions = predictions.tolist()
    msg = predictions[-1]
    return msg

@app.route('/', methods=['GET', 'POST'])
def redirect_to_html():
    if request.method == 'POST':
        selected_option = request.form.get('option')
        return f"You selected: {selected_option}"
    return render_template('index.html')
```

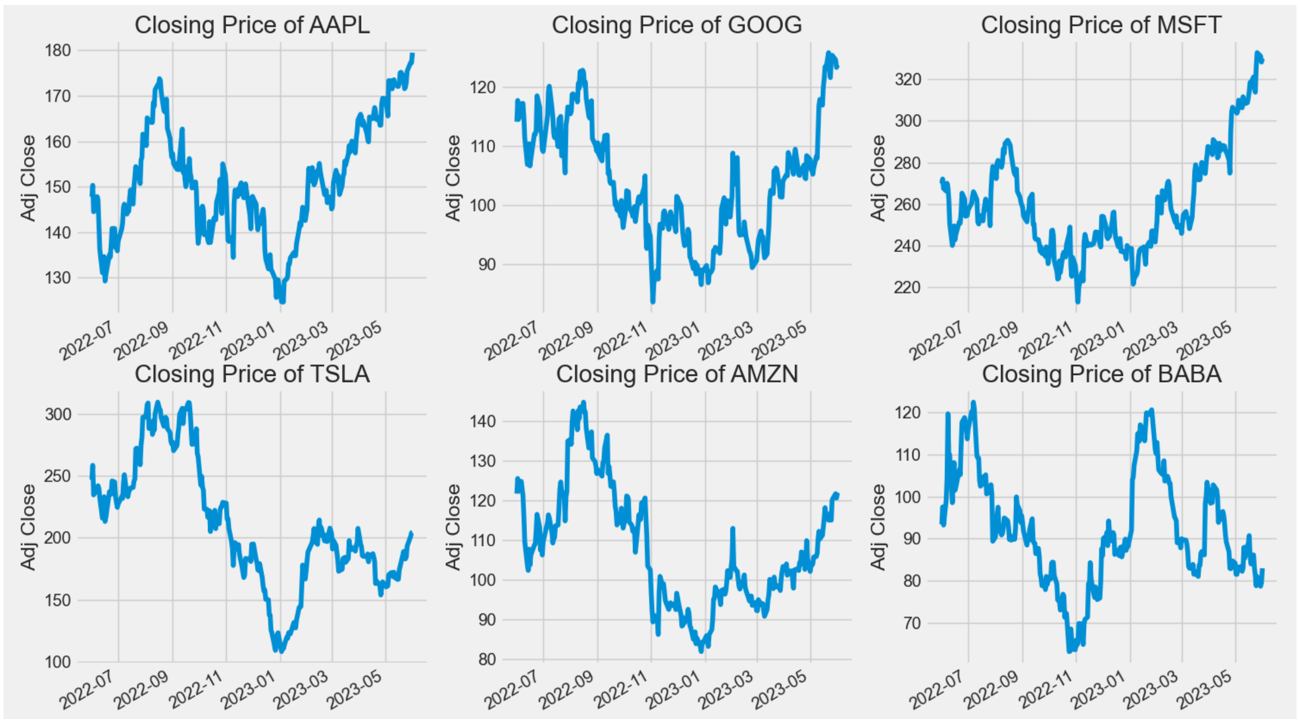
```
if __name__=="__main__":
    app.run(debug=True)
```

APPENDIX II

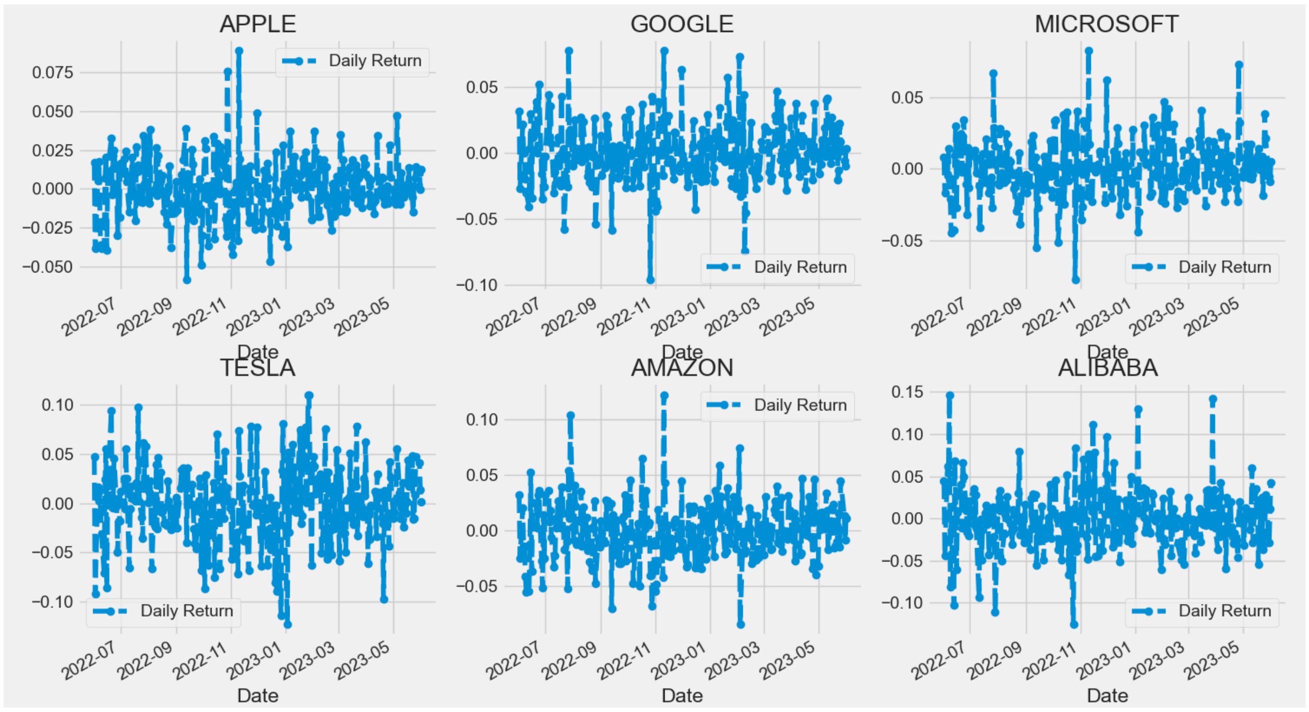
CHAPTER 9

APPENDIX II

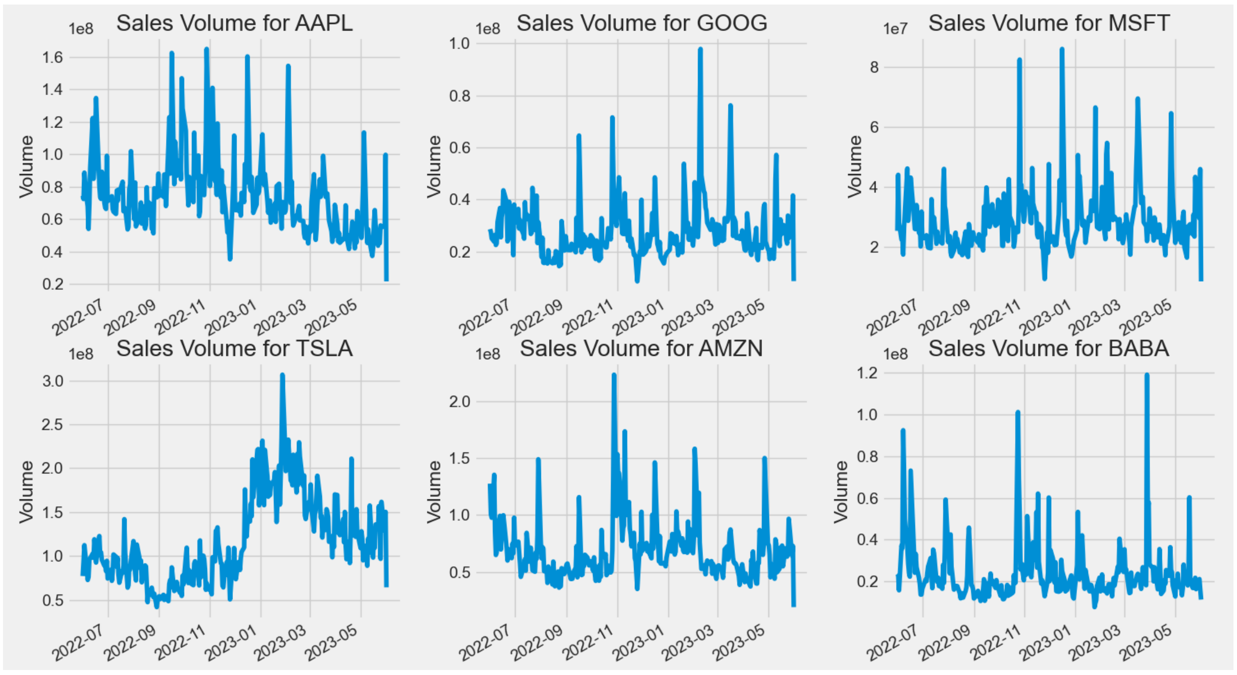
OUTPUT:



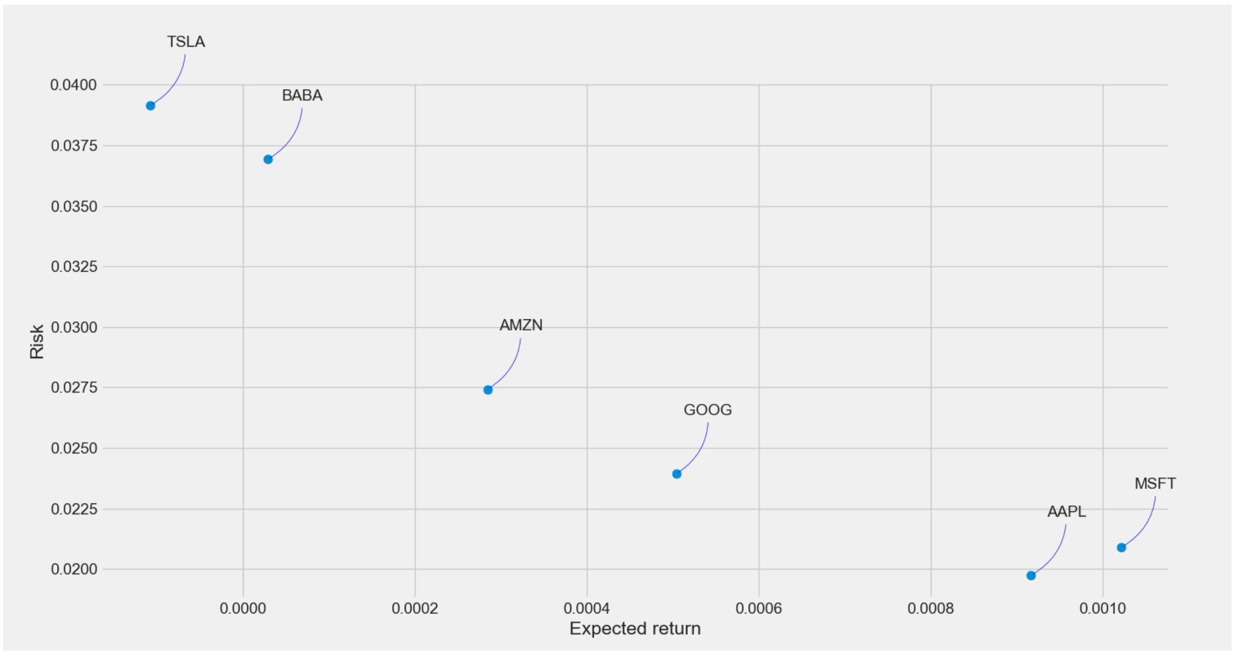
DAILY RETURN



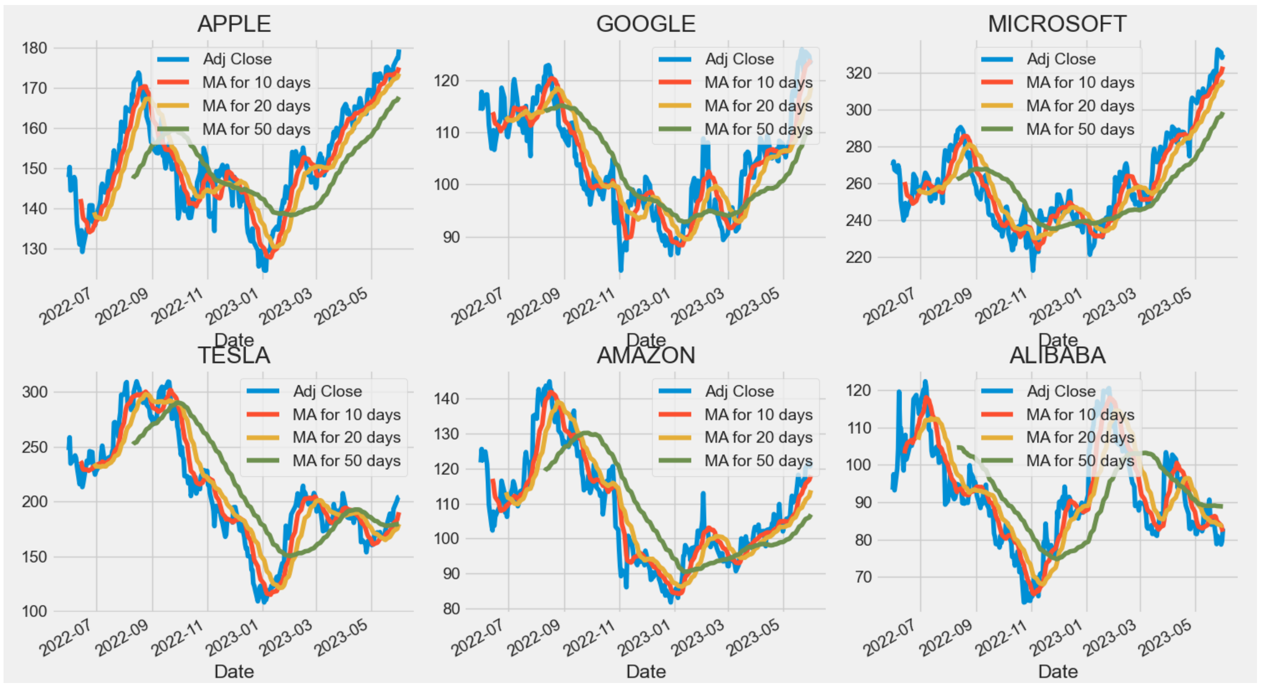
SALES VOLUME



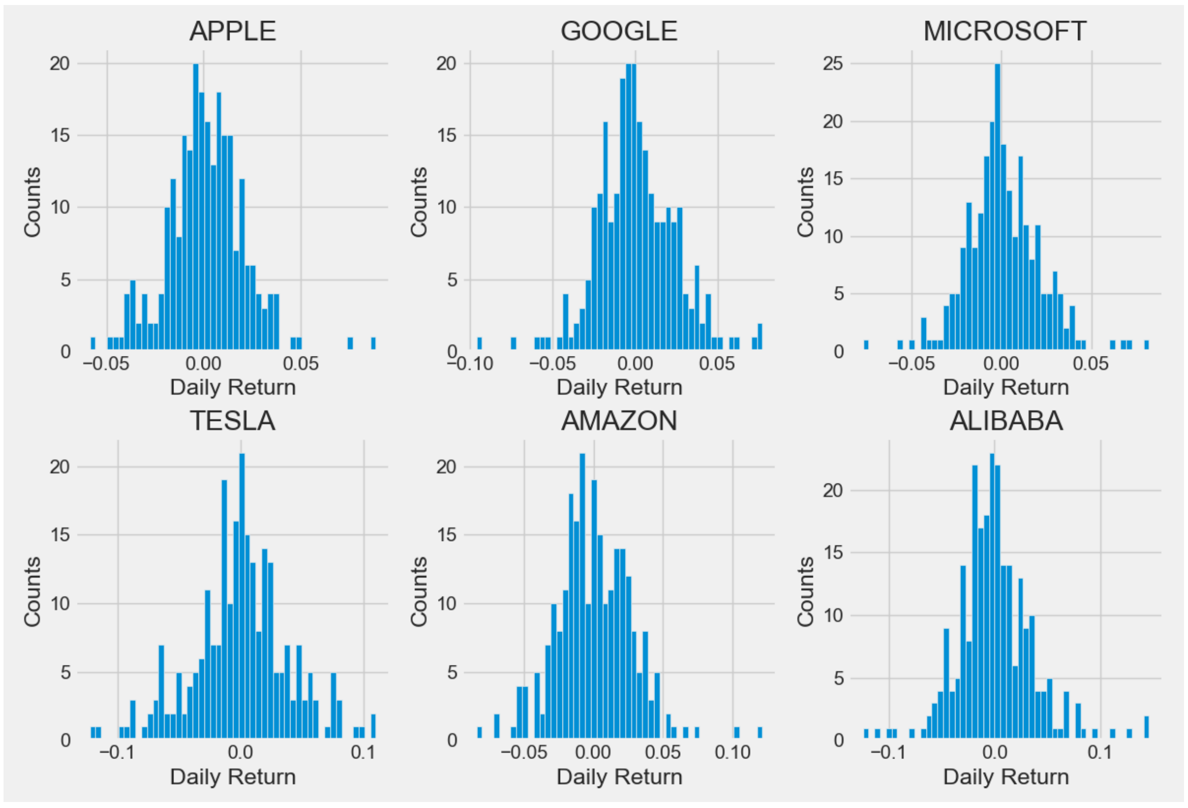
RISK ANALYSIS:



MOVING AVERAGE

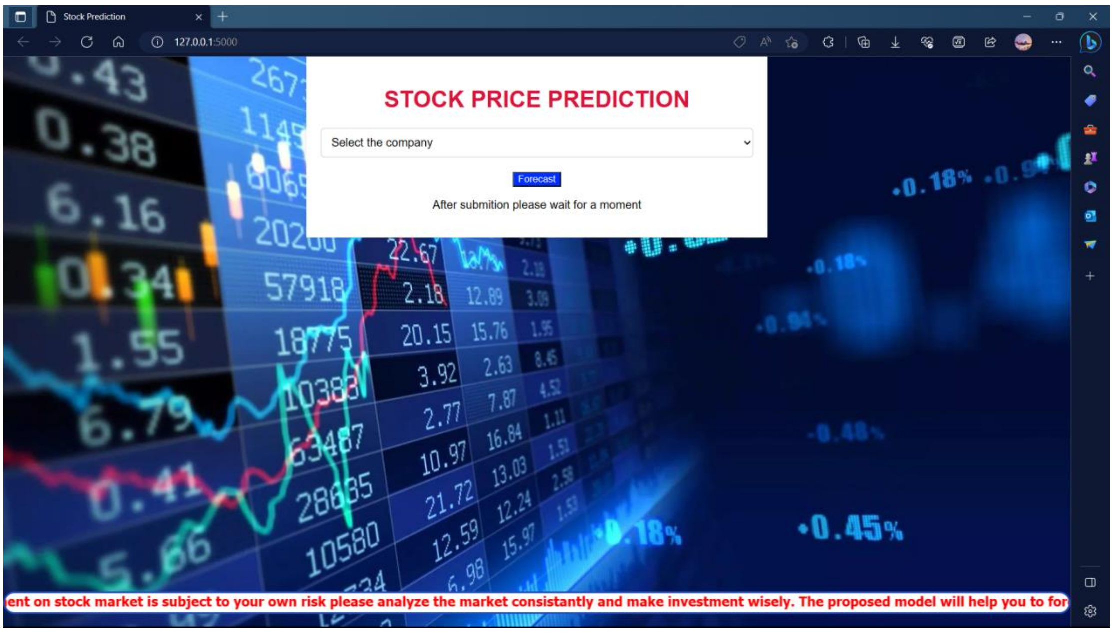


DAILY RETURN

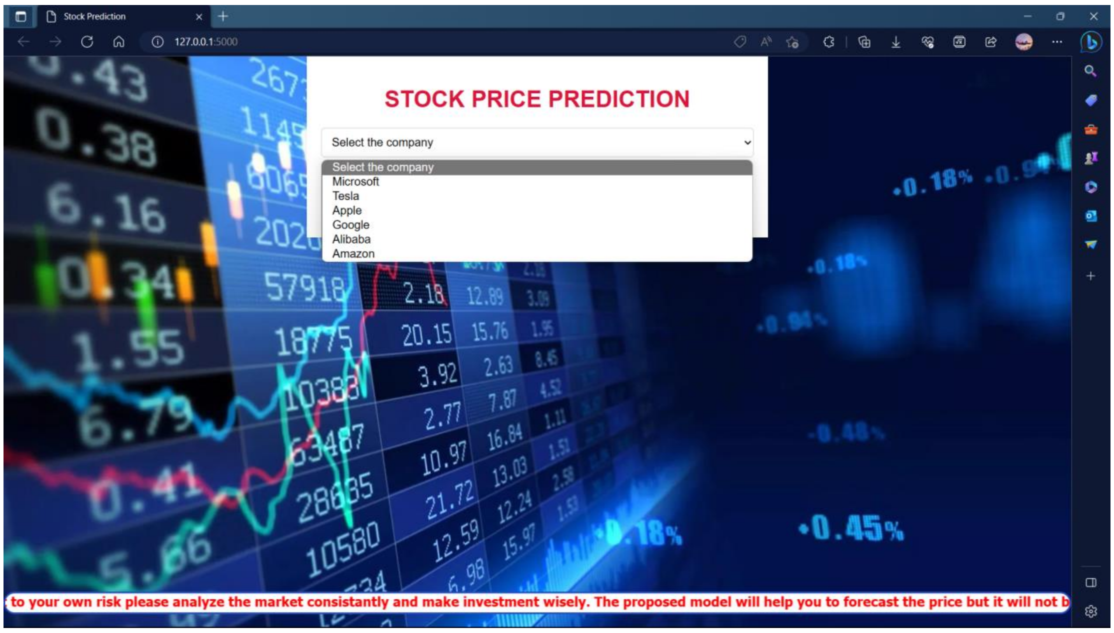


WEBPAGE

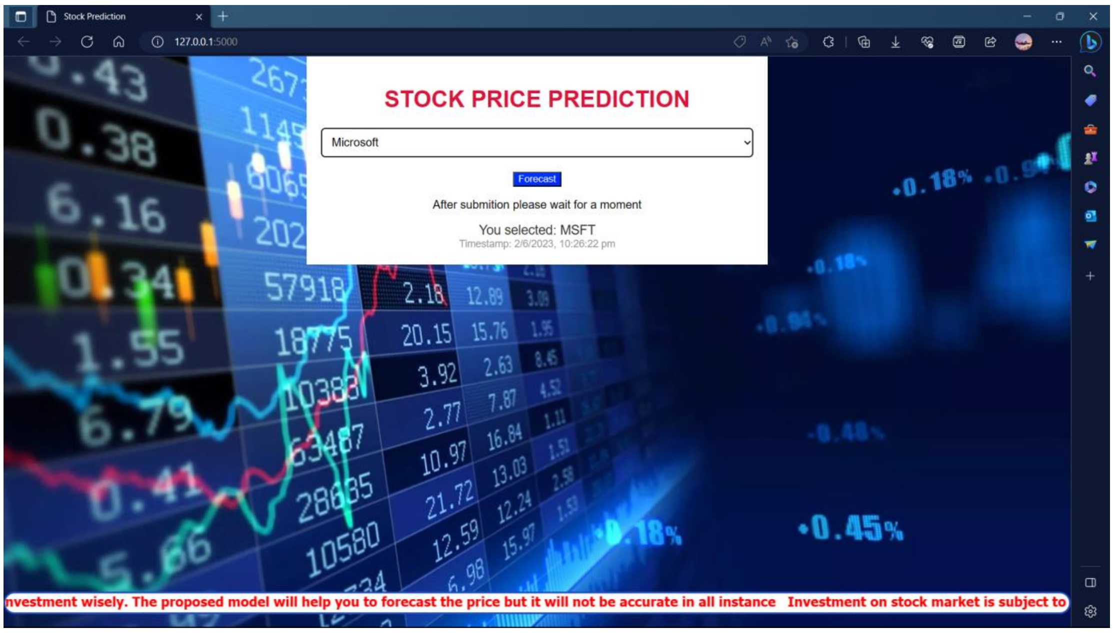
BLANK WEB PAGE WHEN USER VISIT :


Stock Prediction
127.0.0.1:5000
STOCK PRICE PREDICTION
Select the company
Forecast
After submition please wait for a moment
ent on stock market is subject to your own risk please analyze the market consistantly and make investment wisely. The proposed model will help you to for

ALLOW THE USER TO SELECT THE COMPANY FROM THE GIVEN LIST:


Stock Prediction
127.0.0.1:5000
STOCK PRICE PREDICTION
Select the company
Select the company
Microsoft
Tesla
Apple
Google
Alibaba
Amazon
to your own risk please analyze the market consistantly and make investment wisely. The proposed model will help you to forecast the price but it will not b

PROCESS UPON THE SELECTED INPUT :


Stock Prediction

127.0.0.1:5000

STOCK PRICE PREDICTION

Microsoft

Forecast

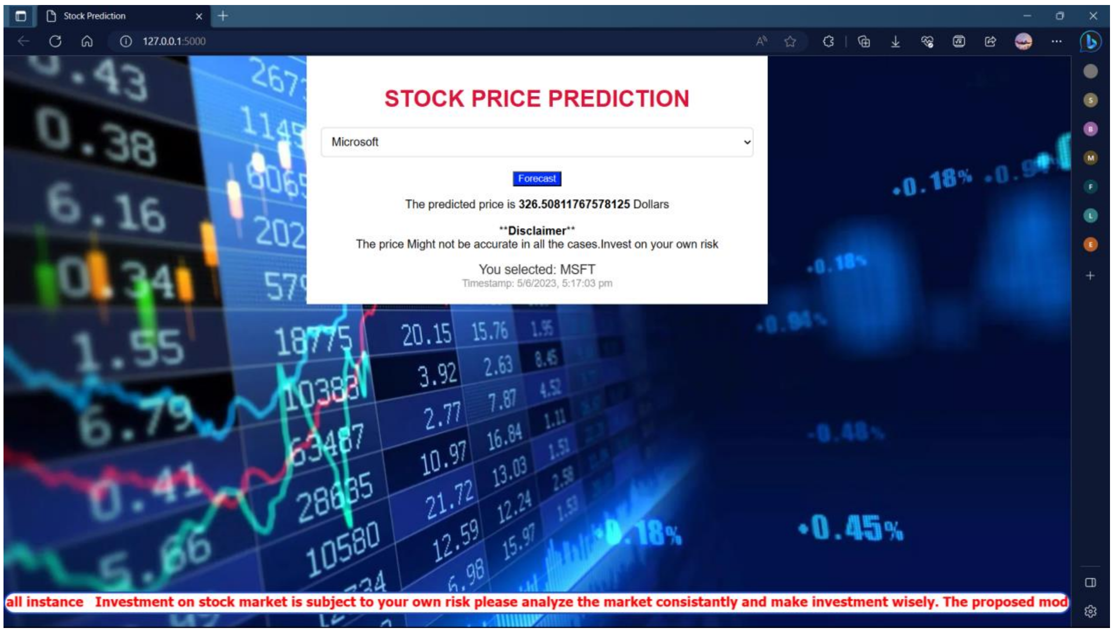
After submition please wait for a moment

You selected: MSFT

Timestamp: 2/6/2023, 10:26:22 pm

nvestment wisely. The proposed model will help you to forecast the price but it will not be accurate in all instance Investment on stock market is subject to

DISPLAY THE RESULT :


Stock Prediction

127.0.0.1:5000

STOCK PRICE PREDICTION

Microsoft

Forecast

The predicted price is **326.50811767578125** Dollars

Disclaimer

The price Might not be accurate in all the cases.Invest on your own risk

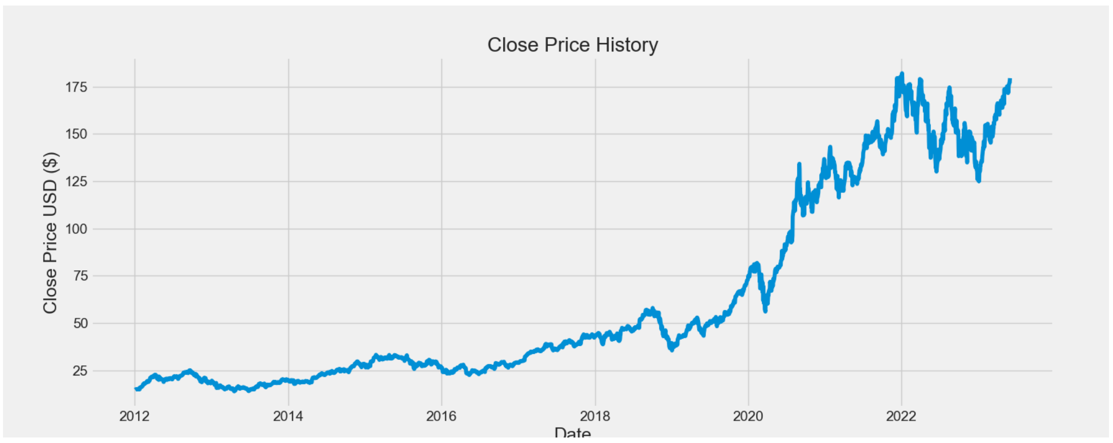
You selected: MSFT

Timestamp: 5/6/2023, 5:17:03 pm

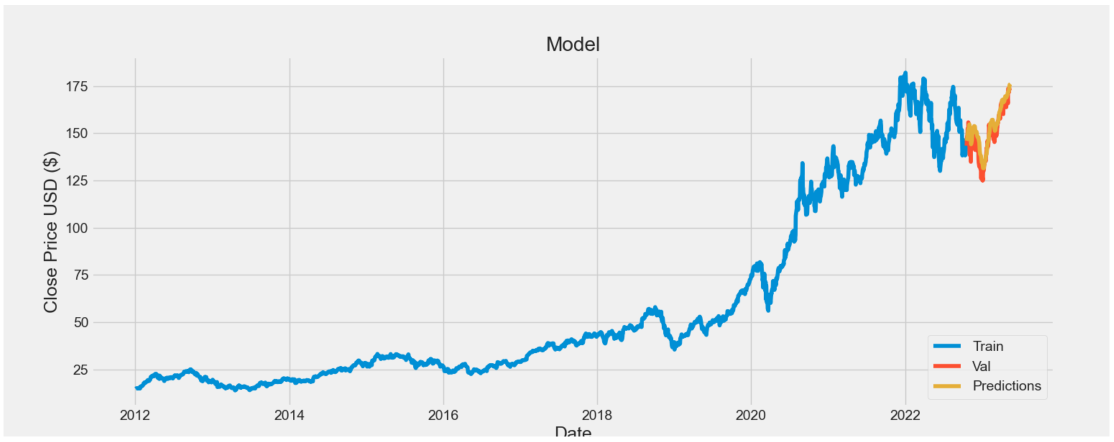
all instance Investment on stock market is subject to your own risk please analyze the market consistantly and make investment wisely. The proposed mod

VALIDATION

DAILY DATA ANALYSIS



MODEL PREDICTION ANALYSIS



REFERENCES

CHAPTER 10

REFERENCES

[1] Umer, M., Awais, M., & Muzammul, M. (2019). Stock market prediction using machine learning (ML) algorithms. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 8(4), 97-116.

[2] Mokhtari, S., Yen, K. K., & Liu, J. (2021). Effectiveness of artificial intelligence in stock market prediction based on machine learning. arXiv preprint arXiv:2107.01031.

[3] Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W., & Wills, G. (2020). A survey on machine learning for stock price prediction: Algorithms and techniques.

[4] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. Journal of Big Data, 7(1), 1-40

[5] Koukaras, P., Nousi, C., & Tjortjis, C. (2022, May). Stock Market Prediction Using Microblogging Sentiment Analysis and Machine Learning. In Telecom (Vol. 3, No. 2, pp. 358-378). MDPI.

[6] Strader, T. J., Rozycki, J. J., Root, T. H., & Huang, Y. H. J. (2020). Machine learning stock market prediction studies: review and research directions. Journal of International Technology and Information Management, 28(4), 63-83.

[7] Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., & Kim, H. C. (2021). Stock market prediction using machine learning techniques: a decade survey

on methodologies, recent developments, and future directions. Electronics, 10(21), 2717.

[8] Umer, M., Awais, M., & Muzammul, M. (2019). Stock market prediction using machine learning (ML) algorithms. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 8(4), 97-116

[9] Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2020). Stock market prediction using machine learning classifiers and social media, news. Journal of Ambient Intelligence and Humanized Computing, 1-24.

[10] Pahwa, K., & Agarwal, N. (2019, February). Stock market analysis using supervised machine learning. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon) (pp. 197-200). IEEE.

[11] Kanade, P. A., Singh, S., Rajoria, S., Veer, P., & Wandile, N. (2020). Machine learning model for stock market prediction. International Journal for Research in Applied Science and Engineering Technology, 8(6), 209-216.

[12] Vazirani, S., Sharma, A., & Sharma, P. (2020, October). Analysis of various machine learning algorithm and hybrid model for stock market prediction using python. In 2020 International conference on smart technologies in computing, electrical and electronics (ICSTCEE) (pp. 203-207). IEEE.

[13]Rao, P. S., Srinivas, K., & Mohan, A. K. (2020). A survey on stock market prediction using machine learning techniques. In ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications (pp. 923-931). Springer Singapore.

[14] Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., & Kim, H. C. (2021). Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. Electronics, 10(21).

[15] Zhang, J., Li, L., & Chen, W. (2021). Predicting stock price using two-stage machine learning techniques. Computational Economics, 57, 1237-1261.

[16] Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh, S., Aich, S., & Kim, H. C. (2021). Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. Electronics, 10(21), 2717.

[17] Rao, P. S., Srinivas, K., & Mohan, A. K. (2020). A survey on stock market prediction using machine learning techniques. In ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications (pp. 923-931). Springer Singapore.

[18] Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., & Salwana, E. (2020). Deep learning for stock market prediction. Entropy, 22(8), 840.

[19] Strader, T. J., Rozycki, J. J., Root, T. H., & Huang, Y. H. J. (2020). Machine learning stock market prediction studies: review and research directions. Journal of International Technology and Information Management, 28(4), 63-83.

[20] Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2020). Stock market prediction using machine learning classifiers and social media, news. Journal of Ambient Intelligence and Humanized Computing, 1-24.