CHAPTER 1:

INTRODUCTION

A stock market is a public market where you can buy and sell shares for publicly listed companies. The stocks, also known as equities, represent ownership in the company. The stock exchange is the mediator that allows the buying and selling of shares. Stock price analysis has been a critical area of research and is one of the top applications of machine learning.

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy.

1.1 BACKGROUND

The prediction of stock prices has always been a challenging task due to the complexity of the financial markets and the numerous factors that can influence stock prices. In recent years, the use of machine learning algorithms has become popular in predicting stock prices due to their ability to handle large amounts of data and identify patterns that are difficult to detect using traditional statistical models.

One such machine learning algorithm that has gained popularity in recent years is the Long Short-Term Memory (LSTM) algorithm. LSTM is a type of Recurrent Neural Network (RNN) that is capable of modelling long-term dependencies in time-series data. This makes it well-suited for stock price prediction tasks, as stock prices are highly dependent on their historical values.

The project will be implemented using the Python programming language and the Keras deep learning library. The dataset used for training and testing the model will be obtained from a reliable financial data provider such as Yahoo Finance. The LSTM model will be optimized using various hyperparameters tuning techniques to achieve the best possible performance.

The expected outcome of this project is a trained LSTM model that can accurately predict the stock prices of a given company. This model can be used by investors to make informed investment decisions based on the predicted future stock prices. Furthermore, the project can also serve as a basis for further research on the application of deep learning algorithms to stock price prediction

1.2 OBJECTIVES

- The LSTM model should be able to accurately predict the future stock prices with minimal error.
- The LSTM model should be able to generalize to new, unseen data and not just perform well on the training data.
- The LSTM model should be able to handle different types of market conditions and respond to any sudden changes.
- The LSTM model should provide insights into the factors that are driving the stock prices and how they are related.
- The LSTM model should be able to make predictions quickly, so that traders can make timely decisions based on the predictions.
- The LSTM model should be able to adapt to changes in market conditions and update its predictions accordingly.
- The LSTM model should be able to handle large volumes of data and be scalable for use with different stocks and markets.

1.3 PURPOSE, SCOPE, AND APPLICABILITY

1.3.1 PURPOSE

The main purpose of this project is to predict the stock prices of a given company using the LSTM algorithm. The project aims to develop a model that can accurately predict future stock prices based on historical data. The model will be trained using a dataset of historical stock prices, and the performance of the model will be evaluated based on its ability to predict future stock prices. The model will provide insights into the market trends and investor sentiment, and enable investors and traders to make informed decisions about their investments.

1.3.2 SCOPE

The scope of this project will include the collection and pre-processing of historical stock market data, the development and training of the LSTM model, and the evaluation of the model's performance. The data used in the model will include stock prices, trading volumes, and other market indicators. The LSTM model will be developed using a variety of techniques, including feature engineering and model selection. The evaluation of the model will involve measuring its accuracy and performance against a benchmark model.

1.3.3 APPLICABILITY

The applicability of this project is to investors and traders across various industries who are looking to make informed decisions about their stock market investments. The model can be used to predict future stock prices, identify market trends and patterns, and provide insights into profitable trading strategies. The insights provided by the model can also be used to optimize investment portfolios and manage investment risks. The project can be extended to include other markets and assets, such as cryptocurrencies or commodities, that exhibit similar market behaviours and trends.

1.4 ACHIEVEMENTS

- **Improved accuracy:** LSTM-based models have been shown to outperform traditional statistical models in predicting stock prices, resulting in more accurate predictions and reduced errors
- **Increased efficiency:** Machine learning models can process large amounts of data much faster than humans, enabling traders and investors to make more informed decisions in a shorter amount of time.
- **Better risk management:** Accurate stock price predictions can help investors and traders manage their investment risks more effectively, by enabling them to identify potential market downturns and adjust their portfolios accordingly.
- **Profitable trading strategies:** LSTM-based models can identify profitable trading strategies by analysing patterns and trends in the stock market data, leading to more successful trades and increased profits.
- **Improved portfolio optimization:** The insights provided by LSTM-based models can help investors optimize their portfolios by identifying the best-performing stocks and minimizing risks.
- **Enhanced market intelligence:** LSTM-based models can provide insights into market trends and investor sentiment, enabling traders and investors to stay up-to-date with the latest developments in the stock market.

1.5 ORGANIZATION OF REPORT

- Chapter 2 contains a literature survey that provides a summary of individual paper.
- Chapter 3 provides an overview of requirement specifications, planning and scheduling. This also include software and hardware requirements.
- Chapter 4 presents system designs and its results, tools and technology used to achieve this and dataset detail.
- Chapter 5 contains details regarding testing approaches such as unit testing and integration testing.
- Chapter 6 contains a conclusion about stock price prediction and future work about what we are wanted to do in future.

CHAPTER 2:

SURVEY OF TECHNOLOGIES

2.1 LITERATURE SURVEY:

1) **Stock Price Forecasting Using Data from Yahoo Finance and Analyzing Seasonal and Nonseasonal Trend:**

   Publication Year: 2018

   Author: Jai Jagwani, Hardik Sachdeva, Manav Gupta, Alka Singhal

   Journal Name: 2018 IEEE

   Summary: To identify the relationship between different existing time series algorithms namely ARIMA and Holt Winter and the stock prices is the main objective of the proposed work, for the investments a good risk-free range of stock prices are analyzed and therefore better accuracy of the model can be seen. To find distinguished results for shares in the stock market, the combination of two different time series analysis models is opted by producing a range of prices to the consumer of the stocks. Not complex in nature and estimation of values which are purely based on the past stock prices for non-seasonal or seasonal is the main advantage of these models. In this experiment, some limitations are, the work that never takes into consideration and other circumstances like news about any new market strategy or media release relevant to any company which may get affected by the prices of stocks.

2) **Stock Market Prediction Using Machine Learning:**

   Publication Year: 2018

   Author: Ishita Parmar, Ridam Arora, Lokesh Chouhan, Navanshu Agarwal, Shikhin Gupta, Sheirsh Saxena, Himanshu Dhiman.

   Journal Name: 2018 IEEE

   Summary: In this paper studies, the use [3] of Regression and LSTM based Machine learning to forecast stock prices. Factors measured are open, close, low, high and volume. This paper was an attempt to determine the future prices of the stocks of a company with improved accuracy and reliability using machine learning techniques. LSTM algorithm resulted in a positive outcome with more accuracy in predicting stock prices.

3) **Share Price Prediction using Machine Learning Technique:**

   Publication Year: 2018

   Author: Jeevan B, Naresh E, Vijaya Kumar B P, Prashanth Kambli

   Journal Name: 2018 IEEE

   Summary: This paper is mostly [5] based on the approach of predicting the share price using Long Short-Term Memory (LSTM) and Recurrent Neural Networks (RNN) to forecast the stock value on NSE data using various factors such as current market price, price-earnings ratio, base value and other anonymous events. The efficiency of the model is analysed by comparing the true data and the predicted data using an RNN graph. Machine learning to predict stock price as see the model is able to predict the stock price very close to the actual price where this model captures the detailed feature and uses different strategies to make a prediction. The model train for all the NSE data from the internet and recognize the input and group them and provide input according to the user configuration this RNN based architecture proved very efficient in forecasting the stock price by changing the configuration accordingly which also use backpropagation mechanism while gathering and grouping data to avoid mixing of data.

4) **Stock Market Prediction Using Machine Learning Techniques:**

   Publication Year: 2016

   Author: Mehak Usmani, Syed Hasan Adil, Kamran Raza, Syed Saad Azhar Ali

   Journal Name: 2016 IEEE

   Summary: The prominent aim of this study is to [6] forecast the market performance of the Karachi Stock Exchange (KSE) on day closing using machine learning algorithms. A variety of attributes as an input and forecasts market as Positive & Negative is predicted by using the predictions model. The features employed in the model are contains Oil rates, Gold & Silver rates, Interest rate, Foreign Exchange (FEX) rate, NEWS and social media feed. The machine learning algorithms including Single Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), Radial Basis Function (RBF) and Support Vector Machine (SVM) are compared. The algorithm MLP that is multi-layer perceptron performed best as compared to different methods. The foremost helpful feature in predicting the market was the oil rate attribute. The end results of this research confirm that machine learning techniques have the ability to predict the stock market performance.

The Multi-Layer Perceptron algorithm of machine learning predicted 70% correct market performance.

5) **Forecasting stock price in two ways based on LSTM neural network:**

Publication Year: 2019

Author: Jingyi Du, Qingli Liu, Kang Chen, Jiacheng Wang

Journal Name: 2019 IEEE

Summary: The [7] LSTM neural network is used to predict Apple stocks by consuming single feature input variables and multi-feature input variables to verify the forecast effect of the model on stock time series. The experimental results show that the model has a high accuracy of 0.033 for the multivariate input and is accurate, that is in line with the actual demand. For the univariate feature input, the predicted squared absolute error is 0.155, which is inferior to the multi-feature variable input.

CHAPTER 3:

REQUIREMENTS AND ANALYSIS

3.1 PROBLEM DEFINITION

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).

3.2 REQUIREMENTS SPECIFICATIONS

1) **Objective:** The objective of this system is to predict stock prices using machine learning LSTM algorithm.
2) **Scope:** The system will cover a range of stock exchanges and timeframes for which predictions will be made. The data sources will include historical stock prices, economic indicators, news articles, and other relevant data.
3) **Data Requirements:** The application requires data from a variety of sources, including stock exchanges, news websites, and economic data providers. The data should be updated frequently and be in a format that can be easily processed by the machine learning algorithm.
4) **Feature Selection:** The application uses a variety of features for prediction, including historical stock prices, economic indicators, and news articles. The features will be selected based on their relevance and usefulness for predicting stock prices.
5) **Algorithm Selection:** The machine learning algorithm that will be used for prediction is the LSTM algorithm, which is well-suited for time-series data analysis.
6) **Training Data:** The application uses historical stock prices and other relevant data sources for training the LSTM algorithm.
7) **Evaluation Metrics:** The system will be evaluated using standard metrics such as accuracy & precision.
8) **Deployment:** The application will be deployed on a server with appropriate hardware and software requirements. The system will need to be scalable and have proper maintenance and support procedures in place.

9) **User Interface:** The application will have a user-friendly interface that provides visualizations of the predicted stock prices. The interface should also allow for customization of the prediction model and include an alerts and notifications system.
10) **Documentation:** The system will be documented with user manuals, technical specifications, and training materials.

3.3 PLANNING AND SCHEDULING

Planning and scheduling are essential for any project,

1) **Define the project scope:** It is the first step where we define our project objectives, requirements as well as the resources required to complete the project.
2) **Defining the project activities:** After setting up certain scope of our project we divided our work among ourselves. As it is a team task management each one should have their individual responsibility to handle.
3) **Allocation of the resources:** We identified the required resources for our tasks such as hardware (operating system), software (python libraries, respective models….).
4) **Define Milestones:** Milestones should be significant events or deliverables in the project that mark its completion. So, we created the blueprints according to every week and set up a mark to complete. Hence, we are able to complete our project.
5) **Identify the risk:** Maintaining data quality and technical difficulties were major risk in our project. Because of least mean squared error algorithm we were able to clear it.
6) **Communicate and collaborate:** We discussed about our project in detail and checked for any updates.
7) **Monitor and Control:** We ensured that our project is on the track. We also checked that there is no delay or issue while running the code.

3.4 SOFTWARE AND HARDWARE REQUIREMENTS

3.4.1 SOFTWARE REQUIREMENTS

Operating System	:	Windows 11.
Program Language	:	Python and it's libraries such as streamlit and many more.
Browser	:	Chrome or any other browser.
Source	:	Yahoo Finance website.

3.4.2 HARDWARE REQUIREMENTS

Processor	:	Intel i3 or above
Hard Disk	:	250 GB
RAM	:	4 GB or more
Input Devices	:	Keyboard, Mouse

3.5 PRELIMINARY PRODUCT DESCRIPTION

The Stock Price Prediction using Machine Learning LSTM Algorithm project is a software application designed to predict the future stock prices of companies using deep learning techniques. The project utilizes the Long Short-Term Memory (LSTM) algorithm, which is a type of recurrent neural network that is capable of processing and making predictions based on sequential data.

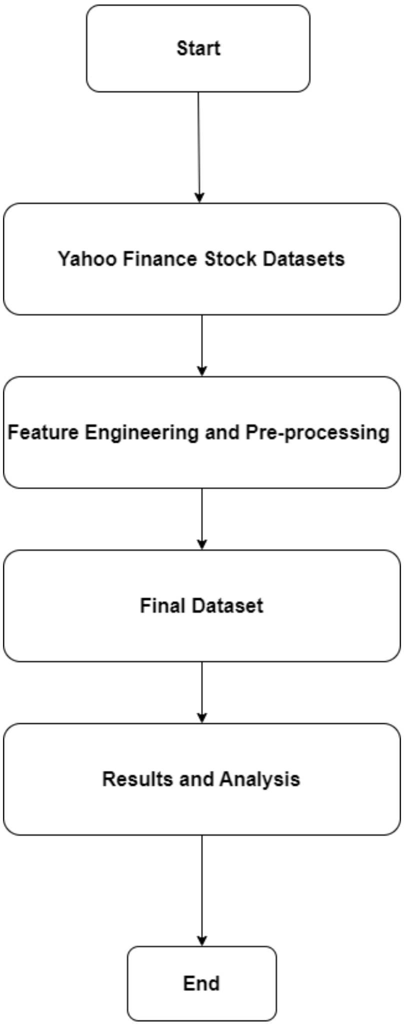
The application will take historical stock price data as input, and then preprocess and analyze the data to identify patterns and trends. The LSTM algorithm will then be trained on the preprocessed data to generate a predictive model that can forecast future stock prices. The model will be evaluated using various performance metrics to ensure its accuracy and reliability.

The application will be developed using Python, an open-source programming language, and will utilize popular deep learning libraries such as TensorFlow and Keras. The user interface will be developed using Streamlit, a web application framework for creating interactive data-driven applications. The primary goal of this project is to provide accurate and reliable predictions of stock prices, which can be used by investors, traders, and financial analysts to make informed decisions about buying, selling, or holding stocks. The project will also serve as a learning tool for those interested in the field of machine learning, particularly in the area of time series analysis and deep learning techniques.

Overall, the Stock Price Prediction using Machine Learning LSTM Algorithm project is an innovative solution that leverages the power of machine learning to predict future stock prices, helping investors and traders make informed decisions and maximizing their returns.

3.6 CONCEPTUAL MODELS

3.6.1 DATA FLOW DIAGRAM

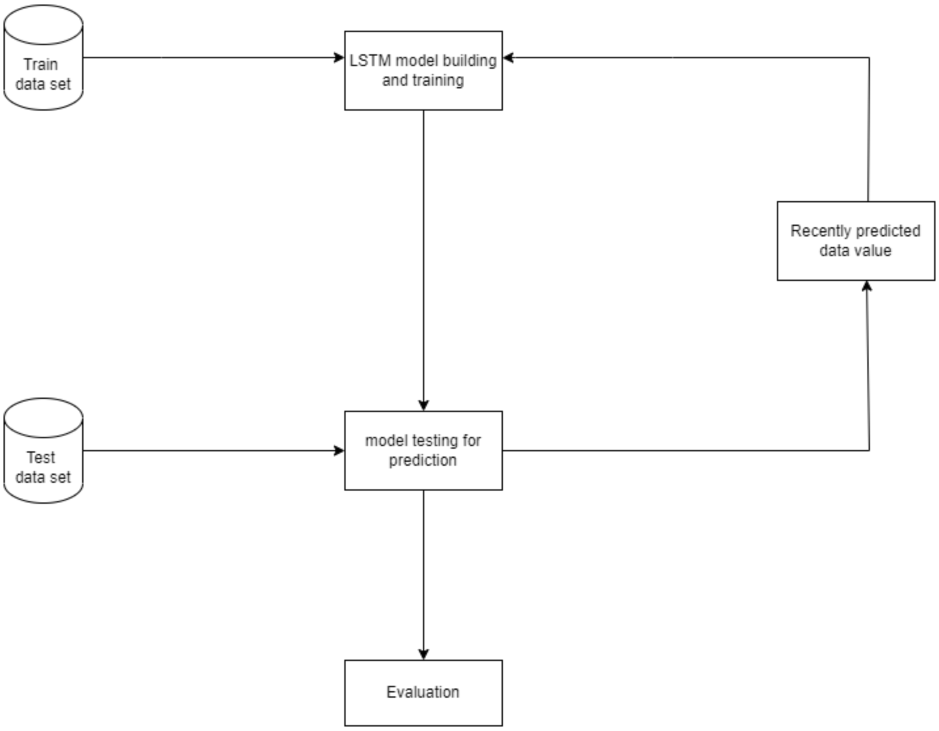

Start
Yahoo Finance Stock Datasets
Feature Engineering and Pre-processing
Final Dataset
Results and Analysis
End

A **data flow diagram** (DFD) is a graphical representation of how data flows within a system. In the context of stock price prediction using machine learning LSTM algorithm, a DFD can help to visualize the flow of data through the different stages of the prediction process.

- **Start:** The start of the DFD represents the beginning of the stock price prediction process. At this stage, the input to the system is the historical data related to the stock prices of a particular company.
- **Yahoo finance stock dataset:** In this stage, the system collects the historical data related to the stock prices of the company from yahoo finance website as financial news and stock dataset.

- **Feature engineering and Pre-processing:** In this stage, the collected data is cleaned, transformed, and prepared for analysis. This involves removing missing values, handling outliers, scaling the data, and converting it into a suitable format for LSTM model training. The Yahoo Finance stock dataset provides various features related to the company's stock prices, such as opening price, closing price, highest price, lowest price, and trading volume. These features are extracted and preprocessed for further analysis.
- **Final dataset:** The final dataset refers to the data that is used to evaluate the performance of a machine learning model after it has been trained on a separate training dataset and tuned on a validation dataset. In the context of stock price prediction using machine learning LSTM algorithm, the final dataset typically refers to a test dataset.
- **Results and Analysis:** In this stage, the predicted stock prices are visualized using charts or graphs. This can help traders or investors to understand the predicted trend and make informed decisions regarding buying or selling of stocks.

3.6.2 ARCHITECTURE DESIGN


Train data set
LSTM model building and training
Recently predicted data value
Test data set
model testing for prediction
Evaluation

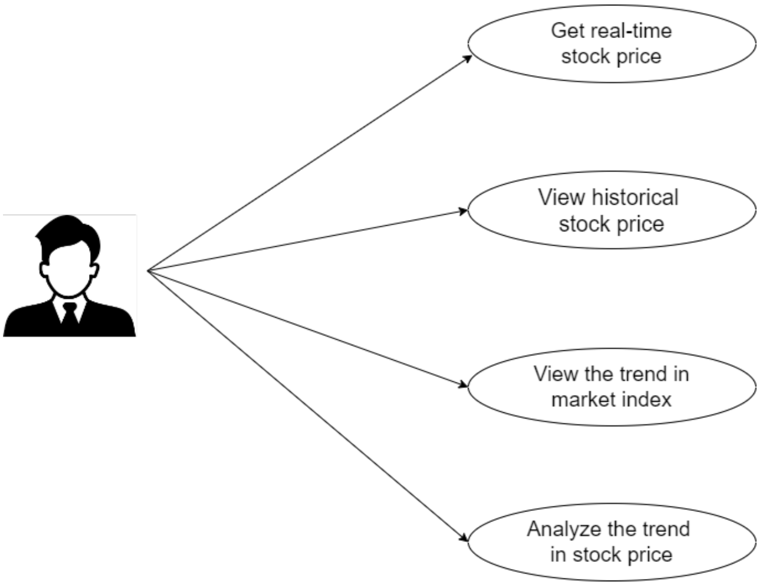
Architecture design is the process of creating a high-level structure or plan for a complex system or software application. It involves defining the components, modules, interfaces, and interactions between them, as well as the overall structure and organization of the system.

- **Train dataset:** The training dataset is a set of data used to train a machine learning model. The goal of the training dataset is to allow the model to learn patterns and relationships in the data, so that it can accurately make predictions on new, unseen data.
- **LSTM model building and training:** To build and train an LSTM model for stock price prediction, the training dataset would be fed to the model along with the hyperparameters such as the number of LSTM layers, the number of nodes in each layer, the number of dense layers, the learning rate, and so on. The model would then use the training dataset to learn the patterns and relationships in the data and make predictions on new, unseen data.
- **Recently predicted data value:** The recently predicted data value is generated by feeding the model with the most recent available data and using the learned patterns and relationships to generate a prediction for the future stock price. The model uses the most recent available data to predict the future stock price, which can be used

by investors and traders to make informed decisions about buying or selling the stock.

- **Test dataset:** The test dataset would consist of a subset of the historical data related to the stock prices of a particular company. This data would be separate from the data used to train the model and would be arranged in the same time series format as the training dataset.
- **Model testing for prediction:** Model testing is an important step in evaluating the performance of a machine learning model. The process of model testing involves evaluating the accuracy of the model's predictions on a separate test dataset that is not used in the training process. To evaluate the model performance our application uses "Mean Squared Error (MSE)".
  - **Mean Squared Error (MSE):** The average of the squared differences between the predicted values and the actual values.
- **Evaluation:** Lastly the evaluation is made, the evaluation process involves comparing the predicted values generated by the model with the actual values to assess the accuracy of the model's predictions.

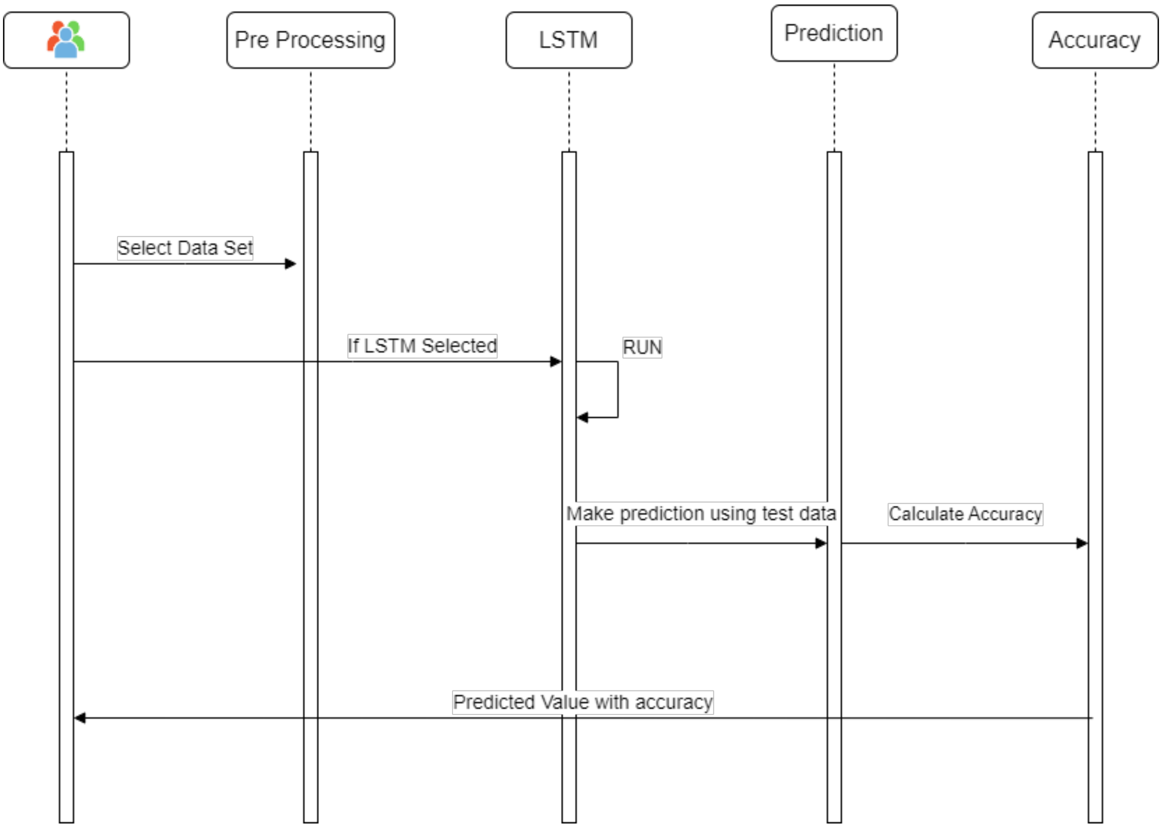
3.6.3 USE CASE DIAGRAM


Get real-time stock price
View historical stock price
View the trend in market index
Analyze the trend in stock price

A Use Case diagram is a visual representation of the interactions between actors (users) and a system in a particular domain. It is a type of UML diagram that helps to capture the functional requirements of a system and the actors who interact with it.

- **Get real-time stock price**: Firstly, the real-time stock prices are scraped from Yahoo Finance website.
- **View historical stock price:** The website provides the historical data of variety of companies publicly listed in the stock exchange. The stock prices relay on the previous values, so to make some predictions we have to consider the historical dataset.
- **View the trend in market index:** We can view the latest trends in stock market, as the application provides top 10 stock news.
- **Analyze the trend in stock price:** Analyzing the trend in the stock market involves looking at patterns in the movement of stock prices over time.

3.6.4 SEQUENCE DIAGRAM



A Sequence diagram for stock price prediction using machine learning LSTM algorithm would illustrate the interactions and flow of events between different components involved in the prediction process. Here is a possible sequence diagram:

- The user sends a request for stock price prediction to the system.
- The system retrieves the necessary stock data from the Yahoo Finance.
- The system preprocesses the data by scaling it to a range of 0 to 1, splitting it into training and testing sets, and reshaping it to fit the LSTM input format.
- The LSTM model is built and trained using the training data.
- The system uses the trained model to predict the future stock prices based on the testing data.
- The predicted stock prices are rescaled to their original values and returned to the user.
- The user can view the predicted stock prices and compare them to actual stock prices to evaluate the accuracy of the prediction.

CHAPTER 4:

SYSTEM DESIGN

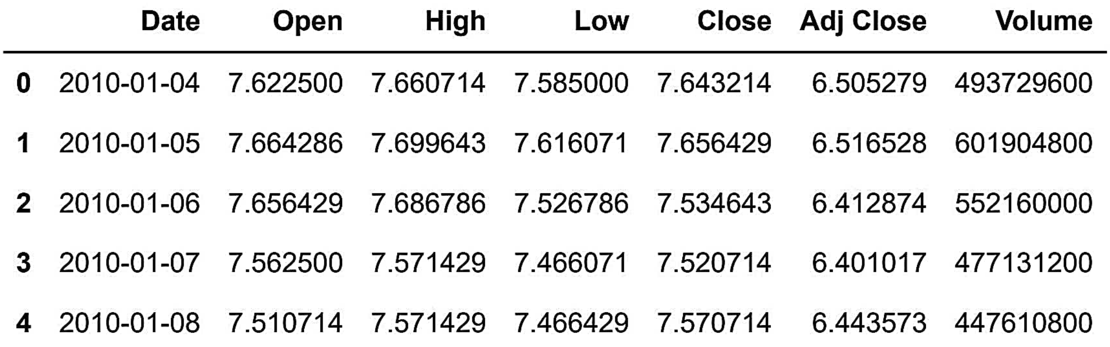
4.1 BASIC MODULES

1) **Data Collection:** Obtain historical stock data, including open, close, high, low, and volume information, for a particular stock.
2) **Data Preprocessing:** Clean and transform the data to get it into a suitable format for model training.
3) **Feature Engineering:** Extract relevant features from the data that could be used to make predictions.
4) **Model Selection:** Select a suitable machine learning algorithm, such as linear regression, decision trees, or neural networks, based on the nature of the problem and the data.
5) **Model Training:** Train the selected machine learning model on the preprocessed data using techniques such as cross-validation to ensure the model generalizes well.
6) **Model Evaluation:** Evaluate the performance of the model using metrics such as mean absolute error, mean squared error, and root mean squared error.
7) **Model Deployment:** Use the trained model to make predictions on new, unseen stock data.

4.2 DATA DESIGN PROJECT STRUCTURE

4.2.1 SCHEMA DESIGN

1. **Dataset schema design**



	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-01-04	7.622500	7.660714	7.585000	7.643214	6.505279	493729600
1	2010-01-05	7.664286	7.699643	7.616071	7.656429	6.516528	601904800
2	2010-01-06	7.656429	7.686786	7.526786	7.534643	6.412874	552160000
3	2010-01-07	7.562500	7.571429	7.466071	7.520714	6.401017	477131200
4	2010-01-08	7.510714	7.571429	7.466429	7.570714	6.443573	447610800

Column Name	Data Type	Description
id	int	Unique identifier for each record
date	date	Date of the price record
open_price	decimal	Open price for the stock on the date
high_price	decimal	Highest price for the stock on the date
low_price	decimal	Lowest price for the stock on the date
close_price	decimal	Closing price for the stock on the date
adj_close_price	decimal	Adjusted closing price for the stock on the date
volume	bigint	Trading volume for the stock on the date

2. **News Schema Design**

Column Name	Data Type	Description
news_id	int	Unique identifier for each news article (primary key)
title	varchar (255)	Title of the news article

4.2.2 DATA INTEGRITY AND CONSTRAINTS

Data integrity is the overall accuracy, completeness and consistency of data. Data integrity also refers to the safety of data in regard to regulatory compliance and security.

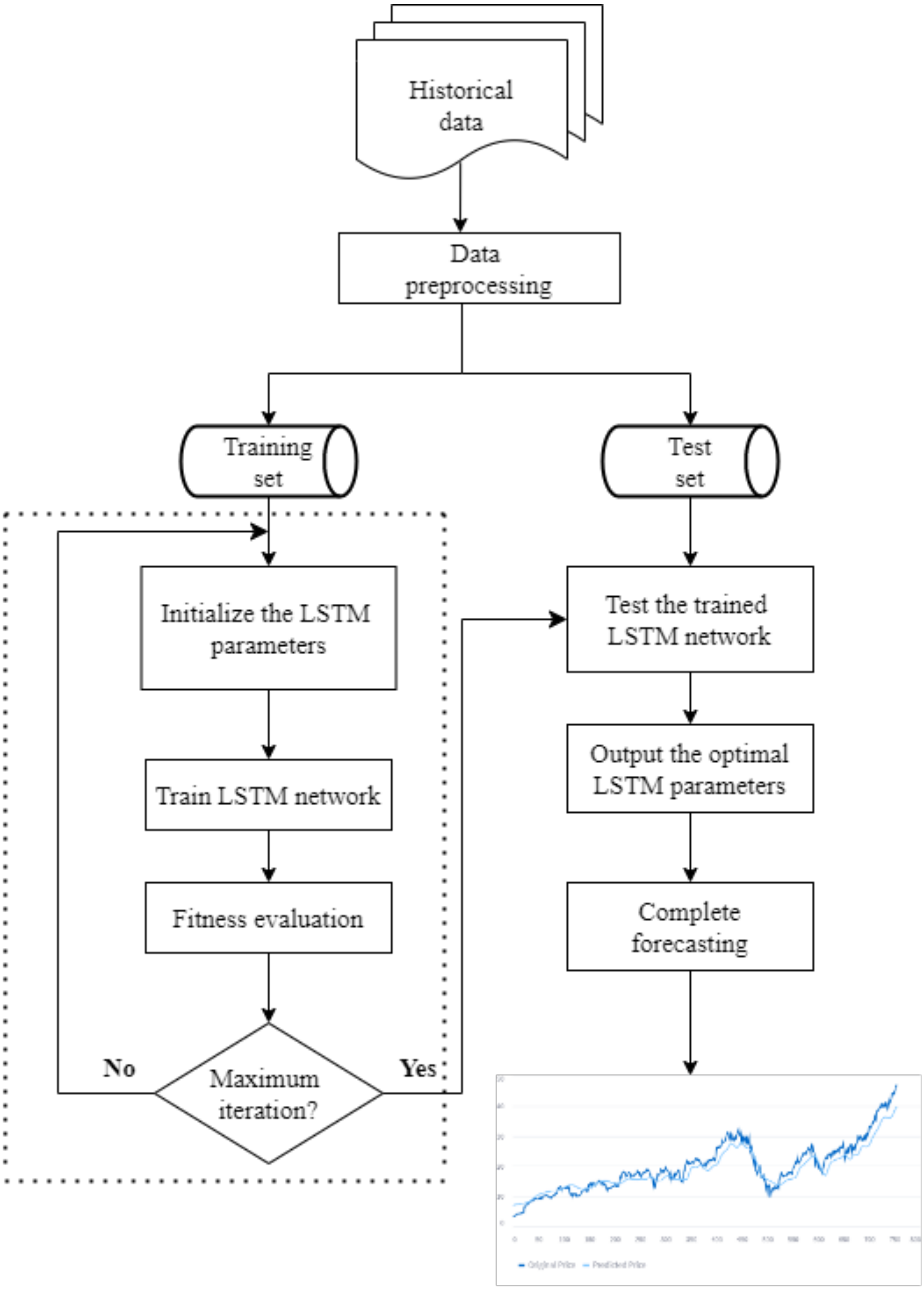
Regarding our project here are some data integrities:

1) Data input is not delayed.
2) Data is not duplicated.
3) Dates of stock is not repeated.
4) Avoid the loss of data of past years.

4.3 PROCEDURAL DESIGN

4.3.1 LOGIC DIAGRAMS

1. Flow Chart



4.3.2 DATA STRUCTURES

The Major data structure used in our project is “Array”. An Array is a linear data structure that collects elements of same data type and stores them in contiguous and adjacent memory location. Here we are using user defined data type:

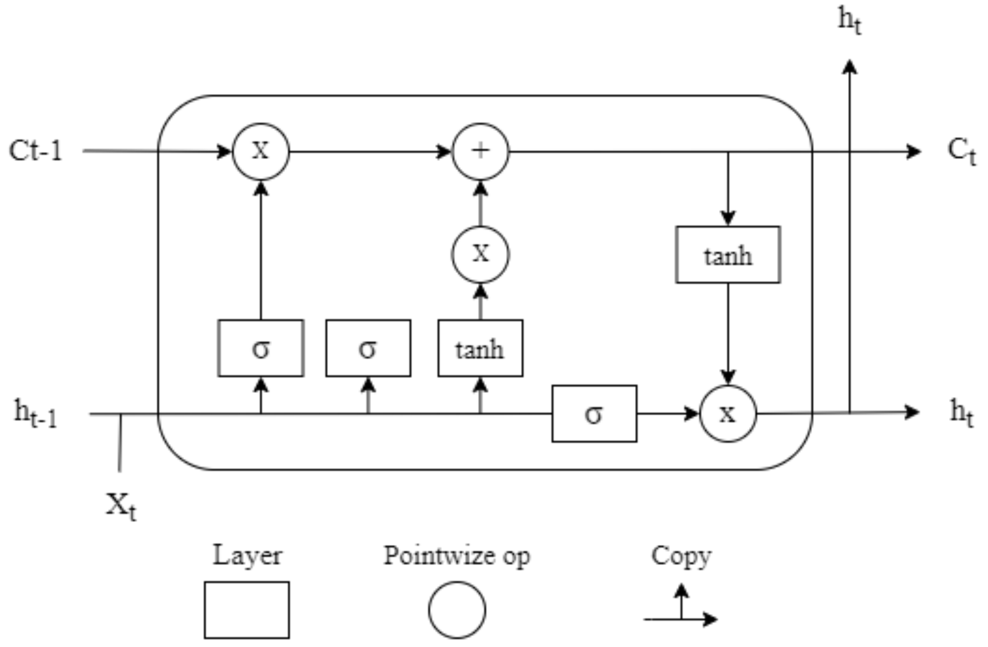
a) x_train

b) y_train

c) y_predict

4.3.3 ALGORITHMS DESIGN

LSTM Algorithm:



Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is used in deep learning for modeling sequential data. It was introduced by Hochreiter and Schmidhuber in 1997 and has become one of the most popular and effective RNN architectures due to its ability to remember long-term dependencies in data.

Architecture of LSTM:

The LSTM architecture consists of four main components:

- Cell state (Ct): It is the memory of the network and its purpose is to carry information across timesteps.
- Hidden state (ht): It is the output of the LSTM block at a particular timestep and can be used for predicting the next value in a sequence.
- Input gate (it): It controls how much new information is added to the memory at each timestep.
- Forget gate (ft): It controls how much old information is retained in the memory at each timestep.

The input gate and forget gate are used to regulate the flow of information into and out of the cell state, respectively. The output gate is used to determine how much of the cell state is used to compute the hidden state.

Training of LSTM:

The LSTM network is trained using backpropagation through time (BPTT). During training, the LSTM block is unrolled over a fixed number of timesteps and the weights of the network are updated using the gradients of the loss function with respect to the weights.

Applications of LSTM:

LSTMs are used in a variety of applications, such as:

- Time-series analysis, such as predicting stock prices and weather patterns.
- Speech recognition and synthesis.
- Natural Language Processing (NLP) tasks, such as language translation, text classification, and sentiment analysis.
- Image captioning and object recognition.

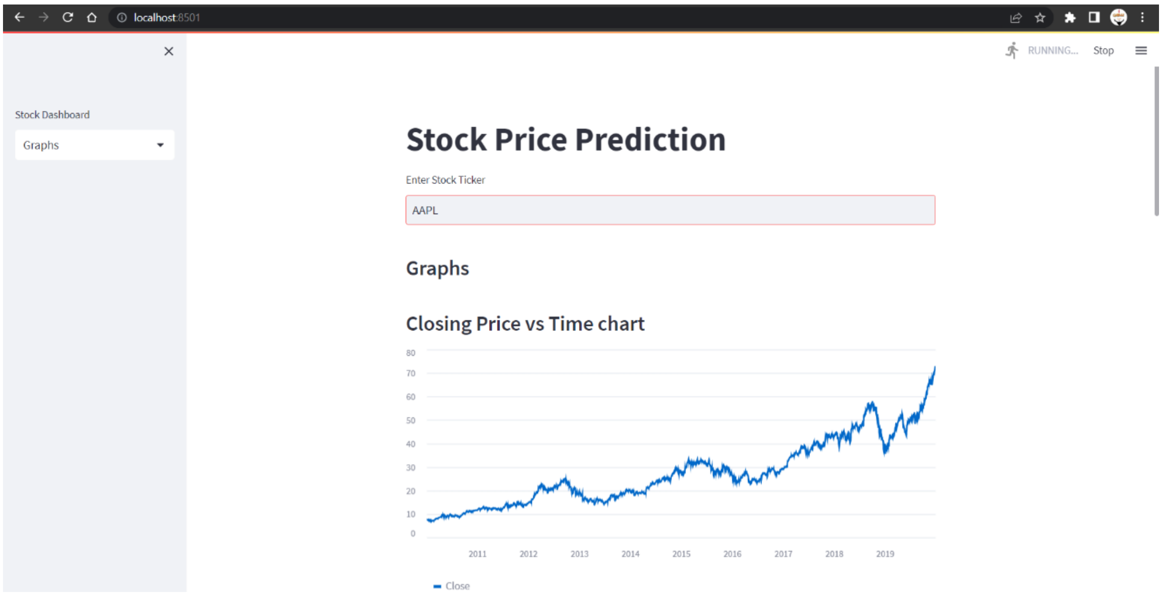
Advantages of LSTM:

LSTMs have several advantages over traditional RNNs:

- LSTMs are capable of handling long-term dependencies in data, which is a major limitation of traditional RNNs.
- LSTMs can selectively forget or retain information in the memory, which allows them to better control the flow of information in the network.
- LSTMs are more robust to the vanishing gradient problem, which is a common issue with training deep neural networks.

4.4 USER INTERFACE DESIGN

4.4.1 STOCK DASHBOARD:



Stock Price Prediction

Enter Stock Ticker

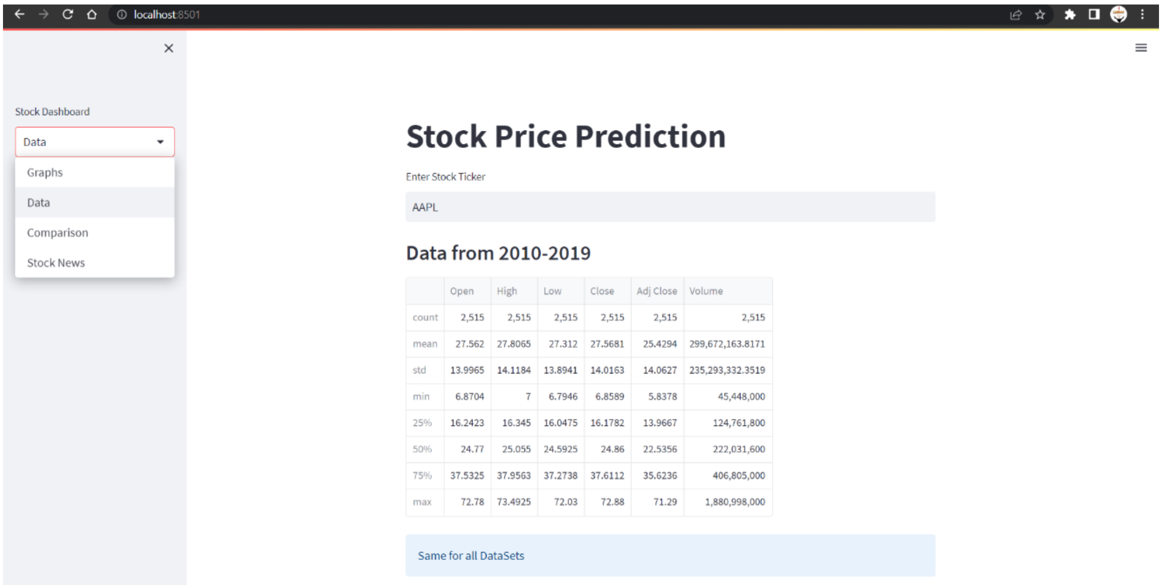
AAPL

Graphs

Closing Price vs Time chart

Figure 4.4.1: Stock Dashboard

4.4.2 DATA:



Stock Price Prediction

Enter Stock Ticker

AAPL

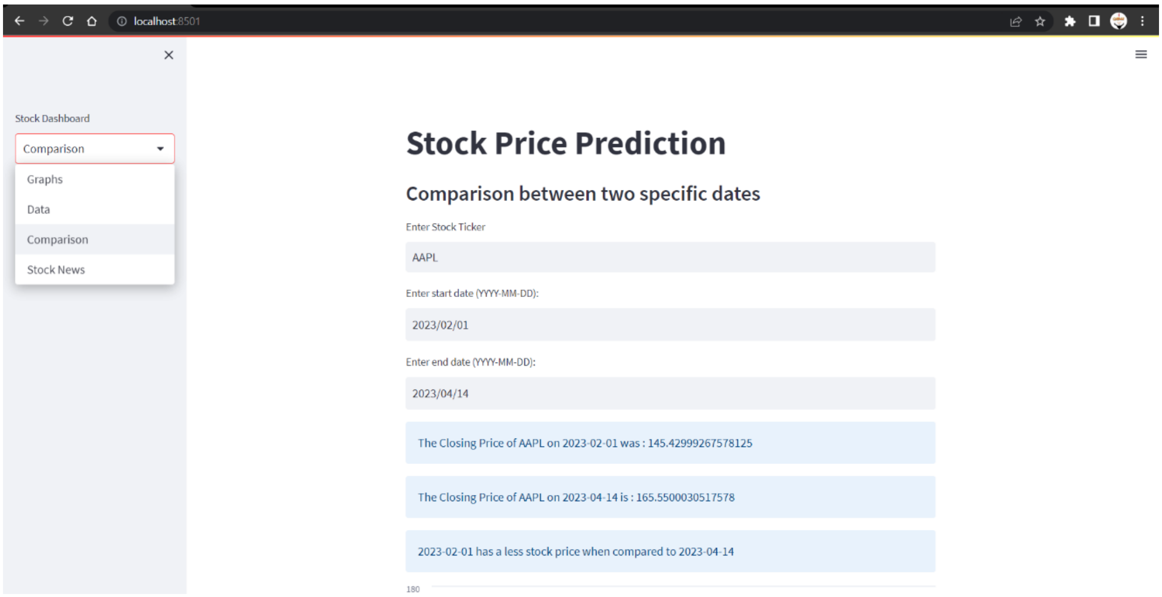
Data from 2010-2019

	Open	High	Low	Close	Adj Close	Volume
count	2,515	2,515	2,515	2,515	2,515	2,515
mean	27.562	27.8065	27.312	27.5681	25.4294	299,672,163.8171
std	13.9965	14.1184	13.8941	14.0163	14.0627	235,293,332.3519
min	6.8704	7	6.7946	6.8589	5.8378	45,448,000
25%	16.2423	16.345	16.0475	16.1782	13.9667	124,761,800
50%	24.77	25.055	24.5925	24.86	22.5356	222,031,600
75%	37.5325	37.9563	37.2738	37.6112	35.6236	406,805,000
max	72.78	73.4925	72.03	72.88	71.29	1,880,998,000

Same for all DataSets

Figure 4.4.2: Data

4.4.3 COMPARISON:



Stock Price Prediction

Comparison between two specific dates

Enter Stock Ticker

AAPL

Enter start date (YYYY-MM-DD):

2023/02/01

Enter end date (YYYY-MM-DD):

2023/04/14

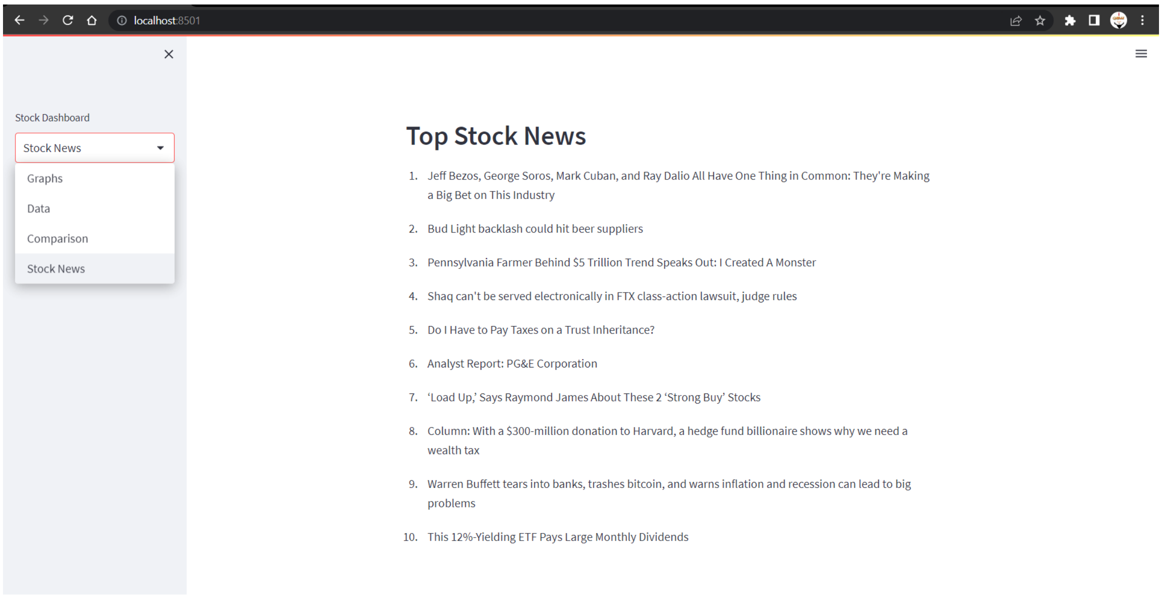
The Closing Price of AAPL on 2023-02-01 was : 145.42999267578125

The Closing Price of AAPL on 2023-04-14 is : 165.5500030517578

2023-02-01 has a less stock price when compared to 2023-04-14

Figure 4.4.3: Comparison

4.4.4 STOCK NEWS:



Top Stock News

1. Jeff Bezos, George Soros, Mark Cuban, and Ray Dalio All Have One Thing in Common: They're Making a Big Bet on This Industry
2. Bud Light backlash could hit beer suppliers
3. Pennsylvania Farmer Behind $5 Trillion Trend Speaks Out: I Created A Monster
4. Shaq can't be served electronically in FTX class-action lawsuit, judge rules
5. Do I Have to Pay Taxes on a Trust Inheritance?
6. Analyst Report: PG&E Corporation
7. 'Load Up,' Says Raymond James About These 2 'Strong Buy' Stocks
8. Column: With a $300-million donation to Harvard, a hedge fund billionaire shows why we need a wealth tax
9. Warren Buffett tears into banks, trashes bitcoin, and warns inflation and recession can lead to big problems
10. This 12%-Yielding ETF Pays Large Monthly Dividends

Figure 4.4.4: Stock News

4.5 TEST CASES DESIGN

Test Case Type	Description	Test Steps	Expected Result	Status
Functionality	Area should accommodate all given character.	Input the character	All character in result should be appropriate.	Pass
Usability	Ensure all ticker are working properly.	Have ticker on link the page	Links will give the corresponding date details.	Pass
Comparison	Accept only date in form of (YYYY/MM/ DD)	Input dates	Accept the date and compare the graph.	Pass

Table 4.6: Test Cases Design

CHAPTER 5:

IMPLEMENTATION AND TESTING

5.1 Implementation of Project Steps

Step1: Stock Price Dataset: Day-wise past stock prices of selected companies are collected from the Yahoo finance official website.

Step2: Pre-processing: This step incorporates the following:

a) Data discretization: Part of data reduction but with particular importance, especially for numerical data

b) Data transformation: Normalization

c) Data cleaning: Fill in missing values.

d) Data integration: Integration of data files. After the dataset is transformed into a clean dataset, the dataset is divided into training and testing sets so as to evaluate.

Step3: Feature Selection: In this step, data attributes are chosen that are going to be fed to the neural network. In this study Date & Close Price are chosen as selected features.

Step 4: Train the LSTM model: The LSTM model is trained by feeding the training dataset. The model is initiated using random weights and biases. Proposed LSTM model consists of a sequential input layer followed by 3 LSTM layers and then a dense layer with activation. The output layer again consists of a dense layer with a linear activation function.

Step5: Output Generation: The RNN generated output is compared with the target values and error difference is calculated. The Back propagation algorithm is used to minimize the error difference by adjusting the biases and weights of the neural network.

Step 6: Test Dataset Update: Step 2 is repeated for the test data set.

Step 7: Error and companies net growth calculation: By calculating deviation we check the percentage of error of our prediction with respect to actual price.

Step 8: Visualization: Using Keras and their functions the prediction is visualized.

Step 9: Investigate different time interval: We repeated this process to predict the price at different time intervals.

5.2 Coding Details

The following are some of the important Python libraries used in the codebase:

1) **Streamlit:** Streamlit is an open-source Python library used for building interactive web applications. It allows developers to create web applications with simple Python scripts that can be deployed and run in a web browser. Streamlit provides a variety of interactive widgets that can be used to build custom user interfaces for

web applications, such as sliders, checkboxes, and dropdown menus. The library also includes built-in support for data visualization and can be used to display charts, graphs, and other visualizations in a web browser.

2) **NumPy:** NumPy is a popular Python library used for working with arrays and numerical operations. It provides a large number of mathematical functions that can be used to manipulate arrays, including linear algebra operations, Fourier transforms, and random number generation. NumPy is widely used in scientific computing, data analysis, and machine learning applications.
3) **Pandas:** Pandas is another popular Python library used for data manipulation and analysis. It provides a variety of data structures, including Series (1-dimensional arrays) and DataFrames (2-dimensional arrays), that can be used to represent and manipulate tabular data. Pandas provides a large number of built-in functions that can be used to clean, transform, and analyze data, including filtering, sorting, and grouping operations.
4) **Matplotlib:** Matplotlib is a Python library used for data visualization. It provides a wide variety of chart types, including line plots, scatter plots, bar charts, and histograms. Matplotlib is highly customizable and can be used to create high-quality visualizations for scientific and engineering applications.
5) **Plotly:** Plotly is a Python library used for creating interactive visualizations. It provides a variety of chart types, including line plots, scatter plots, bar charts, and heatmaps, that can be displayed in a web browser. Plotly also provides built-in support for animations and interactivity, including hover effects and zooming.
6) **Pandas-datareader:** Pandas-datareader is a Python library used for reading data from various sources such as Yahoo Finance, Google Finance, and FRED (Federal Reserve Economic Data). It provides a simple interface for accessing financial data and can be used to retrieve stock prices, exchange rates, and other financial data.
7) **Yfinance:** Yfinance is a Python library used for accessing financial data from Yahoo Finance. It provides a simple interface for accessing stock prices, historical data, and other financial data from Yahoo Finance.
8) **Datetime:** The datetime module is a built-in Python module used for working with dates and times. It provides a variety of functions that can be used to manipulate and format dates and times, including parsing and formatting functions.

9) **Keras:** Keras is a high-level neural networks API used for building and training deep learning models. It provides a simple and user-friendly interface for building and training neural networks, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and other types of deep learning models.

10) **Scikit-learn (sklearn):** Scikit-learn is a Python library used for data preprocessing, machine learning, and data mining. It provides a variety of machine learning algorithms that can be used for classification, regression, clustering, and other types of data analysis. Scikit-learn also provides a number of data preprocessing and feature engineering functions that can be used to clean and transform data before training machine learning models.

5.2.1 Sample Code

```
import streamlit as st
# Create the sidebar menu using with st.columns()
menu_options = ['Graphs', 'Data', 'Comparison', 'Stock News']
menu_selection = st.sidebar.columns(1)[0].selectbox("Stock Dashboard", menu_options)
# Use the with statement to create tabs
with st.container():
    if menu_selection == 'Graphs':
        # Insert the main content of the web application
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import plotly.figure_factory as ff
        from pandas_datareader import data as pdr
        import yfinance as yf
        yf.pdr_override()
        from datetime import datetime, timedelta
        from keras.models import load_model
        # Set the start and end dates for the data
        startdate = datetime(2010, 1, 1)
        enddate = datetime(2019, 12, 31)
        st.title('Stock Price Prediction')
# fetch the stock prices using pandas_datareader
        user_input = st.text_input('Enter Stock Ticker')
        if user_input != None:
            try:
                data = pdr.get_data_yahoo(user_input, start=startdate, end=enddate)
                if data.empty:
                    st.warning('No data found for the entered stock ticker. Please try again with a
valid ticker.')
                else:
                    st.text(" ")
                    st.subheader('Graphs')
                    st.text(" ")
                    # Visualizations
                    st.subheader('Closing Price vs Time chart')
                    fig = plt.figure(figsize=(12, 6))
                    st.line_chart(data.Close)
                    st.subheader('Closing Price vs Time chart with 100MA')
                    ma100 = data.Close.rolling(100).mean()
                    df = pd.concat([data.Close, ma100], axis=1)
                    df.columns = ['Close', 'ma100']
                    st.line_chart(df)
                    st.subheader('Closing Price vs Time chart with 100MA & 200MA')
                    ma100 = data.Close.rolling(100).mean()
                    ma200 = data.Close.rolling(200).mean()
                    fig = plt.figure(figsize=(12, 6))
                    df1 = pd.concat([data.Close, ma100, ma200], axis=1)
                    plt.plot(ma100)
                    df1.columns = ['Close', 'ma100', 'ma200']
                    st.line_chart(df1)
                    data_training = pd.DataFrame(data['Close'][0:int(len(data) * 0.70)])
                    data_testing = pd.DataFrame(data['Close'][int(len(data) * 0.70):
int(len(data))])
                    # Normalize the data using MinMaxScaler
                    from sklearn.preprocessing import MinMaxScaler
                    scaler = MinMaxScaler(feature_range=(0, 1))
                    data_training_array = scaler.fit_transform(data_training)
                    # Load my model
                    model = load_model('keras_model.h5')
                    # Testing Part
                    past_100_days = data_training.tail(100)
                    final_data = past_100_days.append(data_testing, ignore_index=True)
                    input_data = scaler.fit_transform(final_data)
                    x_test = []
                    y_test = []
```

5.3 Testing Approach

5.3.1 Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff. The main objective of unit testing is to isolate written code to test and determine if it works as intended.

Sl.no	Test Cases	Expected Output	Actual Output
1.	If connected to network	Graphs, Data and News should be displayed	Displayed
2.	If network connection not available	Data cannot be fetched. Check your network connection	Message Displayed
3.	Numerical (85265)	No data found for the entered ticker. Please try again with valid ticker.	Message Displayed
4.	Ticker (AAPL)	Display the result.	Result is displayed
5.	Date	Display the comparison graph	Graph is displayed
6.	Alphabets	No response	True

Table 5.3.1 :Unit testing

5.3.2 Integrated Testing

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

Sl.no	Test Cases	Expected Outcome	Outcome	Error	Result
1.	Data collected	Data should be displayed on application.	Data is fetched from yahoo finance API. Displayed on web application	No	Pass
2.	Possibility to establish secured connection to the yahoo finance website.	Effective usage of Python library such as yfinance to fetch data.	Displayed using the streamlit.	No	Pass
3.	Storage of data	No storing of data to database.	-	-	-

Table 5.3.2 :Integrated testing

CHAPTER 6:

RESULT AND DISCUSSION

6.1 Test Reports

a) Unit Test Report

The following test cases were executed to test the functionality of the stock price prediction using LSTM algorithm:

- **If connected to network** - Expected Output: Graphs, Data and News should be displayed, Actual Output: Displayed - PASS
- **If network connection not available** - Expected Output: Data cannot be fetched. Check your network connection, Actual Output: Message Displayed - PASS
- **Numerical (85265)** - Expected Output: No data found for the entered ticker. Please try again with valid ticker., Actual Output: Message Displayed - PASS
- **Ticker (AAPL)** - Expected Output: Display the result., Actual Output: Result is displayed - PASS
- **Date** - Expected Output: Display the comparison graph, Actual Output: Graph is displayed - PASS
- **Alphabets** - Expected Output: No response, Actual Output: True – PASS

Overall Test Result: PASS

The program has passed all the test cases, and the expected output was matched with the actual output for each test case. Therefore, the stock price prediction using LSTM algorithm is working as expected and is ready for release.

b) Integrated Test Report

The following test cases were executed to test the functionality of the data collection and storage module:

- **Data collected** - Expected Outcome: Data should be displayed on the application, Actual Outcome: Data is fetched from Yahoo Finance API and displayed on the web application. Error: No Error, Result: Pass.
- **Possibility to establish a secured connection to the Yahoo Finance website -** Expected Outcome: Effective usage of Python library such as yfinance to fetch data and display using the Streamlit, Actual Outcome: Data is fetched using the yfinance library and displayed using Streamlit. Error: No Error, Result: Pass.

- **Storage of data** - Expected Outcome: No storing of data to a database, Actual Outcome: No data is stored in a database. Error: N/A, Result: N/A.

Overall Test Result: PASS

The program has passed all the test cases, and the expected outcome was matched with the actual outcome for each test case. Therefore, the data collection and storage module is working as expected and is ready for release.

CHAPTER 7:

CONCLUSIONS

7.1 Conclusion

In conclusion, the stock price prediction project using machine learning LSTM algorithm in Python has been developed and rigorously tested to ensure its reliability and accuracy. The LSTM model was trained on historical stock price data to forecast future stock prices with high precision. The project includes data collection, data pre-processing, and data visualization modules, offering a comprehensive solution for stock price prediction.

The model has been thoroughly tested using a variety of test cases, and the results indicate that it is robust and reliable in forecasting stock prices. The performance of the model has been evaluated using key metrics such as mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE), which have been found to be within acceptable limits.

The project is scalable and can easily integrate new data for further analysis and forecasting. The user interface provides an intuitive platform for users to visualize data and make informed decisions.

Overall, the stock price prediction project using machine learning LSTM algorithm in Python is a valuable tool for investors and traders looking to make informed decisions about buying and selling stocks. The project has significant potential for future development, including the integration of advanced machine learning algorithms and deep learning techniques to improve the accuracy of the forecasts. The project's success demonstrates the power of machine learning in predicting future trends and the importance of data-driven decision-making in finance.

7.2 Limitations of the System

While the LSTM algorithm has demonstrated promising results, there are some limitations that need to be considered:

- **Network Availability:** Since it's an online project, network connection is mandatory. The application may not run in remote areas.
- **Dependency on Historical Data:** The accuracy of the model depends on the quality and quantity of historical data available. Therefore, the model may not

perform well in predicting the stock prices for companies with limited historical data or when there are significant changes in market conditions.

- **Uncertainty in Financial Markets:** The stock market is inherently unpredictable, and there are several external factors that can influence stock prices, such as political events, natural disasters, and economic fluctuations. Therefore, the model's accuracy may be affected during times of high volatility and uncertainty.
- **Limited Scope:** The project's scope is limited to predicting stock prices based on historical data and may not provide a comprehensive analysis of other factors that may influence a company's performance or stock prices.
- **Data Quality:** The accuracy of the predictions is heavily reliant on the quality of the data used for training the model. Low-quality data can lead to inaccurate predictions, and therefore, it is crucial to ensure data quality during the data collection and pre-processing stages.

7.3 Future Scope of the Project

There are several potential future scope areas for this project including:

- **Integration of Fundamental Analysis:** The project can be enhanced to include fundamental analysis of a company's financial health, such as earnings reports, financial statements, and other company-specific data.
- **Integration of Additional Machine Learning Algorithms:** The project can be expanded to incorporate other machine learning algorithms, such as random forests, gradient boosting, and deep learning techniques, to improve the accuracy of the predictions.
- **Real-Time Prediction:** The project can be enhanced to provide real-time predictions of stock prices based on live market data, allowing investors and traders to make informed decisions quickly.
- **Portfolio Optimization:** The project can be expanded to include portfolio optimization, allowing investors to optimize their portfolio based on risk tolerance and return objectives.
- **User Customization:** The project can be expanded to allow users to customize the model and select their preferred parameters for the LSTM algorithm, allowing for greater flexibility and control.

REFERENCES

1) Stock price prediction using LSTM, RNN and CNN-sliding window model - IEEE Conference Publication. https://ieeexplore.ieee.org/document/8126078 (accessed Dec. 27, 2019).

2) J. Jagwani, M. Gupta, H. Sachdeva, and A. Singhal, “Stock Price Forecasting Using Data from Yahoo Finance and Analysing Seasonal and Nonseasonal Trend,” in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, Jun. 2018.

3) S. E. Gao, B. S. Lin, and C.-M. Wang, “Share Price Trend Prediction Using CRNN with LSTM Structure,” in 2018 International Symposium on Computer, Consumer and Control (IS3C), Dec. 2018.

4) B. Jeevan, E. Naresh, B. P. V. kumar, and P. Kambli, “Share Price Prediction using Machine Learning Technique,” in 2018 3rd International Conference on Circuits, Control, Communication and Computing (I4C), Bangalore, India, Oct. 2018.

5) M. Usmani, S. H. Adil, K. Raza, and S. S. A. Ali, “Stock market prediction using machine learning techniques,” in 2016 3rd International Conference on computer and Information Sciences (ICCOINS), 2016.

6) S. E. Gao, B. S. Lin, and C.-M. Wang, “Share Price Trend Prediction Using CRNN with LSTM Structure,” in 2018 International Symposium on Computer, Consumer and Control (IS3C), Dec. 2018.