# CRYPTOCURRENCY PRICE PREDICTION USING MACHINE LEARNING AND DEEP LEARNING MODELS

**PROJECT REPORT**

***Submitted by***

**SWETHA P**
**Register No: 20MCA042**

*in the partial fulfillment for the award of the degree*

*of*

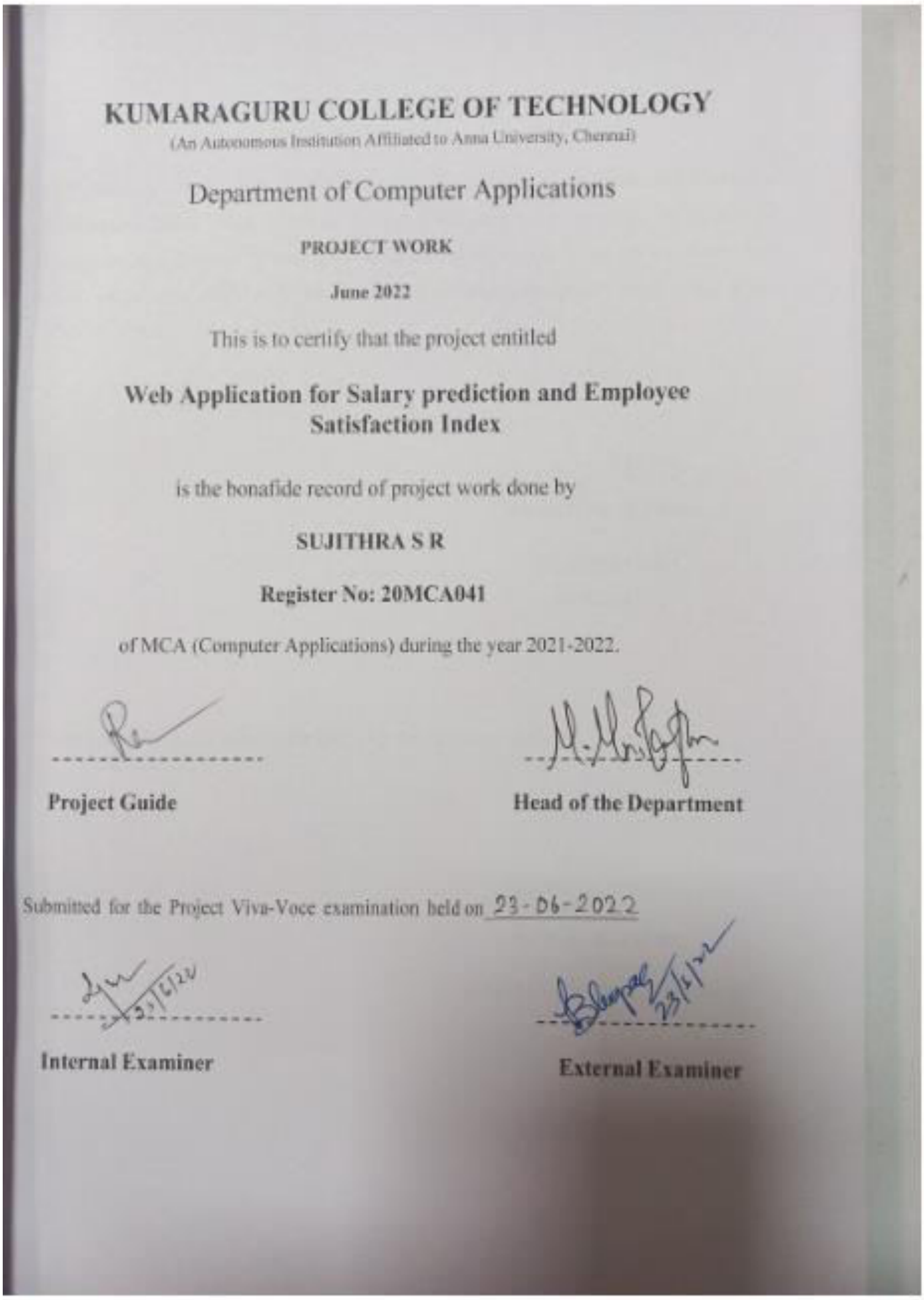
**MASTER OF COMPUTER APPLICATIONS**

*in*

**COMPUTER APPLICATIONS**

**KUMARAGURU COLLEGE OF TECHNOLOGY**
(An Autonomous Institution Affiliated to Anna University, Chennai)
**COIMBATORE - 641 049**

**JUNE 2022**



# KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

## Department of Computer Applications

**PROJECT WORK**

**June 2022**

This is to certify that the project entitled

**Web Application for Salary prediction and Employee Satisfaction Index**

is the bonafide record of project work done by

**SUJITHRA S R**

**Register No: 20MCA041**

of MCA (Computer Applications) during the year 2021-2022.

**Project Guide**

**Head of the Department**

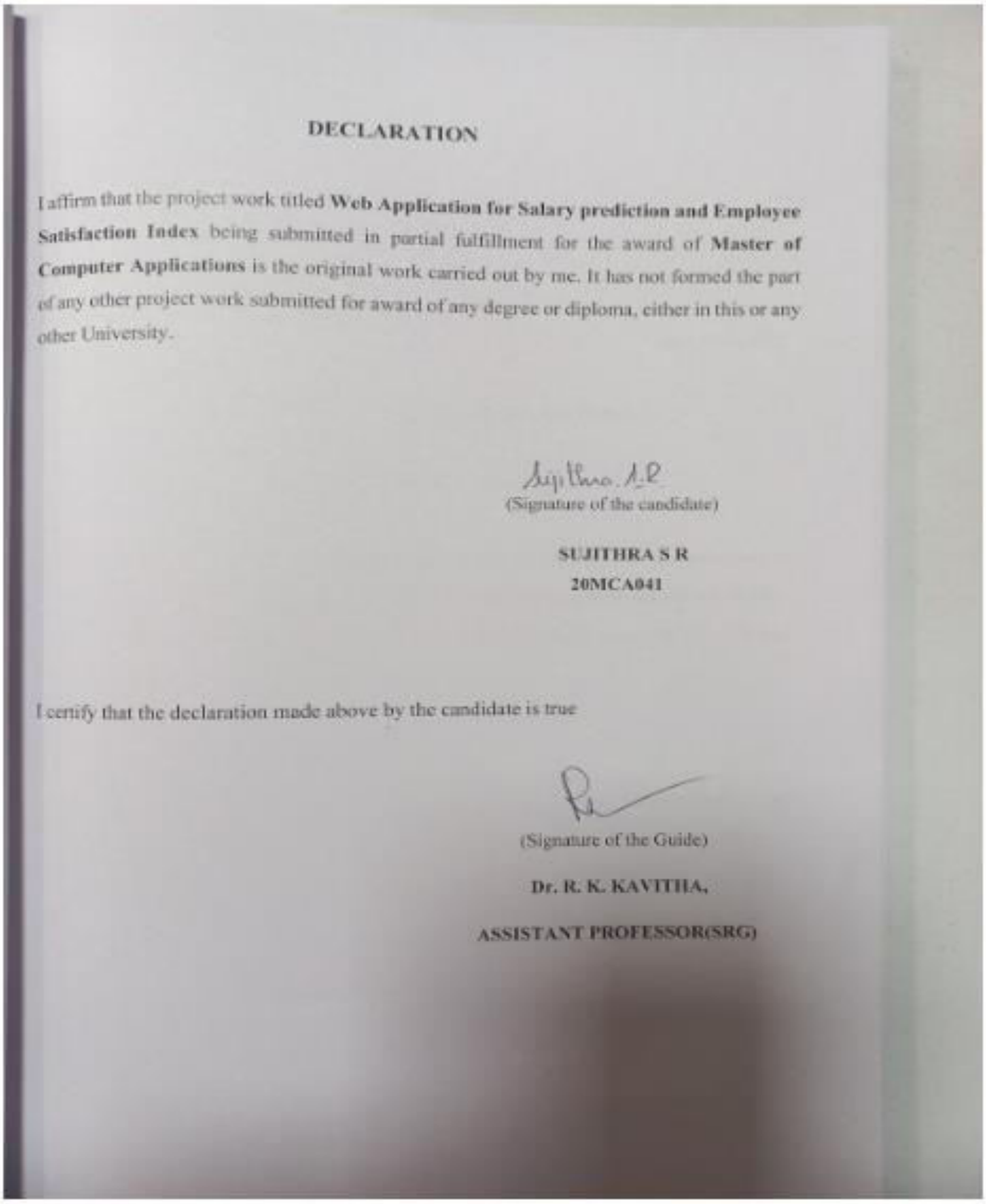
Submitted for the Project Viva-Voce examination held on 23-06-2022

23/6/22

**Internal Examiner**

23/6/22

**External Examiner**



## DECLARATION

I affirm that the project work titled **Web Application for Salary prediction and Employee Satisfaction Index** being submitted in partial fulfillment for the award of **Master of Computer Applications** is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

Sujithra.S.R
(Signature of the candidate)

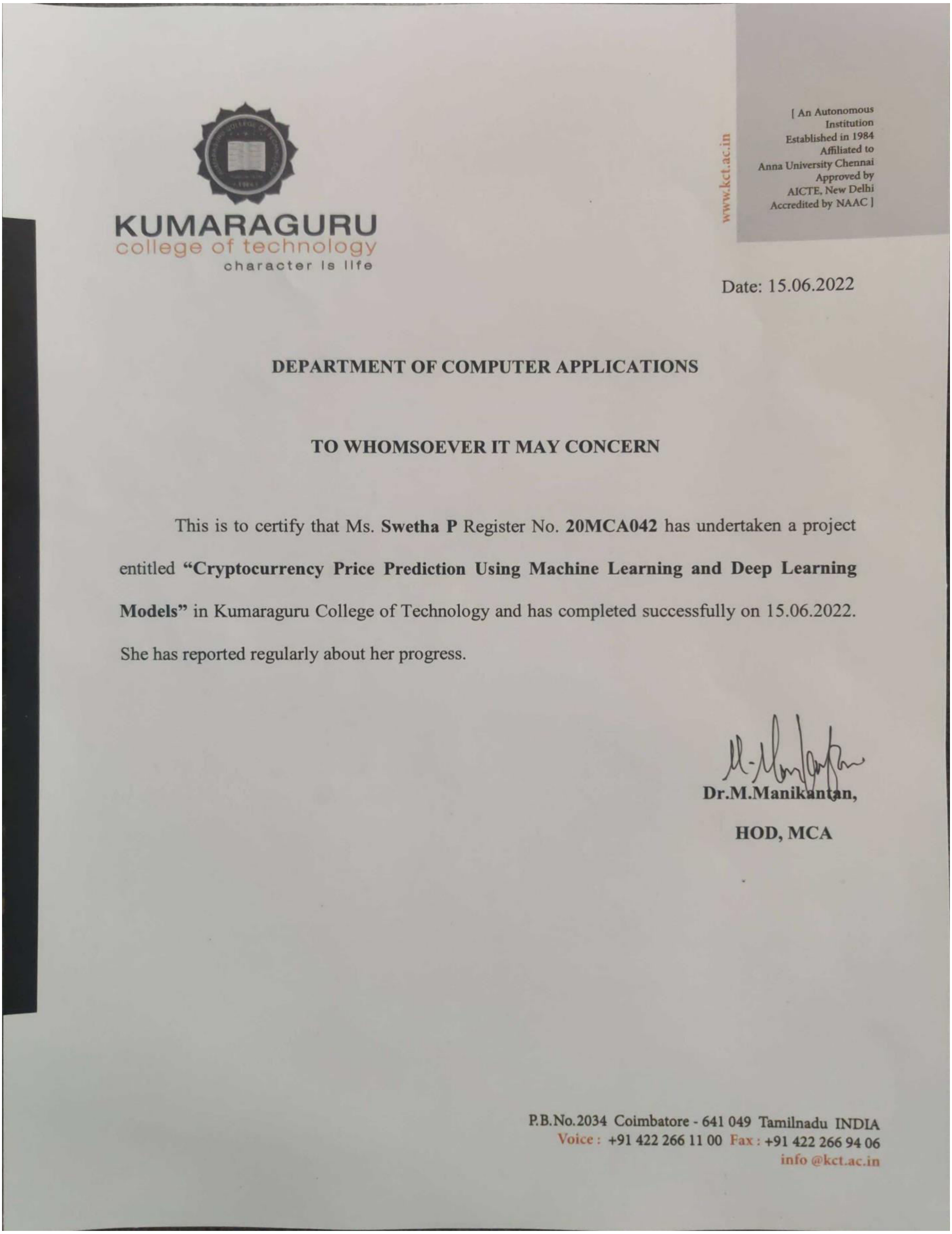
**SUJITHRA S R**
**20MCA041**

I certify that the declaration made above by the candidate is true

(Signature of the Guide)

**Dr. R. K. KAVITHA,**

**ASSISTANT PROFESSOR(SRG)**



KUMARAGURU
college of technology
character is life

www.kct.ac.in

[ An Autonomous
Institution
Established in 1984
Affiliated to
Anna University Chennai
Approved by
AICTE, New Delhi
Accredited by NAAC ]

Date: 15.06.2022

## DEPARTMENT OF COMPUTER APPLICATIONS

## TO WHOMSOEVER IT MAY CONCERN

This is to certify that Ms. **Swetha P** Register No. **20MCA042** has undertaken a project entitled **"Cryptocurrency Price Prediction Using Machine Learning and Deep Learning Models"** in Kumaraguru College of Technology and has completed successfully on 15.06.2022. She has reported regularly about her progress.

**Dr.M.Manikantan,**

**HOD, MCA**

P.B.No.2034 Coimbatore - 641 049 Tamilnadu INDIA
Voice : +91 422 266 11 00 Fax : +91 422 266 94 06
info @kct.ac.in

# ACKNOWLEDGEMENT

I thank the all mighty who has been with me through every walk of life, guarding me and showering the blessings throughout the endeavor to put forth my dissertation.

I sincerely thank **SHRI. SHANKAR VANAVARAYAR**, Joint Correspondent, Kumaraguru College of Technology, for providing the ample facilities to carry out this project.

I sincerely thank **Dr. D. SARAVANAN**, Principal, Kumaraguru College of Technology, forgiving me an opportunity to undertake this full project work.

I am grateful to **Dr. M. MANIKANTAN**, Associate Professor and Head, Department of Computer Applications, for his continuous encouragement and support during the course of this project.

I am obliged to **Dr. P. PARAMESWARI,** Assistant Professor (SRG), Department of Computer Applications, Project Coordinator, for her guidance, support, valuable suggestions and great encouragement during the course of this project.

I sincerely thank my guide **Dr. N. JAYAKANTHAN,** Assistant Professor (SRG), Department of Computer Applications, Project Guide, Kumaraguru College of Technology, for his sincere advice, thought provoking discussions and immense help throughout the project and encouragement given by him.

Finally, I wish to thank my parents, friends and well-wishers who have been a catalyst throughout the development of the project work.

# TABLE OF CONTENTS

# ABSTRACT

The main aim of the "**Cryptocurrency price prediction using Machine learning and Deep learning models**" project is to analyze and develop a model for predicting the prices of three types of cryptocurrencies, namely Bitcoin (BTC), Litecoin (LTC), and Ethereum (ETH), that are widely used cryptocurrencies in the digital market. These cryptocurrencies are under a decentralized control, which is not owned by the government or any other company. Cryptocurrency price forecasting is difficult due to its price volatility and dynamism, so it is important to predict its price for making investments.

Different data sets of Bitcoin, Litecoin and Ethereum are used. With the help of python libraries, the data filtration process is done. After the understanding of the data, the data is trimmed and the required features or attributes best suited for the model is used. Then the model is implemented to predict the price with the help of Machine learning algorithms - linear regression, Prophet and Deep learning algorithm - Long Short-Term Memory (LSTM), by training and testing the model. This project will conclude the best model for predicting cryptocurrencies price and helps investors in taking decision while investing in them. The final results will demonstrate the more optimized model.

**LANGUAGE:** Python

**TOOL:** Google Colab Notebook

**OS**: Windows 11

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATION

| S.NO | ABBREVIATION | DESCRIPTION |
|---|---|---|
| 1 | BTC | BITCOIN |
| 2 | LTC | LITECOIN |
| 3 | ETH | ETHEREUM |
| 4 | LSTM | LONG SHORT TERM MEMORY |

# CHAPTER 1
# INTRODUCTION

## 1.1 COMPANY PROFILE

**Kumaraguru College of Technology (KCT),** Coimbatore is an Engineering College started in 1984 under the auspices of **Ramanandha Adigalar Foundation**, a charitable educational trust of Sakthi Group. It is situated in a sprawling campus of 150 acres in the IT corridor of Coimbatore which in many ways was a front runner in the eco-system.

The able guidance and patronage of Arutselvar Dr. N. Mahalingam, Founder, Sakthi Group along with the efficient administration of Dr.B.K.Krishnaraj Vanavarayar, Chairman, the resourcefulness of Sri. M.Balasubramaniam, Correspondent and the foresightedness of Sri.Shankar Vanavarayar, Joint Correspondent have equipped the college with the excellent facilities such as spacious classrooms, seminar halls, well equipped laboratories, excellent sporting amenities, dedicated high-speed internet connectivity (broadband) and well-qualified faculty. Five Academic Blocks house the different departments. The administrative building, “Dr.Mahalingam Vigyan Bhavan” is an architectural beauty and a land mark in Coimbatore.

Currently the college, as an autonomous institution affiliated to the Anna University, offers 13 under-graduate (B.E., B.Tech.) and 14 post-graduate (M.E., M.Tech., MCA, MBA) programs of study. All the above courses have the approval of the All India Council for Technical Education (AICTE) and all the eligible UG programs have also been accredited by National Board of Accreditation (NBA). In addition, KCT has also been accredited by National Assessment and Accreditation Council (NAAC) of the University Grants Commission (UGC). 9 of the 15 academic departments have been recognized as research centers permitting research leading to Ph.D degree by Anna University.

The value of the education and training imparted by the college is highlighted by the interest shown by leading companies for on-campus recruitments. Our alumni have done us proud by proving their worth in their chosen field of work.

## 1.2 OBJECTIVE

The main objective of the project is to analyse the predictions of the Machine Learning Algorithm - linear regression and Artificial Neural Network Model - Long Short-Term Memory (LSTM) in predicting the value of a Bitcoin, Litecoin and Ethereum. The crypto market is very volatile, and among all the cryptocurrencies in the market, Bitcoin, Litecoin, Ethereum is experienced by most of the investors due to its anonymity and transparency in the system. Cryptocurrencies price depend on a lot of factors like technological progress, internal competition, pressure on the markets to deliver, economic problems, security issues, political factor and more. Their high volatility leads to the great potential of high profit if intelligent investing strategies are taken. But, due to their lack of indexes, cryptocurrencies are relatively not accurately predictable compared to traditional financial predictions like stock market prediction.

This project focus on close price factor in predicting the price. Using Machine learning and deep leaning algorithms, hidden patterns from data is discovered and will result in better efficient predictions. The proposed hybrid models are evaluated using evaluation matrix RMSE of Bitcoin, Ethereum, and Litecoin. Final prediction outcomes of a machine learning model and an artificial neural network model is compared and will help in achieving a reliable prediction model that investors can rely on based on historical cryptocurrency prices.

## 1.3 PROBLEM DESCRIPTION

Most people wants to grow their money by investing in the stock market, but with the growing technology and the introduction of e-money, the better way to make the money to grow is by investing in cryptocurrency. Cryptocurrency is not under the influence of any country or government, for this reason, it can be invested by anyone around the globe without the fear of being imposed of taxes from other countries. The success of cryptocurrency is measured by its huge capitalism growth and price. One of the primary reasons for people to dive into the crypto market is that it's very easy and simple to buy and sell assets via trading platforms such as WazirX, Binance, etc. Compared to traditional exchanges, crypto exchanges are way faster as assets can be bought and sold in a minute wherein traditional exchange method we require a day or two for the process to take place. This is where blockchain technology comes into the picture. Traditional exchanges use convoluted technology which requires checking and rechecking, but with the introduction of blockchain, settling can be done instantly.

One of the major problem with cryptocurrency price predictions is that they lack sufficient analytical support. Accurate predictions can assist cryptocurrency investors towards right investing decisions and lead to potential increased profits. So, there is a need for a model that can predict the future price movement of Cryptocurrencies for making decision before investing in Cryptocurrency market.

# CHAPTER 2

# SYSTEM ANALYSIS

This chapter describes the analysis of the System. Analysis is a detailed study of the various operations performed on the work.

## 2.1 EXISTING WORK:

The previously existing Cryptocurrencies price prediction analysis are only based on Machine Learning model. Although machine learning has been successful in predicting stock market prices through a host of different time series models, its application in predicting cryptocurrency prices has been quite restrictive due to various factors of volatility.

### DRAWBACKS:

- Cryptocurrency price prediction is considered a very challenging task, due to its chaotic and very complex nature.
- Previously used algorithm do not have great time management.
- Other Models (like ARIMA) are not able to capture non-linear patterns of very complicated prediction problems and also they does not has feedback connections.

## 2.2 PROPOSED WORK:

The Proposed work overcomes the drawbacks in the existing system. In the proposed work, the Cryptocurrencies price are predicted using both Machine learning algorithms and Deep learning algorithm. The results of these algorithm predictions are compared and thus, assist the investors in taking right decision before investing in Cryptocurrencies.

### ADVANTAGES:

- The best model in Cryptocurrencies price prediction with high accuracy can be known by comparing ML model and Deep learning model.
- The Future price movement of Cryptocurrencies are predicted.
- LSTM model are able to learn long-term dependencies by utilizing feedback connections in order to remember past network cell states.
- LSTM shows remarkable performance on time series forecasting and also it is also able to create a controlled information flow.
- Deep learning model result only minor error rate with highest accuracy.

## 2.3 LITERATURE REVIEW

The prediction system using Log regression, SVM, ANN, and random forest was proposed and shows that SVM has the best accuracy regarding a time-scale activity consisting of daily, 15-, 30- and 60- minutes return. Although SVM does tend to show better results out of the 4 algorithms, the prediction system can still show better results when Deep learning concepts are applied (1).

The prediction of crypto coins using the SVM and SVM-PSO method is suggested, where they used the day trading method to predict the values of ETH, BTC, XEM, XRP, XLM, LTC. SVM-PSO shows the optimized results. Performance accuracy of different Classifiers differs from coin to coin. However, this paper works only with a machine learning algorithm, and hence the data can be further improved by implementing the Deep Learning concept (2).

The study in (3) on Bitcoin daily price prediction with high-dimensional data reveals that logistic regression and linear discriminant analysis achieve an accuracy of 66%. On the other hand, surpassing (a sophisticated machine learning algorithm) outperforms the benchmark results for daily price prediction, with statistical techniques and machine learning algorithms having the greatest accuracies of 66% and 65.3%, respectively.

A prediction model was proposed using four major algorithms, Gradient Boosted Tree, Neural Network, Ensemble Learning Method (with the best accuracy of 92.4%), and KNN model (With the least accuracy due to noisy random features and extreme volatility)(4).

The prediction of Bitcoin price using a transaction graph is proposed. The experiment consists of the Baseline, Logistic Regression, SVM, and Neural Network model with an accuracy of 53.4%, 54.3%, 53.7 and 55.1%. The feature selection in this paper is based on the Bitcoin block chain network which tends to be the least informative feature for the prediction of the Bitcoin price (5).

Machine learning (ML) is a type of artificial intelligence that can predict the future based on past data. ML-based models have various advantages over other forecasting models as prior research has shown that it not only delivers a result that is nearly or exactly the same as the actual result, but it also improves the accuracy of the result (6). Examples of machine learning include neural networks (NN), support vector machines (SVM), and deep learning.

The authors of (7) employ the traditional support vector machine and linear regression methods to forecast Bitcoin values. This research takes into account a time series prediction made up of everyday

A linear regression model was used to predict the various cryptocurrency price using the open, low, and high cost. The experiment shows an accuracy of 99.3%. This paper does consist of a high accuracy rate but the data set used is comparatively small for a model to work on a real-time chart (8).

The authors of (9) used machine learning techniques to address both a multiple regression technique that relies on highly correlated characteristics and a deep learning mechanism that uses a conjugate gradient mechanism in conjunction with a linear search for BTC price prediction.

Predicting the crypto currency prices using sentimental analysis and Machine learning concepts like SVM and Random Forest on ETH, BTC, and XRP with BTC being the highest accuracy of 0.72. This accuracy rate is very low since machine learning algorithms were applied and it can be improved by testing with deep learning models (10).

# CHAPTER 3
# SYSTEM SPECIFICATION

The system specification is a technical specification of requirements for the analysis work.

## 3.1 HARDWARE REQUIREMENTS

| | | |
|---|---|---|
| Processor | : | Intel core i3 |
| Hard disk | : | 4 GB RAM |
| Mouse | : | Compatible mouse |
| Monitor | : | LCD monitor |

## 3.2 SOFTWARE REQUIREMENTS

| | | |
|---|---|---|
| Operating System | : | Windows 11 |
| Language | : | Python |
| Tool | : | Google Colab Notebook |

# CHAPTER 4
# SOFTWARE DESCRIPTION

## 4.1 LANGUAGE DESCRIPTION

### PYTHON

Python is a high-level, interpreted, interactive, and object-oriented scripting language created by Guido Van Rossum. Python is designed to be highly readable. Python consistently ranks as one of the most popular programming languages. Python is a multi-functional, maximally interpreted programming language with several advantages that are often used to streamline massive, and complex data sets.

### Characteristics of Python

Python has a number of distinguishing characteristics that make it the best option for data analysis.

- Simplicity
- Readability
- Flexibility
- Huge library collection
- Graphics and Visualization
- Built-in data analytics tool

Python is an internationally recognized programming language that can assist users in managing data. Python is the most popular language among data analysts and data scientists due to its extensive set of graphical options and visualization tools that make data more available. Furthermore, it is constantly evolving and becoming more powerful, multi-featured, and versatile.

### PYTHON LIBRARIES

A Python library is a reusable chunk of code that can be include in the programs or projects. Compared to languages like C++ or C, Python libraries do not pertain to any specific context in Python.

The libraries that are used in this project are as follows:

**• Tensorflow**

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

**• Keras**

Keras is high-level API wrapper for the low-level API, capable of running on top of TensorFlow, CNTK, or Theano. Keras High-Level API handles the way we make models, defining layers, or set up multiple input-output models. In this level, Keras also compiles our model with loss and optimizer functions, training process with fit function.

**• Scikit-learn**

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

**• NumPy**

NumPy, which stands for Numerical Python, is 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.

**• Matplotlib**

Matplotlib is one of the most popular Python packages used for data visualization. It is a crossplatform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

**• Pandas**

A pandas is an open-source, fast, powerful, flexible and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.

## 4.2 TOOL DESCRIPTION

### GOOGLE COLAB NOTEBOOK

Google is quite aggressive in AI research. Over many years, Google developed AI framework called TensorFlow and a development tool called Colaboratory. Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook. Colab supports GPU and it is totally free. The reasons for making it free for public could be to make its software a standard in the academics for teaching machine learning and data science. It may also have a long term perspective of building a customer base for Google Cloud APIs which are sold per-use basis. Irrespective of the reasons, the introduction of Colab has eased the learning and development of machine learning applications.

# CHAPTER 5
# DATASET DETAILS

## 5.1 DATA COLLECTION

Data collection is the first step to take in order to start any project. It is defined as the procedure of collecting, measuring, and analyzing accurate insights for research using standard validation techniques. The most important objective of data collection is ensuring that the gathered information is rich in content and reliable for statistical analysis, so that data-driven decisions can be made efficiently and effectively. In this project, the historical datas of Bitcoin, Litecoin and Ethereum are gathered from Yahoo finance website, which is one of the best site for business.

## 5.2 DATASET DESCRIPTION

This section gives the detailed description about the dataset. The dataset consists of Bitcoin-2684 instances, Litecoin-2706 instances and Ethereum- 1662instances, all with 9 attributes.

| ATTRIBUTE | DATA DESCRIPTION |
|---|---|
| DATE | Date wise data recorded. |
| OPEN | It represent the first price level during the specified interval |
| CLOSE | It represent the last price level during the specified interval. |
| HIGH | It represent the highest reached price during the specified interval. |
| LOW | It represent the lowest reached price during the specified interval. |
| VOLUME | Total amount traded during the period. |
| TIMESTAMP | Timestamps are in Unix time in which data record |
| WEIGHTED PRICE | Average of the coin price |

**Table 5.2.1 Description of attributes in dataset**

## 5.3 SYSTEM FLOW DIAGRAM

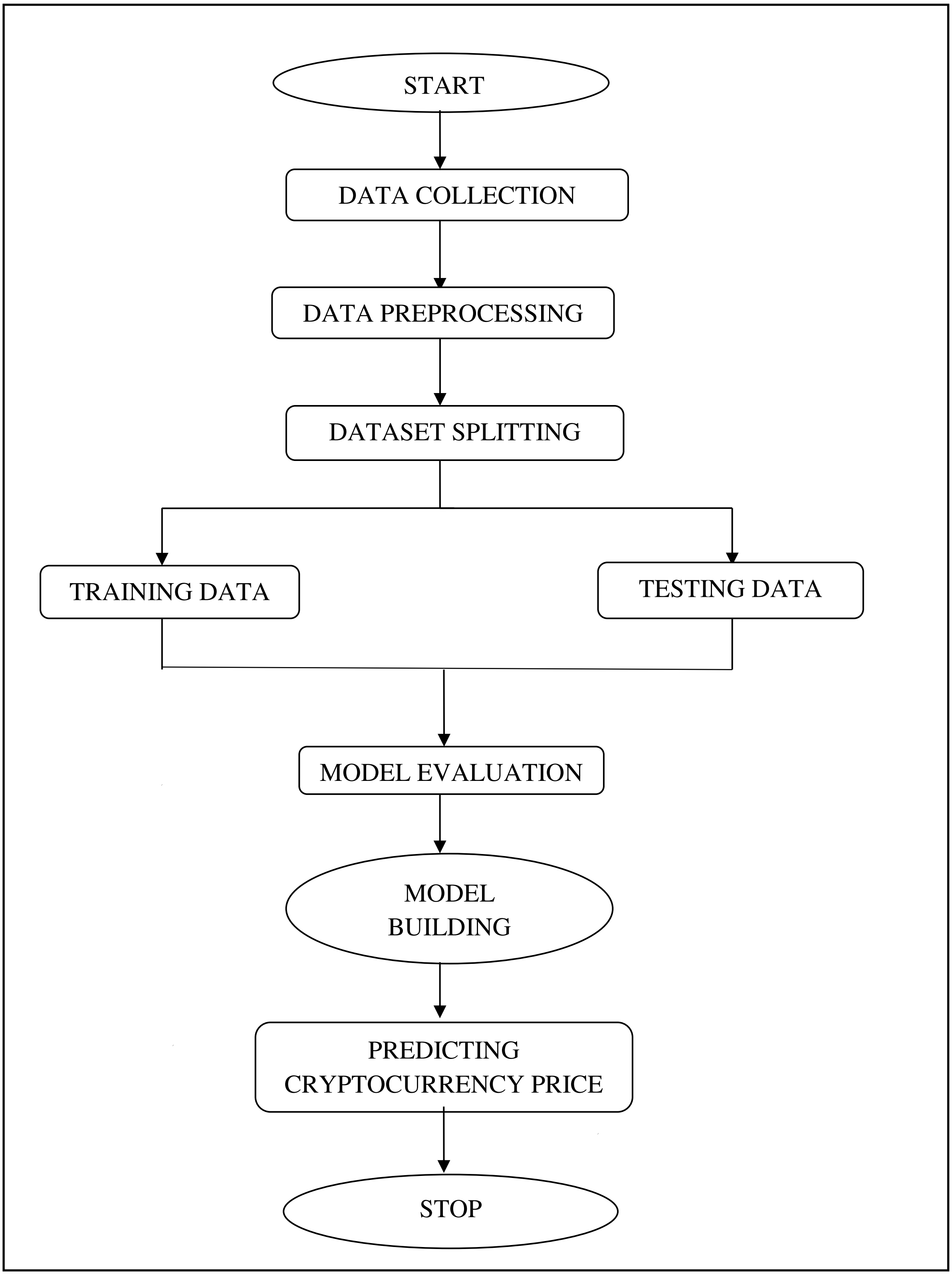


**Figure 5.3.1 System flow diagram**

# CHAPTER 6
# METHODOLOGY

## 6.1 MACHINE LEARNING ALGORITHM

Machine learning is a core sub-area of artificial intelligence, it enables computers to get into a mode of self-learning without being explicitly programmed. When exposed to new data, these computer programs are enabled to learn, grow, change, and develop by themselves. Using Machine learning algorithm in time series forecasting make forecasting faster, more precise, and more efficient in the long run. ML has proven to help better process both structured and unstructured data flows, swiftly capturing accurate patterns within massifs of data.

### 6.1.1 Linear Regression

This technique is used to identify the relationship between dependent and independent variables and is leveraged to predict future outcomes. When one dependent and one independent variable is used, then it is called the simple linear regression. Linear regression is a statistical tool used to help predict future values from past values. It is commonly used as a quantitative way to determine the underlying trend and when conditions are over extended. Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable X predict is called the dependent variable. The variable Y using to predict the other variable's value is called the independent variable.

### 6.1.2 Prophet

It is an algorithm to build forecasting models for time series data. Unlike the traditional approach, it tries to fit additive regression models a.k.a. ‘curve fitting’. The best thing about this algorithm is that it is very flexible when it comes to the data that is fed to the algorithm. It have NAs and don’t need to have all the dates and times lined up. And, it works pretty reasonably by default, without setting any parameters explicitly.

## 6.2 DEEP LEARNING ALGORITHM

Deep learning is a subset of a Machine Learning algorithm that uses multiple layers of neural networks to perform in processing data and computations on a large amount of data. Deep learning algorithm works based on the function. Deep learning algorithms run data through several “layers” of neural network algorithms, each of which passes a simplified representation of the data to the next layer.

### 6.2.1 Long Short-Term Memory (LSTM):

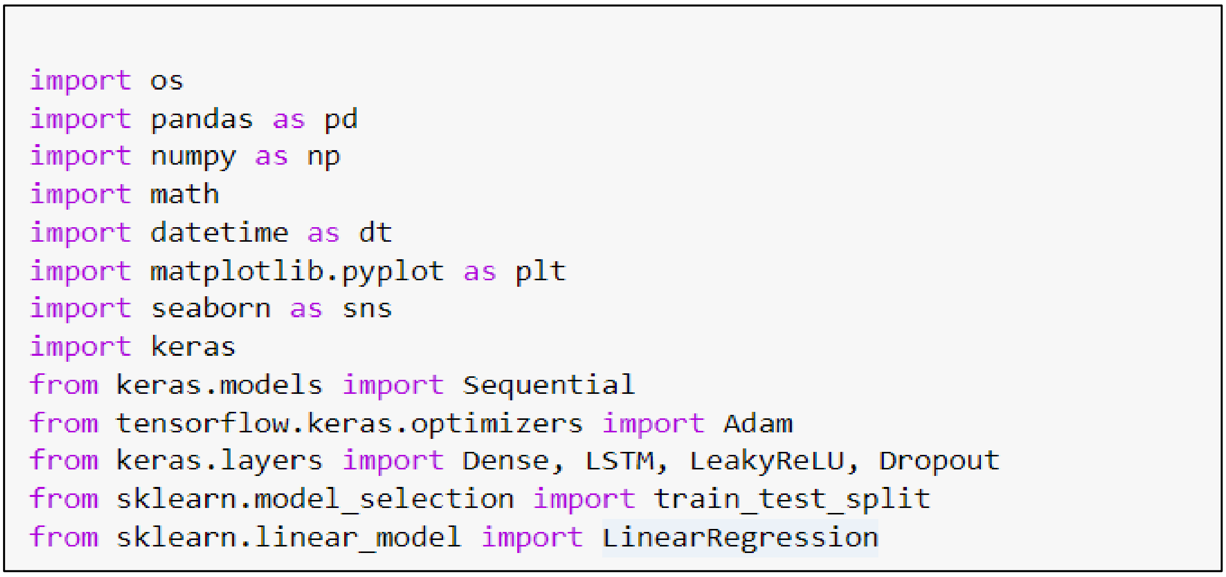
It is a deep learning concept or particularly a Recurrent Neural Network concept that avoids the vanishing gradient problem. The main reason for using this algorithm is that it avoids the back propagation error from vanishing or exploding, instead, these errors can flow backward through an unlimited number of virtual layers unfolded in space. LSTM mainly works on time series graphs with data sets that consist of events that occur thousands or millions of discrete-time steps earlier. It works with given long delays between significant events and can also handle signals with a mixture of low and high-frequency components. Over a lot of researchers have used LSTM to predict time series related data sets for stock prediction and have achieved greater or higher accuracy compared to other algorithms. LSTM is capable of recognizing context-sensitive language unlike any other previous models based on Hidden Markov Models (HMM) and other similar concepts.

# CHAPTER 7

# IMPLEMENTATION

## 7.1 IMPORTING LIBRARIES

In Python, the import keyword is used to make code in one module available in another. Imports in Python are important for structuring the code effectively. Python Libraries are set of useful functions that eliminate the need for writing codes from scratch. Python libraries Plays a vital role in developing machine learning, data science, data visualization, image and data manipulation applications, loading the dataset. The packages imported in the project is shown in figure 7.1.1.

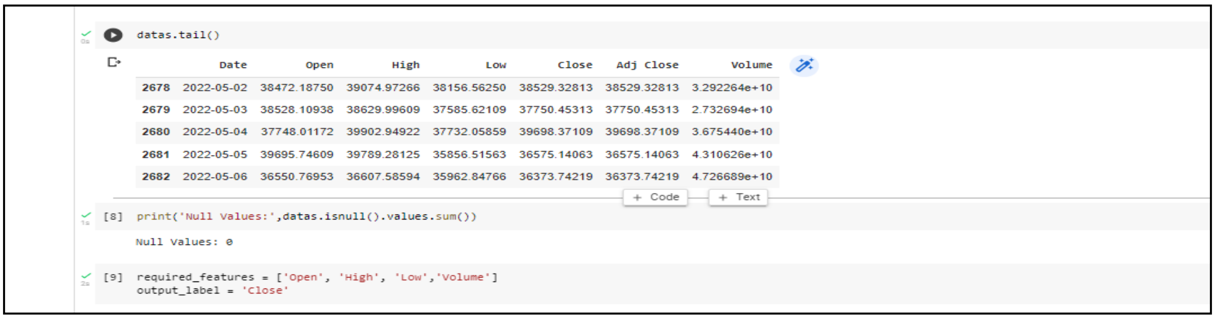


```
import os
import pandas as pd
import numpy as np
import math
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
import keras
from keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from keras.layers import Dense, LSTM, LeakyReLU, Dropout
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

**Figure 7.1.1 Imported library packages**

## 7.2 DATA PREPROCESSING

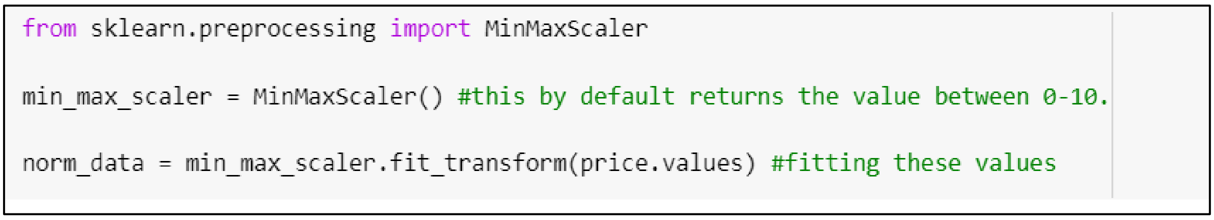
The dataset has no missing values. Handling missing values is important because machine learning algorithms do not support data with missing value. Normalize the function into an array using minmax scalar feature to represent the data as normalized data.



```
datas.tail()

            Date         Open         High          Low        Close    Adj Close        Volume
2678  2022-05-02  38472.18750  39074.97266  38156.56250  38529.32813  38529.32813  3.292264e+10
2679  2022-05-03  38528.10938  38629.99609  37585.62109  37750.45313  37750.45313  2.732694e+10
2680  2022-05-04  37748.01172  39902.94922  37732.05859  39698.37109  39698.37109  3.675440e+10
2681  2022-05-05  39695.74609  39789.28125  35856.51563  36575.14063  36575.14063  4.310626e+10
2682  2022-05-06  36550.76953  36607.58594  35962.84766  36373.74219  36373.74219  4.726689e+10
```

```
[8] print('Null Values:',datas.isnull().values.sum())

    Null Values: 0
```

```
[9] required_features = ['Open', 'High', 'Low','Volume']
    output_label = 'Close'
```



```
from sklearn.preprocessing import MinMaxScaler

min_max_scaler = MinMaxScaler() #this by default returns the value between 0-10.

norm_data = min_max_scaler.fit_transform(price.values) #fitting these values
```

**Figure 7.2.1 Preprocessing data**

## 7.3 VISUALIZATION

The demo of the model can be visualized using Exploratory Data Analysis. Data Visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

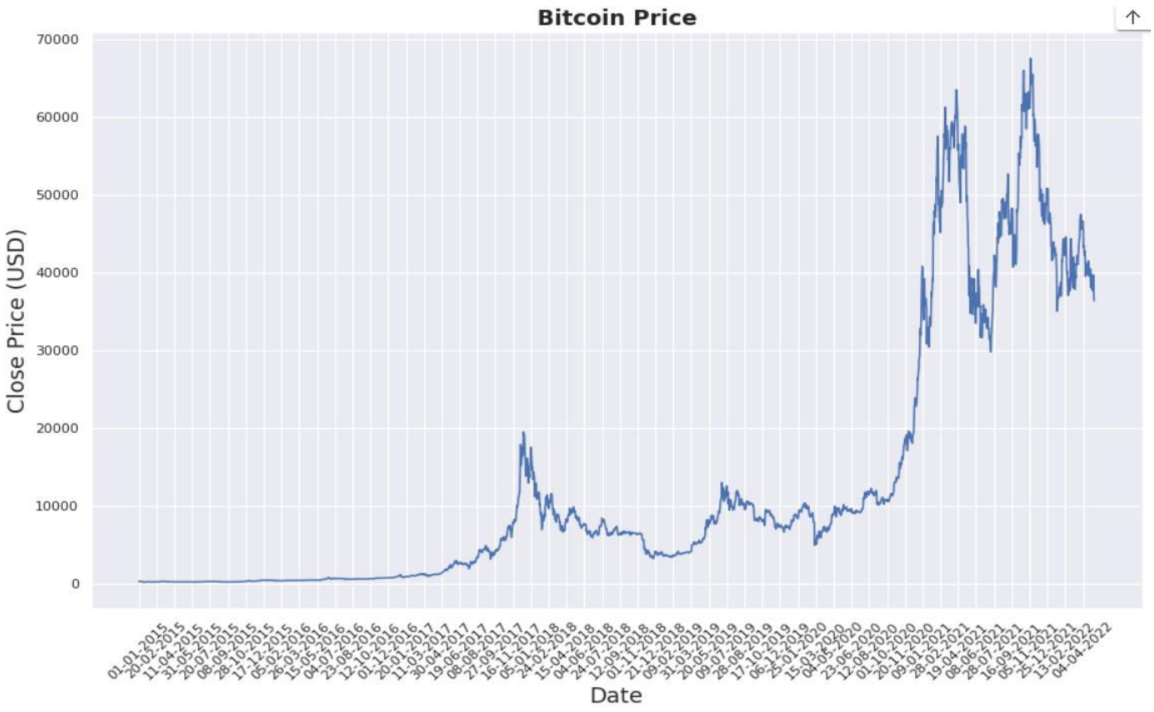


**Figure 7.3.1 Close price visualization of BTC**

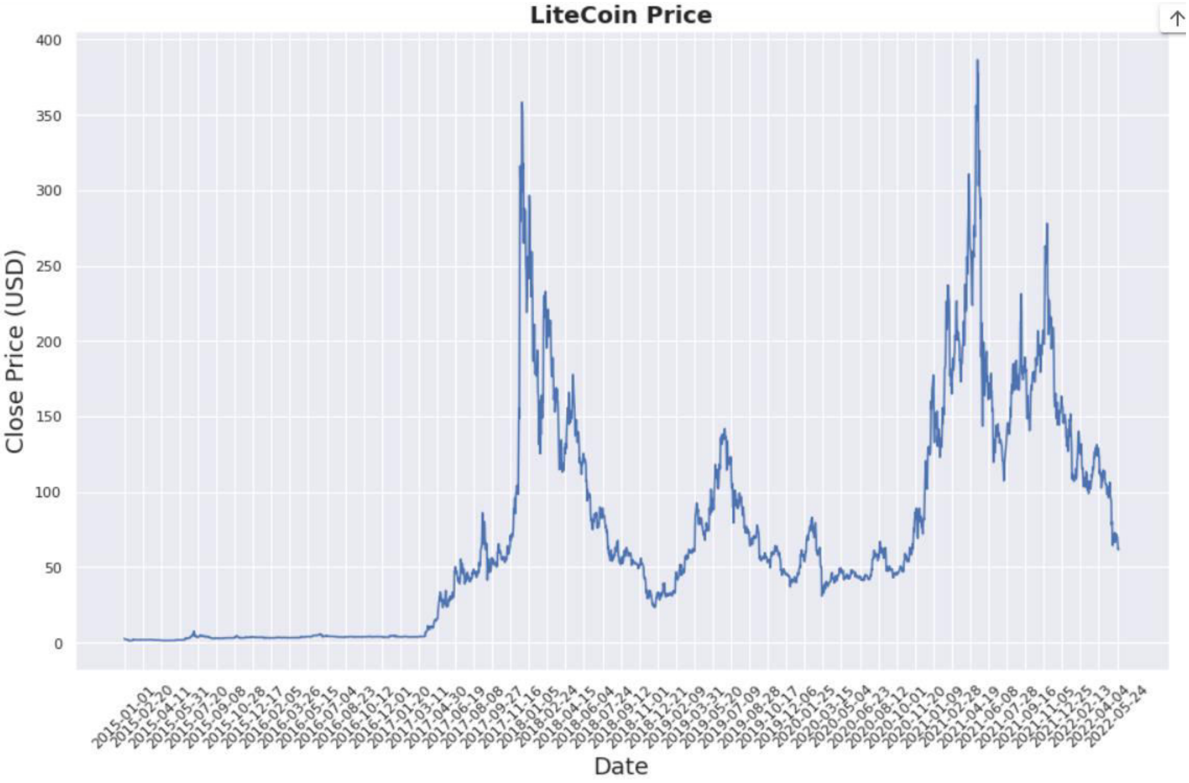


**Figure 7.3.2 Close price visualization of LTC**

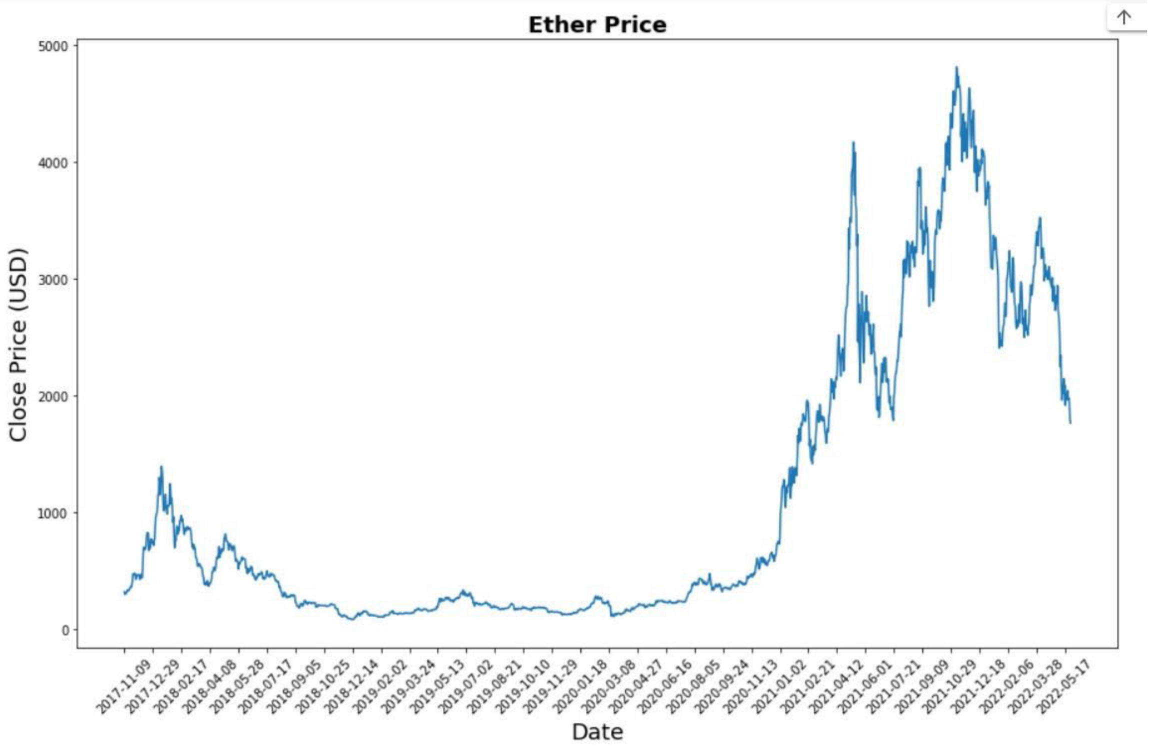
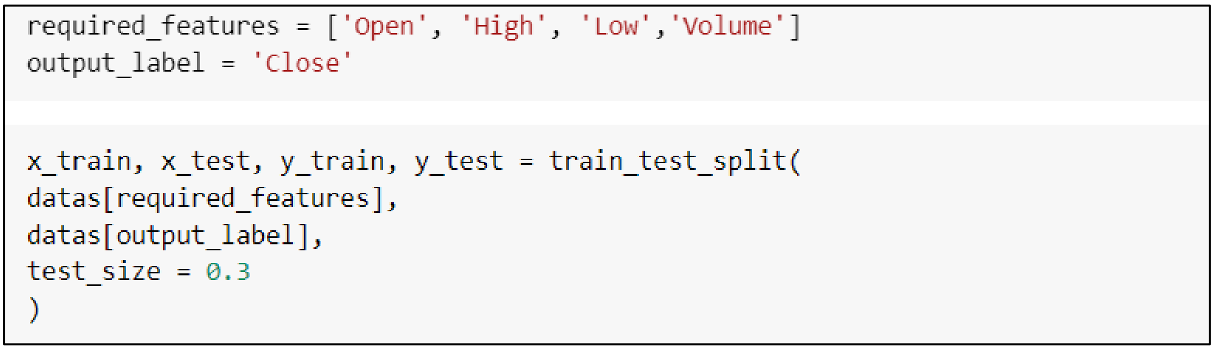


**Figure 7.3.3 Close price visualization of ETH**

## 7.4 TRAINING AND TESTING DATA USING MACHINE LEARNING MODEL

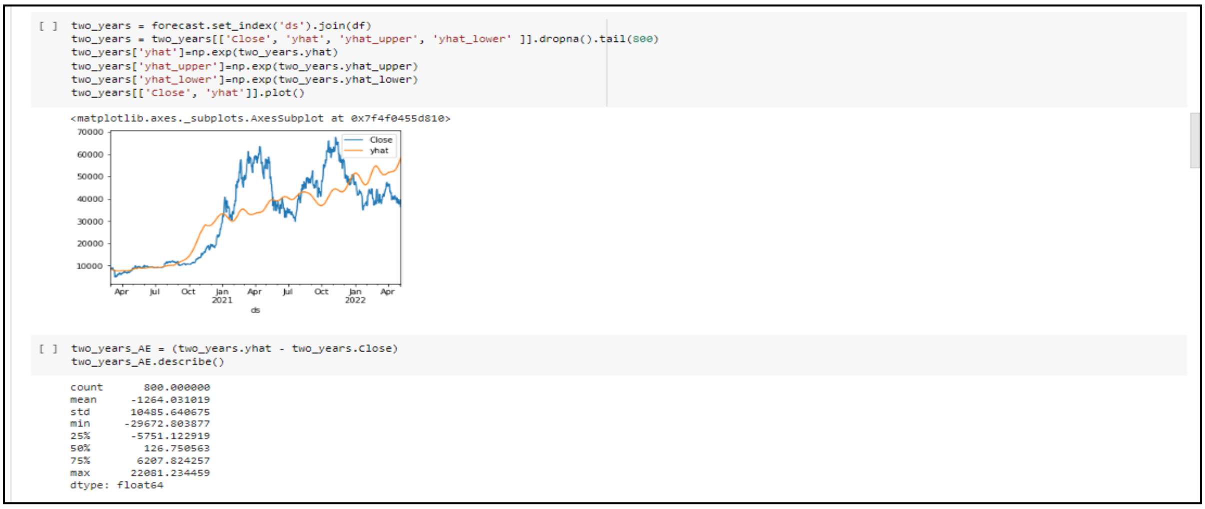
Training data and testing data are the two important concepts in machine learning algorithms. Training data builds up the machine learning algorithm. After the model is built, the testing data validates and makes accurate predictions. In supervised learning problems, each observation consists of an observed output variable and one or more observed input variables.



```
required_features = ['Open', 'High', 'Low','Volume']
output_label = 'Close'
```

```
x_train, x_test, y_train, y_test = train_test_split(
datas[required_features],
datas[output_label],
test_size = 0.3
)
```

**Figure 7.4.1 Training and Testing Linear regression model**



```
[ ]  two_years = forecast.set_index('ds').join(df)
     two_years = two_years[['Close', 'yhat', 'yhat_upper', 'yhat_lower' ]].dropna().tail(800)
     two_years['yhat']=np.exp(two_years.yhat)
     two_years['yhat_upper']=np.exp(two_years.yhat_upper)
     two_years['yhat_lower']=np.exp(two_years.yhat_lower)
     two_years[['Close', 'yhat']].plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4f0455d810>
```

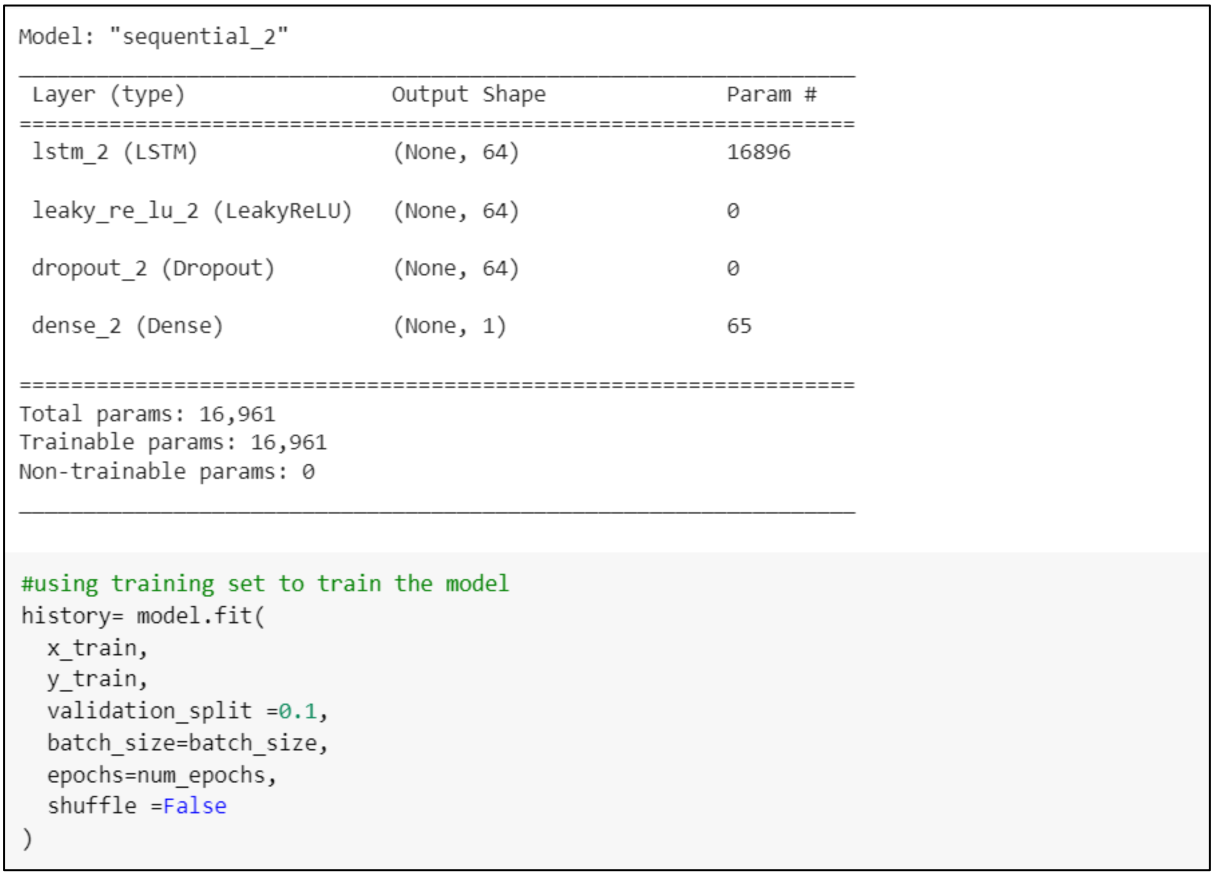
```
[ ]  two_years_AE = (two_years.yhat - two_years.Close)
     two_years_AE.describe()
```

```
count       800.000000
mean      -1264.031019
std       10485.640675
min      -29672.803877
25%       -5751.122919
50%         126.750563
75%        6207.824257
max       22081.234459
dtype: float64
```

**Figure 7.4.2 Training and Testing Prophet model**

## 7.5 TRAINING AND TESTING DATA USING DEEP LEARNING MODEL

A simple method that use to split the historic dataset into train and test datasets. It calculates the index of the split point and separates the data into the training datasets with of the observations that can use to train our model, leaving the remaining for testing the model.

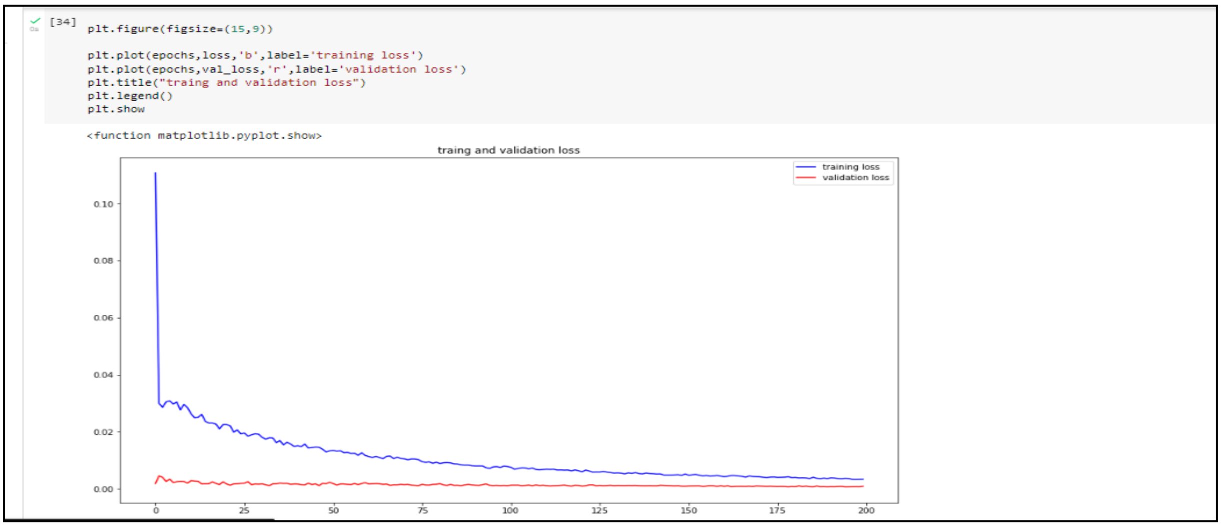


```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_2 (LSTM)               (None, 64)                16896

 leaky_re_lu_2 (LeakyReLU)   (None, 64)                0

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 16,961
Trainable params: 16,961
Non-trainable params: 0
_________________________________________________________________
```

```
#using training set to train the model
history= model.fit(
  x_train,
  y_train,
  validation_split =0.1,
  batch_size=batch_size,
  epochs=num_epochs,
  shuffle =False
)
```

**Figure 7.5.1 Training and Testing LSTM model**

The number of epochs is a hyper parameter that defines the number times of learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset had an opportunity to update the internal model parameters. An epoch is comprised of one or more batches. This model is trained with 250 epochs.



```
[34] plt.figure(figsize=(15,9))

     plt.plot(epochs,loss,'b',label='training loss')
     plt.plot(epochs,val_loss,'r',label='validation loss')
     plt.title("traing and validation loss")
     plt.legend()
     plt.show

     <function matplotlib.pyplot.show>
```

**Figure 7.5.2 Training and Validation loss**

## 7.6 MODEL BUILDING

Building a model in machine learning is creating a mathematical representation by generalizing and learning from training data. Then, the built machine learning model is applied to new data to make predictions and obtain results. The model build can be either a regression model or a classification model based on the target variable which is known as the Y variable. If the target variable has a quantitative value, then a regression model should be built. In this project, the target variable has a quantitative value, so regression models are built using various regression algorithms.

## 7.7 MODEL EVALUATION

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during the initial research phases, and it also plays a role in model monitoring. The evaluation metrics used here are R squared (R2), Root Mean Squared Error (RMSE).

### R SQUARED (R2)

The R2 score is a very important metric that is used to evaluate the performance of a regression-based machine learning model. It is pronounced as R squared and is also known as the coefficient of determination. It works by measuring the amount of variance in the predictions explained by the dataset. Simply put, it is the difference between the samples in the dataset and the predictions made by the model. If the value of the r squared score is 1, it means that the model is perfect and if its value is 0, it means that the model will perform badly on an unseen dataset. This also implies that the closer the value of the r squared score is to 1, the more perfectly the model is trained.

### ROOT MEAN SQUARED ERROR (RMSE)

As RMSE is clear by the name itself, it is a simple square root of mean squared error. The lower value of MAE, MSE, and RMSE implies higher accuracy of a regression model. For an ideal model, MSE/RMSE/MAE should be 0.

This section shows the results obtained from linear regression, prophet and long short-term memory (LSTM) algorithms using three types of popular cryptocurrency: BTC, ETH, and LTC. Based on the criteria, all of the models applied to three types of currencies can be considered good models. But it was discovered that the linear regression model's R2 score rate is very high as shown in Table 7.7.1 when compared to the other Machine Learning model. The LSTM model, on the other hand shows a minor RMSE rate as shown in Table 7.7.3. This, in turn, demonstrates that the Deep learning model is more optimized than the machine learning model. Simulation results from those models will indicate that there are few occasions where the forecast result differs from the actual results.

| **CRYPTOCURRENCY** | **LINEAR REGRESSION (R2 SCORE)** |
|---|---|
| BTC | 0.99 |
| ETH | 0.98 |
| LTC | 0.99 |

**Table 7.7.1 R2 score of Cryptocurrency using linear regression**

| **CRYPTOCURRENCY** | **PROPHET (R2 SCORE)** |
|---|---|
| BTC | 0.66 |
| ETH | 0.91 |
| LTC | 0.29 |

**Table 7.7.2 R2 score of Cryptocurrency using Prophet**

| **CRYPTOCURRENCY** | **LSTM (RMSE)** |
|---|---|
| **BTC** | 0.2590 |
| **ETH** | 0.2609 |
| **LTC** | 0.2234 |

**Table 7.7.3 RMSE score of Cryptocurrency using LSTM**

# CHAPTER 8

# CONCLUSION AND FUTURE ENHANCEMENT

## 8.1 CONCLUSION

In this project, three types of machine learning algorithm are constructed and used for predicting the future price movement of three types of cryptocurrency - BTC, ETH, and LTC. Performance measures were conducted to test the accuracy of different models and then compared the actual and predicted prices. Finally, the project reveals that the best accuracy rate is shown in Long Short-Term Memory model than Linear Regression model. This project used the feature close only, hence the result may differ if we tend to take various other features into considerations. Because the crypto market is volatile and influenced by social media and other external factors, data sets cannot be the only reason for forecasting. As technology advances, new data can be collected, analyzed, and practiced, resulting in better results.

## 8.2 FUTURE ENHANCEMENT

- Investigate other factors that might affect the prices of the cryptocurrency market, and can focus on the effect that social media in general and tweets in particular can have on the price and trading volume of Cryptocurrencies.
- To work on a better User Interface so that people can access these data easily and effortlessly.
- Implementing more algorithms to find out the best method for predicting the crypto currency price with highest possible accuracy.

# CHAPTER 9

# APPENDICES

## 9.1 SOURCE CODE:

### IMPORTING THE LIBRARIES

```
import os
import pandas as pd
import numpy as np
import math
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import MinMaxScaler
import keras
from keras.models import Sequential
from tensorflow.keras.optimizers import Adam
from keras.layers import Dense, LSTM, LeakyReLU, Dropout
from fbprophet import Prophet
```

### LOADING THE DATASET

```
datas=pd.read_csv('/content/BTC_USD data.csv')
datas=pd.read_csv('/content/LTC_USD data.csv')
datas=pd.read_csv('/content/ETH_USD data.csv')
```

### DATA EXPLORATION

```
datas.describe()
datas=datas.sort_values('Date')
datas.shape
datas.head()
datas.tail()
```

## DATA PREPROCESSING

```
required_features = ['Open', 'High', 'Low','Volume']
output_label = 'Close'
min_max_scaler = MinMaxScaler() #this by default returns the value between 0-10.
norm_data = min_max_scaler.fit_transform(price.values) #fitting these values
norm_data
print(f'real: {price.values[0]}, normalized:{norm_data[0]}')
print(f'real: {price.values[500]}, normalized:{norm_data[500]}')
print(f'real: {price.values[900]}, normalized:{norm_data[900]}')
```

## VISUALIZATION

```
#Creating a dataframe for Closing Price (USD) column as it's our target variable to predict.
price= datas[['Close']]
plt.figure(figsize = (15,9))
plt.plot(price)
plt.xticks (range(0, datas.shape[0],50), datas[ 'Date'].loc[::50], rotation=45)
plt.title("Bitcoin Price", fontsize=18, fontweight='bold')
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price (USD)', fontsize=18)
plt.show()
```

## SPLITTING THE DATA INTO TRAIN, TEST FOR LINEAR REGRESSION

```
x_train, x_test, y_train, y_test = train_test_split(
datas[required_features],
datas[output_label],
test_size = 0.3
)
```

## LINEAR REGRESSION MODEL BUILDING AND EVALUATION

```
model = LinearRegression()
model.fit(x_train, y_train)
future_set = datas.shift(periods=40).tail(40)
```

## LINEAR REGRESSION PREDICTION

```
prediction = model.predict(future_set[required_features])
plt.figure(figsize = (12, 7))
plt.plot(datas["Date"][-400:-60], datas["Close"][-400:-60], color='goldenrod', lw=2)
plt.plot(future_set["Date"], prediction, color='deeppink', lw=2)
plt.title("Bitcoin Price Prediction using Linear Regression", size=25)
plt.xlabel("Timestamp", size=20)
plt.ylabel("$ Price", size=20)
```

## PROPHET MODEL BUILDING AND EVALUATION

```
%matplotlib inline
color = sns.color_palette()
df = df.iloc[::-1]
df.plot()
ml_df = df.reset_index().rename(columns={'Date':'ds', 'Close':'y'})
ml_df['y'] = np.log(ml_df['y'])
model = Prophet()
model.fit(ml_df);
future = model.make_future_dataframe(periods=365)
forecast = model.predict(future)
two_years = forecast.set_index('ds').join(df)
two_years = two_years[['Close', 'yhat', 'yhat_upper', 'yhat_lower' ]].dropna().tail(800)
two_years['yhat']=np.exp(two_years.yhat)
two_years['yhat_upper']=np.exp(two_years.yhat_upper)
two_years['yhat_lower']=np.exp(two_years.yhat_lower)
two_years[['Close', 'yhat']].plot()
two_years_AE = (two_years.yhat - two_years.Close)
two_years_AE.describe()
mean_squared_error(two_years.Close, two_years.yhat)
mean_absolute_error(two_years.Close, two_years.yhat)
```

## PROPHET PREDICTION

```
prediction = model.predict(future_set[required_features])
```

```
plt.figure(figsize = (12, 7))
plt.plot(datas["Timestamp"][-400:-60], datas["Weighted_price"][-400:-
60], color='goldenrod', lw=2)
plt.plot(future_set["Timestamp"], prediction, color='deeppink', lw=2)
plt.title("Ether Price Prediction using Linear Regression", size=25)
plt.xlabel("Timestamp", size=20)
plt.ylabel("$ Price", size=20)
```

## SPLITTING THE DATA FOR LSTM

```
def univariate_data(dataset, start_index, end_index, history_size, target_size):
  data = []
  labels = []
  start_index= start_index + history_size
  if end_index is None:
    end_index= len(dataset) - target_size
  for i in range(start_index, end_index):
    indices = range(i-history_size, i)
    #Reshape data from (history size,) to (history_size, 1)
    data.append(np.reshape(dataset[indices], (history_size, 1)))
    labels.append(dataset[i+target_size])
  return np.array(data), np.array(labels)
past_history=5 #using 5 days of data to learn to predict the next point in the time series 'futur
e_target'
future_target=0
TRAIN_SPLIT=int(len(norm_data)*0.8)
x_train,y_train= univariate_data(norm_data,0,TRAIN_SPLIT,past_history,future_target)
x_test,y_test= univariate_data(norm_data,0,TRAIN_SPLIT,past_history,future_target)
y_test.shape
```

## LSTM MODEL BUILDING AND EVALUATION

```
num_units= 64 #Number of neurons
learning_rate = 0.0001
activation_function = 'sigmoid'
adam = Adam(lr=learning_rate)
```

```
loss_function = 'mse'
batch_size = 5
num_epochs = 250
#Initialize the RNN
model= Sequential()
#In Keras we can simply stack multiple layers on top of each other, for this we need to initialize the model as Sequential()
model.add(LSTM(units= num_units, activation =activation_function, input_shape=(None, 1)))
model.add(LeakyReLU(alpha =0.5))
model.add(Dropout(0.1))
#This layer will help to prevent overfitting by ignoring randomly selected neurons during training, and hence reduces the sensitivity to the specific weights of individual neurons
model.add(Dense(units = 1))
#fully conneceted layer
#Compiling the RNN
model.compile(optimizer=adam, loss=loss_function)
model.summary()
```

**TRAINING THE LSTM MODEL**

```
#using training set to train the model
history= model.fit(
  x_train,
  y_train,
  validation_split =0.1,
  batch_size=batch_size,
  epochs=num_epochs,
  shuffle =False
)
loss=history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(loss))
plt.figure(figsize=(15,9))
plt.plot(epochs,loss,'b',label='training loss')
plt.plot(epochs,val_loss,'r',label='validation loss')
plt.title("traing and validation loss")
plt.legend()
plt.show
a=min_max_scaler.inverse_transform(y_test)
b=min_max_scaler.inverse_transform(model.predict(x_test))
```

**LSTM PREDICTION**

```
original=pd.DataFrame(a)
predictions = pd.DataFrame(b)
sns.set(rc={'figure.figsize':(11.7+2,8.27+2)})
ax = sns.lineplot (x = original.index, y =original[0], label= "Test Data", color ='royalblue')
ax = sns.lineplot(x = predictions.index, y =predictions[0], label="Prediction", color ='tomato')
ax.set_title('Bitcoin price', size =14, fontweight='bold')
ax.set_xlabel("Days", size = 14)
ax.set_ylabel("Cost (USD)", size = 14)
ax.set_xticklabels('',size=10)
from sklearn.metrics import mean_squared_error
from math import sqrt
x = np.array(original[0]).ravel()
y = np.array(predictions[0]).ravel()
rmse = sqrt(mean_squared_error(x, y))
print('RMSE: %.4f' % (rmse*0.0001))
```

## 9.2 SCREENSHOTS

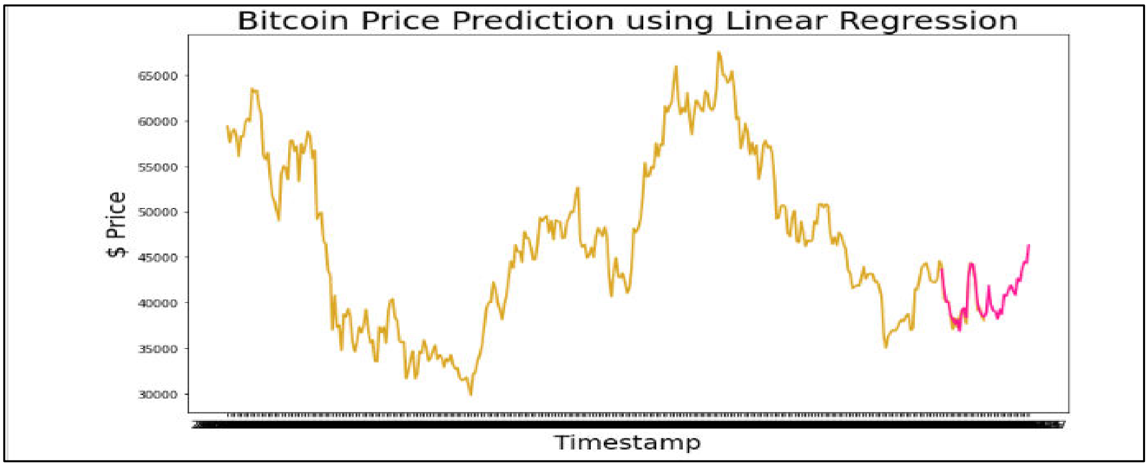


**Figure 9.2.1 Bitcoin price prediction using Linear regression**

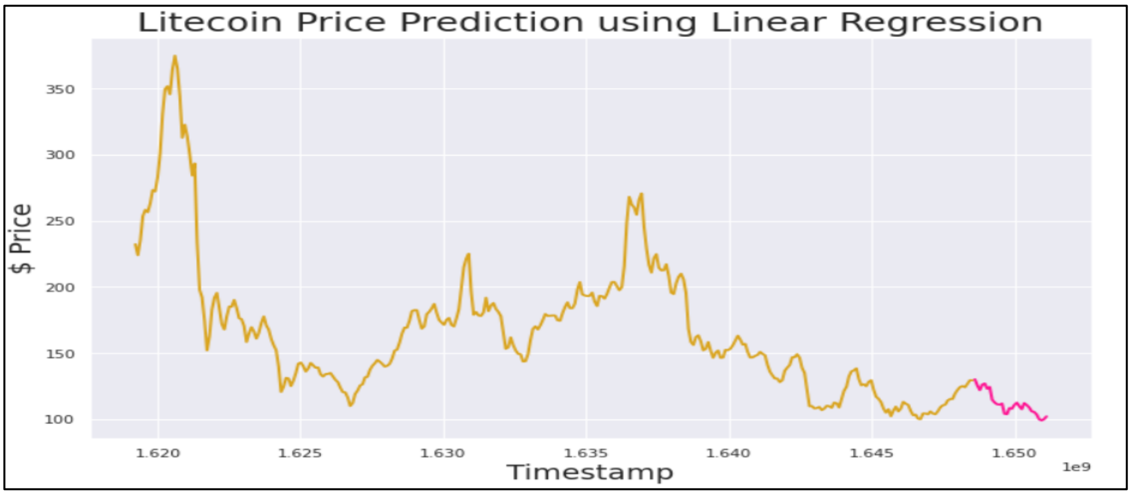


**Figure 9.2.2 Litecoin price prediction using Linear regression**

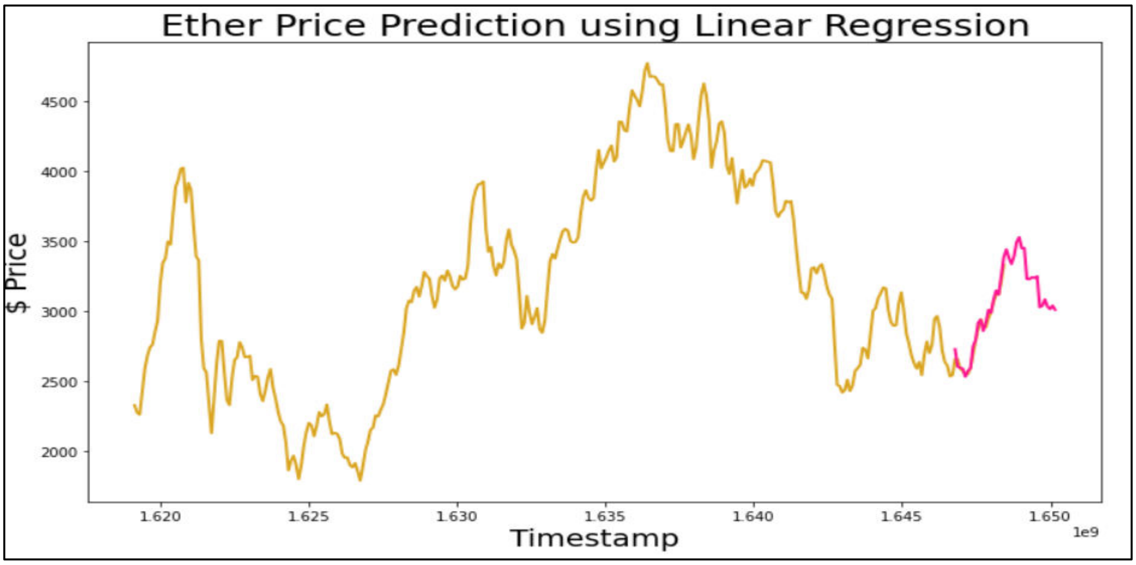


**Figure 9.2.3 Ether price prediction using Linear regression**

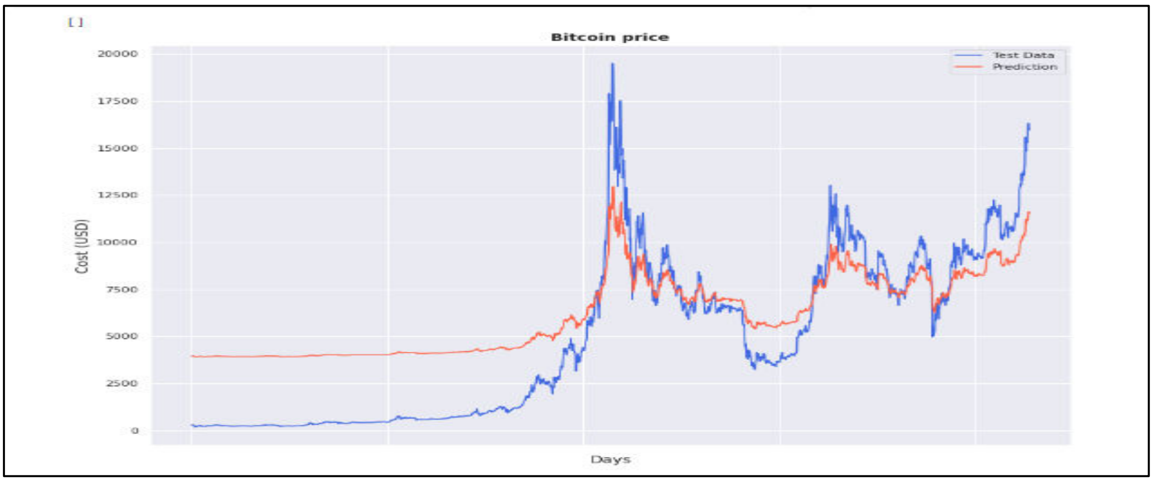


**Figure 9.2.4 Bitcoin price prediction using LSTM**

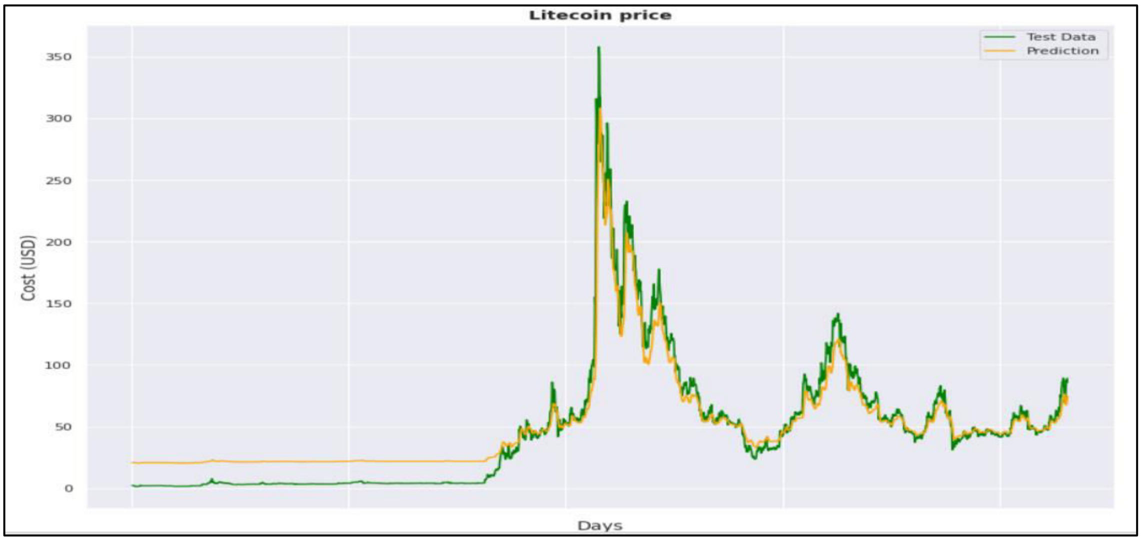


**Figure 9.2.5 Litecoin price prediction using LSTM**

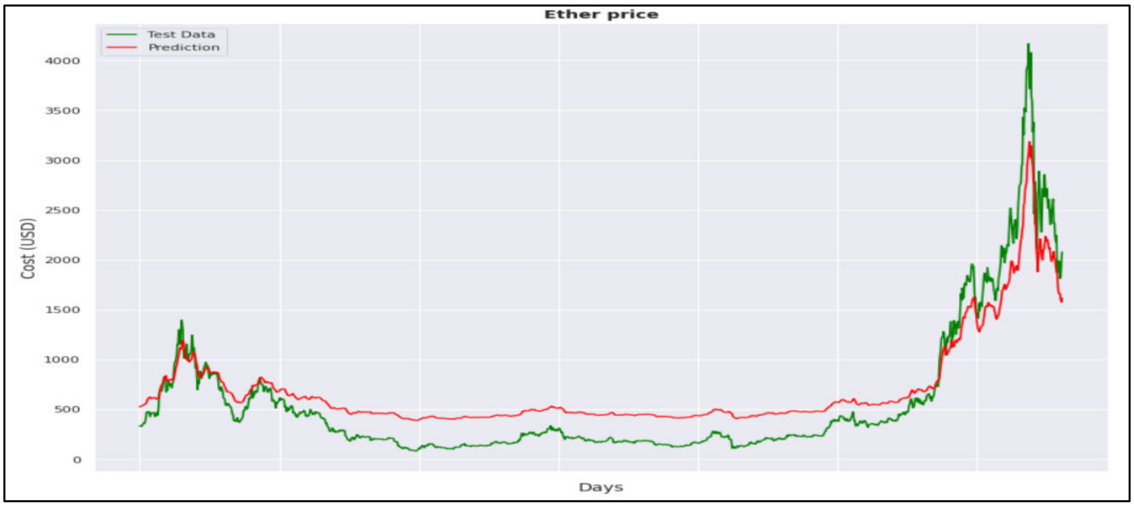


**Figure 9.2.6 Ether price prediction using LSTM**

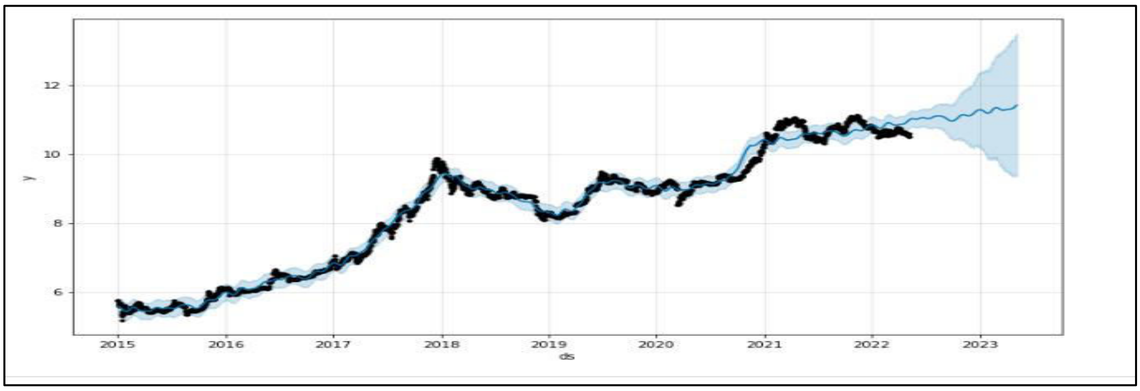


**Figure 9.2.7 Bitcoin price prediction using Prophet**

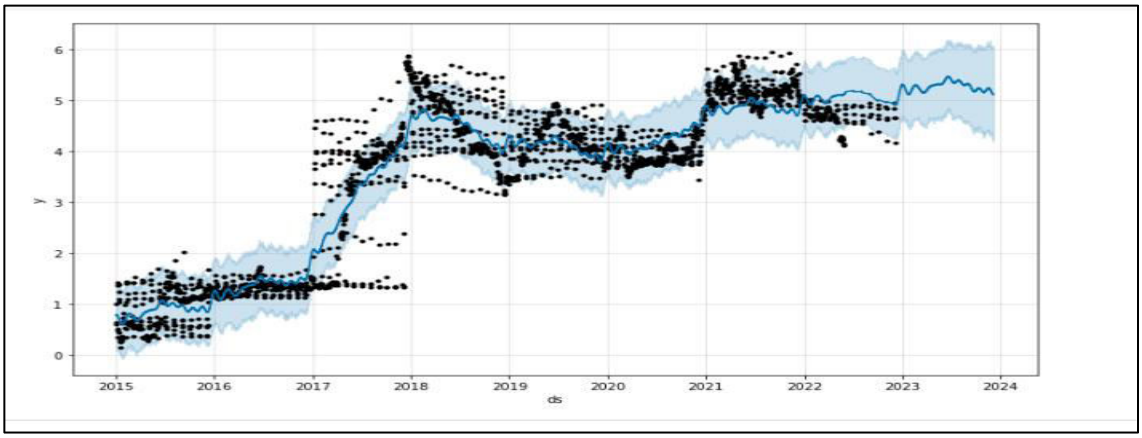


**Figure 9.2.8 Litecoin price prediction using Prophet**

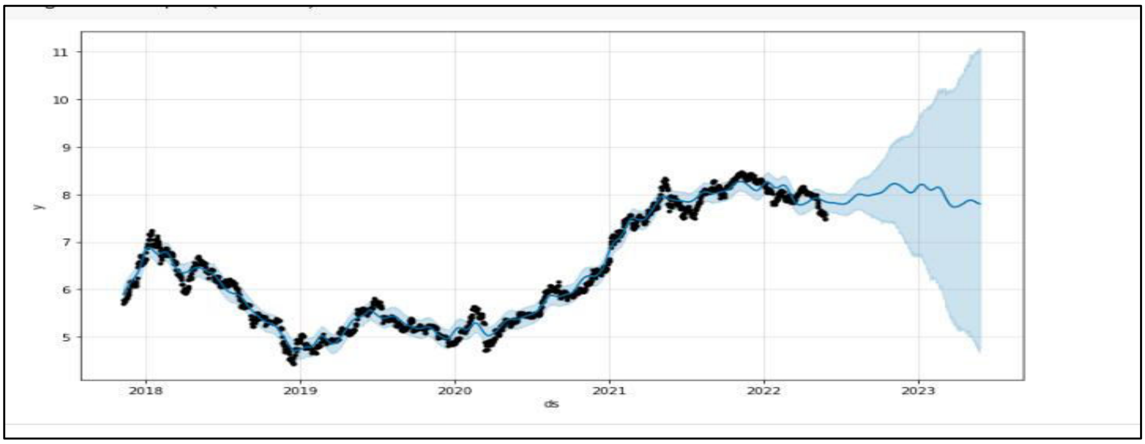


**Figure 9.2.9 Ether price prediction using Prophet**

# CHAPTER 10

# REFERENCES

## JOURNAL ARTICLE:

1. Akyildirim E, Goncu A, Sensoy A. Prediction of cryptocurrency returns using machine learning. Annals of Operations Research. 2021;297(1-2):3–36.
2. An optimized support vector machine (SVM) based on particle swarm optimization (PSO) for cryptocurrency forecasting. Procedia Computer Science. 2019;163:427–433.
3. Chen, Z.; Li, C.; Sun, W. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. J. Comput. Appl. Math. 2019, 365, 112395.
4. Chowdhury R, Rahman MA, Rahman MS, Mahdy MRC. An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning. Physica A: Statistical Mechanics and its Applications.
5. Greaves A, Au B. Using the bitcoin transaction graph to predict the price of bitcoin. 2015.
6. Hitam, N.A.; Ismail, A.R. Comparative Performance of Machine Learning Algorithms for Cryptocurrency Forecasting. Indones. J. Electr. Eng. Comput. Sci. 2018, 11, 1121–1128.
7. Karasu. S, Altan. A, Sarac. Z, Hacioglu. R, Prediction of Bitcoin prices with machine learning methods using time series data. In Proceedings of the 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018.
8. Mittal R, Arora S, Bhatia MP. Automated cryptocurrencies prices prediction using machine learning. ICTACT Journal on Soft Computing. 2018;8(4):1758– 1761.
9. Saad, M.; Mohaisen, A. Towards characterizing blockchain-based cryptocurrencies for highly-accurate predictions. In Proceedings of the IEEE INFOCOM—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018.
10. Valencia F, Gómez-Espinosa A, Valdés-Aguirre B. Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning. Entropy. 2019;21(6):589.

## BOOKS:

1. Amjad. M and Shah. D, "Trading Bitcoin and Online Time Series Prediction," in NIPS 2016 Time Series Workshop, 2017.

2. Chen. R and Lazer. M, "Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement," Stanford Computer Science, no. 229, 2011, p. 15.
3. Dixon. M, Klabjan. D and Bang. J. H, “Classification-based financial markets prediction using deep neural networks”, ArXiv,2017
4. Garcia. D and Schweitzer. F, "Social signals and algorithmic trading of Bitcoin", Royal Society Open Science, vol. 2, no. 9, 2015.