CRYPTOCURRENCY PRICE PREDICTION

MINI PROJECT REPORT (19CS67C)

Submitted by

KARTHIKEYAN.I (1912014)

RAGUL.T N (1912021)

VAIDYANATHAN.K (1912033)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Estd :1984

NATIONAL ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University – Chennai)

K.R. NAGAR, KOVILPATTI – 628 503

November 2022

BONAFIDE CERTIFICATE

This is to certify that this project **CRYPTOCURRENCY PRICE PREDICTION SYSTEM** is the bonafide work of **KARTHIKEYAN I (REGISTER NO: 1912014), RAGUL T N (REGISTER NO: 1912021)** and **VAIDYANATHAN K (REGISTER NO: 1912033)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.GR Hemalakshmi	**Dr.V. GOMATHI M.Tech.,Ph.D.,**
Assistant Professor(Sr. Grade)	**Professor & Head**
Department of CSE,	**Department of CSE,**
National Engineering College	National Engineering College
(An Autonomous Institution)	(An Autonomous Institution)
K.R.Nagar, Kovilpatti: 628503.	K.R.Nagar, Kovilpatti: 628503.

Submitted to the **MINI PROJECT (19CS67C)** Viva-Voce Examination held at **NATIONAL ENGINEERING COLLEGE, K.R.NAGAR, KOVILPATTI** on…………

Internal Examiner **External Examiner**

Abstract

Cryptocurrencies are a digital way of money in which all transactions are held electronically. It is a soft currency that doesn't physically exist in the form of hard notes. Here, we are emphasizing the difference between fiat currency which is decentralized, and that without any third-party intervention all virtual currency users can get the services. To the best of our knowledge, our target is to implement efficient deep learning-based prediction models, specifically long short-term memory (LSTM) and gated recurrent unit (GRU) to handle the price volatility of bitcoin and to obtain high accuracy. Our study involves comparing these two-time series deep learning techniques and proving the efficacy in forecasting the price of bitcoin. One of the most important and famous applications of blockchain technology, cryptocurrency has attracted extensive attention recently. Empowered by blockchain technology, all the transaction records of cryptocurrencies are irreversible and recorded in blocks. These transaction records containing rich information and complete traces of financial activities are publicly accessible, thus providing researchers with unprecedented opportunities for data mining and knowledge discovery in this area. Networks are a general language for describing interacting systems in the real world, and a considerable part of existing work on cryptocurrency transactions is studied from a network perspective.

TABLE OF CONTENTS

LIST OF FIGURES

CRYPTOCURRENCY PRICE PREDICTION

MINI PROJECT REPORT (19CS67C)

Submitted by

KARTHIKEYAN.I (1912014)

RAGUL.T N (1912021)

VAIDYANATHAN.K (1912033)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

NATIONAL ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University – Chennai)

K.R. NAGAR, KOVILPATTI – 628 503

November 2022

BONAFIDE CERTIFICATE

This is to certify that this project **CRYPTOCURRENCY PRICE PREDICTION SYSTEM** is the bonafide work of **KARTHIKEYAN I (REGISTER NO: 1912014), RAGUL T N (REGISTER NO: 1912021)** and **VAIDYANATHAN K (REGISTER NO: 1912033)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.GR Hemalakshmi	**Dr.V. GOMATHI M.Tech.,Ph.D.,**
Assistant Professor(Sr. Grade)	**Professor & Head**
Department of CSE,	**Department of CSE,**
National Engineering College	National Engineering College
(An Autonomous Institution)	(An Autonomous Institution)
K.R.Nagar, Kovilpatti: 628503.	K.R.Nagar, Kovilpatti: 628503.

Submitted to the **MINI PROJECT (19CS67C)** Viva-Voce Examination held at **NATIONAL ENGINEERING COLLEGE, K.R.NAGAR, KOVILPATTI** on…………

Internal Examiner **External Examiner**

Abstract

Cryptocurrencies are a digital way of money in which all transactions are held electronically. It is a soft currency that doesn't physically exist in the form of hard notes. Here, we are emphasizing the difference between fiat currency which is decentralized, and that without any third-party intervention all virtual currency users can get the services. To the best of our knowledge, our target is to implement efficient deep learning-based prediction models, specifically long short-term memory (LSTM) and gated recurrent unit (GRU) to handle the price volatility of bitcoin and to obtain high accuracy. Our study involves comparing these two-time series deep learning techniques and proving the efficacy in forecasting the price of bitcoin. One of the most important and famous applications of blockchain technology, cryptocurrency has attracted extensive attention recently. Empowered by blockchain technology, all the transaction records of cryptocurrencies are irreversible and recorded in blocks. These transaction records containing rich information and complete traces of financial activities are publicly accessible, thus providing researchers with unprecedented opportunities for data mining and knowledge discovery in this area. Networks are a general language for describing interacting systems in the real world, and a considerable part of existing work on cryptocurrency transactions is studied from a network perspective.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Bitcoin has been growing extremely rapidly since 2017. It has frequently been experiencing price hikes, therefore attracting numerous investors. Its popularity has grown to a level that in addition to many private companies, a country has recently announced to be accepting it as a form of payment. Not to mention, researchers from central banks have been analyzing and debating on it since at least 2014.

The major factor distinguishing Bitcoin from other types. of currency is its decentralization. That is, unlike other currencies, Bitcoin transactions are not processed and/or supervised by any government. Its money supply increases over time, though not linearly, by a reward process called "mining". Computers engage in solving mathematical equations through brute force, receiving bitcoins as a reward. Indeed, the classic supply and demand laws determine the exchange rate of Bitcoin prices to other currencies.

The algorithm performs very well for sequential data such as time series, speech, text, financial data, audio, video, weather, and more. RNNs are able to form a much deeper understanding of a sequence and its context compared to other algorithms.In an RNN, the information goes through a cycle. When making a decision, it considers the current input and also what it has learned from the inputs it has received previously. there is a lot of controversy over innovative developments. Cryptocurrencies are digital currencies where transactions can be done by online transactions, unlike the common currency, cryptocurrency is designed based on cryptography.

Therefore, as an emerging and interdisciplinary research area, increasing research efforts have been devoted to the analysis and mining of cryptocurrency transactions from a network perspective. Studies in this area not only advance theories and applications of graph data mining techniques on financial systems but also benefit the development of financial security and regulation technologies of blockchain-based cryptocurrencies. In this paper, we aim to provide a comprehensive review and summary of existing literature and state-of-the-art techniques in this area, with a focus on modeling, profiling, and prediction issues of cryptocurrency transaction networks. In particular, since Bitcoin (Nakamoto, 2008) and Ethereum (Wood et al., 2014) are the two largest and relatively mature blockchain systems, much of existing research focuses on these two systems.

Due to the open and transparent nature of blockchain, cryptocurrency transaction records containing rich information and complete traces of financial activities are publicly accessible, thus providing researchers with unprecedented opportunities for data mining in this area. The main value of analyzing and mining the transaction data of cryptocurrencies is twofold: (1) Transaction records in traditional financial scenarios are relatively unexplored in existing studies as these transaction records are usually not publicly accessible for the sake of security and interest. Through analysis and mining of cryptocurrency transaction records, we can extensively explore the trading behaviors, wealth distribution, and generative mechanism of a transaction system, as well as infer reasons for fluctuations in the financial market of cryptocurrencies. This study can also provide a reference for knowledge discovery in other financial systems. (2) Due to the anonymity of blockchain systems and the lack of authority, various types of cybercrimes have arisen in the blockchain ecosystem during recent years. Extracting information from the transaction records can help track cryptocurrency transactions and identify illegal behaviors, thereby establishing effective regulation and building a healthier blockchain ecosystem.

While the history of implementation of artificial neural network methods to predict other financial processes (e.g., stocks prices) is lengthy, the literature of prediction of cryptocurrency prices is not as much due to its novel nature. Despite this, over the last few years a growing number of researchers have attempted to create hybrid models of artificial neural networks that predict the price and/or price volatility of cryptocurrency prices, with a major focus on Bitcoin.

1.1 SYSTEM SPECIFICATIONS

Non-Functional requirements

Hardware Requirements:

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

Software Requirements:

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

Functional requirements

Functional requirements describe what the software should do (the functions). Think about the core operations.Because the "functions" are established before development, functional requirements should be written in the future tense. In developing the software for Stock Price Prediction,

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract.They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. Some of the functional requirements could include the following:

- The software shall accept the tw_spydata_raw.csv dataset as input.
- The software shall use LSTM architecture as the main component of the software.
- It processes the given input data by producing the most possible outcomes .

Non-Functional requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, "how fast does the website load?" Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non Functional requirements in Software Engineering allows you to impose constraints or restrictions on the design of the system across the various agile backlogs.

One of the NFR examples is Employees never allowed to update their salary information. Such an attempt should be reported to the security administrator. Functional Requirement is a verb while Non-Functional Requirement is an attribute The advantage of Non-functional requirement is that it helps you to ensure good user experience and ease of operating the software

Product properties

- **Usability**: It defines the user interface of the software in terms of the simplicity.
- **Efficiency**: maintaining the possible highest accuracy in the closing stock prices.
- **Performance**: It is a quality attribute of responsiveness to user interactions with it.
- **Usability**: It defines the user interface of the software in terms of the simplicity.
- **Efficiency**: maintaining the possible highest accuracy in the closing stock prices.

CHAPTER 2
LITERATURE SURVEY

Sina E. Charandabi,Kamyar Kamyar(2021) [1] -The purpose of the study is to examine whether the application of deep learning-based dual-stage Partial Least Square-Structural Equation Modeling (PLS-SEM) & Artificial Neural Network (ANN) analysis enable better in-depth research results as compared to single-step PLS-SEM approach and to excavate factors which can predict behavioral intention to adopt cryptocurrency.For the remaining majority of the paper, recent cryptocurrency price data of Bitcoin, Ethereum, Tether, Dogecoin, and Binance coin was used to train a machine learning model of Feed Forward Neural Networks to predict future prices for each of the datasets. Further and in conclusion, the results are discussed, and the efficiency and accuracy of these models are evaluated.

Ahmed M. Khedr,Ifra Arif,Pravija Raj P V(2021) [2] – Traditional statistical methods, require a lot of statistical assumptions that could be unrealistic, leaving machine learning as the best technology in this field.This article provides a comprehensive summary of the previous studies in the field of cryptocurrency price prediction from 2010 to 2020. The discussion presented in this article will help researchers to fill the gap in existing studies and gain more future insight. Integrated Moving Average (ARIMA) model to predict the prices of the three major cryptocurrencies-Bitcoin, XRP and Ethereum-using daily, weekly and monthly time series. The results demonstrated that ARIMA outperforms most other methods in predicting cryptocurrency prices on a daily time series basis in terms of mean absolute error

Edwin, Lipo Wang (2017) [3] - Described The features of Bitcoin and the next day change in the price of Bitcoin using an Artificial Neural Network ensemble. The features of Bitcoin and the next day change in the price of Bitcoin using an Artificial

Neural Network ensemble approach called Genetic Algorithm-based Selective Neural Network Ensemble, constructed using Multi-Layered Perceptron as the base model for each of the neural network in the ensemble.This paper explores the relationship between the features of Bitcoin and the next day change in the price of Bitcoin using an Artificial Neural Network ensemble approach called Genetic Algorithm based Selective Neural Network Ensemble, constructed using Multi-Layered Perceptron as the base model for each of the neural network in the ensemble.

Yeray Mezquita, Ana Belén Gil-González, Javier Prieto, Juan Manuel Corchado (2021) [4]- This paper proposes a platform based on blockchain technology and the multi-agent system paradigm to allow for the creation of an automated peer-to-peer electricity market in micro-grids. The use of a permissioned blockchain network has multiple benefits as it reduces transaction costs and enables micro-transactions. They are gradually spreading to other services where a decentralized, reliable and immutable model makes sense. One of the fields where the use of Blockchain Technologies is spreading the most is in Smart Contracts, computer programs which are executed in the blockchains establishing a collection of clauses between the participating parties that agree to interact with each other and that are executed automatically at the moment in which these clauses are fulfilled

Lloyd Kasal, Mihir Shetty , Tanmay Nayak , Ramanath Pai , Shilpa B(2022) [5]- proposed to analyze in cryptocurrency values, numerous different kinds of neural networks may be utilized. The most successful of them all has been determined to be LSTM. The key factors used are available price, close price, high price, low price, volume and market cap with the interdependencies amid some cryptocurrencies, thus centers on evaluating vital features that influence the trade’s unpredictability by applying the model to increase the effectiveness of this process.

Ahmed M. Khedr Ifra Arif Pravija Raj P V Magdi El-Bannany (2021) [6]- proposed that traditional statistical methods, although simple to implement and interpret, require a lot of statistical assumptions.This article provides a comprehensive summary of the previous studies in the field of cryptocurrency price prediction from 2010 to 2020. The discussion presented in this article will help researchers to fill the gap in existing studies and gain more future insight.Cryptocurrency price prediction is one of the trending areas among researchers. Research work in this field uses traditional statistical and machine‑learning techniques, such as Bayesian regression, logistic regression, linear regression, support vector machine, artificial neural network, deep learning, and reinforcement learning. No seasonal effects exist in cryptocurrency, making it hard to predict using a statistical approach

Caporale, Guglielmo Maria Plastun, Alex (2018) [7],- A trading robot approach is then used to establish whether these statistical anomalies can be exploited to generate profits. The results suggest that a strategy based on counter-movements after overreactions is not profitable, whilst one based on inertia appears to be profitable but produces outcomes not statistically different from the random ones.The results suggest that a strategy based on counter-movements after overreactions does not generate profits in the FOREX and the commodity markets, but in some cases it can be profitable in the US stock market. By contrast, a strategy exploiting the "inertia anomaly" produces profits in the case of the FOREX and the commodity markets, but not in the case of the US stock market.

Al-Yahyaee KH, Mensi W(2020) [8] -Multifractality, long-memory process, and efficiency hypothesis of six major cryptocurrencies using the time-rolling MF-DFA approach. The causality between cryptocurrencies and economic factors is undirected. Interestingly, our findings show that cryptocurrencies are insignificant correlations with economic factors. The result implies that cryptocurrencies can not be assumed as financial assets to hedge systematic risks from economic factors. This study assesses the efficiency of Bitcoin market compared to gold, stock and foreign exchange markets. By applying a MF-DFA approach, the study found that the long-memory feature and multifractality of the Bitcoin market was stronger and Bitcoin was therefore more inefficient than the gold, stock and currency markets.

Magdi El-Bannany,Saadat M. Alhashmi, Meenu Sreedharan(2016) [9] - Cryptocurrency price prediction using traditional statistical and machine-learning techniques,this type of model approach called Genetic Algorithm based Selective Neural Network Ensemble.Wide fluctuations in cryptocurrency prices motivate the urgent requirement for an accurate model to predict its price. Cryptocurrency price prediction is one of the trending areas among researchers. Research work in this field uses traditional statistical and machine‑learning techniques. Traditional statistical methods, although simple to implement and interpret, require a lot of statistical assumptions that could be unrealistic, leaving machine learning as the best technology in this field, being capable of predicting price based on experience

Ujan Mukhopadhyay;Anthony Skjellum;Oluwakemi;Hambolu;Jon Oakley;Richard Brooks(2022), [10] proposed that Cryptocurrencies require strong, secure mining algorithms. In this paper we survey and compare and contrast current mining techniques as used by major Cryptocurrencies. Mining adds records of past transactions to the distributed ledger known as Blockchain, allowing users to reach secure, robust consensus for each transaction. Mining also introduces wealth in the form of new units of currency. Cryptocurrencies lack a central authority to mediate transactions because they were designed as peer-to-peer systems. Cryptocurrencies lack a central authority to mediate transactions because they were designed as peer-to-peer systems. They rely on miners to validate transactions. Cryptocurrencies require strong, secure mining algorithms. In this paper we survey and compare and contrast current mining techniques as used by major Cryptocurrencies.

Carmine Ventre,Michail Basios,Leslie Kanthan,David Martinez-Rego(2022), [11] This survey analyzes the research distribution that characterize the cryptocurrency Research and distribution among properties . This also analyzes datasets, research trends and distribution among research objects (contents/properties) and technologies, concluding with some promising opportunities that remain open in cryptocurrency trading. This paper also analyses datasets, research trends and distribution among research objects (contents/properties) and technologies, concluding with some promising opportunities that remain open in cryptocurrency trading.

Sina E. Charandabi;Kamyar Kamyar (2021), [12] The purpose of this survey paper is to present and compare multiple research papers that employed multiple neural networks.Researchers would be allowed to decide the more suitable direction of work to be taken in order to provide more accurate alternatives to the field.The academic community has similarly spent considerable efforts in researching cryptocurrency trading. This paper seeks to provide a comprehensive survey of the research on cryptocurrency trading, by which we mean any study aimed at facilitating and building strategies to trade cryptocurrencies. This paper explains the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python.

Saeed Alzahrani; Tugrul U. Daim(2019) [13] -Paper suggests that the main factors driving the adoption decision, the acceptance by businesses as a payment method, the fast transfer of funds.This aims at providing an in-depth analysis of the factors influencing the adoption of cryptocurrency as well as the ranking of these influencing factors based on the quantification of the users' judgments.Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period of time to predict the result in the next time unit. Many time series prediction algorithms have shown their effectiveness in practice

Pravija Raj ,Magdi El-Bannany,Saadat M.Alhashmi,Meenu Sreedharan(2021) [14] –Research work in this field uses traditional statistical and machine-learning techniques, making it hard to predict using a statistical approach. such as Bayesian regression, logistic regression, linear regression, support vector machine, artificial neural network, deep learning, and reinforcement learning. No seasonal effects exist in cryptocurrency, making it hard to predict using a statistical approach.. Many factors are incorporated and considered when developing an ATS, for instance, trading strategy to be adopted, complex mathematical functions that reflect the state of a specific stock, machine learning algorithms that enable the prediction of the future stock value, and specific news related to the stock being analyzed.

George S.Atsalakis(2021) [15] –Fuzzy modeling demonstrating that the closed-loop or feedback control technique can cope with uncertainties associated with the dynamic behavior of the price of Bitcoin, and achieve positive returns. This study proposes a computational intelligence technique that uses a hybrid Neuro-Fuzzy controller, namely PATSOS, to forecast the direction in the change of the daily price of Bitcoin.This study proposes a computational intelligence technique that uses a hybrid Neuro-Fuzzy controller, namely PATSOS, to forecast the direction in the change of the daily price of Bitcoin. The proposed methodology outperforms two other computational intelligence models, the first being developed with a simpler neuro-fuzzy approach, and the second being developed with artificial neural networks. Furthermore, the investment returns achieved by a trading simulation, based on the signals of the proposed model, are 71.21% higher than the ones achieved through a naive buy-and-hold strategy. The performance of the PATSOS system is robust to the use of other cryptocurrencies.

CHAPTER 3

EMPATHY AND CUSTOMER JOURNEY MAP

3.1 EMPATHY MAP

Empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community. Empathy Map (EM) is a method that assists in designing business models according to customer perspectives. It goes beyond demographic characteristics and develops a better understanding of the customer's environment, behavior, aspirations, and concerns. The EM"s goal is to create a degree of empathy for a specific person.

According to Bratsberg, the EM is a user-centered approach, i.e., the focus is on understanding the other individual by looking at the world through his or her eyes. When the stakeholders understand the user, they are able to understand how small changes in design can have a big impact on users.In the first version of the EM, Matthews, proposed four different areas that should be covered when making an Empathy Map of a person.Empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Empathy mapping is a simple workshop activity that can be done with stakeholders, marketing and sales, product development, or creative teams to build empathy for end users. After Bland improved the EM by including Pain and Gain areas Figure 3.2 shows the empathy map of the proposed system.

- See – what the user sees in his/her environment;

- Say and Do – what the user says and how s/he behaves in public;
- Think and Feel– what happens in the user‟s mind;
- Hear –how the environment influences the user;
- Pain– the frustrations, pitfalls, and risks that the user experiences,
- Gain –what the user really wants and what can be done to achieve his/her goals.

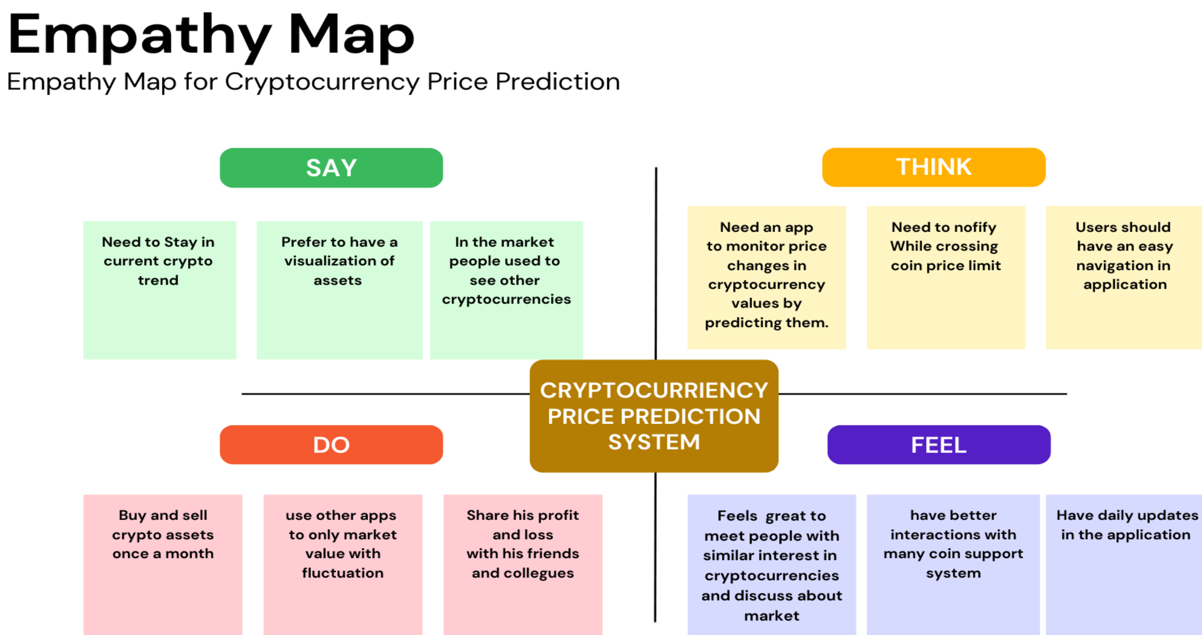


Figure 3.1 Empathy Map for Crypto price prediction

3.2 CUSTOMER JOURNEY MAP

A customer journey map is a very simple idea: a diagram that illustrates the steps your customer(s) go through in engaging with your company, whether it be a product, an online experience, retail experience, or a service, or any combination. The more touch points you have, the more complicated — but necessary — such a map becomes. Sometimes customer journey maps are "cradle to grave," looking at the entire arc of engagement. Here, for example, is a customer journey timeline that includes first engaging with a customer (perhaps with advertising or in a store), buying the product or service, using it, and sharing the experience with others (in person or online), and then finishing the journey by upgrading, replacing, or choosing a competitor (re-starting the journey with another company) the Figure 3.3 (a) and 3.3 (b) shows the customer journey map of the proposed system

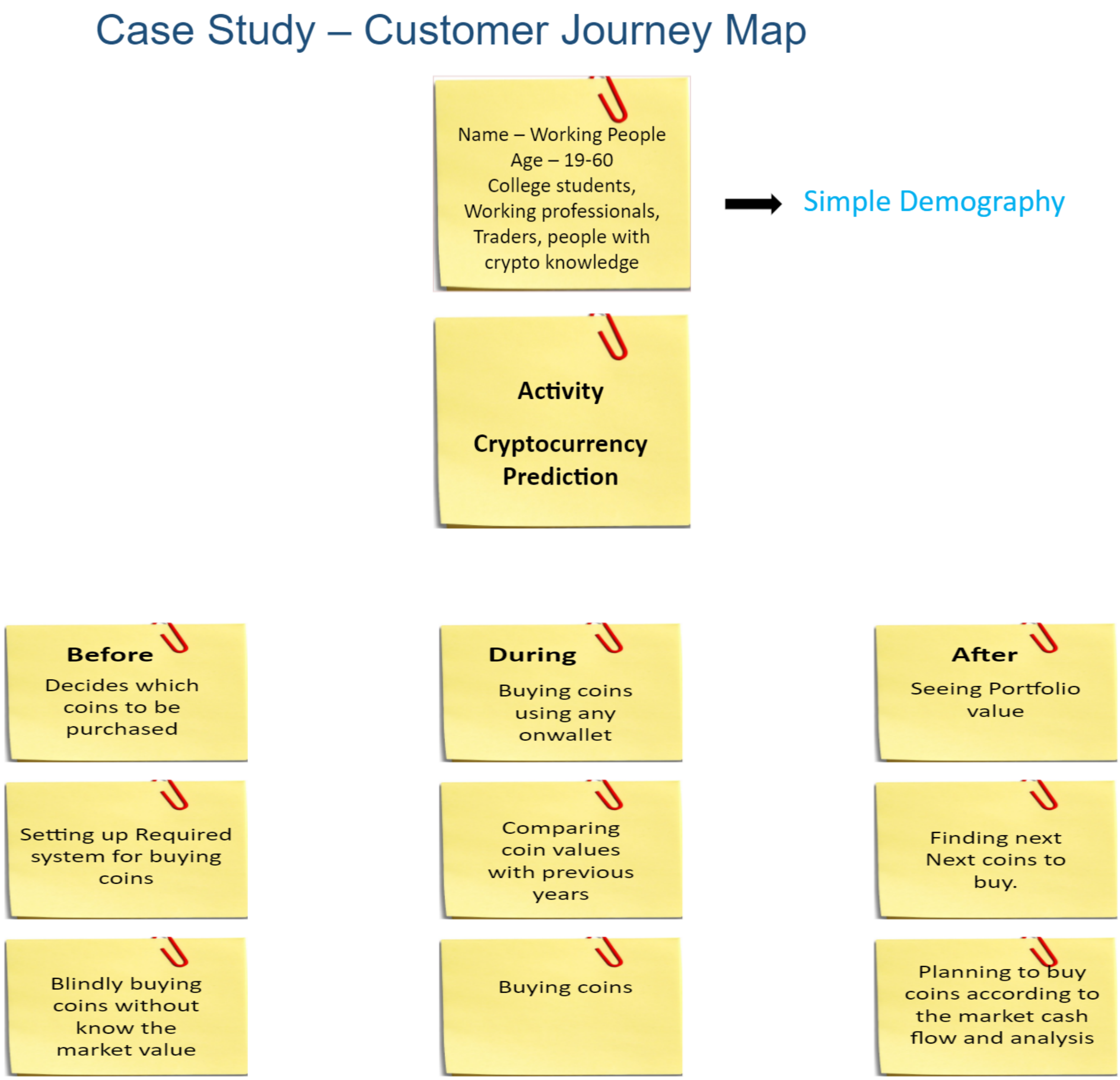


Figure 3.2 Customer Journey Map

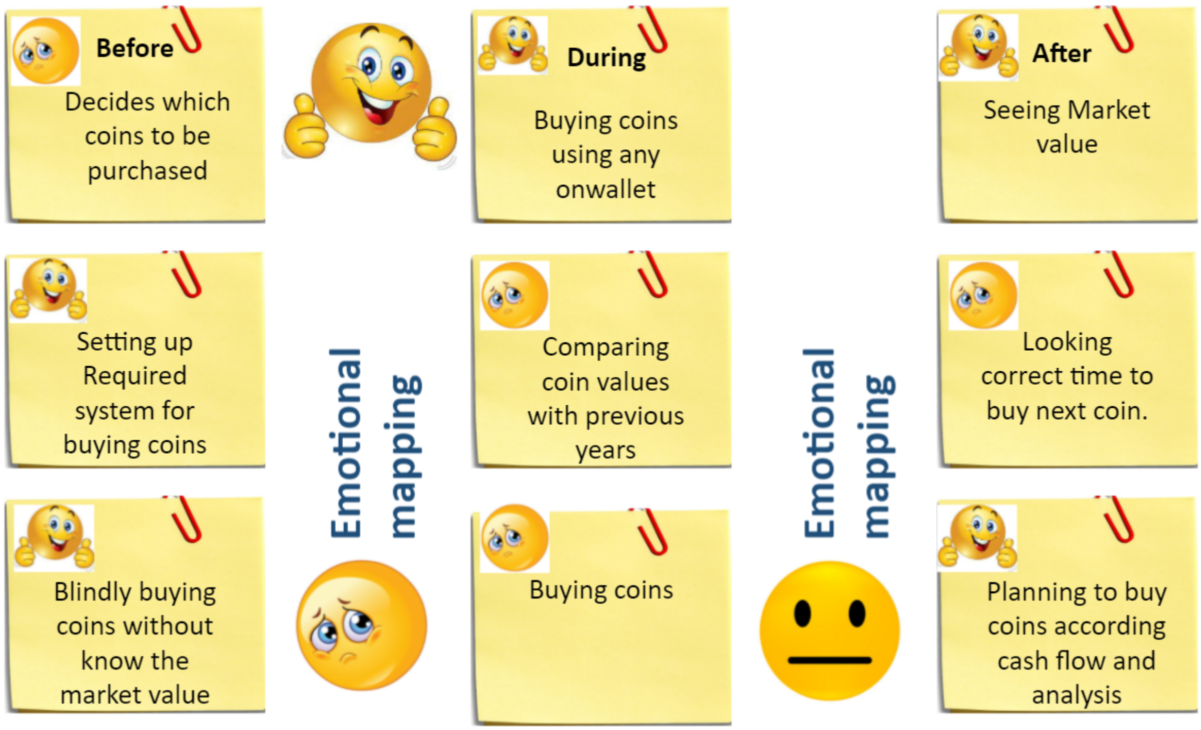


Figure 3.3 Emotional Mapping

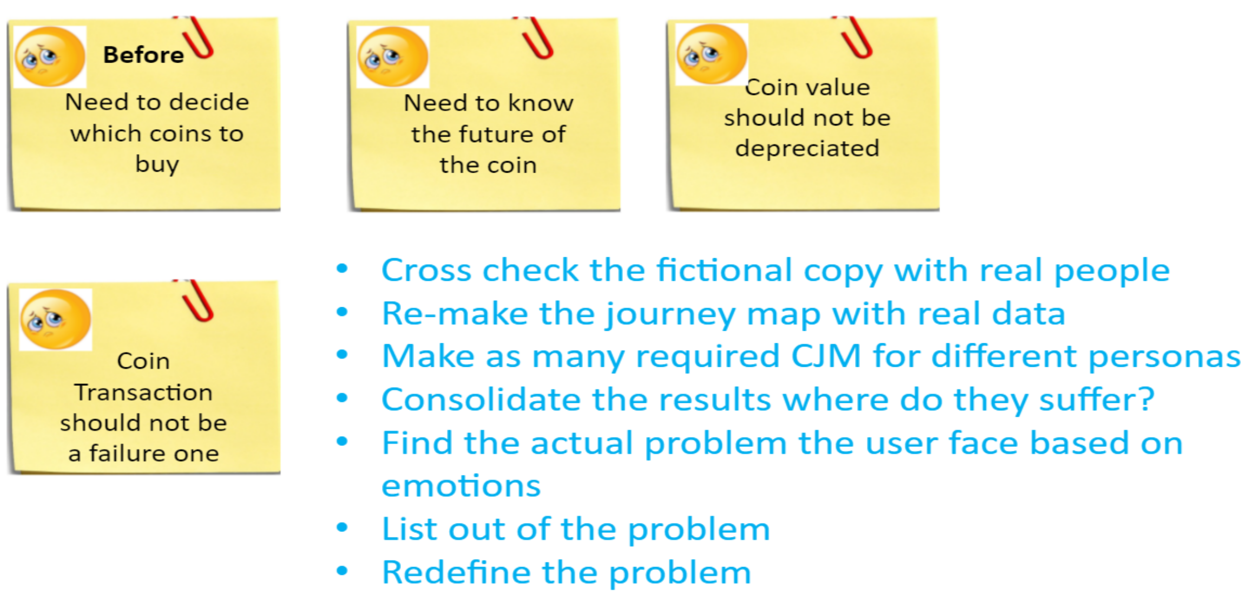


Figure 3.2 Customer Journey Map Result

CHAPTER - 4

PROPOSED METHODOLOGY

4.1 Proposed Architecture

The LSTM model is used to capture the time-dependent aspects of the prices of cryptocurrencies, and an embedding network is presented to capture the hidden representations from linked cryptocurrencies. Both networks are employed in conjunction with each other. The developed technique was used to show the future fluctuations of the prices of the different types of cryptocurrencies over a period of 180 days as a form of long-term forecasting. In the real-world cryptocurrency market, we experimentally demonstrated the usefulness of our LSTM model. In addition, LSTM showed state-of-the-art performance that was superior to those of all other existing models. The Preprocessing of the data for our model is shown below in figure 4.1. The process of preparing the raw data and making it suitable for a machine learning model.

The most basic workflow for data mining, and therefore machine learning, can be divided into six steps. In the first step data acquisition has to be mentioned, as insufficient or biased data can lead to wrong results. In machine learning or data mining this data usually has to be quite big, as patterns might only emerge with thousands or millions of individual data points.The result is heavily dependent on the complexity of these clusters, as is in regression the result dependent on the complexity of a curve. It is the first and crucial step while creating a machine learning model.As illustrated preprocessing data for machine learning is something of an art form and requires careful consideration of the raw data in order to select the correct strategies and preprocessing techniques. In data mining and machine learning an abundance of models and algorithms can be found, but most fundamentally these are divided into supervised and unsupervised learningThe 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.

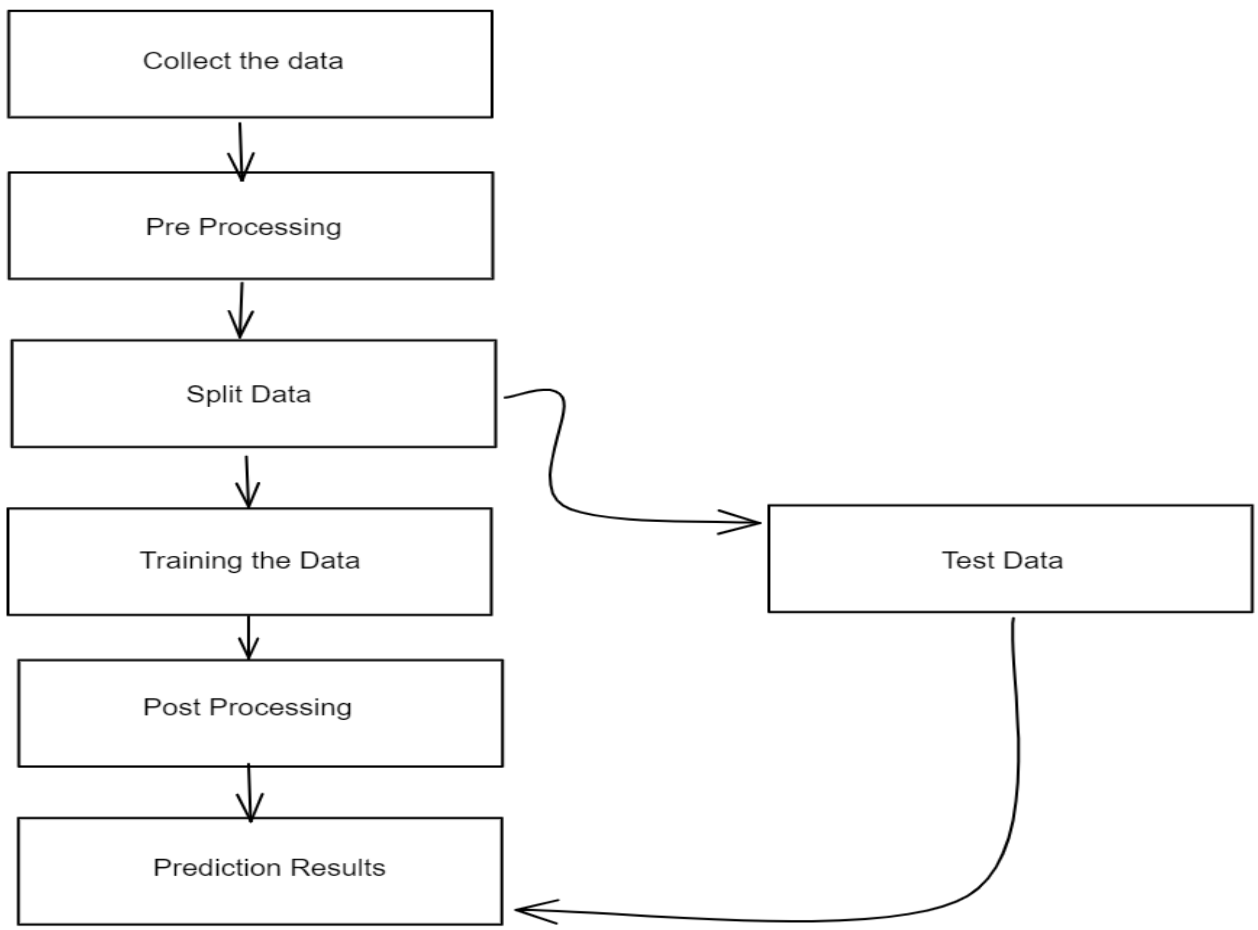


Fig 4.1 Building the model

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. As all Bitcoin transactions are recorded in a public ledger that is accessible to the public, developers have created tools that can predict Bitcoin prices on the basis of the transaction identifier, sender, receiver, value, and timestamp included in each transaction. Bitcoin prices are based on identifying the edges that occur most frequently in the transaction network. This strategy obtained positive results. According to the findings of another study many indicators have been utilized to forecast the development of Bitcoin prices over time. These indicators include blockchain data (e.g., the number of transactions per block, median confirmation time, hash rate, and level of difficulty) and macroeconomic variables.These indicators include blockchain data (e.g., the number of transactions per block, median confirmation time, hash rate, and level of difficulty) and macroeconomic variables. The proposed system provide a precise explanation of the problem and discuss four distinct types of characteristics. This work has the potential to contribute to the advancement of research on cryptocurrencies and to supply investors with more tools for conducting investment assessments. To overcome the problem of price fluctuation prediction.

The proposed system provide a precise explanation of the problem and discuss four distinct types of characteristics. This work has the potential to contribute to the advancement of research on cryptocurrencies and to supply investors with more tools for conducting investment assessments. To overcome the problem of price fluctuation prediction.

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 a Long short-term memory network (LSTM).The LSTM is made up of four neural networks and numerous memory blocks known as cells in a chain structure.

A conventional LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. The flow of information into and out of the cell is controlled by three gates, and the cell remembers values over arbitrary time intervals.

The LSTM algorithm is well adapted to categorize, analyze, and predict time series of uncertain duration. LSTMs provide us with a large range of parameters such as learning rates, and input and output biases. Hence, no need for fine adjustments.. In order to build a machine learning model, it is the first and most important stage. As demonstrated, selecting the appropriate strategies and preprocessing approaches for machine learning needs careful examination of the raw data and is somewhat of an art form.The historical data that were collected from the Internet and used as experimental data in this study…

4.2Model Architecture

The LSTM Algorithm uses a three-gate technique to store the state of the network. Memory loss from a concealed state can be controlled by means of the forget gate (ft), the first gate. When a new piece of information is received, an input gate determines how much of it is to be kept in the current cell state.

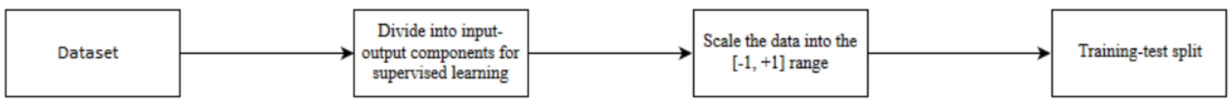


Fig 4.2 Training of the Data

The Overall architecture of our Model is shown below in figure 4.3. Here, after applying both the models for bitcoin prediction, we can determine which model is much more accurate for the future fulfillment of our target and select appropriate parameters to obtain a better performance. In this work, we have proposed deep learning mechanisms such as LSTM and GRU which are the latest and efficient techniques for the forecasting of bitcoin price. As bitcoin is the most popular cryptocurrency, the price volatility issue should be handled within a short period of time.The LSTM model is used to capture the time-dependent aspects of the prices of cryptocurrencies, and an embedding network is presented to capture the hidden representations from linked cryptocurrencies. Both networks are employed in conjunction with each other. The developed technique was used to show the future fluctuations of the prices of the different types of cryptocurrencies over a period of 180 days as a form of long-term forecasting. In the real-world cryptocurrency market, we experimentally demonstrated the usefulness of our LSTM model. In addition, LSTM showed state-of-the-art performance that was superior to those of all other existing models.The LSTM Algorithm uses a three-gate technique to store the state of the network. Memory loss from a concealed state can be controlled by means of the forget gate (ft), the first gate. When a new piece of information is received, an input gate (it=) determines how much of it is to be kept in the current cell state.

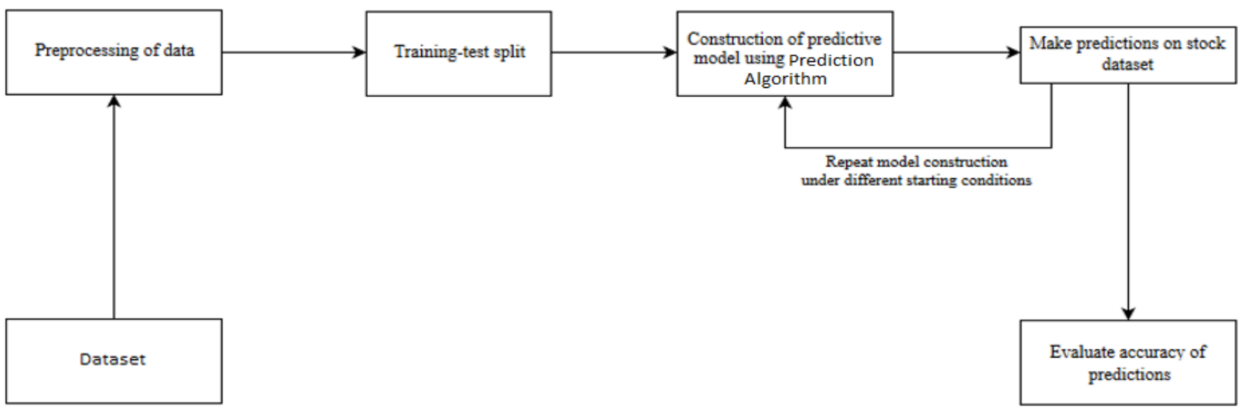


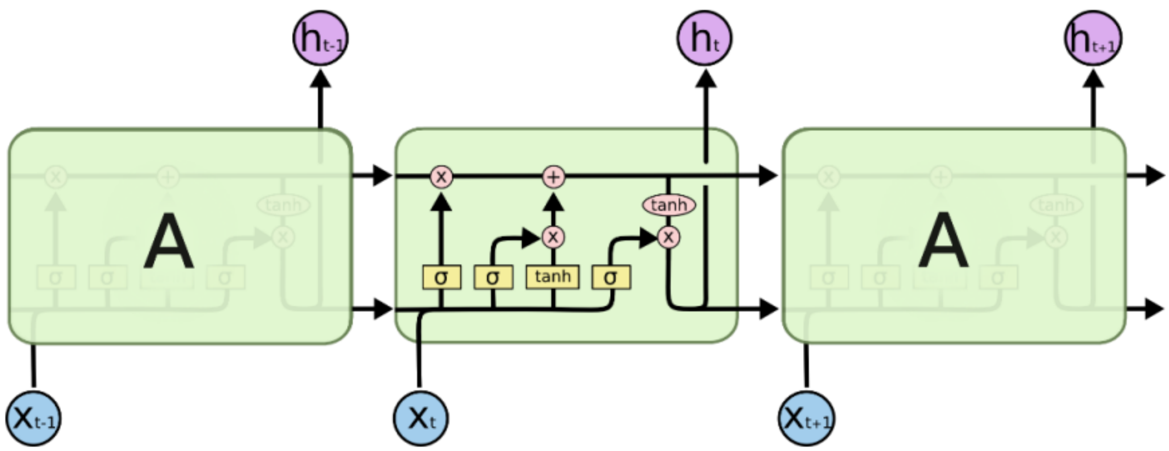
Fig 4.3 Overall Architecture

Firstly, we collect the data set from the online source: Yahoo Finance. The data set represents the Bitcoin price in United States Dollars (USD). The dataset includes all the information about bitcoin prices from 27th October, 2015 to 30th October, 2021. After the dataset has been filtered and cleaned, we need to generate a lag plot of the time series data. A lag in a time series data defines how much a data point is falling behind in time from another data point. A lag plot is a special type of scatter plot in which the X-axis represents the dataset with some time units behind or ahead as compared to the Y-axis. The difference between these time units is called lag or lagged and it is represented by *k*.

Lag plots are put into use to analyse and find out whether the time series data follows any pattern. They are essential for searching patterns like trends, randomness, and seasonality. The plot can be brought about by the representation of time series data in x-axis, and the lag of the time series data points in the y-axis. We are plotting lag plots for a minute lag, an hourly lag, daily lag, weekly lag, and a monthly lag.

4.3 LSTM Architecture

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



`

Fig 4.4 LSTM Architecture

LSTM (Long Short Term Memory) is another type of module provided for RNN. LSTM was created by Hochreiter & Schmidhuber (1997) and later developed and popularized by many researchers. Like RNN, the LSTM network (LSTM network) also consists of modules with recurrent consistency.

LSTM is an updated version from RNN, the difference is the connection between the hidden layers of RNN. The explanation structure of RNN is shown in Figure 5.2 RNN & LSTM have a similar structure, the other difference is the memory cell of the structure hidden layer. And the design of three special gates effectively solve the gradient problems. The LSTM memory structure of the hidden layer.

Machine learning is concerned with the development of algorithms that automatically improve by practice. Ideally, the more the learning algorithm is run, the better the algorithm becomes. It is the task of the learning algorithm to create a classifier function from the training data presented. The performance of this built classifier is then measured by applying it to previously unseen data. Artificial Neural Networks (ANN) are inspired by biological learning systems and loosely model their basic functions. Neurons are simple units accepting a vector of real-valued inputs and producing a single real-valued output. The most common standard neural network type are feed-forward neural networks. Here sets of neurons are organized in layers: one input layer, one output layer, and at least one intermediate hidden layer. Feed-forward neural networks are limited to static classification tasks. Therefore, they are limited to provide a static mapping between input and output. To model time prediction tasks we need a so-called dynamic classifier.

The proposed system provides a precise explanation of the problem and discusses four distinct types of characteristics. This work has the potential to contribute to the advancement of research on cryptocurrencies.

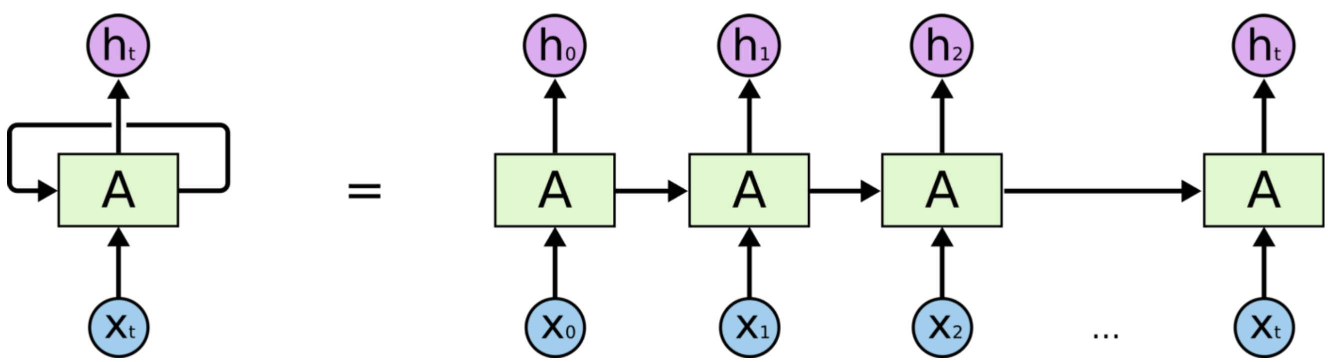


Fig 4.5 RNN Gates

In Figure something explains the RNN has shortcomings, the shortcomings can be seen in the input X_0, X_1 has a very large range of information X_t,, X_{t+1} so that when $_{1+1}$ requires information those that are relevant X_0, X_1 to RNN are unable to learn to link information because of old memory saved will be increasingly useless as time goes by because it is overwritten or replaced with new memory, this problem was discovered by Bengio, et al.(1994) . Unlike the RNN, LSTM does not have the disadvantage that LSTM can manage the memory at each input by using memory cells and gate units.

4.4 Importing Libraries

A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. A module allows you to log ically org anize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you canbind and reference. Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code.It makes Python Programming simpler and convenient for the programmer.Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc. Python libraries make it very easy for us to handle the data and perform typical and complex tasks with a single line of code.

A Python statement can access variables in local namespace and in the global namespace. If a local and a global variable have the same name, the local variable shadows the global variable. Each Function Has its own local namespace. Class methods follow the same scoping rule as ordinary functions. Python: Make educated guesses on whether variables are local or global. It assumes that any variable assigned a value ina function is local. Therefore, in order to assign value to a global variable within a function, you must first use the global statement. The statement global VarName tells PythonthatVarName is a global variable. Pythonstops searching the local namespace for the variable. For example, we define a variable Money in the g global namespace. Within The functionMoney, we assignMoney a value, therefore Python Assumes Money as a local variable. However, we assessed the value of the local variable Money before setting it, so anUnboundLocalError is the result. Uncommenting the global statement fixes the problem.

- **Pandas** – This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.
- **Numpy** – Numpy arrays can perform large computations in a concise time.
- **Matplotlib**– This library is used to draw visualizations.
- **Sklearn** – This module contains multiple libraries having pre-implemented functions to perform tasks from data preprocessing to model development.

```
import os
import pandas as pd
import numpy as np
import math
import datetime as dt

# For Evalution we will use these library

from sklearn.metrics import mean_squared_error, mean_absolute_error, explained_variance_score, r2_score
from sklearn.metrics import mean_poisson_deviance, mean_gamma_deviance, accuracy_score
from sklearn.preprocessing import MinMaxScaler

# For model building we will use these library

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import LSTM


# For PLotting we will use these library

import matplotlib.pyplot as plt
from itertools import cycle
import plotly.graph_objects as go
import plotly.express as px
```

Fig 4.6 Importing Libraries

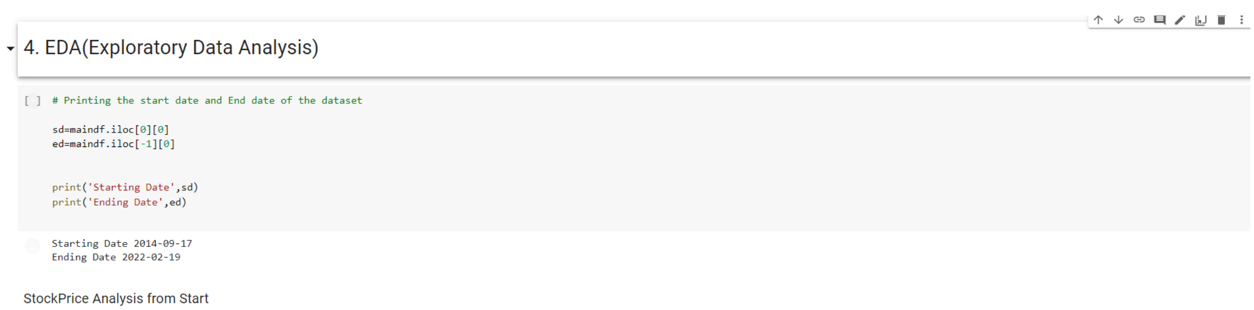
The data that we are going to use for this article can be downloaded from Yahoo Finance. For training our algorithm, we will be using the Apple stock prices from 1st January 2013 to 31 December 2021. This is where the power of LSTM can be utilized. LSTM (Long Short-Term Memory network) is a type of recurrent neural network capable of remembering the past information and while predicting the future values, it takes this past information into account.

4.5 Exploratory Data Analysis

In statistics, exploratory data analysis (EDA) is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling and thereby contrasts traditional hypothesis testing.

EDA is applied to investigate the data and summarize the key insights.It will give you the basic understanding of your data, it's distribution, null values and much more.You can either explore data using graphs or through some python functions.

It focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values, and making transformations of variables as needed. EDA encompasses IDA.


4. EDA(Exploratory Data Analysis)

```
# Printing the start date and End date of the dataset

sd=maindf.iloc[0][0]
ed=maindf.iloc[-1][0]


print('Starting Date',sd)
print('Ending Date',ed)
```

```
Starting Date 2014-09-17
Ending Date 2022-02-19
```

StockPrice Analysis from Start

Fig 4.7 Exploratory Data Analysis

The next step is training, followed by testing the dataset. We train our model, using the algorithm and the features taken into account to assist our model, to predict the future price of the crypto currency. Moving on to the testing part, we test the data to measure the accuracy of the algorithm that our model is using to predict the price of the Bitcoin.Finally after the processes of training with the help of the data set features and testing, we evaluate the accuracy of our model. We compare the predicted price of the crypto currency, at a given time period with the real world Bitcoin price at that particular period of time, and evaluate the accuracy and efficiency of our model.

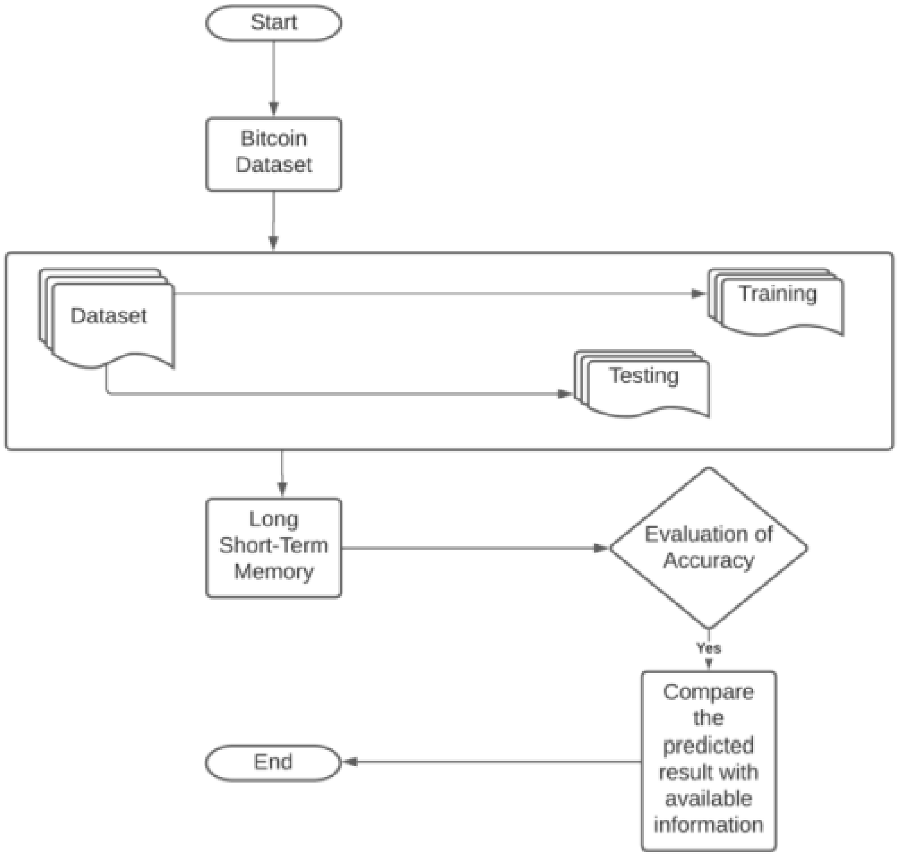


Fig 4.8 LSTM Flow

For instance, comparisons of a feed forward artificial neural network and the RNN's Long Short-Term Memory (LSTM) and Gated Recurrent Network (GRU) was conducted by Apaydan for streamflow prediction. The research showed that the two recurrent models were superior in getting the peak discharges and low discharges.The data conversion module, aiming at converting time series data to supervised learning sequences for later use, and it solves the multivariate problem with high dimensions by utilizing periodicity of the data and transforming rows of data to one row. This module includes two operations, data preprocessing and data converting. The second data converting operation is aimed at merging several rows to one, changing time series data into supervised learning sequences according the periodicity of the data.

4.6 SOFTWARE USED

4.5.1 PYTHON:

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. The language of choice for this project was Python. This was a straightforward call for many reasons.

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

However, Python is not without its errors. Python is a dynamically written language and packages are an area unit infamous for Duck writing. This may be frustrating once a package technique returns one thing that, for instance, looks like an array instead of being an actual array. Plus the standard Python documentation did not clearly state the return type of a method, this can’t lead without a lot of trials and error testing otherwise happening in a powerfully written language.

4.5.2 NUMPY :

It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. Numpy is a python package which provides scientific and higher level mathematical abstractions wrapped in python. It is the core library for scientific computing that contains tools for integrating C, strong n-dimensional array objects, C++ etc. It is also useful in random number capability, linear algebra etc. Numpy's array type augments the Python language with an efficient data structure used for numerical work, e.g., manipulating matrices.This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences.NumPy uses a multidimensional array object, and has functions and tools for working with these arrays.

Itis a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding The powerful n-dimensional array in NumPy speeds-up data processing. NumPy can be easily interfaced with other Python packages and provides tools for integrating with other programming languages like C, C++ etc.

4.5.3 GOOGLE COLAB

Google Colaboratory allows you to execute Python in a browser without configuring Python in your local system. The Python code is run from a notebook environment similar to Jupyter notebooks with execution and text cells. The Google colab is an open-source web application that enables making and sharing documents that contain visualizations, narrative text, live code and equations. Uses include: data , data visualization, data transformation, statistical modeling, machine learning, numerical simulation, data cleaning and much more.

It is a cloud service based on Jupyter Notebooks for disseminating machine learning education and research. It provides a runtime fully configured for deep learning and free-of-charge access to a robust GPU. This work presents a detailed analysis of Colaboratory regarding hardware resources, performance, and limitations. This analysis is performed through the use of Colaboratory for accelerating deep learning for computer vision and other GPU-centric applications. After you have created a new notebook, you will see an empty code cell. Python code can be entered into these code cells and executed at any time by either clicking the Play button to the left of the code cell or by pressing The chosen test-cases are a parallel tree-based combinatorial search and two computer vision applications: object detection/classification and object localization/segmentation. The hardware under the accelerated runtime is compared to a mainstream workstation and a robust Linux server equipped with 20 physical cores. Results show that the performance reached using this cloud service is equivalent to the performance of the dedicated testbeds, given similar resources. Thus, this service can be effectively exploited to accelerate not only deep learning but also other classes of GPU-centric applications. Clicking on the folder icon will give you the visualization of the file structure There should be a jpg file, if you do not see it, click the refresh button The file is temporarily stored, and will be removed once you end your session.Finally, several strengths and limitations of this cloud service are discussed, which might be useful for helping potential users.

With these benefits come serious potential risks. By connecting to a local runtime, you are allowing the Colaboratory frontend to execute code in the notebook using the local resources on your machine.

CHAPTER - 5

RESULT ANALYSIS

5.1 Experimental Results:

Firstly, we collect the data set from the online source: Yahoo Finance. The data set represents the Bitcoin price in United States Dollars (USD). The dataset includes all the information about bitcoin prices from 27th October, 2015 to 30th October, 2021.However, it needs to investigate further to enhance the accuracy of the deep learning-based prediction models by considering different parameters in addition to the previous one. Features such as political system, public relations, and market policy of a country can affect and determine the price volatility of cryptocurrency

2. Importing Library

```
# First we will import the necessary Library

import os
import pandas as pd
import numpy as np
import math
import datetime as dt

# For Evalution we will use these library

from sklearn.metrics import mean_squared_error, mean_absolute_error, explained_variance_score, r2_score
from sklearn.metrics import mean_poisson_deviance, mean_gamma_deviance, accuracy_score
from sklearn.preprocessing import MinMaxScaler

# For model building we will use these library

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import LSTM


# For PLotting we will use these library

import matplotlib.pyplot as plt
from itertools import cycle
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
```

Fig 5.1 : Importing Library

3. Loading Dataset

We can use this link to download bitcoin dataset from yahoo finance

```
# Load our dataset
# Note it should be in same dir


maindf=pd.read_csv('BTC-USD.csv')
```

```
print('Total number of days present in the dataset: ',maindf.shape[0])
print('Total number of fields present in the dataset: ',maindf.shape[1])
```

```
Total number of days present in the dataset:  2713
Total number of fields present in the dataset:  7
```

```
maindf.shape
```

```
(2713, 7)
```

```
maindf.head()
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100

Fig 5.2 : Loading Dataset

Figure below shows the pre-processing result to load the dataset into the machine and algorithm, and then shown the last day's close price data of bitcoin before we train and test and predict the results.

The second step involves filtering and cleaning the data set. This involves removing all the incomplete data from the rows i.e Null Values. It also involves filtering out unnecessary features present in the data collected. For our model, we will only use the columns labeled: Date, Price, Open, High, and Low.

```
maindf.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2713 entries, 0 to 2712
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       2713 non-null   object
 1   Open       2713 non-null   float64
 2   High       2713 non-null   float64
 3   Low        2713 non-null   float64
 4   Close      2713 non-null   float64
 5   Adj Close  2713 non-null   float64
 6   Volume     2713 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 148.5+ KB
```

```
[ ] maindf.describe()
```

	Open	High	Low	Close	Adj Close	Volume
count	2713.000000	2713.000000	2713.000000	2713.000000	2713.000000	2.713000e+03
mean	11311.041069	11614.292482	10975.555057	11323.914637	11323.914637	1.470462e+10
std	16106.428891	16537.390649	15608.572560	16110.365010	16110.365010	2.001627e+10
min	176.897003	211.731003	171.509995	178.102997	178.102997	5.914570e+06
25%	606.396973	609.260986	604.109985	606.718994	606.718994	7.991080e+07
50%	6301.569824	6434.617676	6214.220215	6317.609863	6317.609863	5.098183e+09
75%	10452.399414	10762.644531	10202.387695	10462.259766	10462.259766	2.456992e+10
max	67549.734375	68789.625000	66382.062500	67566.828125	67566.828125	3.509679e+11

Fig 5.3 : Fixing Range for data

The next step is training, followed by testing the dataset. We train our model, using the algorithm and the features taken into account to assist our model, to predict the future price of the crypto currency. Moving on to the testing part, we test the data to measure the accuracy of the algorithm that our model is using to predict the price of the Bitcoin.

Finally after the processes of training with the help of the data set features and testing, we evaluate the accuracy of our model. We compare the predicted price of the crypto currency, at a given time period with the real world Bitcoin price at that particular period of time, and evaluate the accuracy and efficiency of our model.

Checking for Null Values

```
print('Null Values:',maindf.isnull().values.sum())
```

```
Null Values: 0
```

```
print('NA values:',maindf.isnull().values.any())
```

```
NA values: False
```

```
# If dataset had null values we can use this code to drop all the null values present in the dataset

# maindf=maindf.dropna()
# print('Null Values:',maindf.isnull().values.sum())
# print('NA values:',maindf.isnull().values.any())
```

```
# Final shape of the dataset after dealing with null values

maindf.shape
```

```
(2713, 7)
```

Fig 5.4 : Checking for Null values

4. EDA(Exploratory Data Analysis)

```
# Printing the start date and End date of the dataset

sd=maindf.iloc[0][0]
ed=maindf.iloc[-1][0]


print('Starting Date',sd)
print('Ending Date',ed)
```

```
Starting Date 2014-09-17
Ending Date 2022-02-19
```

StockPrice Analysis from Start

Fig 5.5 : Exploratory Data Analysis

EDA is applied to investigate the data and summarize the key insights.It will give you the basic understanding of your data, it's distribution, null values and much more.You can either explore data using graphs or through some python functions.

Analysis of year 2014

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

y_2014 = maindf.loc[(maindf['Date'] >= '2014-09-17')
                    & (maindf['Date'] < '2014-12-31')]

y_2014.drop(y_2014[['Adj Close','Volume']],axis=1)
```

	Date	Open	High	Low	Close
0	2014-09-17	465.864014	468.174011	452.421997	457.334015
1	2014-09-18	456.859985	456.859985	413.104004	424.440002
2	2014-09-19	424.102997	427.834991	384.532013	394.795990
3	2014-09-20	394.673004	423.295990	389.882996	408.903992
4	2014-09-21	408.084991	412.425995	393.181000	398.821014
...	...	...	...	...	...
100	2014-12-26	319.152008	331.424011	316.627014	327.924011
101	2014-12-27	327.583008	328.911011	312.630005	315.863007
102	2014-12-28	316.160004	320.028015	311.078003	317.239014
103	2014-12-29	317.700989	320.266998	312.307007	312.670013
104	2014-12-30	312.718994	314.808990	309.372986	310.737000

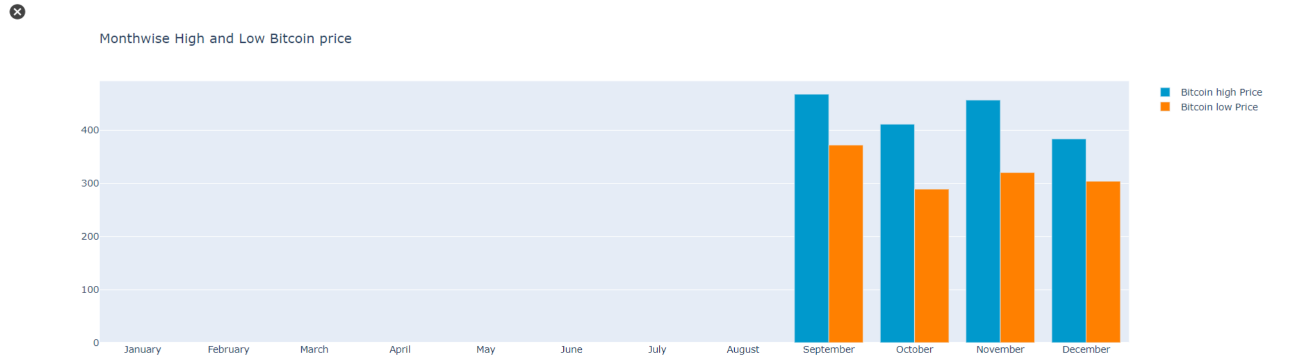
105 rows × 5 columns

```
monthvise= y_2014.groupby(y_2014['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise
```

Fig 5.6 : Analysis of year 2014

For Step Split and training the data, we divided the 4 years period of data for training and 1 year for testing. Split at 1462 data in. While splitting the data into train and validation, we cannot use random splitting since that will destroy the time component.

So we set the last year's data into a test and the 9 years' data before that into the dataset.



Note that we only have few months in 2014 so the rest of the months are not plotted since we do not have the data

Fig 5.7 : Year 2014 Monthwise bitcoin price

```
names = cycle(['Bitcoin Open Price','Bitcoin Close Price','Bitcoin High Price','Bitcoin Low Price'])

fig = px.line(y_2014, x=y_2014.Date, y=[y_2014['Open'], y_2014['Close'],
                                          y_2014['High'], y_2014['Low']],
             labels={'Date': 'Date','value':'Bitcoin value'})
fig.update_layout(title_text='Bitcoin analysis chart', font_size=15, font_color='black',legend_title_text='Bitcoin Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

Fig 5.8 : Bitcoin analysis chart 2014

Analysis of Year 2015

↳ 5 cells hidden

Analysis of Year 2016

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

y_2016 = maindf.loc[(maindf['Date'] >= '2016-01-01')
                     & (maindf['Date'] < '2017-01-01')]

y_2016.drop(y_2016[['Adj Close','Volume']],axis=1)
```

	Date	Open	High	Low	Close
471	2016-01-01	430.721008	436.246002	427.515015	434.334015
472	2016-01-02	434.622009	436.062012	431.869995	433.437988
473	2016-01-03	433.578003	433.743011	424.705994	430.010986
474	2016-01-04	430.061005	434.516998	429.084015	433.091003
475	2016-01-05	433.069000	434.182007	429.675995	431.959991
...	...	...	...	...	...
832	2016-12-27	908.354004	940.047974	904.255005	933.197998
833	2016-12-28	934.830994	975.921021	934.830994	975.921021
834	2016-12-29	975.125000	979.396973	954.502991	973.497009
835	2016-12-30	972.534973	972.534973	934.833008	961.237976
836	2016-12-31	960.627014	963.742981	947.236023	963.742981

366 rows × 5 columns

Fig 5.9 : Analysis of year 2015 and 2016

```
        marker_color='rgb(0, 153, 204)'
))
fig.add_trace(go.Bar(
        x=monthvise_low.index,
        y=monthvise_low,
        name='Stock low Price',
        marker_color='rgb(255, 128, 0)'
))

fig.update_layout(barmode='group',
                  title=' Monthwise High and Low stock price')
fig.show()
```

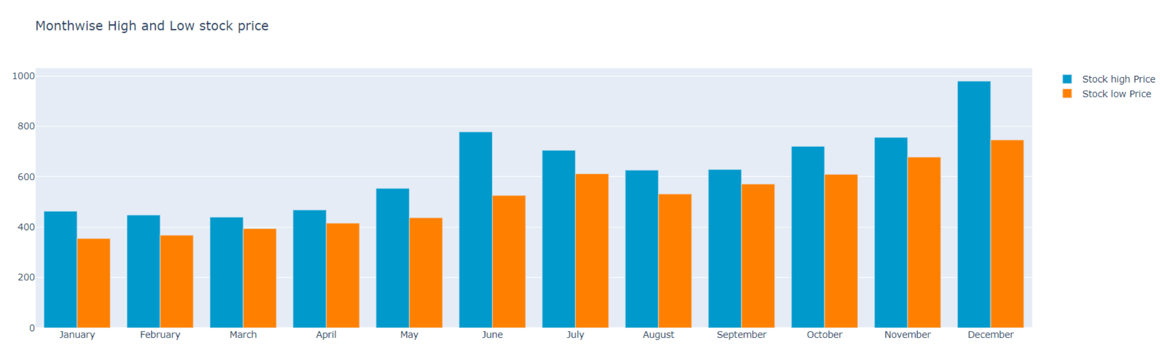


Fig 5.10 : Year 2015 and 2016 Monthwise bitcoin price

```python
fig = px.line(y_2016, x=y_2016.Date, y=[y_2016['Open'], y_2016['Close'],
                                         y_2016['High'], y_2016['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15, font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

Fig 5.11 : Bitcoin analysis chart 2015 and 2016

Analysis of Year 2017

```python
maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2017 = maindf.loc[(maindf['Date'] >= '2017-01-01')
                     & (maindf['Date'] < '2018-01-01')]

y_2017.drop(y_2017[['Adj Close','Volume']],axis=1)
```

	Date	Open	High	Low	Close
837	2017-01-01	963.658020	1003.080017	958.698975	998.325012
838	2017-01-02	998.617004	1031.390015	996.702026	1021.750000
839	2017-01-03	1021.599976	1044.079956	1021.599976	1043.839966
840	2017-01-04	1044.400024	1159.420044	1044.400024	1154.729980
841	2017-01-05	1156.729980	1191.099976	910.416992	1013.380005
...	...	...	...	...	...
1197	2017-12-27	16163.500000	16930.900391	15114.299805	15838.500000
1198	2017-12-28	15864.099609	15888.400391	13937.299805	14606.500000
1199	2017-12-29	14695.799805	15279.000000	14307.000000	14656.200195
1200	2017-12-30	14681.900391	14681.900391	12350.099609	12952.200195
1201	2017-12-31	12897.700195	14377.400391	12755.599609	14156.400391

365 rows × 5 columns

```python
monthvise= y_2017.groupby(y_2017['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise
```

Fig 5.12 : Bitcoin analysis of year 2017 and 2018

```python
names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low Price'])

fig = px.line(y_2017, x=y_2017.Date, y=[y_2017['Open'], y_2017['Close'],
                                          y_2017['High'], y_2017['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15, font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

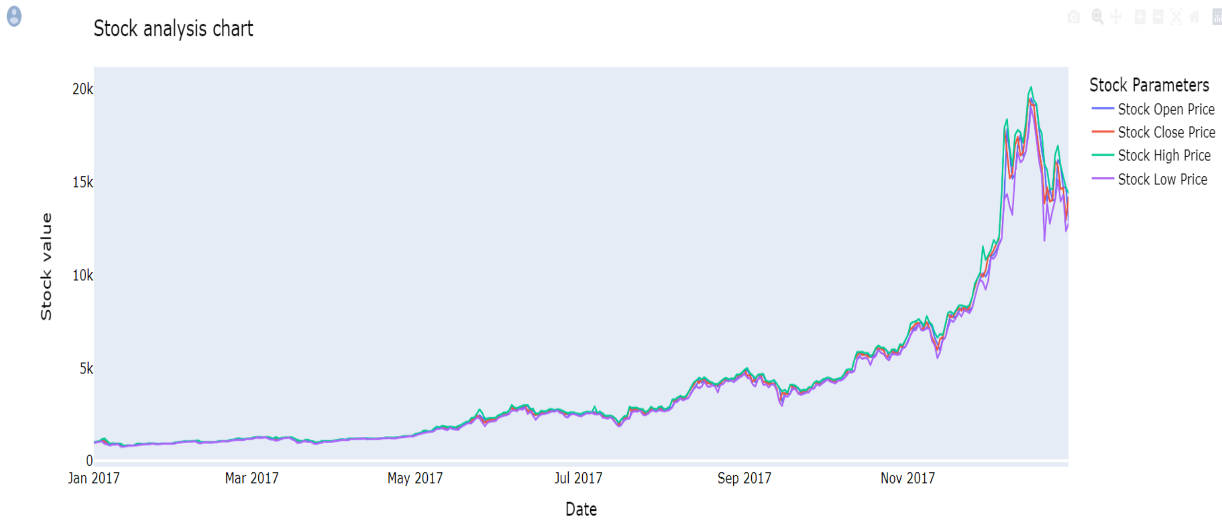


Fig 5.13 : Bitcoin analysis of year 2017

Analysis of Year 2018

```python
maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2018 = maindf.loc[(maindf['Date'] >= '2018-01-01')
                     & (maindf['Date'] < '2019-01-01')]

y_2018.drop(y_2018[['Adj Close','Volume']],axis=1)
```

	Date	Open	High	Low	Close
1202	2018-01-01	14112.200195	14112.200195	13154.700195	13657.200195
1203	2018-01-02	13625.000000	15444.599609	13163.599609	14982.099609
1204	2018-01-03	14978.200195	15572.799805	14844.500000	15201.000000
1205	2018-01-04	15270.700195	15739.700195	14522.200195	15599.200195
1206	2018-01-05	15477.200195	17705.199219	15202.799805	17429.500000
...	...	...	...	...	...
1562	2018-12-27	3854.688477	3874.416992	3645.448486	3654.833496
1563	2018-12-28	3653.131836	3956.135986	3642.632080	3923.918701
1564	2018-12-29	3932.491699	3963.758789	3820.408691	3820.408691
1565	2018-12-30	3822.384766	3901.908936	3797.219238	3865.952637
1566	2018-12-31	3866.839111	3868.742920	3725.867432	3742.700439

365 rows × 5 columns

```python
[ ] monthvise= y_2018.groupby(y_2018['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise
```

Fig 5.14 : Bitcoin analysis of year 2018

```
names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low Price'])

fig = px.line(y_2018, x=y_2018.Date, y=[y_2018['Open'], y_2018['Close'],
                                          y_2018['High'], y_2018['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15, font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

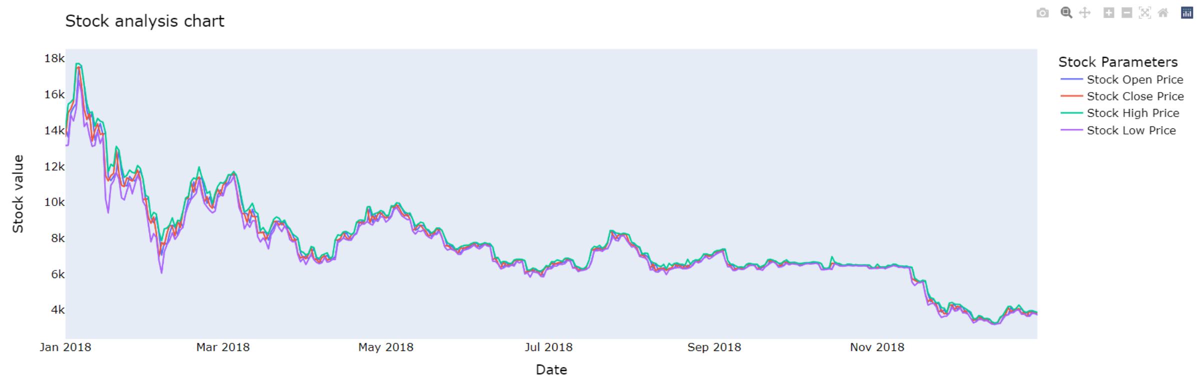


Fig 5.15 : Stock analysis of 2014 - 2018

Analysis of Year 2019

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

y_2019 = maindf.loc[(maindf['Date'] >= '2019-01-01')
                     & (maindf['Date'] < '2020-01-01')]

y_2019.drop(y_2019[['Adj Close','Volume']],axis=1)
```

	Date	Open	High	Low	Close
1567	2019-01-01	3746.713379	3850.913818	3707.231201	3843.520020
1568	2019-01-02	3849.216309	3947.981201	3817.409424	3943.409424
1569	2019-01-03	3931.048584	3935.685059	3826.222900	3836.741211
1570	2019-01-04	3832.040039	3865.934570	3783.853760	3857.717529
1571	2019-01-05	3851.973877	3904.903076	3836.900146	3845.194580
...	...	...	...	...	...
1927	2019-12-27	7238.141113	7363.529297	7189.934082	7290.088379
1928	2019-12-28	7289.031250	7399.041016	7286.905273	7317.990234
1929	2019-12-29	7317.647461	7513.948242	7279.865234	7422.652832
1930	2019-12-30	7420.272949	7454.824219	7276.308105	7292.995117
1931	2019-12-31	7294.438965	7335.290039	7169.777832	7193.599121

365 rows × 5 columns

```
[ ] monthvise= y_2019.groupby(y_2019['Date'].dt.strftime('%B'))[['Open','Close']].mean()
    new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
                 'September', 'October', 'November', 'December']
    monthvise = monthvise.reindex(new_order, axis=0)
    monthvise
```

Fig 5.16 : Analysis of year 2019

```
names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low Price'])

fig = px.line(y_2019, x=y_2019.Date, y=[y_2019['Open'], y_2019['Close'],
                                          y_2019['High'], y_2019['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15, font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

Fig 5.17 : Bitcoin analysis of 2019

Analysis of Year 2020

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

y_2020 = maindf.loc[(maindf['Date'] >= '2020-01-01')
                   & (maindf['Date'] < '2021-01-01')]

y_2020.drop(y_2020[['Adj Close','Volume']],axis=1)
```

	Date	Open	High	Low	Close
1932	2020-01-01	7194.892090	7254.330566	7174.944336	7200.174316
1933	2020-01-02	7202.551270	7212.155273	6935.270020	6985.470215
1934	2020-01-03	6984.428711	7413.715332	6914.996094	7344.884277
1935	2020-01-04	7345.375488	7427.385742	7309.514160	7410.656738
1936	2020-01-05	7410.451660	7544.497070	7400.535645	7411.317383
...	...	...	...	...	...
2293	2020-12-27	26439.373047	28288.839844	25922.769531	26272.294922
2294	2020-12-28	26280.822266	27389.111328	26207.640625	27084.808594
2295	2020-12-29	27081.810547	27370.720703	25987.298828	27362.437500
2296	2020-12-30	27360.089844	28937.740234	27360.089844	28840.953125
2297	2020-12-31	28841.574219	29244.876953	28201.992188	29001.720703

366 rows × 5 columns

```
[ ] monthvise= y_2020.groupby(y_2020['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
monthvise
```

Fig 5.18 : Analysis of year 2020

```python
names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low Price'])

fig = px.line(y_2020, x=y_2020.Date, y=[y_2020['Open'], y_2020['Close'],
                                        y_2020['High'], y_2020['Low']],
              labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15, font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

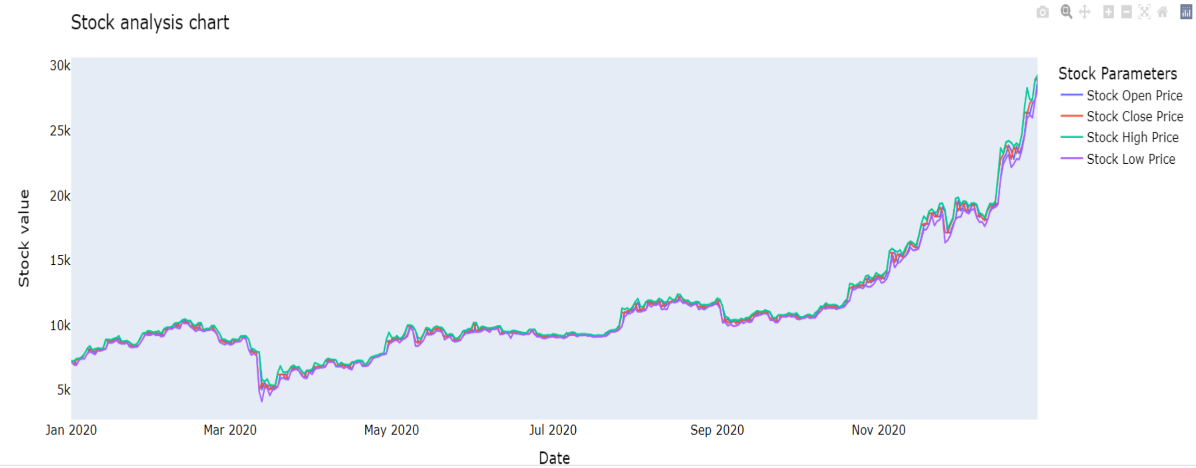


Fig 5.19 : Bitcoin analysis of 2020

Analysis of Year 2021

```python
maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_2021 = maindf.loc[(maindf['Date'] >= '2021-01-01')
                    & (maindf['Date'] < '2021-12-31')]

y_2021.drop(y_2021[['Adj Close','Volume']],axis=1)
```

	Date	Open	High	Low	Close
2298	2021-01-01	28994.009766	29600.626953	28803.585938	29374.152344
2299	2021-01-02	29376.455078	33155.117188	29091.181641	32127.267578
2300	2021-01-03	32129.408203	34608.558594	32052.316406	32782.023438
2301	2021-01-04	32810.949219	33440.218750	28722.755859	31971.914063
2302	2021-01-05	31977.041016	34437.589844	30221.187500	33992.429688
...	...	...	...	...	...
2657	2021-12-26	50428.691406	51196.378906	49623.105469	50809.515625
2658	2021-12-27	50802.609375	51956.328125	50499.468750	50640.417969
2659	2021-12-28	50679.859375	50679.859375	47414.210938	47588.855469
2660	2021-12-29	47623.871094	48119.742188	46201.496094	46444.710938
2661	2021-12-30	46490.605469	47879.964844	46060.312500	47178.125000

364 rows × 5 columns

```python
[ ] monthvise= y_2021.groupby(y_2021['Date'].dt.strftime('%B'))[['Open','Close']].mean()
    new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
                 'September', 'October', 'November', 'December']
    monthvise = monthvise.reindex(new_order, axis=0)
    monthvise
```

Fig 5.20 : Analysis of year 2021

```python
names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low Price'])

fig = px.line(y_2021, x=y_2021.Date, y=[y_2021['Open'], y_2021['Close'],
                                          y_2021['High'], y_2021['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15, font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

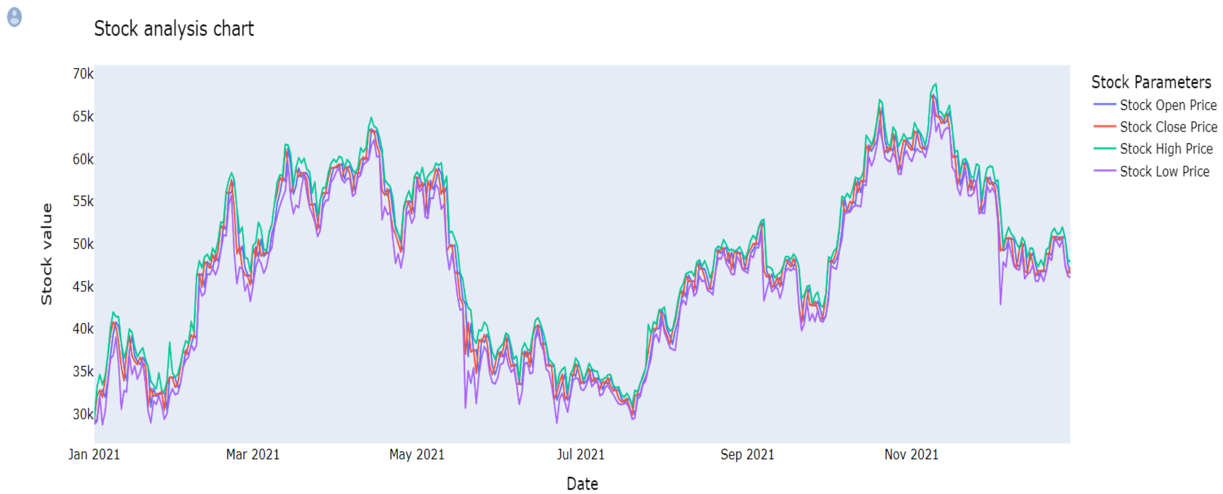


Fig 5.21 : Bitcoin analysis of 2021

```python
    marker_color='crimson'
))
fig.add_trace(go.Bar(
    x=monthvise.index,
    y=monthvise['Close'],
    name='Stock Close Price',
    marker_color='lightsalmon'
))

fig.update_layout(barmode='group', xaxis_tickangle=-45,
                  title='Monthwise comparision between Stock open and close price')
fig.show()
```

Fig 5.22 : Month wise comparison of 2021

```python
names = cycle(['Stock Open Price','Stock Close Price','Stock High Price','Stock Low Price'])

fig = px.line(y_2022, x=y_2022.Date, y=[y_2022['Open'], y_2022['Close'],
                                        y_2022['High'], y_2022['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15, font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

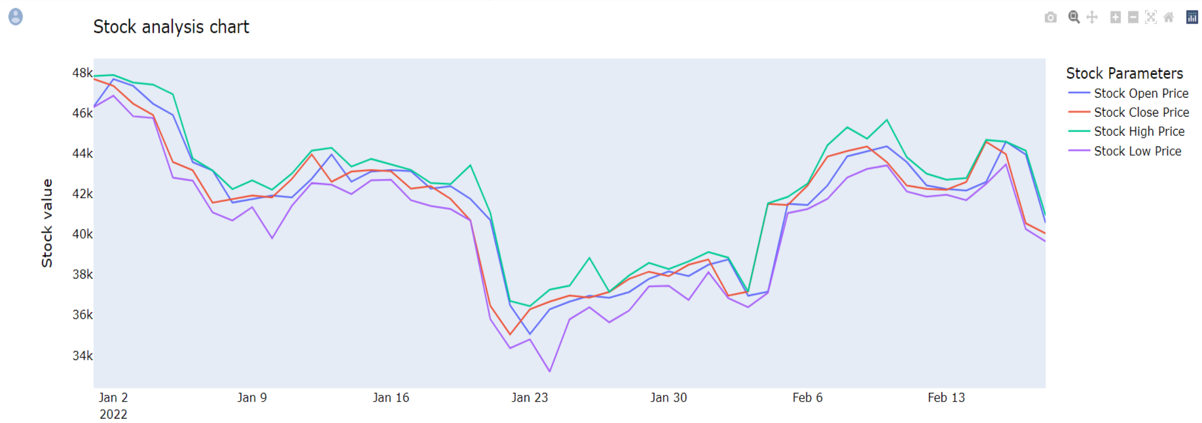


Fig 5.23 :Stock analysis chart of 2019 - 2021

Overall Analysis from 2014-2022

```python
maindf['Date'] = pd.to_datetime(maindf['Date'], format='%Y-%m-%d')

y_overall = maindf.loc[(maindf['Date'] >= '2014-09-17')
                     & (maindf['Date'] <= '2022-02-19')]

y_overall.drop(y_overall[['Adj Close','Volume']],axis=1)
```

	Date	Open	High	Low	Close
0	2014-09-17	465.864014	468.174011	452.421997	457.334015
1	2014-09-18	456.859985	456.859985	413.104004	424.440002
2	2014-09-19	424.102997	427.834991	384.532013	394.795990
3	2014-09-20	394.673004	423.295990	389.882996	408.903992
4	2014-09-21	408.084991	412.425995	393.181000	398.821014
...	...	...	...	...	...
2708	2022-02-15	42586.464844	44667.218750	42491.035156	44575.203125
2709	2022-02-16	44578.277344	44578.277344	43456.691406	43961.859375
2710	2022-02-17	43937.070313	44132.972656	40249.371094	40538.011719
2711	2022-02-18	40552.132813	40929.152344	39637.617188	40030.976563
2712	2022-02-19	40022.132813	40246.027344	40010.867188	40126.429688

2713 rows × 5 columns

```python
monthvise= y_overall.groupby(y_overall['Date'].dt.strftime('%B'))[['Open','Close']].mean()
new_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August',
             'September', 'October', 'November', 'December']
monthvise = monthvise.reindex(new_order, axis=0)
```

Fig 5.24 : Overall Analysis of 2014 - 2022

```python
fig = px.line(y_overall, x=y_overall.Date, y=[y_overall['Open'], y_overall['Close'],
                                              y_overall['High'], y_overall['Low']],
             labels={'Date': 'Date','value':'Stock value'})
fig.update_layout(title_text='Stock analysis chart', font_size=15, font_color='black',legend_title_text='Stock Parameters')
fig.for_each_trace(lambda t:  t.update(name = next(names)))
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)

fig.show()
```

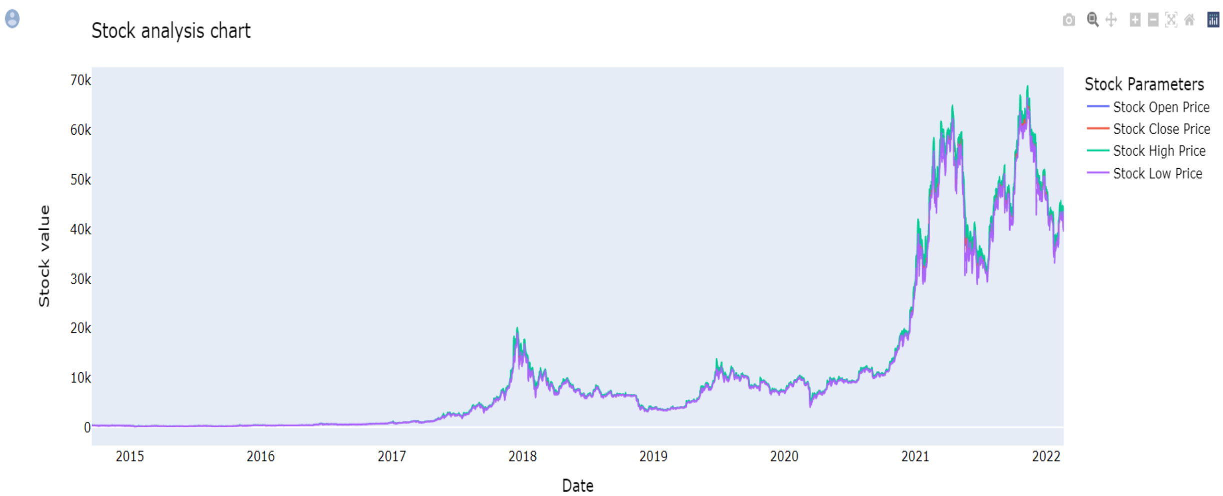


Fig 5.25 : Stock Analysis of overall prices

5. Building LSTM Model

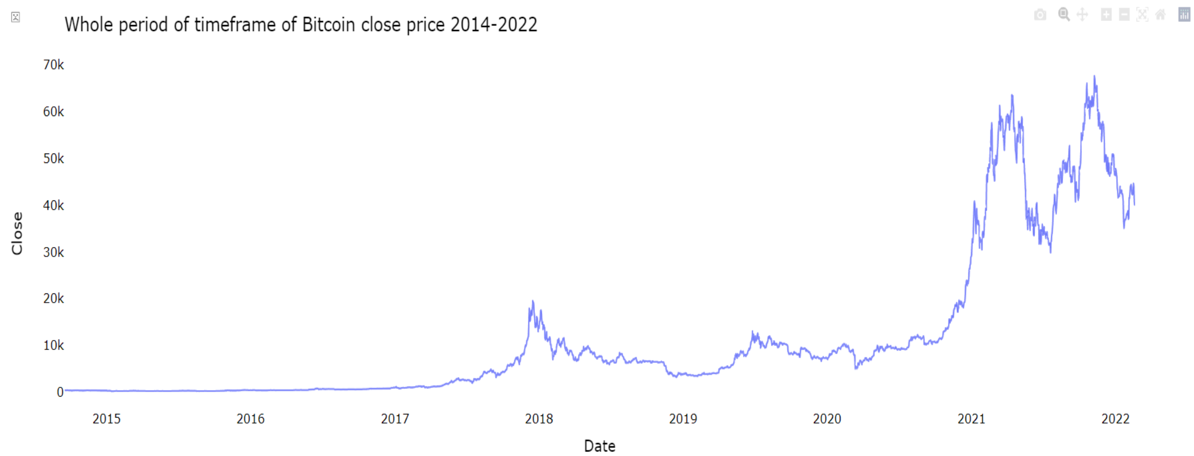
- First Step is Preparing Data for Training and Testing
- Here we are just considering 1 year data for training data
- Since Bitcoin price has drastically flucated from 200 dollar in year 2014 to 15000 dollar in year 2018 to 3000 dollar in year 2019(theses values are apporx) so we will just consider 1 Year to avoid this type of flucation in the data.
- As we want to predict Close Price of the Bitcoin so we are just Considering Close and Date

```python
# Lets First Take all the Close Price
closedf = maindf[['Date','Close']]
print("Shape of close dataframe:", closedf.shape)
```

```
Shape of close dataframe: (2713, 2)
```

```python
fig = px.line(closedf, x=closedf.Date, y=closedf.Close,labels={'date':'Date','close':'Close Stock'})
fig.update_traces(marker_line_width=2, opacity=0.8, marker_line_color='orange')
fig.update_layout(title_text='Whole period of timeframe of Bitcoin close price 2014-2022', plot_bgcolor='white',
                  font_size=15, font_color='black')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

Fig 5.26 : LSTM Model



Now we will Take data of just 1 Year

```
closedf = closedf[closedf['Date'] > '2021-02-19']
close_stock = closedf.copy()
print("Total data for prediction: ",closedf.shape[0])
```

```
Total data for prediction:  365
```

Fig 5.27 : Timeframe of bitcoin price

```
fig = px.line(closedf, x=closedf.Date, y=closedf.Close,labels={'date':'Date','close':'Close Stock'})
fig.update_traces(marker_line_width=2, opacity=0.8, marker_line_color='orange')
fig.update_layout(title_text='Considered period to predict Bitcoin close price',
                  plot_bgcolor='white', font_size=15, font_color='black')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
```

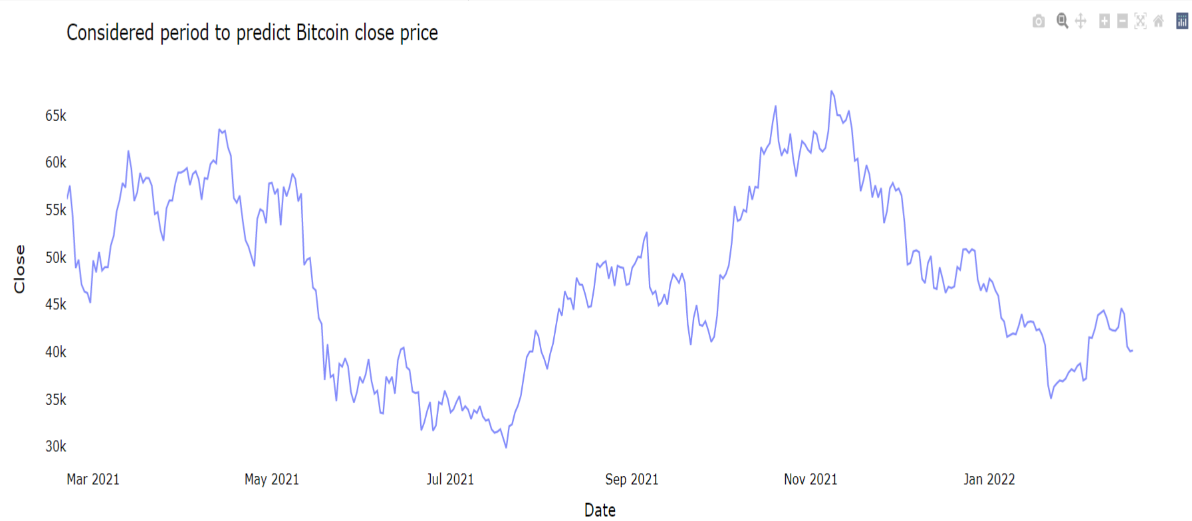


Fig 5.28 : Considered period to Predict bitcoin

Model Evaluation

```
# Transform back to original form

train_predict = scaler.inverse_transform(train_predict)
test_predict = scaler.inverse_transform(test_predict)
original_ytrain = scaler.inverse_transform(y_train.reshape(-1,1))
original_ytest = scaler.inverse_transform(y_test.reshape(-1,1))
```

Evaluation metrices RMSE, MSE and MAE

[ ] 1 cell hidden

Variance Regression Score

```
print("Train data explained variance regression score:",
      explained_variance_score(original_ytrain, train_predict))
print("Test data explained variance regression score:",
      explained_variance_score(original_ytest, test_predict))
```

```
Train data explained variance regression score: 0.9138865055226862
Test data explained variance regression score: 0.9361372329698047
```

R square score for regression

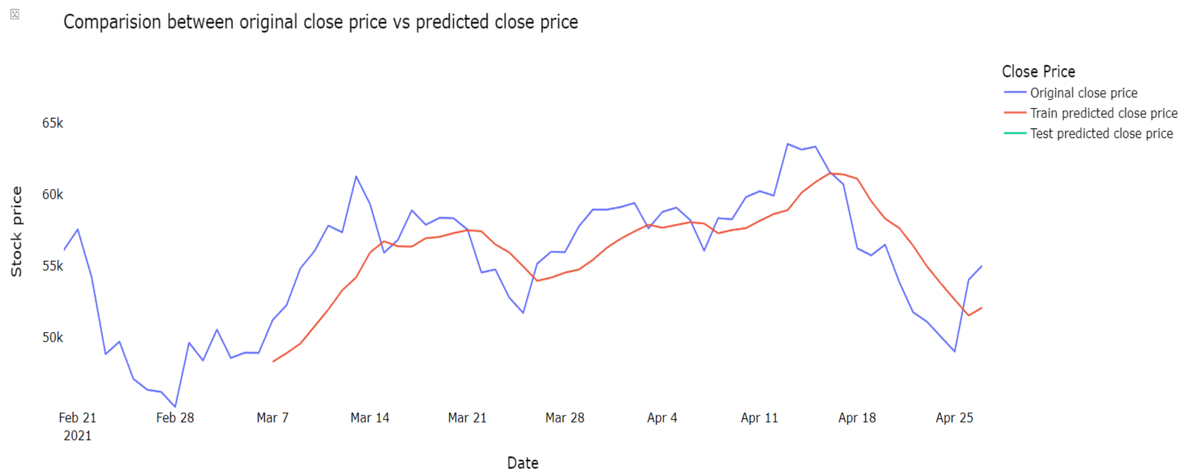
```
print("Train data R2 score:", r2_score(original_ytrain, train_predict))
print("Test data R2 score:", r2_score(original_ytest, test_predict))
```

```
Train data R2 score: 0.9138763695789057
Test data R2 score: 0.9361329036067648
```

Fig 5.29 : Model Evaluation

```
fig.show()
```

```
Train predicted data:  (365, 1)
Test predicted data:  (365, 1)
```



Predicting next 30 days

Fig 5.30 : Comparison between original and predicted price

Long short-term memory is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points, but also entire sequences of data.

5.2 Final Output :

The Figure below shows pre-processing results to load the dataset into machine and algorithm, and then the last day close price data of bitcoin before we train and test and predict the results. For Step Split and training the data, we divided the 4 years period of data for training and 1 year for testing. Split at 1462 data in. While splitting the data into train and validation, we cannot use random splitting since that will destroy the time component. So we set the last year's data into a test and the 9 years' data before that into the training dataset.

Based on figures 6 that result from prediction using LSTM shown by the graph with epoch 500, model dropout 0, and, Red Line is a result for the close prediction, the Blue and Green line is from Data Training. The price result is above $12600 for next days based on the model.

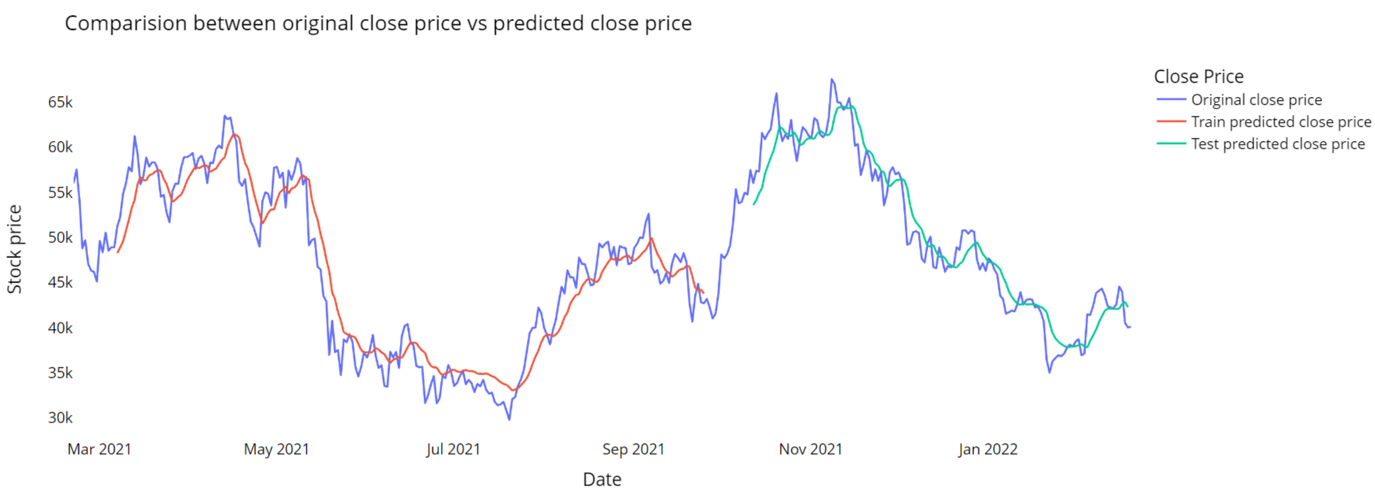


Fig 5.31 : Prediction Graph for next 30 days

CHAPTER - 6

EXPECTED OUTCOME PROOF

Our Proposed system was submitted at the 2023 Ieee Third International Conference On Technology, Engineering.

2023 IEEE THIRD INTERNATIONAL CONFERENCE ON TECHNOLOGY, ENGINEERING, MANAGEMENT FOR SOCIETAL IMAPCT USING MARKETING, ENTREPRENEURSHIP AND TALENT : Submission (168) has been created. External Inbox ×

Microsoft CMT <email@msr-cmt.org> 1:00 PM (0 minutes ago)
to me

Hello,

The following submission has been created.

Track Name: TRACK 1: Applications of Technology and Engineering

Paper ID: 168

Paper Title: CryptoCurrency Price prediction

Abstract:
Cryptocurrencies are a digital way of money in which all transactions are held electronically. It is a soft currency which doesn't exist in the form of hard notes physically. Here, we are emphasizing the difference of fiat currency which is decentralized in that without any third-party intervention all virtual currency users can get the services. To the best of our knowledge, our target is to implement the efficient deep learning-based prediction models, specifically long short-term memory (LSTM) and gated recurrent unit (GRU) to handle the price volatility of bitcoin and to obtain high accuracy. Our study involves comparing these two-time series deep learning techniques and proved the efficacy in forecasting the price of bitcoin.

Created on: Sat, 26 Nov 2022 07:30:19 GMT

Last Modified: Sat, 26 Nov 2022 07:30:19 GMT

Authors:
- 1912021@nec.edu.in (Primary)
- 1912014@nec.edu.in
- 1912033@nec.edu.in
- hema@nec.edu.in

Secondary Subject Areas: Not Entered
Submission Files: CRYPTO - PAPER.pdf (2 Mb, Sat, 26 Nov 2022 07:30:05 GMT)

Fig 6.1 IEEE submission proof

CHAPTER - 7

CONCLUSION

7.1 CONCLUSION AND FUTURE WORKS

LSTMs are a complex area of deep learning.LSTMs are often referred to as fancy RNNs. Vanilla RNNs do not have a cell state. They only have hidden states and those hidden states serve as the memory for RNNs. Meanwhile, LSTM has both cell states and hidden states.Our Proposed model has been succeeded to provide the result prediction bitcoin from yahoo finance stock market.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. Our model with time series techniques can produce the results and the results can predict the price for the next days with split the data to train and test that we mention in the article above.

Based on our experimental results and investigation regarding to our research questions about cryptocurrency price problem, we conclude that cryptocurrency prices follow almost a random walk process while few hidden patterns may probably exist in, where an intelligent framework has to identify them in order for a prediction model to make accurate and reliable forecasts. Therefore, new sophisticated algorithmic methods, alternative approaches, new validation metrics should be explored.

References:

[1] Sina E. Charandabi,Kamyar Kamyar(2021) -Prediction of Crypto currency Price Index Using Artificial Neural Networks, Proposed hybrid artificial neural network models to predict prices of cryptocurrency.

[2] Ahmed M. Khedr,Ifra Arif,Pravija Raj P V(2021) – Traditional statistical methods require a lot of statistical assumptions that could be unrealistic, leaving machine learning as the best technology in this field.

[3] Edwin, Lipo Wang (2017) - DescribedThe features of Bitcoin and the next day change in the price of Bitcoin using an Artificial Neural Network ensemble.

[4] Yeray Mezquita, Ana Belén Gil-González, Javier Prieto, Juan Manuel Corchado (2021)-described that the solutions proposed are for the detection of patterns and anomalies in cryptocurrency ecosystems.

[5] Lloyd Kasal, Mihir Shetty , Tanmay Nayak , Ramanath Pai , Shilpa B(2022)-proposed to analyze in cryptocurrency values, numerous different kinds of neural networks may be utilized.

[6] Ahmed M. Khedr Ifra Arif Pravija Raj P V Magdi El-Bannany (2021)- proposed that traditional statistical methods, although simple to implement and interpret, require a lot of statistical assumptions

[7] Caporale, Guglielmo Maria Plastun, Alex (2018),-Cryptocurrency market Bitcoin overreaction momentum abnormal returns contrarian strategy trading strategy trading robot.

[8] Al-Yahyaee KH, Mensi W(2020) -Multifractality, long-memory process, and efficiency hypothesis of six major cryptocurrencies using the time-rolling MF-DFA approach.

[9] Magdi El-Bannany,Saadat M. Alhashmi, Meenu Sreedharan(2016) -Cryptocurrency price prediction using traditional statistical and machine-learning techniques.

[10] Ujan ;Skjellum;Brooks(2022), proposed that Cryptocurrencies require strong, secure mining algorithms. In this paper we survey and compare and contrast current mining techniques as used by major Cryptocurrencies.

[11] Carmine Ventre,Michail Basios,Leslie Kanthan,David Martinez-Rego(2022), This survey analyzes the research distribution that characterize cryptocurrency Research.

[12] Sina E. Charandabi;Kamyar Kamyar (2021),The purpose of this survey paper is to present and compare multiple research papers that employed multiple neural networks.

[13] Saeed Alzahrani; Tugrul U. Daim(2019) -Paper suggests that the main factors driving the adoption decision, the acceptance by businesses as a payment method, the fast transfer of funds.

[14] Pravija Raj, Sreedharan(2021) –Research work in this field uses traditional statistical and machine-learning techniques.

[15] George S.Atsalakis(2021) –Fuzzy modeling demonstrating that the closed-loop or feedback control technique can cope with uncertainties associated with the dynamic behavior of the price.