ANNADA COLLEGE,

VINOBA BHAVE UNIVERSITY,

HAZARIBAG (825301)

PROJECT TITLE:

STOCK MARKET PRICE PREDICTION WITH MACHINE LEARNING

SUBMITTED BY:

NAME: DEWESH CHOPRA

UNIV. ROLL NO.: 200219022377

UNIV. REGN. NO.: BCA19041984/2019

SESSION: 2019-2022

ANNADA COLLEGE, HAZARIBAG
VINOBA BHAVE UNIVERSITY, HAZARIBAG (825301)

STOCK MARKET PRICE PREDICTION WITH MACHINE LEARNING

A PROJECT SUBMITTED
IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF THE DEGREE
OF

BACHELOR OF COMPUTER APPLICATION

By

DEWESH CHOPRA

(UNIV. ROLL NO. - 200219022377)

Under the guidance of

MR. KANCHAN RAJU

DEPARTMENT OF BCA

ANNADA COLLEGE, VINOBA BHAVE UNIVERSITY, HAZARIBAG (825301)

(2019-2022)

BONAFIDE CERTIFICATE

This is to certify that, the project report entitled "**Stock Market Price Prediction with Machine Learning**", submitted by **DEWESH CHOPRA (Univ. roll no. - 200219022377)**, in partial fulfillment of the requirement, for the award of the degree of **Bachelor of Computer Application**, in the **Department of BCA, Annada College, Vinoba Bhave University, Hazaribag,** is a bonafide record of the work, carried out under my guidance and supervision.

SIGNATURE

MR. KANCHAN RAJU

Senior Trainer,

Sai Coding Solutions Pvt. Ltd.,

Circular Road,

Lalpur, Ranchi

Certificate of the Department of Bachelor of Computer Application (BCA), Annada College, Hazaribag

This is to certify that, the project report entitled "**Stock Market Price Prediction with Machine Learning**", submitted by **DEWESH CHOPRA (Univ. roll no. - 200219022377)**, in partial fulfillment of the requirement, for the award of the degree of **Bachelor of Computer Application**, in the **Department of BCA, Annada College, Vinoba Bhave University, Hazaribag,** is a bonafide record of the work, carried out under my guidance and supervision.

DEWESH CHOPRA has worked under my supervision and guidance, and no part of this report has been submitted for the award of any other degree, diploma, fellowship or other similar titles, or prizes, and that the work has not been published in any journal or magazine.

(Project in-charge) **(Course coordinator)**

CERTIFICATE FOR PROJECT

This is to certify that, this is a bona fide record, of the project work entitled **"Stock Market Price Prediction with Machine Learning",** done satisfactorily at **Annada College, Vinoba Bhave University, Hazaribag,** by **DEWESH CHOPRA,** in partial fulfillment of the requirement, for the award of the degree of **Bachelor of Computer Application.**

This report or similar report on the topic has not been submitted, for any other examination and doesn't form part of any other course undergone by the candidate.

(INTERNAL GUIDE) **(EXTERNAL GUIDE)**

(UNDER THE GUIDANCE OF)

EXAMINER'S CERTIFICATION

This is to certify that, this project report entitled "**Stock Market Price Prediction with Machine Learning**", submitted by **DEWESH CHOPRA (Univ. roll no. - 200219022377)**, in partial fulfillment of the requirement, for the award of the degree of **Bachelor of Computer Application**, in the **Department of BCA, Annada College, Vinoba Bhave University, Hazaribag,** is approved, and is acceptable in quality and form.

SIGNATURE **SIGNATURE**

NAME: **NAME:**

(Internal Examiner) **(External Examiner)**

MY MAJOR PROJECT CERTIFICATE

Table of Contents

ACKNOWLEDGEMENT

This project work has been an intellectually invigorating experience for me. I am sure that the knowledge and experience gathered during the course of this work will make me stand in good stead in feature.

With immense pleasure and due respect, I express my sincere gratitude to the Project in-charge, Annada College, Vinoba Bhave University, Hazaribag, for all his support and co-operation in successfully completing this thesis work by providing excellent facilities.

I am also highly grateful to the Department coordinator, **Mr. Sanjeev Kr. Baxi**, for his ever-helping attitude and encouragement to excel in studies, besides he has been a source of inspiration during my entire period of BCA.

I would like to take this opportunity to extend my sincere gratitude and thanks to my pioneer, **Mr. Kanchan Raju**, firstly for coming up with such an innovative thesis idea. He has not only made us to work but guided us to orient toward research. It has been real pleasure working under his guidance and it is chiefly his encouragement and motivation that has made this thesis a reality.

Last, but not the least, I am heartly thankful to my parents for showering their blessings forever during my entire life and also to my family members and friends for providing me great support and feedback.

DECLARATION

I, **Dewesh Chopra (200219022377)**, hereby declare that the work, which is being presented in the dissertation, entitled "**Stock Market Price Prediction with Machine Learning**", in partial fulfillment, of the requirement, for the award of the degree of "**Bachelor of Computer Application**", submitted in Annada College, Vinoba Bhave University, Hazaribag, is an authentic record of our work carried out under the guidance of **Mr. Kanchan Raju**.

I have not submitted the matter embodied in this dissertation for the award of any other degree.

Chapter 1: Introduction

Purpose

Stock Market Price is known for being volatile, dynamic and nonlinear. Accurate stock price prediction is extremely challenging because of multiple (macro and micro) factors, such as politics, global economic conditions, unexpected events, a company's financial performance, and so on.

But, all of this also means that there's a lot of data to find patterns in. So, financial analysts, researchers, and data scientists keep exploring analytics techniques to detect stock market trends. This gave rise to the concept of algorithmic trading, which uses automated, pre-programmed trading strategies to execute orders.

Scope

Despite the volatility, stock prices aren't just randomly generated numbers. So, they can be analysed as a sequence of discrete-time data; in other words, time-series observations taken at successive points in time (usually on a daily basis). Time series forecasting (predicting future values based on historical values) applies well to stock forecasting.

Because of the sequential nature of time-series data, we need a way to aggregate this sequence of information. From all the potential techniques, the most intuitive one is MA with the ability to smooth out short-term fluctuations.

Stock analysis: fundamental analysis vs. technical analysis

When it comes to stocks, fundamental and technical analyses are at opposite ends of the market analysis spectrum.

- Fundamental analysis:

    - Evaluates a company's stock by examining its intrinsic value, including but not limited to tangible assets, financial statements, management effectiveness, strategic initiatives, and consumer behaviors; essentially all the basics of a company.
    - Being a relevant indicator for long-term investment, the fundamental analysis relies on both historical and present data to measure revenues, assets, costs, liabilities, and so on.
    - Generally speaking, the results from fundamental analysis don't change with short-term news.
- Technical analysis:
    - Analyzes measurable data from stock market activities, such as stock prices, historical returns, and volume of historical trades; i.e. quantitative information that could identify trading signals and capture the movement patterns of the stock market.
    - Technical analysis focuses on historical data and current data just like fundamental analysis, but it's mainly used for short-term trading purposes.
    - Due to its short-term nature, technical analysis results are easily influenced by news.
    - Popular technical analysis methodologies include moving average (MA), support and resistance levels, as well as trend lines and channels.

For our exercise, we'll be looking at technical analysis solely and focusing on the Simple MA and Exponential MA techniques to predict stock prices. Additionally, we'll utilize LSTM (Long Short-Term Memory), a deep learning framework for time-series, to build a predictive model and compare its performance against our technical analysis.

As stated in the disclaimer, stock trading strategy is not in the scope of this article. I'll be using trading/investment terms only to help you better understand the analysis, but this is not financial advice. We'll be using terms like:

- Trend indicators: statistics that represent the trend of stock prices,

- Medium-term movements: the 50-day movement trend of stock prices.

Problem Statement

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

Chapter 2: Literature survey

2.1 INTRODUCTION TO LITERATURE SURVEY

In this section we will review Key Terms, Technical Indicators and probabilistic models in depth
Stock market prediction is an area of research which walks hand in hand with all the other businesses in the world. Hence, it is known as 'Mother of all Businesses'. Before trying to jump into analysis we need to understand the following terms.

2.1.1 Key Terms

2.1.1.1 Interval

Millions of trades take place every second. Hence, for analysis we need to classify these trades based on the interval at which they took place. We can divide these trades into intra-day and long-term intervals. Intra-day can be sub-classified into 1 minute, 2, 5, 10, 15, 30 and 60 minutes. Long term intervals can be classified into daily, weekly, monthly and so on.

2.1.1.2 Tick Prices

At every interval any stock will have 4 types of prices associated with it. High price, Low price, Open Price and Close price. High price is the highest value at which it was traded in that particular interval. Low price is the lowest price which it reached in that interval. Open price is the first trade which took place at that particular interval and Close price the price at which the stock was last traded at that interval

2.1.1.3 Trend

At any instance of time, any stock will have either higher demand than supply or lower demand than available supply. Hence, we can classify the Trend into two types namely Bearish and Bullish. A stock is said to be in Bullish Trend if it has higher demand than its supply at that instance of time. If the stock has higher available supply when compared with

demand, it is said it be in Bearish Trend.

2.1.2 Data Representation

Data need to be represented in some format before we can start analyzing it. In this section we will discuss how can we represent data for better interpretation.

2.1.2.1 Japanese Candle

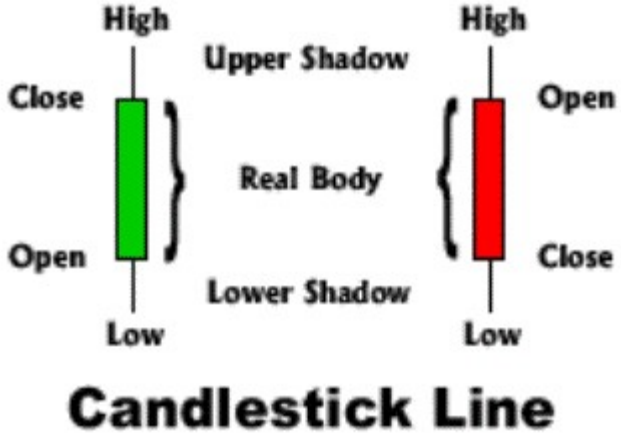


Fig. 1: Japanese Candle

Figure 1 represents two different candle sticks. The green candle represents gain, and red candle represents loss. This candle stick always contains all attributes associated with an interval (Example: tick prices). Since it represents an interval what we are looking at, we will refer to a candle stick as an interval in the rest of this report.

2.1.2.2 Line Graph

Line graph is a graphical way of representing data points in stock market based only on close price or current trading price of the stock.

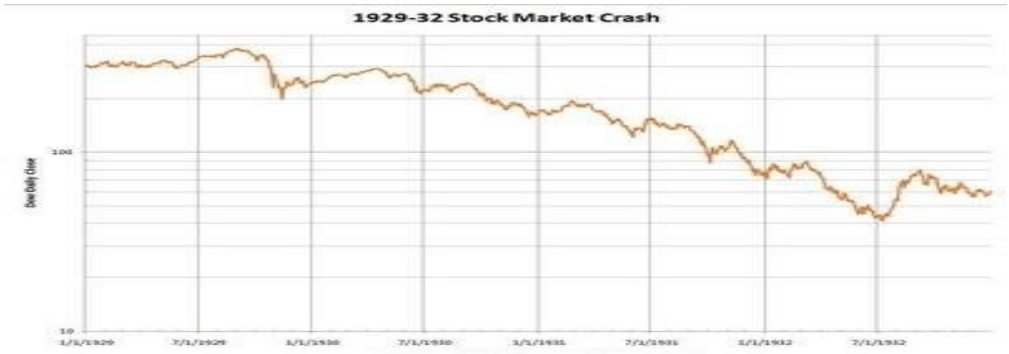


Fig. 2: Line Graph

2.1.3 Accuracy

Accuracy is defined as the result of number of right decisions made divided by total number of decisions.

2.1.3.1 Accuracy based on Close Price

When we calculated accuracy, if we relied on next interval's close price for decision verification, we have mentioned it as Accuracy based on Close price in the rest of the report. When we make a Buy call, to verify based on close price, we check if next interval's close price is above current interval's close price. If so, we consider the decision as a right decision. Also, if we make a Short sell, to verify the decision based on close price, we check if next interval's close price is lesser than that of current interval's close price. If so, we consider the decision as right decision.

2.1.3.2 Accuracy based on Low/High Price

When calculating accuracy, if relied on next interval's low/high price for decision verification, we have mentioned it as Accuracy based on high/low price in the rest of the report. When we make a Buy call, to verify the decision based on high price, we check if current interval's close price is lesser than next interval's high price. If so, we consider the decision as a right decision. Similarly, if we make a Short sell, to verify the decision based on low price, we check if next interval's low price is lesser than current interval's close price. If so, we consider this decision as right decision.

2.1.3.3 Short Sell

Short sell is a type of trade where we sell the stock first, before buying it

and buy it at a later point of time. Investors perform this trade if they think stock market is going to follow Bearish trend.

2.1.3.4 Buy Call

Buy sell is a type of trade where investors invest by buying the stock first and selling it at a later point of time. They perform this trade if they think, the stock in which they are investing will follow Bullish trend and can yield them profit.

2.1.4 Technical Indicators

Technical Indicators are the properties associated with any stock based on its tick prices. Calculation of each indicator is mentioned in the following subsections.
Interpretation of each of the indicator is explained in detail in section 3.0.

2.1.4.1 Simple Moving Average (SMA)

Simple Moving Average is calculated exclusively based on close price of the stock which we are trying to analyze. For instance, if we need to calculate SMA of 'x' intervals we need to get close prices of previous 20 intervals and divide it by 'x'. Hence the first available SMA value will correspond to x^{th} interval. To calculate we add all close prices starting from the current interval looking back for n number of intervals. In this case n stands for number of intervals for which we need to calculate SMA.

2.1.4.2 Exponential Moving Average (EMA)

Exponential Moving Average is also calculated based on close price of the stock which we are analyzing. If we need to calculate EMA of 'x' intervals, we first need to calculate SMA, till x^{th} interval and for every subsequent interval we calculate EMA based on the following formula.

$$EMA = \frac{((Current\ Close - SMA) * 2 * (interval + 1))}{(interval\ + 1)}$$

Hence, the first available EMA value will be corresponding to x^{th} interval. EMA moves hand in hand with price of stock. Higher the number of intervals we choose to calculate EMA, higher the stability of EMA. Hence, we chose 2, 5, 10, 20, 50 and 100 intervals for calculating EMA.

2.1.4.3 Relative Strength Index (RSI)

Relative Strength Index is calculated based on SMA and close price of the stock for the given interval. We must get familiar with the following terms to better understand the calculation of RSI: Gain, Loss. If the close price of a stock at a given interval is greater than its open price, the stock resulted in Gain, vice versa it resulted in Loss.
Here are the formulae to calculate RS and RSI.

$$\text{RS} = \text{Average Gain} / \text{Average Loss}$$

$$RSI = 100 - \frac{100}{1 + RS}$$

RSI indicates the strength of the current trend. Higher the value of interval we choose, we get stable RSI values. We need to find out a threshold value. If RSI falls below its threshold, it is an indication of sellers taking over buyers. If RSI value rises over its threshold, it indicates that buyers are taking over sellers and stock prices will go high.

2.1.4.4 Bollinger Band (BB)

Bollinger Band is calculated based on Standard Deviation and close price of the stock at a given interval. Bollinger Bands are calculated based on the following formulae:

$$Standard\ Deviation = \sqrt{\frac{Close - SMA(interval)^2}{Interval}}$$

$$UpperBand = SMA[20] + (2 * SD[20])$$

$$LowerBand = SMA[20] - (2 * SD[20])$$

Bollinger Band's values provide an insight on how much more the stock can rise if it is in Bullish trend. Or, how much can it fall, if the stock is in Bearish trend.

2.1.4.5 Fast Stochastic (FS)

Fast Stochastic is the technical indicator which involves two values, namely, %k and %d. For a given interval these values are calculated based on current close price of the stock, lowest low price in the look-back period and highest high price of the stock in the look-back period. Lowest low and highest high are the lowest price and highest price at which the stock was traded in the given look-back period with respect to given interval respectively.
Here are the formulae to calculate %k and %d:

$$\%k = \frac{CurrentClose - LowestLow}{HighestHigh - LowestLow}$$

$$\%d = SMA(interval) of \%k$$

At a given interval %k's value falls below %d's value, market takes Bearish trend.
Otherwise, if %k's value rises above %d's value it indicates Bullish trend. However, it is very difficult to identify how long will the trend predicted from FS is valid.

2.1.4.6 Moving Average Convergence Divergence (MACD)

This indicator is calculated based on exponential moving average. To calculate MACD we first need to compute EMA of 9 intervals and EMA of 26 intervals. Now starting from 27th interval till 35th interval subtract EMA (26) from EMA (9) and store the result for further processing. This value represents upper MACD indicator. With these values now calculate EMA

of previously stored results starting from 36th interval.
This represents lower MACD indicator. Minimum number of intervals needed for

MACD is 35 and is its only limitation.

Whenever lower MACD value falls below upper MACD value it shows a trend reversal from Bullish to Bearish. If lower MACD value rises above its upper MACD value it indicates a trend reversal from Bearish to Bullish.

2.1.4.7 Williams Average (WA)

Williams Average is calculated based on current close, highest high and lowest low (2.1.2.5). Here is the formula to calculate Williams Average: In order to calculate this indicator we first subtract current close from highest high and store the result and call it HC. Now we subtract lowest low from highest high, store it and call it HL. With these results we now calculate Williams Average by the following formula.

$$WA = \frac{HC}{HL} * -100$$

WA indicates oversold and overbought conditions. [4] Whenever it reaches close to 0 it shows oversold condition. If WA value is closer to 100 it shows overbought condition. We need to identify the optimal threshold for which this indicator gives accurate results.

2.1.4.8 Fibonacci Series

Fibonacci Series is calculated using the following formula: [4, 5, 16]

$$n^{th} Fibonacci\ number = \frac{\emptyset^n - (-\emptyset^n)}{\sqrt{5}}$$

Fibonacci series helps us to identify support level or resistance level of a given stock. Whenever we make a decision based on certain indicator we can use this indicator to identify we made the right decision before making a trade.

2.1.4.9 Rate Of Change (ROC)

To calculate ROC, we need to know the current trading price of the

current interval and close price of the interval which is look-ahead parameter intervals previous to current interval. Here is the formula to calculate ROC.

$$Rate\ Of\ Change = \frac{Current\ Price\ +\ \ Close(interval)}{Close\ (interval)} * 100$$

Rate of change is directly proportional to the trend of the stock market. If ROC is lesser than 20 it is an indication that the market is in bearish trend. If ROC is greater than 80 it is a strong signal to buy stocks.

2.1.4.10 Artificial Neural Networks (ANN)

This model is a multi-level perceptron model which feeds forward the result of every computation to the next level of computation. The author has modified the approach of feed-forward model by adding pseudo input variables into ANN. First version of ANN propagates back and forth to provide final results. Author in his approach has argued that, by modifying the model he could minimize the back and forth movement of the data to increase optimization of computer's computational power. ANN's are always very less susceptible to noise points in the data. If a model is trained based on noise points accuracy of the prediction would decrease drastically due to increase in false positives. Using this model author is making the decision to hold, sell or buy stocks at given time. He used a Genetic algorithm to select the data from the given data set. This selected data will be given as an input to ANN which would classify the labels based on the weight of computed results. In his results he proves that 2 layer ANN is more feasible than a single layer perceptron. The author has improved the model by embedding pattern matching with basic ANN. Pattern matching algorithms are always highly susceptible to noise. However, for given patterns the algorithm will detect the re-occurrence if any. Hence, by embedding this advantage with ANN author has opened the doors of fortunes. This improved model can now be used for prediction on data that contains extreme noise (stock market often is the case). He has also proved the same with his results in the publication. [22]

2.1.4.11 Hidden Markov Model (HMM)

Hidden Markov Models are statistical models that are used to determine the most likely sequence of occurrence given a sequence of data. This model consists of transition matrix, observation matrix and initial set of observation distribution, denoted as A, B and pi respectively. This model is used to uncover hidden information behind any set of sequences that has happened. In stock market ticks of respective intervals are taken into consideration for computing initial set of observations and observation distribution. Based on the computed results we then calculate the transition matrix. It is the transition matrix, which author relied on to calculate the likeliness of the trading sequence. When he trained this model with ticks whose results we already know since they are training records. He then calculated its likeliness. Now he calculates likeliness for a sequence whose results we don't know yet. It is based on this likeliness; we predict the future trend for that particular stock. The author has used tick prices of current interval as 4 of his observations. He also considered close price of the next interval, relative to the current interval, as fifth observation for training his model. Based on these data he computes A, B and pi as discussed previously. This model now can be used to predict next most likely observation.

Chapter 3: Methodology

3.1 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).

3.1.1 Time Series Analysis

Time series analysis involves the sequential plotting of a set of observations or data points at regular intervals of time. By studying previous outcomes and their progression over time, we can make future predictions with the help of these studied observations.

The approach to problems related to time series analysis is unique in its own way. Most machine learning problems utilize a dataset with certain outcomes to be predicted, such as class labels. This statement means that most machine learning tasks usually utilize an independent and dependent variable for the computation of a particular question.
This procedure involves machine learning algorithms analysing a dataset (say, XX) and using the predictions (YY) to form an assessment and approach to solving the problem statement. Most supervised machine learning algorithms perform in this manner. However, time series analysis is unique because it has only one variable: *time*. We will dive deeper into how to solve the stock market price prediction task with deep learning in the next part of this article. For now, our primary objective will be understanding the terms and important concepts required for approaching this task. Let us begin!

The ordered sequence of data points in regular intervals of time constitutes the primary elements of time series analysis. Time series analysis has a wide range of applications in the practical world. Hence, the study of time series is significant to understand, or at the very least, to gain basic knowledge about what you can expect in the near future. Apart from the stock market price prediction model that we will build in the next part of these articles, time series analysis has applications in economic forecasting, sales forecasting, budgetary analysis, process and quality control, detection of weather patterns, inventory and utility studies, and census analysis, among many others.

Understanding previous behavioural patterns of data elements is critical. Consider an example of business or economic forecasting. When you can extract useful information from the previous patterns, you can plan the future accordingly. More often than not, the predictions made with the help of time series analysis and forecasting yield good results. These will help users to plan and evaluate current accomplishments.

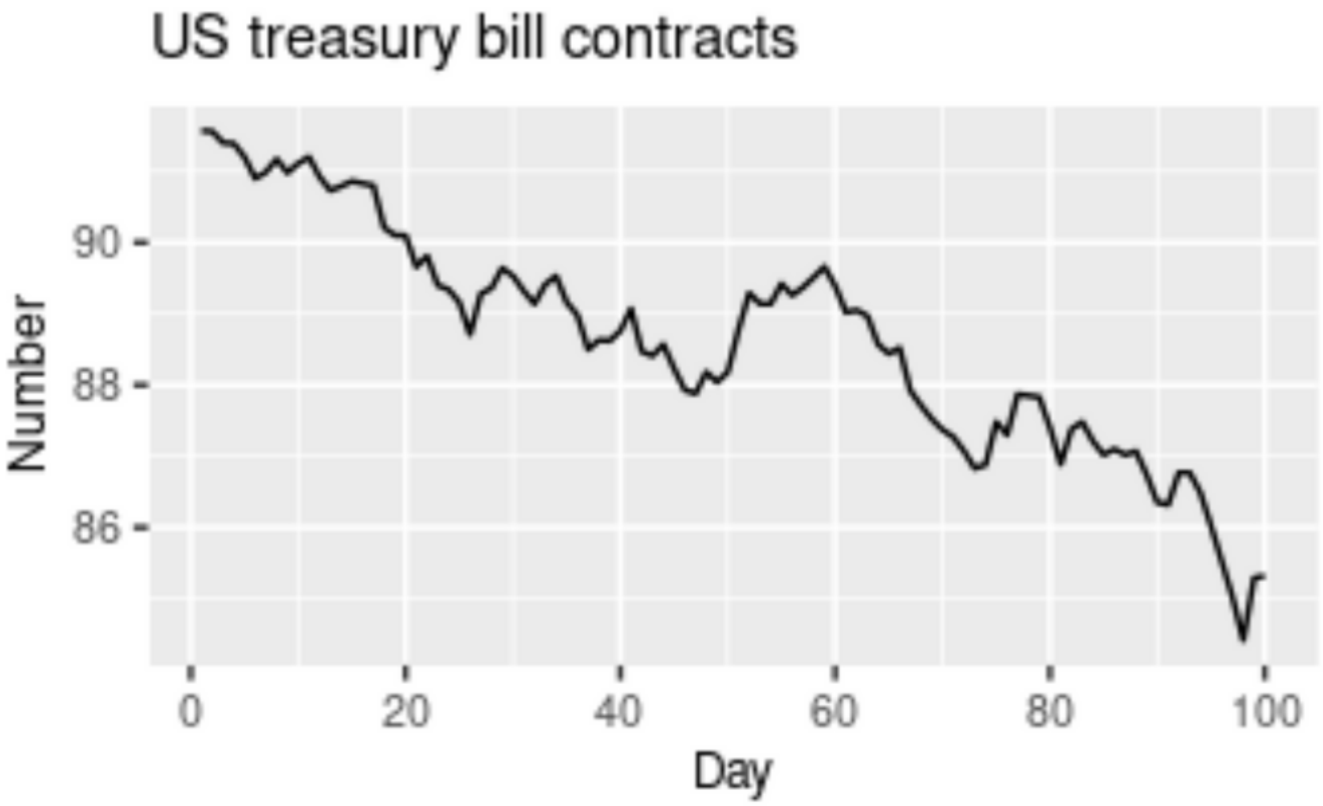
3.1.1.1 Essential components of time series analysis

Time series analysis is all about the collection of previous data to study the patterns of various trends. By conducting a detailed analysis of these time series forecasting patterns, we can determine the future outcomes with the help of our constructive deep learning models. While there are other methods to determine the realistic results of future trends, deep learning models are an outstanding approach to receive some of the best predictions for each concept.

Let's now focus on the components of time series forecasting, namely Trend, Seasonality, Cyclicity, and Irregularity. These four components of time series analysis refer to types of variations in their graphical representations. Let us look at each of these concepts individually and try to gain more intuition behind them with the help of some realistic examples.

Trend

A Trend in time series forecasting is defined as a long period of time with a consistent increase or decrease in the data. There may be slight fluctuations in the data points and elements at various instances of time, but the overall variation and direction of change remain constant for a longer duration of time. When a Trend goes from a long duration of constant increase to a long duration of constant decrease, this is often referred to as a "Changing Direction" trend.
There a few terminologies used to define the type of trend that we are dealing with in time series forecasting. A slightly or moderately increasing trend over a long duration of time can be referred to as an uptrend, while a slightly or moderately decreasing trend over a long duration of time is called a downtrend. If the trend is following a consistent pattern of gradually increasing or gradually decreasing, and there is not much effect in the overall pattern of the graph, then this trend can be referred to as a horizontal or stationary trend. The graphical representation shown in the above image has a downward trajectory over a long period of time. Hence, this image shows the representation of a downward trend, also called a downtrend.
Let us consider an example to understand the concept of trend better. Successful companies like Amazon, Apple, Microsoft, Tesla, and similar tech giants have a reasonably performing upward stock price curve. We have learned that we can determine this upward rise in stock prices as an uptrend. While successful companies have an uptrend, some companies that are not performing that well in the stock market have downward trajectories in stock prices, or a downtrend. Companies that are making a neutral profit rate with decent amounts of profits and losses at regular intervals are determined as a horizontal or stationary trend.

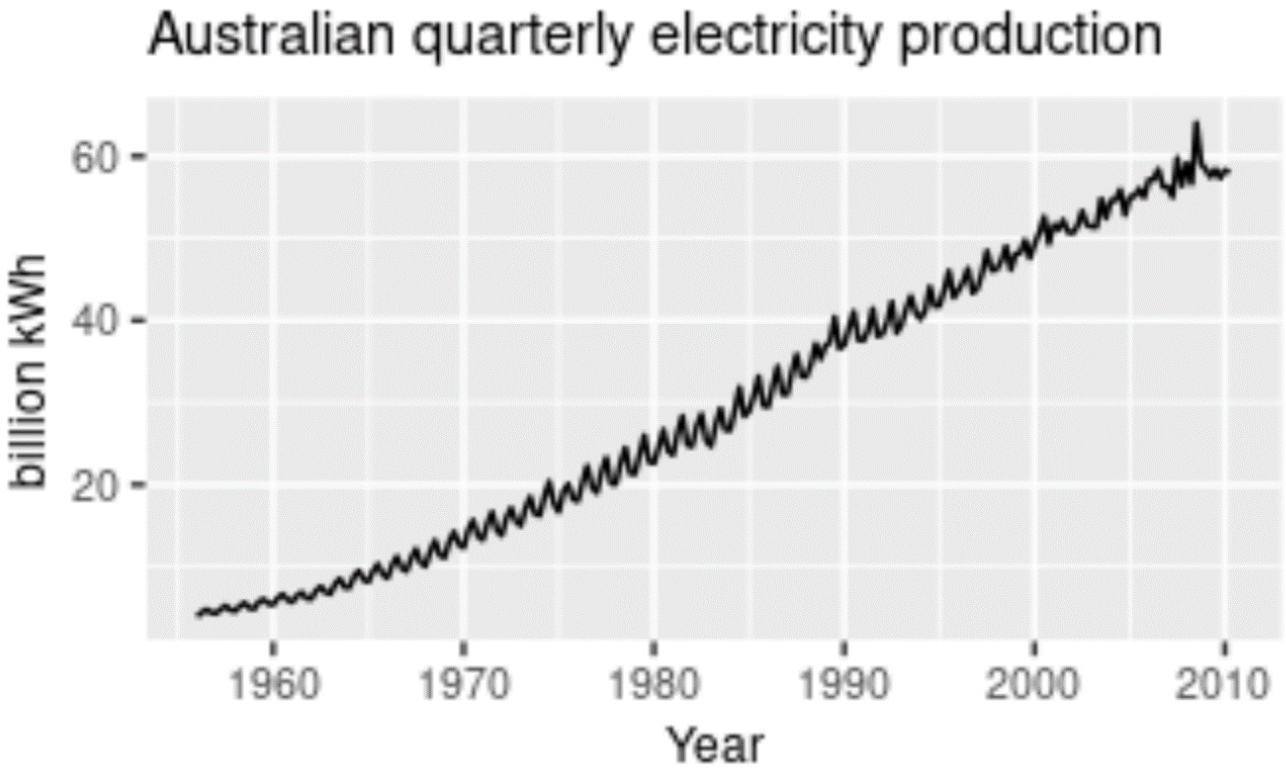


Seasonality

A pattern that is affected or influenced by seasonal frequency variations are termed to be seasonal. This component of time series analysis can vary from time stamps like quarterly, monthly, or half-yearly. However, an important point to note is that these fluctuations usually take place within the period of a year. Whenever the frequency is fixed and known and also occurs on a timely basis, usually within a year, this component of time series analysis is known as seasonality.

To consider a realistic example, think about the sale of certain seasonal fruits. Fruits like watermelon will have increased sales during the summer season, while during the winter seasons, the sales of watermelons will gradually reduce. Similarly, seasonal fruits like apple have higher sales during the winter seasons in comparison to the other seasons. Ice cream and tender coconuts are other examples of food products that have increased sales during the summer season while experiencing a dip in sales during other seasons. Apart from food products, a period or month like April might have higher shares of investment and experience a dip in

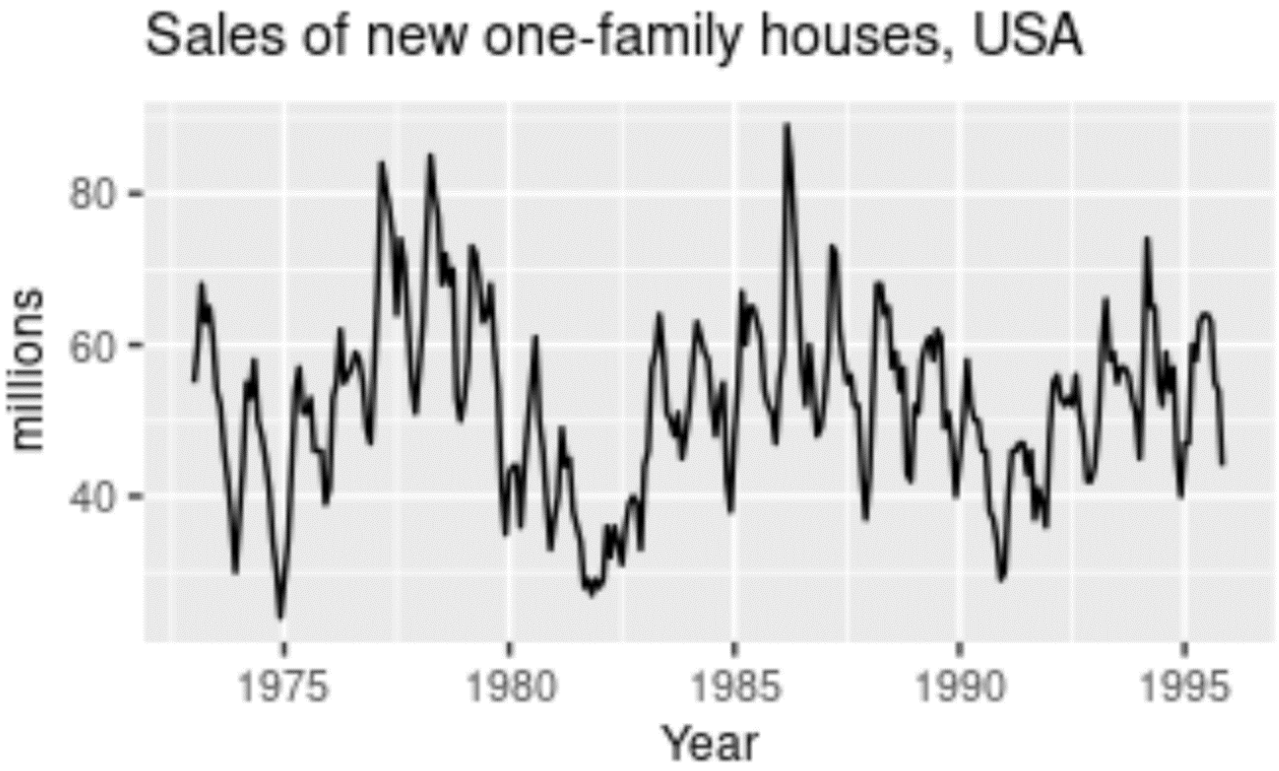
shares until six months later in October, where it has a peak rise. This pattern can also be regarded as a seasonality.



Cyclicity

When a pattern exhibits a rise and fall of mixed frequencies, and the graphical representation has peaks and troughs that occur randomly over a period of time, it is called a cyclic component. The duration of these occurrences usually ranges over the period of at least one year. Stock prices of certain companies that are hard to predict usually have cyclic patterns where they flourish during a certain period, while having lower profits at other times. Cyclic trends are some of the hardest for our deep learning models to predict. The graphical representation above shows the cyclic behaviour of house sales over the span of two decades.
A great realistic example of cyclic behavioural patterns is when a person decides to invest in their own start-up. During the set-up and progression of start-ups, every business experiences a cyclic phase. These cyclic fluctuations often occur in business cycles. Usually, the phases of a start-up would include the investment stage, which would have a slightly negative impact on our prices. The next phases could include your

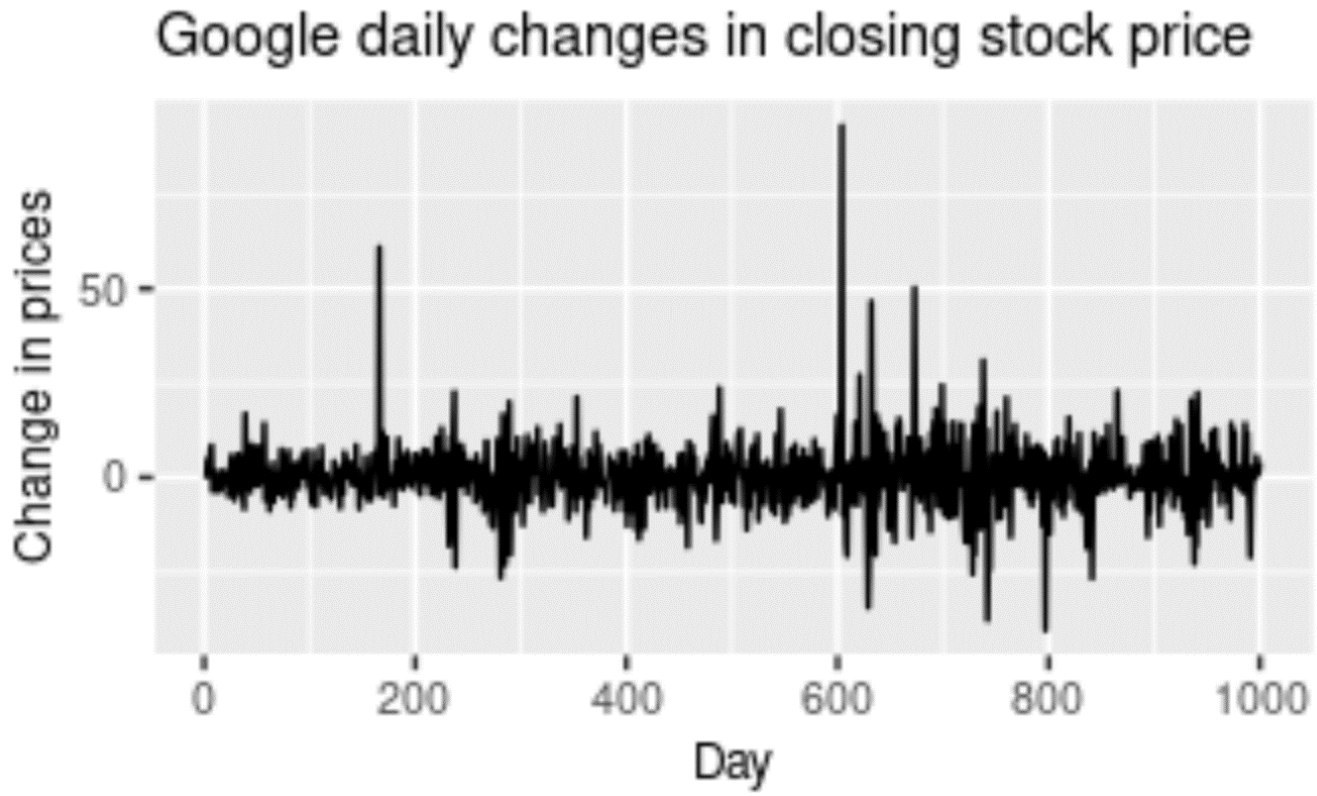
marketing and profitable stages, where you start to earn profits from your successful start-up. Here, you experience an increase in the graphical curve. However, you will eventually experience a depreciation phase as well. These will show lesser profits until you make constant improvements and investments. This procedure begins the cyclic stage again, lasting for periods of a few years. Rinse and repeat.



Irregularity

Irregularity is a component that is almost impossible to make accurate predictions for with a deep learning model. Irregularity, or random variations (as the name suggests), involves an abnormal or atypical pattern where it becomes hard to deduce the occurrences of the data elements with respect to time. Above is a graphical representation of the Google Stock Price changing rapidly with an irregular and random pattern; this is hard to read. Despite the information and data patterns present, the modeling procedure for such kinds of representations will be hard to crack. The primary objective of the model is to predict future possibilities based on previous outcomes. Hence, models for irregular patterns are slightly harder to construct.

To provide a realistic example for irregular patterns, let us analyze the current status of the world, where many businesses and other industries are affected on a large scale. The global pandemic is a great example of irregular activity that is impossible for anyone to predict. These disturbances that occur due to a natural calamity or phenomenon will affect the trading prices, stock price charts, companies, and businesses. It is not possible for the model you construct to detect the occurrences of these situational tragedies. Hence, irregular patterns are an interesting component of time series forecasting to analyze and study.



Understanding stationary and non-stationary series

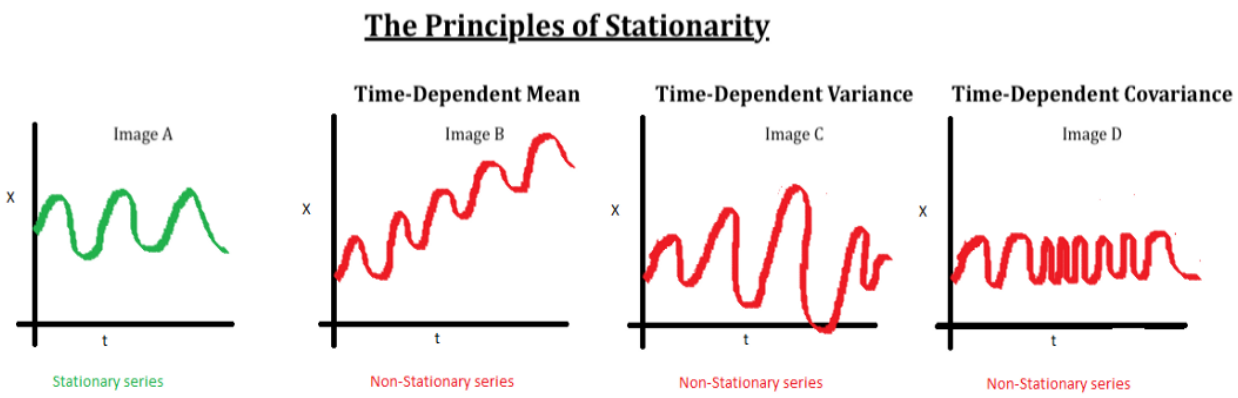


Fig. 3: Principles of Stationarity

Another significant aspect of time series analysis that we will encounter in this article is the concept of stationarity. When we have a certain number of data points or elements, and the mean and variance of all these data points remain constant with time, then we call it a stationary series. However, if these elements vary with time, we call it a non-stationary series. Most forecasting tasks, including stock market prices or cryptocurrency graphs, are non-stationary. Unfortunately, the results obtained by non-stationary patterns are not efficient. Hence, they need to be converted into a stationary pattern.

The entire data preparation process for stock market price prediction in the next article is explained with code snippets. Here, we will discuss a few other topics of importance. Several methods for converting non-stationary series into stationary series include *differencing* and *transformation*. Differencing involves subtracting two consecutive data points from the higher to lower order, while transformation involves transforming or diverging the series. Typically, a log transform is used for this process. For further information on this topic, refer to the link provided in the image source above.

To test for stationarity in Python, the two main methods that are utilized include *rolling statistics* and *the ADCF Test*. Rolling statistics are more of a visual technique that involves plotting the moving average or moving variance to check if it varies with time. *The Augmented Dickey-Fuller test*

(ADCF) is used to give us various values that can help in identifying stationarity. These test results comprise some statistics and critical values. They are used to verify stationarity.

3.1.2 Long short-term memory network

Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. RNNs have issues with the transfer of long-term data elements due to the possibility of exploding and vanishing gradients. The fix to these issues is offered by the Long short-term memory (LSTM) model, which is also a recurrent neural network (RNN) architecture that is used to solve many complex deep learning problems. LSTMs are especially useful for our task of Stock Price Prediction Using Deep Learning.

The LSTM architecture, with its effective mechanism of memory cells, is extremely useful for solving complex problems and making overall efficient predictions with higher accuracy. LSTMs learn when to forget and when to remember the information provided to them. The basic anatomy of the LSTM structure can be viewed in three main steps.

The first stage is the cell state that contains the basic and initial data to be recollected. The second stage is the hidden state that consists of mainly three gates: the forget gate, input gate, and output gate. We will discuss these gates shortly. Finally, we have a looping stage that reconnects the data elements for computation at the end of each time step.

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 neuralnetwork with automatic encoder.
- LSTM is used instead of RNN to avoid exploding and vanishing gradients.

- The historical stock data table contains the information of opening price, the highestprice, lowest price, closing price, transaction date, volume and so on.
- The accuracy of this LSTM model used in this project is 57%.

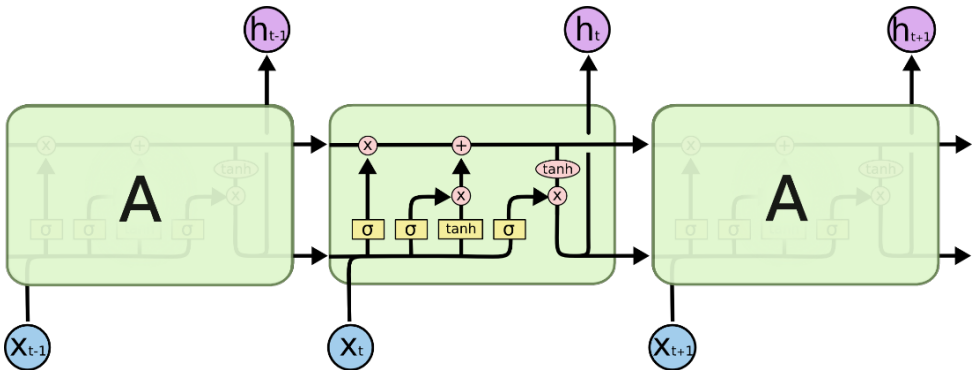


Fig. 4: LSTM Architecture

Forget Gate

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

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

Input Gate

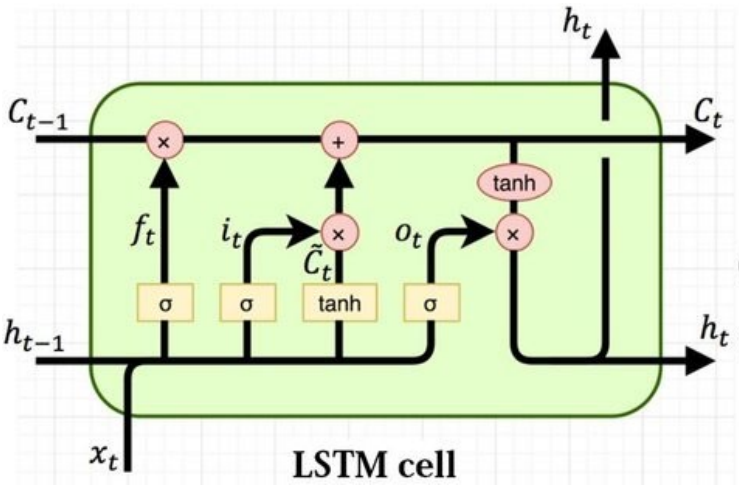
1. Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from hi-1 and x_t.
2. Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
3. Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation.
4.

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 valuesthat 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.

The mathematical computations for LSTMs can be interpreted in many ways. The images shown below are two of the methods in which the mathematical calculations of LSTMs are carried out. Sometimes the bias function is ignored, as shown in the second image. However, we will not cover the intricate details of the first image. You can check out more on that representation from the link provided in the image source. The mathematical equations in consideration of the first LSTM image are pretty accurate for computation purposes.



$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$
$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$
$$h_t = \tanh(C_t) * o_t$$

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \tanh_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

Where,

X(t)X(t): input vector to the LSTM unit

- F(t)F(t): forget gate's activation vector
- I(t)I(t): input/update gate's activation vector
- O(t)O(t): output gate's activation vector
- H(t)H(t): hidden state vector, also known as output vector of the LSTM unit
- c˜(t)c̃(t): cell input activation vector
- C(t)C(t): cell state vector
- WW, UU, and BB: weight matrices and bias vector parameters which need to be learned during training.

Algorithm

- # LSTM
- Inputs: dataset
- Outputs: RMSE of the forecasted data
-
- # Split dataset into 75% training and 25% testing data
- size = length(dataset) * 0.75
- train = dataset [0 to size]
- test = dataset [size to length(dataset)]
-
- # Procedure to fit the LSTM model
- # Procedure LSTMAlgorithm (train, test, train_size, epochs)
- X = train
- y = test
- model = Sequential ()

```
    model.add (LSTM (50), stateful=True)
    model. compile (optimizer='adam', loss='mse')
    model.fit (X, y, epochs=epochs, validation_split=0.2)
    return model

# Procedure to make predictions
# Procedure getPredictonsFromModel (model, X)
    predictions = model.predict(X)
    return predictions

epochs = 100
neurons = 50
predictions = empty
```

3.2 SYSTEM ARCHITECTURE

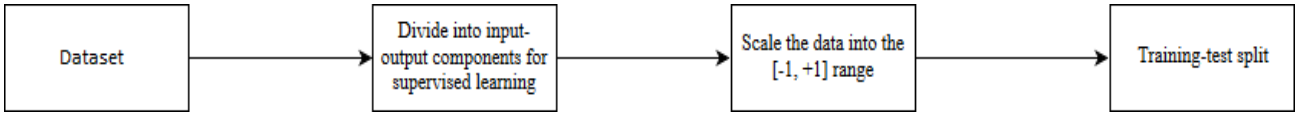


Fig. 5: Pre-processing of data

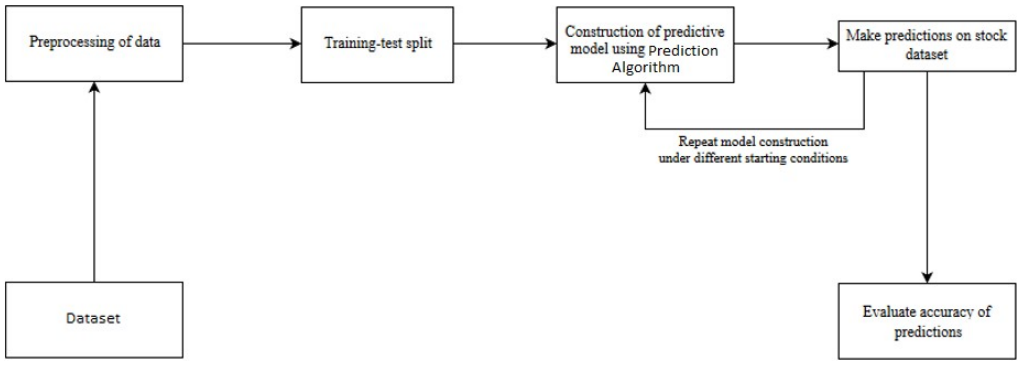


Fig. 6: Overall Architecture

Chapter 4: Design

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

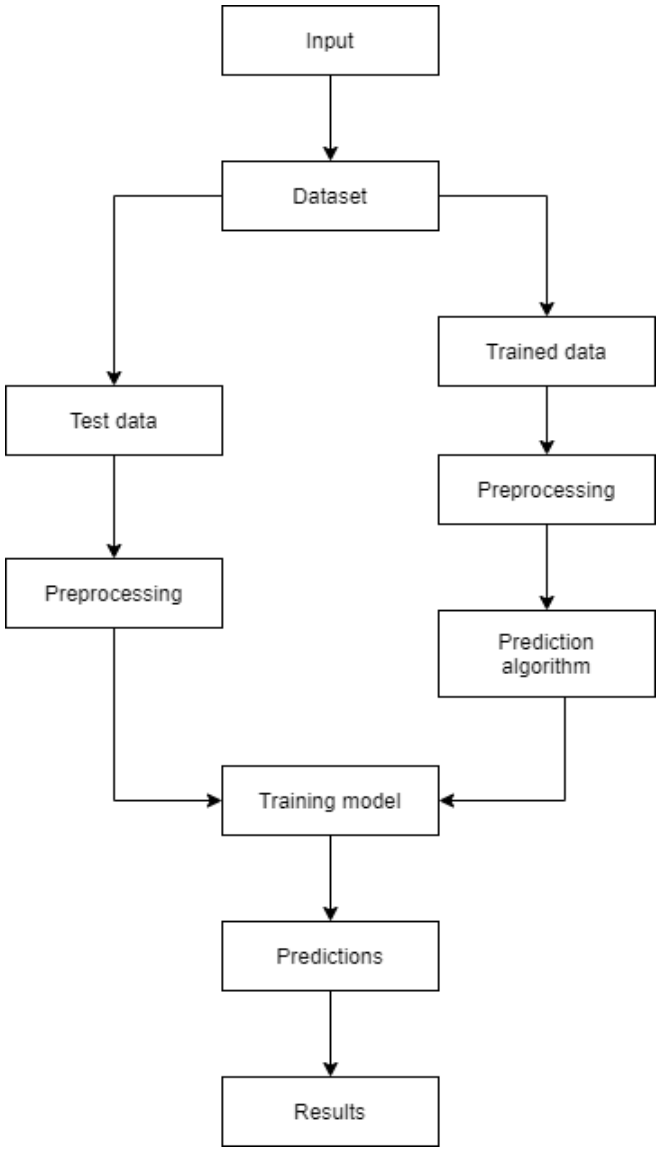


Fig. 7: Training and prediction

4.1 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 overtime.

4.1.1 Use Case Diagram

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

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

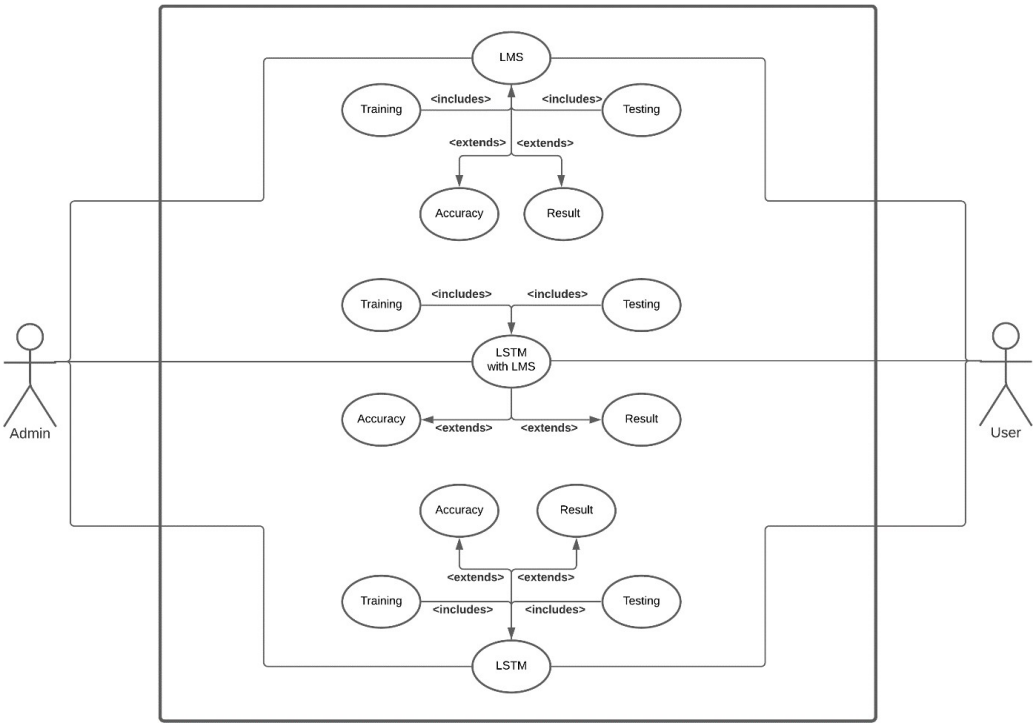


Fig. 8: Using LMS, LSTM and LSTM with LMS in the system

4.1.2 Sequence Diagram

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

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

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

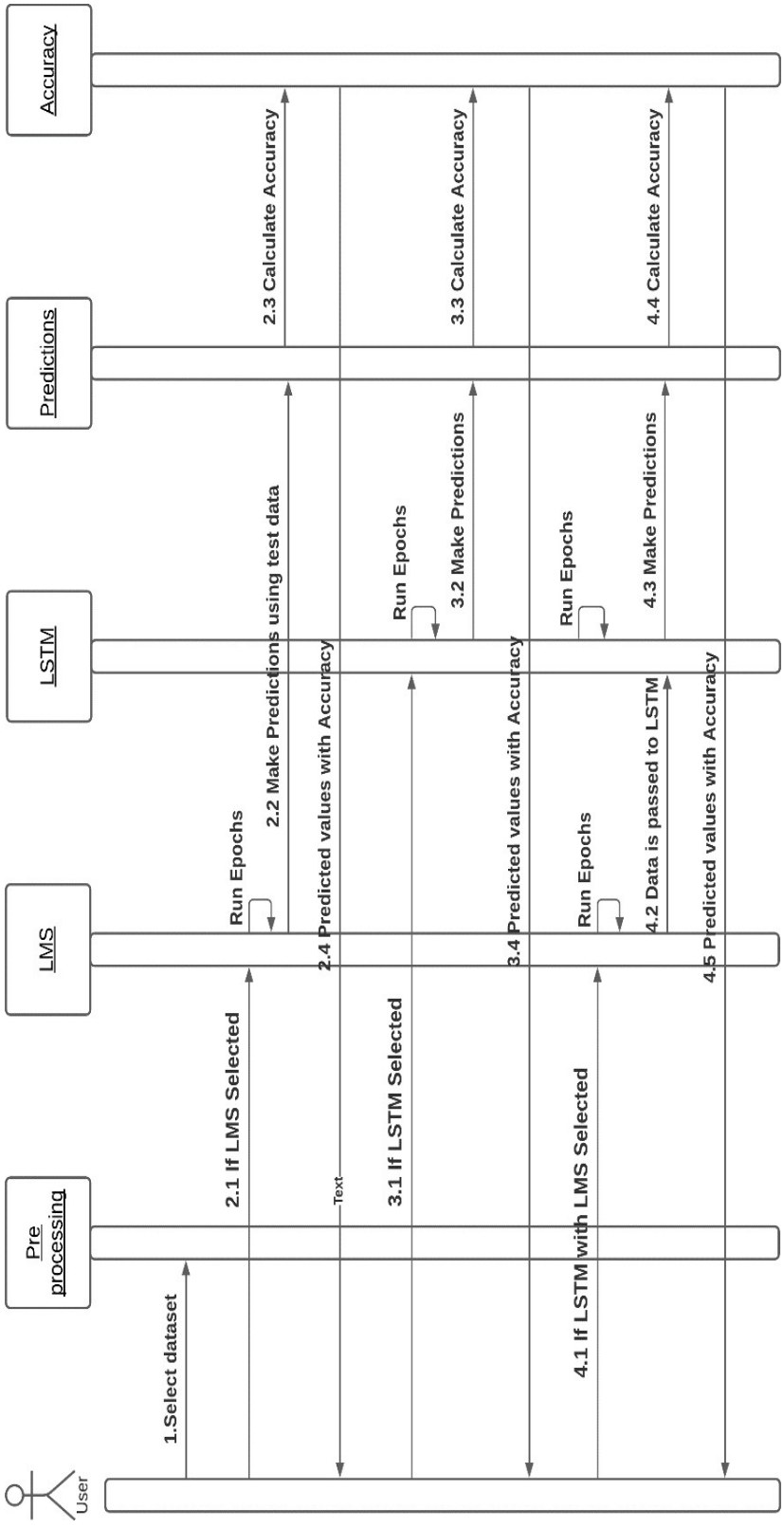


Fig. 9: Execution based on model selection

4.1.3 Activity Diagram

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

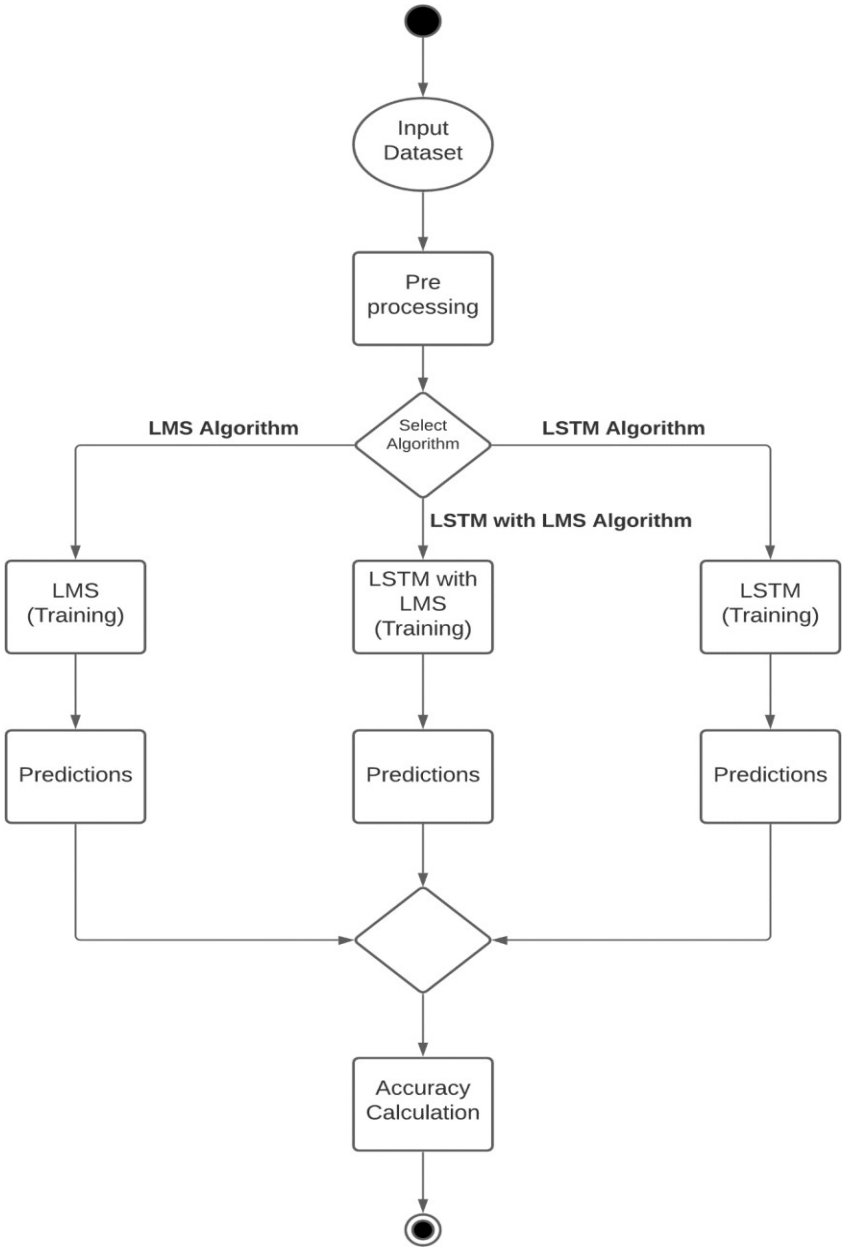


Fig. 10: Execution based on algorithm selection

4.1.4 Collaboration Diagram

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

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

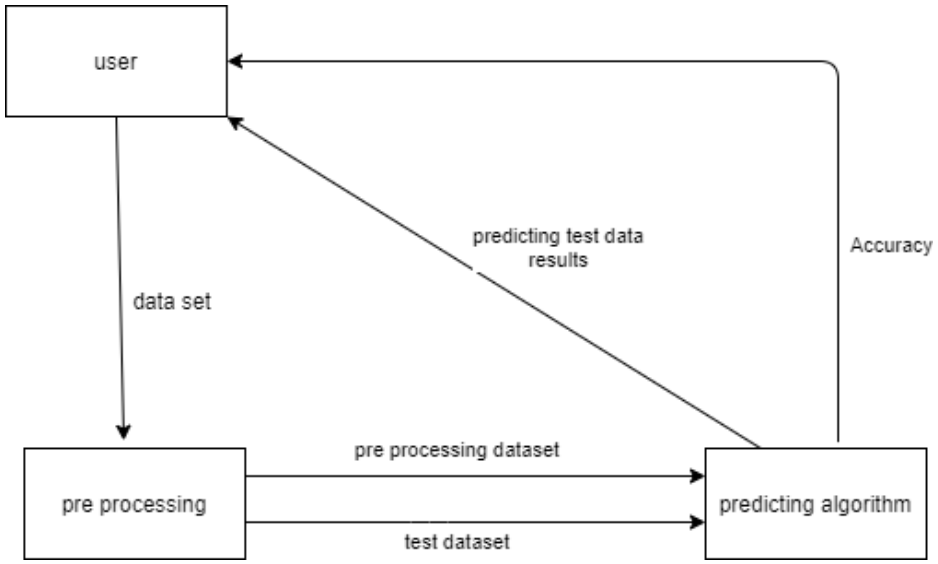


Fig. 11: Data transfer between modules

4.1.5 Flow Chart

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

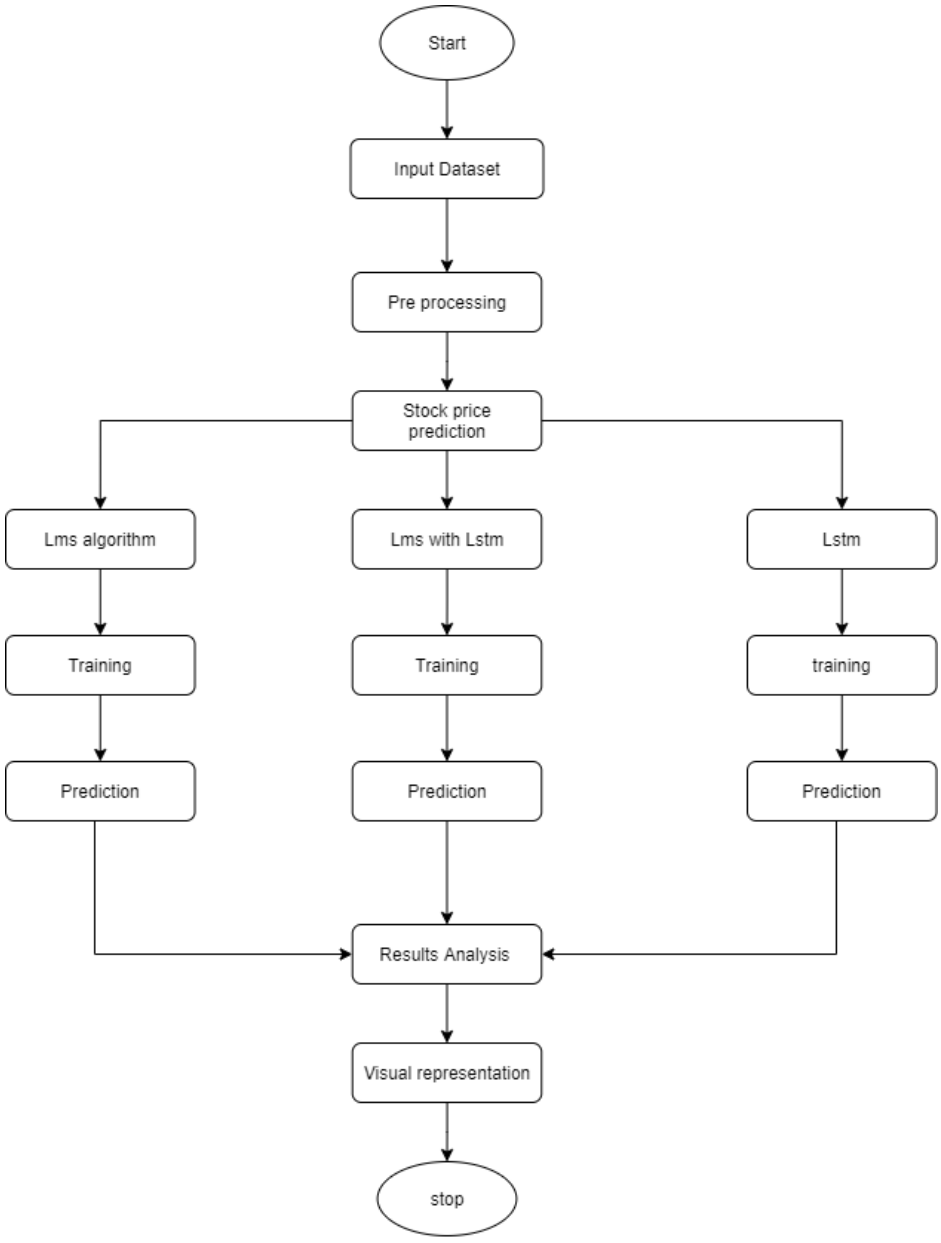


Fig. 12: Flow of execution

4.1.6 Component Diagram

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

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

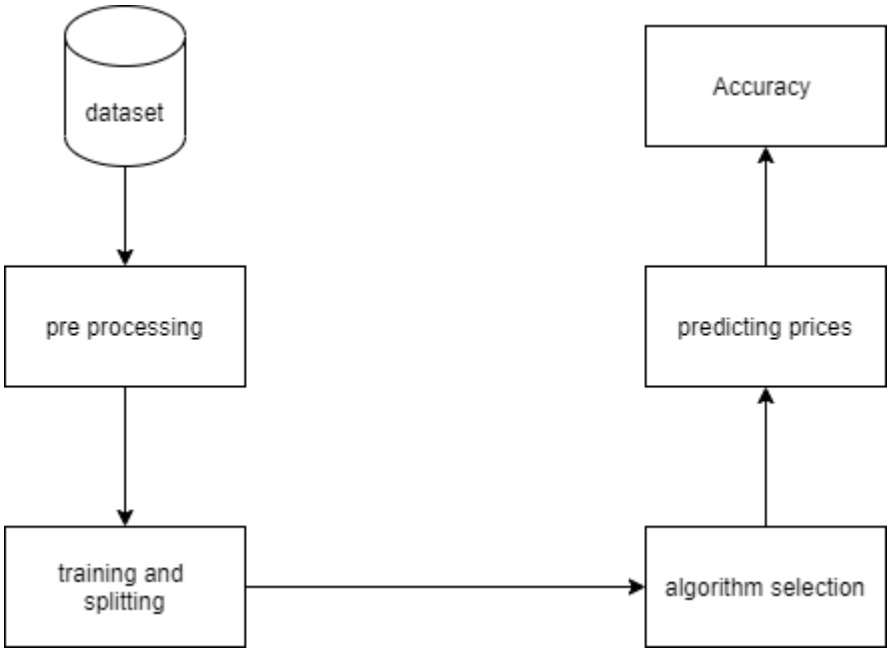


Fig. 13: Components present in the system

Chapter 5: Experiment analysis

5.1 SYSTEM CONFIGURATION

This project can run on commodity hardware.

We ran the entire project on an Intel i5 processor, with 8 GB RAM, 2 GB Nvidia Graphic Processor.

It also has 2 cores which runs at 1.7 GHz and 2.1 GHz respectively.

Hardware Requirements

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

Software Requirements

Python 3.5 and Jupyter Notebook for data pre-processing and prediction.

Windows 7 and above or Linux based OS or MAC OS

Functional requirements

Functional requirements describe what the software should do (the

functions). Think about them as the core operations.

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

- The software shall accept the tw_spydata_raw.csv dataset as input.
- The software 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 thesoftware.
- 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. Theyrepresent some of the core functions.

Non-Functional requirements

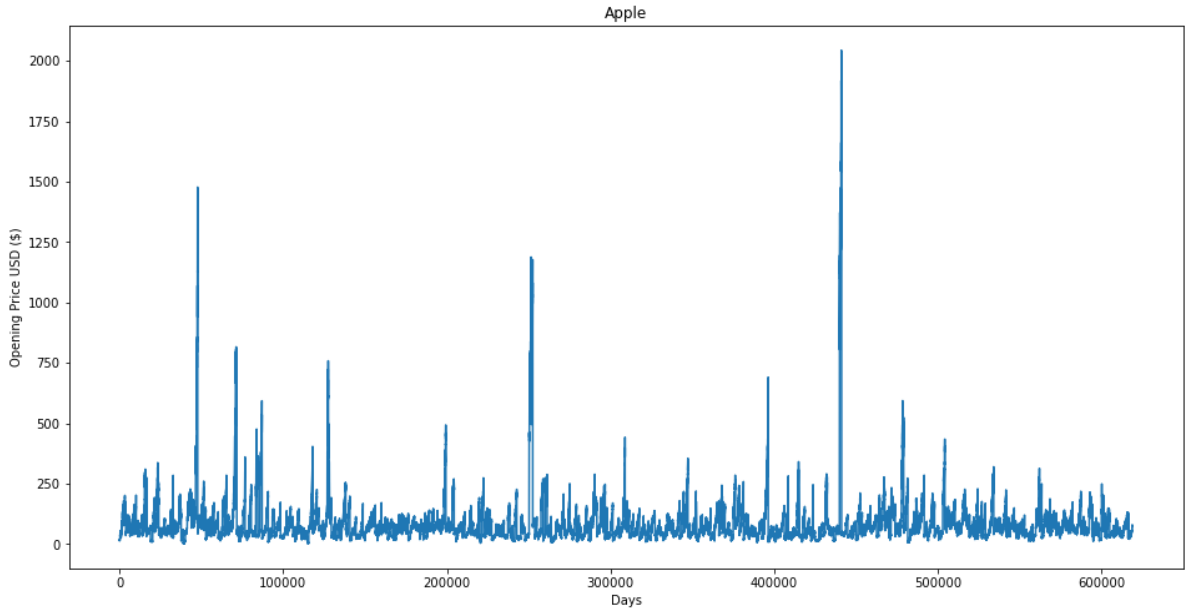
Usability: It defines the user interface of the software in terms of simplicity of understanding the user interface of stock prediction software, for any kind of stock trader and other stakeholders in stock market.

Efficiency: maintaining the possible highest accuracy in the closing stock prices in shortest time with available data.

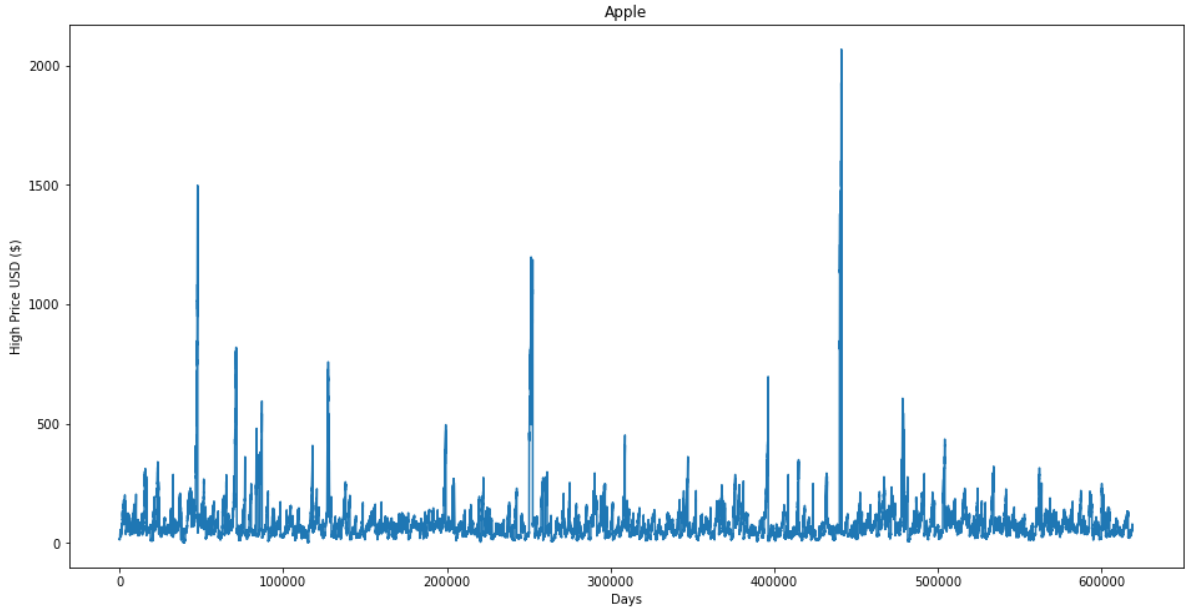
Performance: It is a quality attribute of the stock prediction software that describes the responsiveness to various user interactions with it.

5.2 SAMPLE CODE

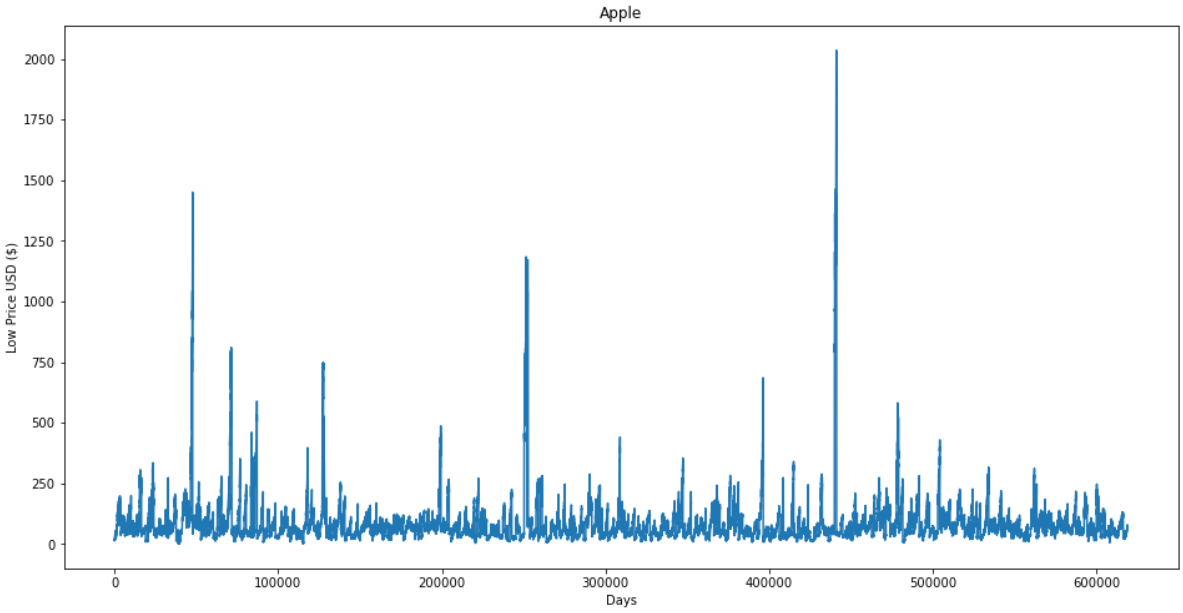
```
# Importing all necessary libraries.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Using data from Apple's stock.
df = pd.read_csv('all_stocks_5yr.csv')

df.head()
df.info()
df.describe()
df.shape

# Visualizing the opening prices of the data.
plt.figure(figsize=(16,8))
plt.title('Apple')
plt.xlabel('Days')
plt.ylabel('Opening Price USD ($)')
plt.plot(df['open'])
plt.show()
```



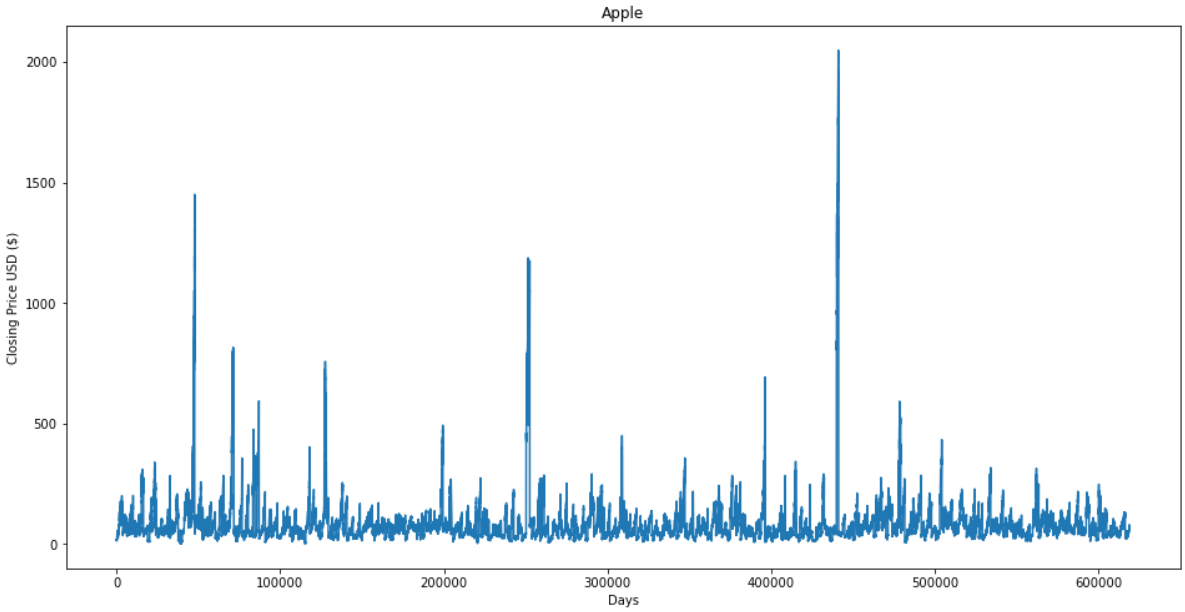

```
# Visualizing the high prices of the data.
plt.figure(figsize=(16,8))
plt.title('Apple')
plt.xlabel('Days')
plt.ylabel('High Price USD ($)')
plt.plot(df['high'])
plt.show()
```



```
# Visualizing the low prices of the data.
plt.figure(figsize=(16,8))
plt.title('Apple')
plt.xlabel('Days')
plt.ylabel('Low Price USD ($)')
plt.plot(df['low'])
plt.show()
```



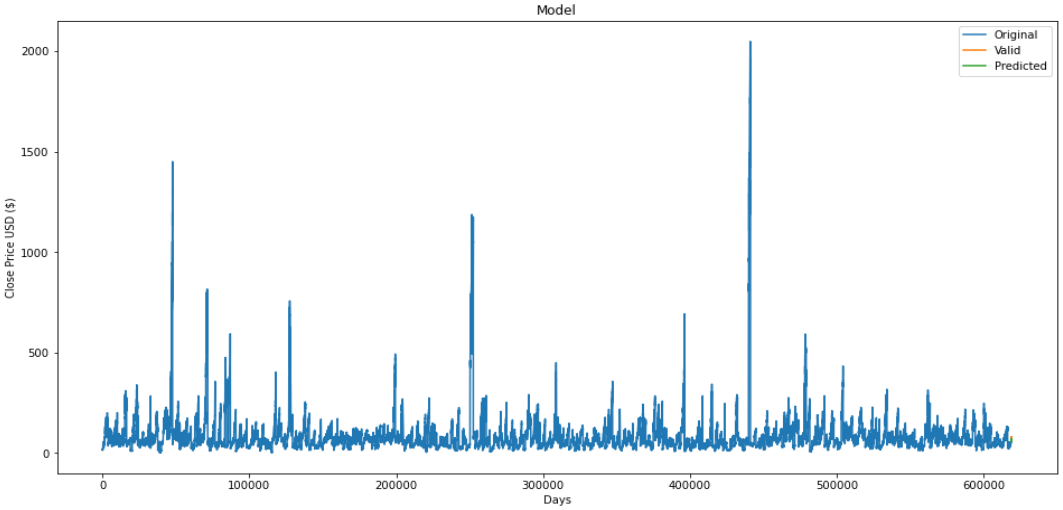

```
# Visualizing the closing prices of the data.
plt.figure(figsize=(16,8))
plt.title('Apple')
plt.xlabel('Days')
plt.ylabel('Closing Price USD ($)')
plt.plot(df['close'])
plt.show()
```

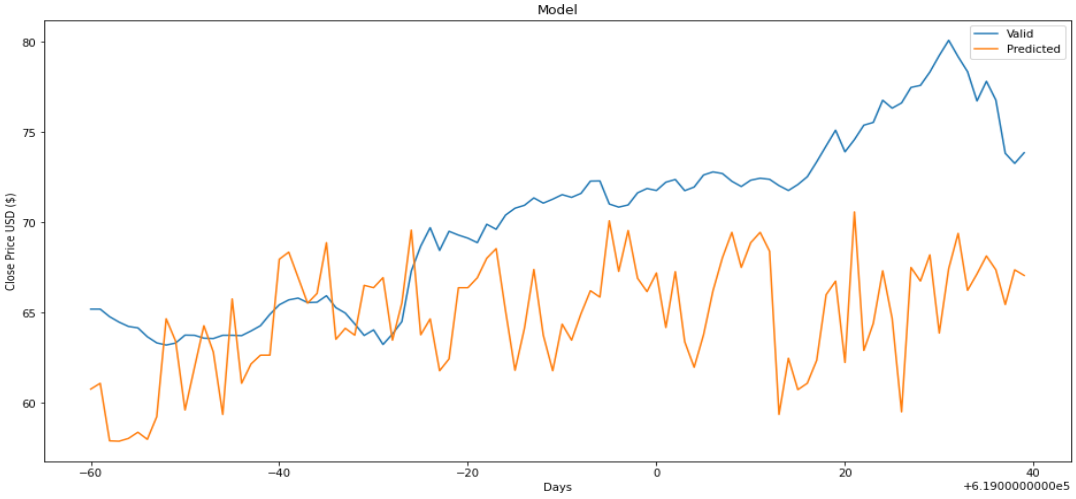



```
df2 = df['close']

df2.tail()

df2 = pd.DataFrame(df2)

df2.tail()

# Prediction 100 days into the future.
future_days = 100
df2['Prediction'] = df2['close'].shift(-future_days)

df2.tail()

X = np.array(df2.drop(['Prediction'], 1))[:-future_days]
print(X)

y = np.array(df2['Prediction'])[:-future_days]
print(y)
```

```
# train test splitting
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)

from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import LinearRegression

# Implementing Linear and Decision Tree Regression Algorithms.
tree = DecisionTreeRegressor().fit(x_train, y_train)
lr = LinearRegression().fit(x_train, y_train)

x_future = df2.drop(['Prediction'], 1)[:-future_days]
x_future = x_future.tail(future_days)
x_future = np.array(x_future)
x_future

tree_prediction = tree.predict(x_future)
print(tree_prediction)

lr_prediction = lr.predict(x_future)
print(lr_prediction)

predictions = tree_prediction
valid = df2[X.shape[0]:].copy()
valid['Predictions'] = predictions

plt.figure(figsize=(16,8))
plt.title("Model")
plt.xlabel('Days')
plt.ylabel('Close Price USD ($)')
```

```
plt.plot(df2['close'])
plt.plot(valid[['close','Predictions']])
plt.legend(["Original","Valid","Predicted"])
plt.show()

plt.figure(figsize=(16,8))
plt.title("Model")
plt.xlabel('Days')
plt.ylabel('Close Price USD ($)')
plt.plot(valid[['close','Predictions']])
plt.legend(["Valid","Predicted"])
plt.show()
```





Chapter 6: Conclusion and future work

6.1 CONCLUSION

In this project, we are predicting closing stock price of any given organization.
We developed an application for predicting closing stock price using LMS and LSTM algorithms for prediction.
We are using a pre-trained model which guarantees 85%-90% of accuracy.

6.2 FUTURE WORK

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