

ETHEREUM PRICE PREDICTION SYSTEM

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

SADAM DIVYAMANI (Reg. No. 37110654)

SADASIVUNI NIKHITA (Reg. No. 37110655)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

MARCH – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY) Accredited with —All grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **SADAM DIVYAMANI (Reg. No. 37110654)** and **SADASIVUNI NIKHITA (Reg. No. 37110655)** who carried out the project entitled —**ETHEREUM PRICE PREDICTION SYSTEM** under my supervision from **August 2020** to **March 2021**.

Internal Guide

Dr.N.Srinivasan,M.E.,Ph.d.,

Head of the Department

Submitted for Viva voice Examination held on

Internal Examiner

External Examiner

DECLARATION

I **SADAM DIVYAMANI** hereby declare that the Project Report entitled **—ETHEREUM PRICE PREDICTION SYSTEM**” is done by me under the guidance of **Dr.N.Srinivasan,M.E.,Ph.d.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide, **Dr.N.Srinivasan,M.E.,Ph.d., Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Cryptocurrency is a form of payment that can be exchanged online for goods and services. More than 6,700 different cryptocurrencies are traded publicly, according to CoinMarketCap.com, a market research website. Largest trading cryptocurrencies are Bitcoin, Ethereum, tether etc.,

Vitalik Buterin, a Russian-Canadian entrepreneur, founded Ethereum in 2015. It's a blockchain protocol that enables people all over the world to send and receive money without the need for a third-party intermediary. This project predicts the fluctuations of the ethereum cryptocurrency and aims to connect the ethereum dataset, machine learning in a single project. We used To model the token price time-series data, we used LSTM neural networks. Finally, some closing thoughts and possible study directions are offered. This paper proposes a methodology and a detailed approach to ethereum price fluctuations.

KEYWORDS: Cryptocurrency, Ethereum, LSTM, Deep learning, Time-series

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 Digital Cash	1
	1.2 Cryptocurrency	1
	1.3 History Of Cryptocurrency	1
	1.4 Types of Cryptocurrency	2
	1.5 Ethereum Price Prediction	5
2	LITERATURE SURVEY	6
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	8
	3.1 Objective	8
	3.2 Scope Of the Project	8
4	EXPERIMENTAL OR MATERIALS,METHODS AND ALGORITHM	10
	4.1 System Requirments	10
	4.2 System Architecture	10
	4.3 Algorithm Used	11
	4.3.1 Long short-Term Memory	12
	4.4 Methodology	13
	4.4.1 Data Preprocessing	13
	4.4.2 Data Visualization	15
	4.4.3 Data Normalization	17
	4.4.3.1 Autocorrelation and Partial autocorrelation	17
	4.4.3.2 Statistics and P-Value	19
	4.4.4 Building LSTM Model	21

5	RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS	23
	5.1 Result	23
	5.2 Discussion and Performance Analysis	24
6	SUMMARY AND CONCLUSION	26
	6.1 Summary	26
	6.2 Conclusion	26
	REFERENCES	27
	APPENDIX	28
	A. PAPER ACCEPTANCE MAIL	
	B. PLAGARISM REPORT	
	C. JOURNAL PAPER	
	D. SOURCE CODE	

LIST OF FIGURES

Figure no	Figure name	Page no
1.1	Ethereum Logo	5
4.1	System Architecture	11
4.2	LSTM neural network	13
4.3	Importing packages and dataset	14
4.4	Exploring Dataset	15
4.5	Price By year wise	15
4.6	Year wise report on %change	15
4.7	Monthly total for stock values	16
4.8	Volume by year	16
4.9	Autocorrelation and partial Autocorrelation	18
4.10	P-Value	20
4.11	Dividing Dataset	21
4.12	Building and fitting the LSTM model	21
5.1	Making Predictions	22
5.2	Comparing result	23

LIST OF ABBREVIATIONS

LSTM – Long Short-Term Memory

ETH - Ethereum

GRU - Gated Recurrent Unit

RNN – Recurrent Neural Network

ACF – Autocorrelation Function

Chapter - 1

INTRODUCTION

1.1 Digital Cash

An ideal payment system would be one in which monetary value could be transferred electronically via cash data files. Such cash data files retain the advantages of physical cash but would be able to circulate freely on electronic networks. A data file of this type could be sent via email or social media channels. A specific feature of electronic data is that it can be copied any number of times at negligible cost. This feature is highly undesirable for money. If cash data files can be copied and the duplicates used as currency, they cannot serve as a payment instrument. This problem is termed the —double spending problem.¶

1.2 Cryptocurrency

A **cryptocurrency**, **crypto currency** or **crypto** is a digital asset designed to work as a medium of exchange wherein individual coin ownership records are stored in a ledger existing in a form of computerized database using strong cryptography to secure transaction records, to control the creation of additional coins, and to verify the transfer of coin ownership. It typically does not exist in physical form (like paper money) and is typically not issued by a central authority. Cryptocurrencies typically use decentralized control as opposed to centralized digital currency and central banking systems. When a cryptocurrency is minted or created prior to issuance or issued by a single issuer, it is generally considered centralized. When implemented with decentralized control, each cryptocurrency works through distributed ledger technology, typically a blockchain, that serves as a public financial transaction

1.3 History Of Cryptocurrency

The story of these virtual coins begins with one person: **the cryptographer David Chaum**. In 1983, the American developed a cryptographic system called **eCash**. Twelve years later, he developed another system, **DigiCash** that used cryptography to make economic transactions confidential.

However, the first time the idea or term "cryptocurrency" was coined was in **1998**. That year, **Wei Dai** began to think about developing a new payment method that used a cryptographic system and whose main characteristic was decentralization.

1.4 Types of Cryptocurrency

Here's a list of cryptocurrency types—representing the most popular industry crypto projects (at present):

1. Bitcoin

Possibly the —Kleenex or —Coca Colall of all crypto, in that its name is the most recognizable and the most closely associated with the cryptocurrency system. There are currently more than 18.5 million Bitcoin tokens in circulation, against a present capped limit of 21 million.

2. Bitcoin Cash

Introduced in 2017, Bitcoin Cash is one of the most popular types of cryptocurrency on the market. Its main difference with the original Bitcoin is its block size: 8MB. Compare that to the original Bitcoin's block size of just 1MB. What that means for users—faster processing speeds.

3. Litecoin

Litecoin is increasingly used in the same breath as Bitcoin, and it functions practically the same way. It was created in 2011 by Charlie Lee, a former employee of Google. He designed it to improve on Bitcoin technology, with shorter transaction times, lower fees, more concentrated miners.

4. Ethereum

Unlike Bitcoin, Ethereum focuses not as much on digital currency as it does on decentralized applications (phone apps). You could think of Ethereum as an app store.

The platform is looking to return control of apps to its original creators, and take away that control from middlemen (like Apple, for instance). The only person who can make changes to the app would be the original creator. The token used here is called Ether, which is used as currency by app developers and users.

5. Ripple

Ripple is one type of cryptocurrency on the list, but it's not Blockchain-based . It's not meant so much for individual users as it is for larger companies and corporations, moving larger amounts of money (its coinage is known as XRP) across the globe.

It's more well-known for its digital payment protocol more than for its XRP crypto. That's because the system allows for transfer of monies in any form, be it dollars or even Bitcoin (or others). It claims to be able to handle 1,500 transactions per second (tps). Compare this with Bitcoin, which can handle 3-6 tps (not including scaling layers). Ethereum can handle 15 tps.

6. Stellar

Stellar focuses on money transfers, and its network is designed to make them faster and more efficient, even across national borders. It was designed by Ripple co-founder Jed McCaleb in 2014 and is operated by a non-profit organization called Stellar.org .

Its goal is to assist developing economies that may not have access to traditional banks and investment opportunities. It doesn't charge users or institutions for using its Stellar network, and covers operating costs by accepting tax-deductible public donations.

7. NEO

Formerly called Antshares and developed in China, NEO is very aggressively looking to become a major global crypto player. Its focus is smart contracts (digital contracts) that allow users to create and execute agreements without the use of an intermediary.

It's going after its main competition, Ethereum, but NEO lead developer Erik Zhang mentioned on a Reddit AMA that NEO has three distinct advantages—better architecture, more developer-friendly smart contracts, and digital identity and digital assets for easier integration into the real world.

Ethereum, on the other hand, uses its own programming languages that developers must first learn before creating smart contracts on its platform.

8. Cardano

Cardano aka ADA is used to send and receive digital funds. It claims to be a more balanced and sustainable ecosystem for cryptocurrencies, and the only coin with a —scientific philosophy and research-driven approach. It

That means that it undergoes especially rigorous reviews by scientists and programmers. It was founded by Charles Hoskinson, who is also the co-founder of Ethereum.

9. IOTA

Launched in 2016, IOTA stands for Internet of Things Application. Unlike most other Blockchain technologies, it doesn't actually work with a block and chain; it works with smart devices on the Internet of Things (IoT).

All you need to do to use it is to verify two other previous transactions on the IOTA ledger, which is called the Directed Acyclic Graph (DAG), but IOTA creators call it The Tangle.

According to Coin Central , this means the devices need to be able to purchase more electricity, bandwidth, storage, or data when they need them, and sell those resources when they don't need them.

1.5 Ethereum Price Prediction

Ethereum is vastly different from other cryptocurrencies such as Bitcoin, Litecoin, or even Ripple. Ethereum is a blockchain protocol for smart contracts, and Ether, also called ETH, is the native cryptocurrency that powers the Ethereum blockchain.



Fig 1.1 : Ethereum logo

Ethereum runs smart contracts on what it calls the Ethereum Virtual Machine, which is essentially a supercomputer built on the blockchain. Because of its unique structure and being the first of its kind, Ethereum price predictions can be extremely high, with so many investors and industry experts believing in Ethereum growth in the long-term.

While anyone can see that Ethereum as an investment holds much promise and will surely reward investors with all the support and potential ETH has, it's not always clear exactly what prices that Ethereum may reach and when. Technical analysis from the industry's best analysts has been gathered to help understand if some of the price predictions from experts match what the charts are saying.

Chapter – 2

Literature survey

In [1] According to the author, machine learning and sentimental analysis, which were previously used for Bitcoin, can be used to predict the course of price changes in the evolving cryptocurrency sector. Using Twitter data, market data, or both, we analyzed and compared the performance of three prediction models: MLPs, SVMs, and RFs for Bitcoin, Ethereum, Ripple, and Litecoin.

[2] The observations of Gerlof Bremmer studies have found such set of abilities for Litecoin or even Ripple, and not just for Ethereum, as per their review. Conversely, this survey had shown that linkage could also be used to assess today's price or sales position with those specificity also for similar cryptocurrencies. However, models used do not have a high enough degree of accuracy to be used for trading. Change in prices of cryptocurrency is a daily process.

[3] GUSTAV SEGERSTEDT et..al Says that the accuracy of predictions can be improved by decreasing the sampling period of data. However, there doesn't seem to be any clear trend on how changing the neurons per LSTM layer affects prediction accuracy.

[4] According to Jaysing Bhosale et al., they compare three cryptocurrencies—Bitcoin, Ethereum, and Litecoin—in terms of their volatility and stability in recent times, as well as their patterns.

[5] Rituparna Ghosh et..al did research on which is a batter cryptocurrency among

Bitcoin and Ethereum They compare the two blockchain technologies and build a portfolio of Bitcoin and Ether that will deliver the highest return with the lowest risk over a five-year investment of a million dollars.

- [6] Yang Li, Zibin Zheng et.al, Indicated in their research as Assess the situation of Crypto market dispersion prediction as in sector of virtual currency. The results show that the designed features are the best fit for our presented problem statement and our proposed LSTM model which achieves the state of superior performance among every baseline. Our study may pave the way towards a real trading environment for the investors within the future.

Chapter - 3

Aim and Scope of the present investigation

3.1 objective

Ethereum is a cryptocurrency protocol that's vastly different than Bitcoin, Litecoin, or other payment technologies, giving it additional value and potential. This guide helps understand that potential value through price predictions from experts and analysts alike.

Ethereum runs smart contracts on what it calls the Ethereum Virtual Machine, which is essentially a supercomputer built on the blockchain. Because of its unique structure and being the first of its kind, Ethereum price predictions can be extremely high, with so many investors and industry experts believing in Ethereum growth in the long-term.

Ethereum projected growth as a network is also high, as the blockchain is not only used for smart contracts and transactions, but actual crypto assets are often built on Ethereum's blockchain, giving it added value as speculation around the future of the crypto market is positive.

Because there is so much positivity surrounding cryptocurrencies and Ethereum potential is so high, many are left wondering —will Ethereum rise like Bitcoin,¹¹ and —how high can the price of Ethereum go?¹² This Ethereum price prediction guide will help paint a clear picture of estimated growth and the projected value of Ethereum over the years.

3.2 Scope of the Project

While anyone can see that Ethereum as an investment holds much promise and will surely reward investors with all the support and potential ETH has, it's not always clear exactly what prices that Ethereum may reach and when. Technical analysis from the industry's best analysts has been gathered to help understand if some of the price predictions from experts match what the charts are saying.

Ethereum has already claimed a new all-time high in 2021, and sky is now the limit it

seems. There is still risk of a double top pattern if a severe correction happens at current levels, but a breakout higher is due any day. Most analysts are expecting Ethereum price to double now that a new record is set, however, the height of the previous base structure is targeting a 1300% rise from the point of breakout. That would take Ethereum price to as high as \$25,000 per ETH.

Continuing to follow along with Bitcoin's trajectory isn't all positive for Ethereum. Continuing the roadmap the fractal presents would take Ethereum to another bear market come 2024 and begin to break out starting in 2025 into a new bull market once again.

By using the historical data, we can predict the price of ethereum cryptocurrency.

Chapter – 4

Experimental or materials, methods and algorithms used

4.1 System Requirments

- Hardware
 - 4GB RAM
 - 1TB Hard disk
- Software Requirements
 - Google Colab
- Construction
 - Bidirectional LSTM,
 - python libraries like pandas matplotlib.

4.2 System Architecture

Figure 4.1., shows the system architecture of ethereum price prediction system

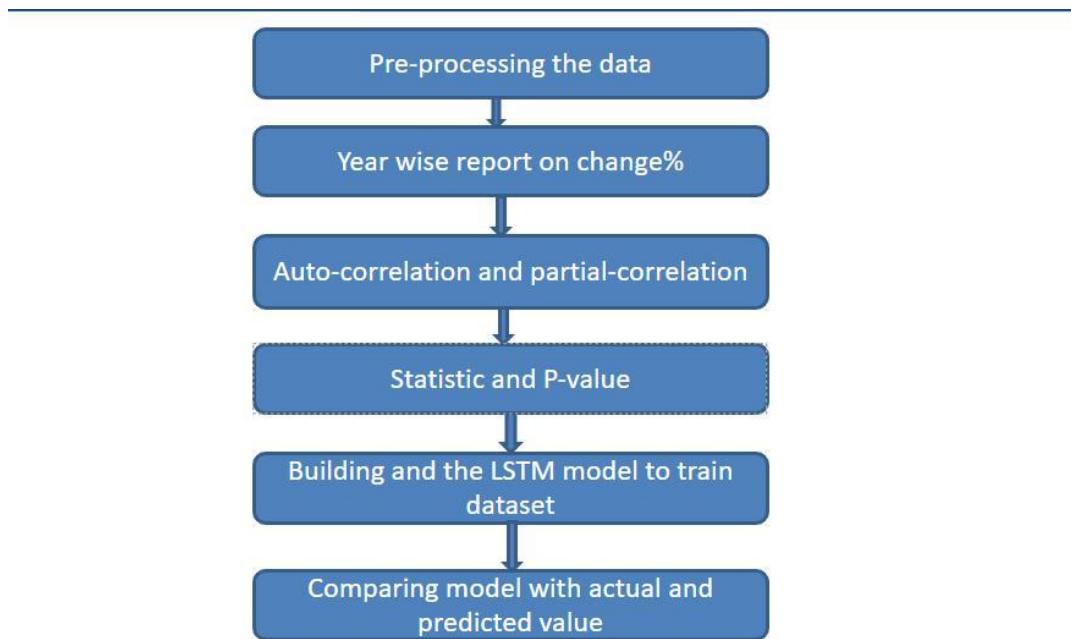


Figure 4.1 : System architecture

4.3 Algorithm Used

The main objective of this project is to develop a prediction model which predicts Ether crypto currency price. Ether is the second widely used crypto currency with \$60 billion market capitalization as on date. Price prediction of stocks from stock market has been well studied both in industry and academia but crypto currency price prediction is just gaining momentum. Specifically Ether price prediction is inadequately studied using Gated Recurrent Unit (GRU). GRU is a type of neural network which is mostly used for time series model and sequence models. Long Short-Term Memory (LSTM) is a predecessor of GRU which has the same accuracy but not so efficient to train. For this reason GRU architecture was used in this work and the results show the capability to handle volatile price index and remembering information and dependency from the past. Prediction from GRU model has good prediction accuracy in the test set. In this project data was scraped from a crypto currency website called CoinMarketCap and split into training set and test set. Training data was used to train the model and test data was used to check the accuracy of the model's prediction.

4.3.1 Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture^[1] used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition,^[2] speech recognition^{[3][4]} and anomaly detection in network traffic or IDSs (intrusion detection systems).

A common LSTM unit is composed of a **cell**, an **input gate**, an **output gate** and a **forget gate**. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with

the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications

In figure 4.2 shows the structure of LSTM neural network

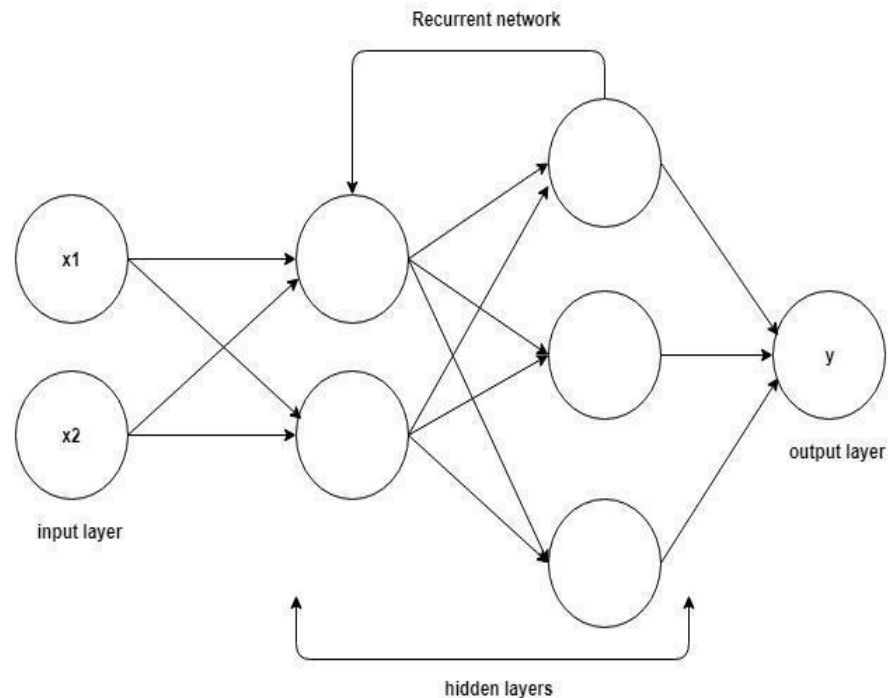


Figure 4.2 : LSTM neural network

4.4 Methodology

The process done in the ethereum price prediction system is divided into modules.

There are:

- Data Preprocessing
- Data visualization
- Data normalization
- Building Model
- Result Analysis

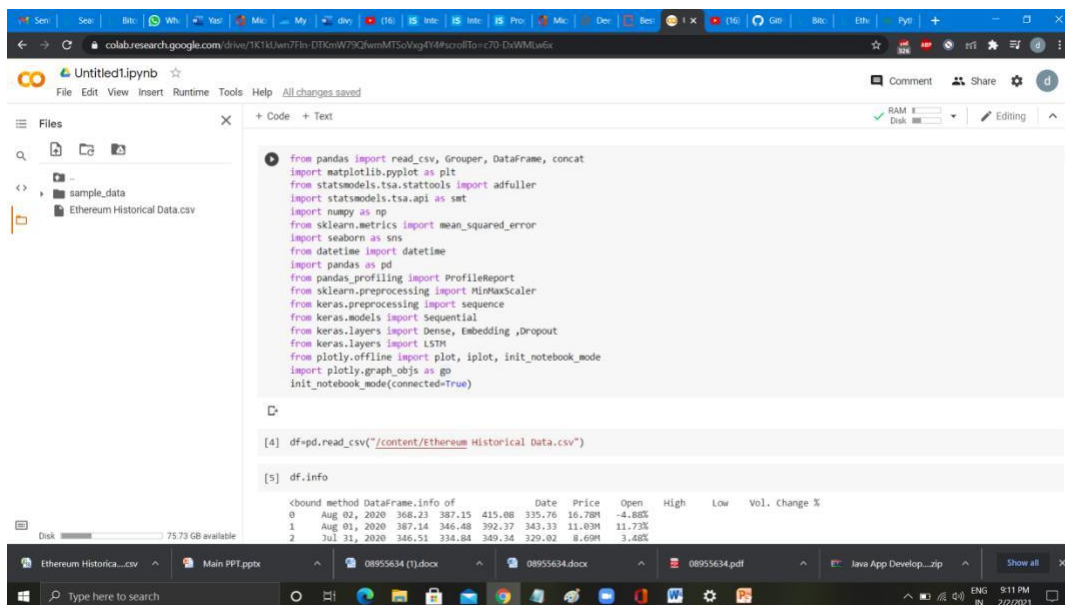
4.4.1 Data preprocessing

Import the important libraries for the ethereum price prediction such as pandas, numpy, Visualization libraries like seaborn, matplotlib, plotly and deep learning libraries like kears from tensorflow.

Collected the dataset from kaggle which is historical dataset for ethereum price prediction consist of 1600 records contains seven columns of price, date, high, low, open, volume, %change in price.

In figure 4.3 we can see that packages are imported and dataset is also readed to the notebook.

In figure 4.4, we can see that, we are exploring through the dataset to find the missing values and treat them accordingly.



```
from pandas import read_csv, Grouper, DataFrame, concat
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller
import statsmodels.tsa.api as smt
import numpy as np
from sklearn.metrics import mean_squared_error
import seaborn as sns
from datetime import datetime
import pandas as pd
from pandas_profiling import ProfileReport
from sklearn.preprocessing import MinMaxScaler
from keras.preprocessing import sequence
from keras.models import Sequential
from keras.layers import Dense, Embedding, Dropout
from keras.layers import LSTM
from plotly.offline import plot, iplot, init_notebook_mode
import plotly.graph_objs as go
init_notebook_mode(connected=True)

[4] df=pd.read_csv("/content/Ethereum Historical Data.csv")

[5] df.info
```

	Date	Price	Open	High	Low	Vol. Change %
0	Aug 01, 2020	368.23	387.15	415.08	335.76	16.78%
1	Aug 01, 2020	387.14	346.48	392.37	343.33	11.03%
2	Jul 31, 2020	346.51	334.84	349.34	329.02	8.69%

Figure 4.3 : Importing packages and dataset

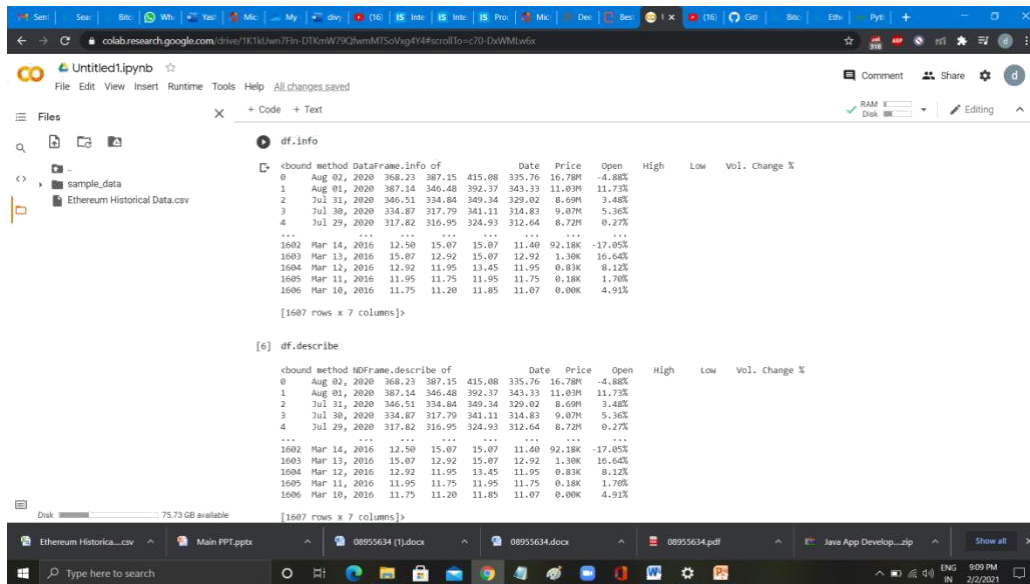


Figure 4.4 : Exploring Dataset

4.4.2 Data Visualization

Data visualization is used to explore the data visually to find the patterns between the variables in the dataset.

In figure 4.5, we can see the graph plot between price and year to find the the yeat when the highest price for ethereum is recorded.

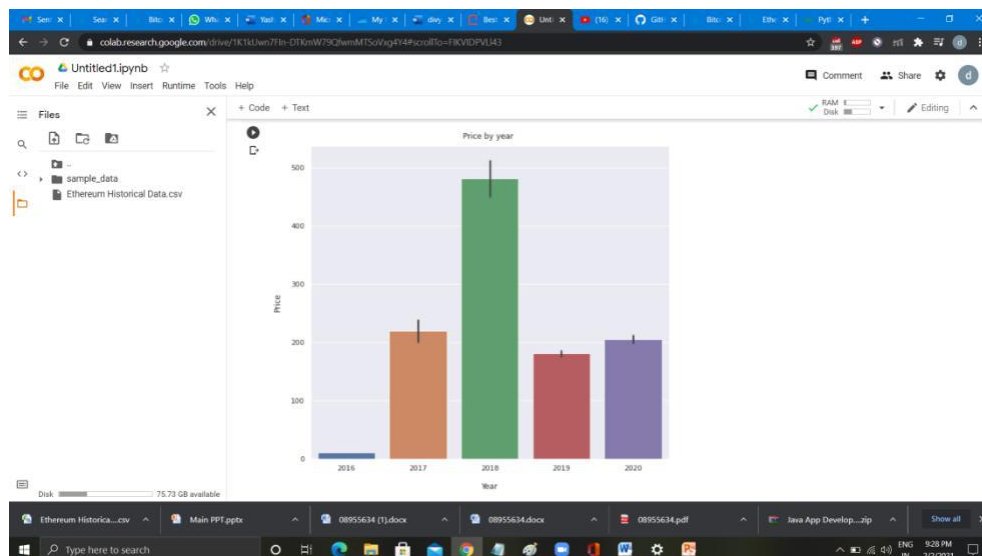


Figure 4.5 : Price by year wise

In below figure 4.6 shows the graph between dates and change in % to show the year wise report on percentage change in prices.

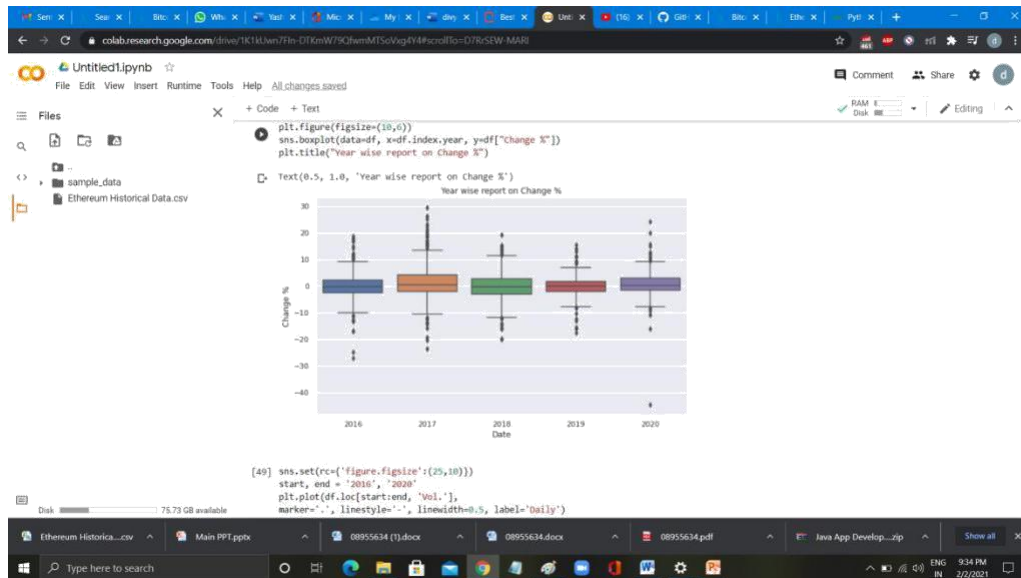


Figure 4.6 : year wise report on %change

In below figure 4.7 shows the four sub graphs of showing monthly total for stock values.

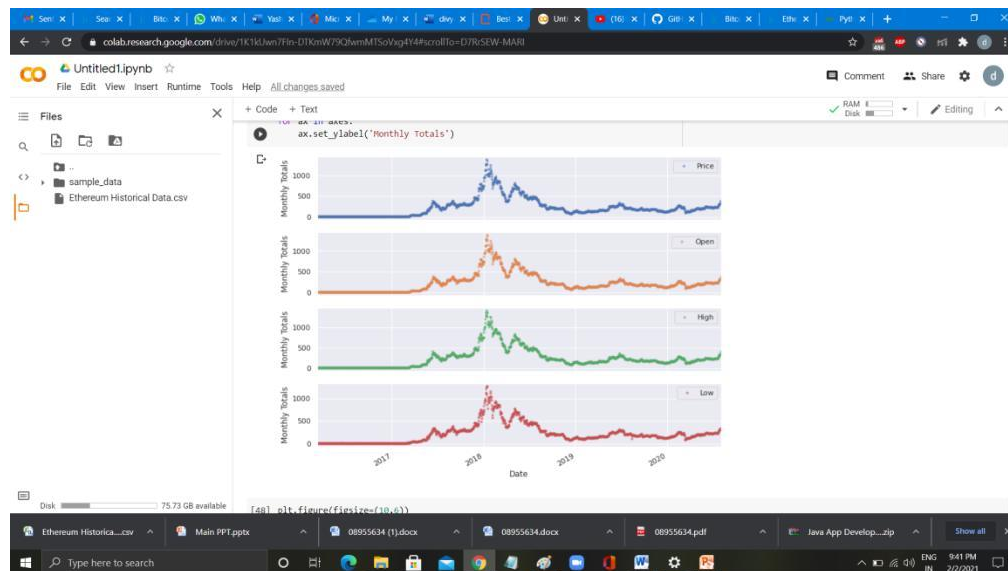


Figure 4.7 : Monthly total for stock values

In the below figure 4.8 shows the graph between date and volume which shows the number of transactions are happening in each day in a year.

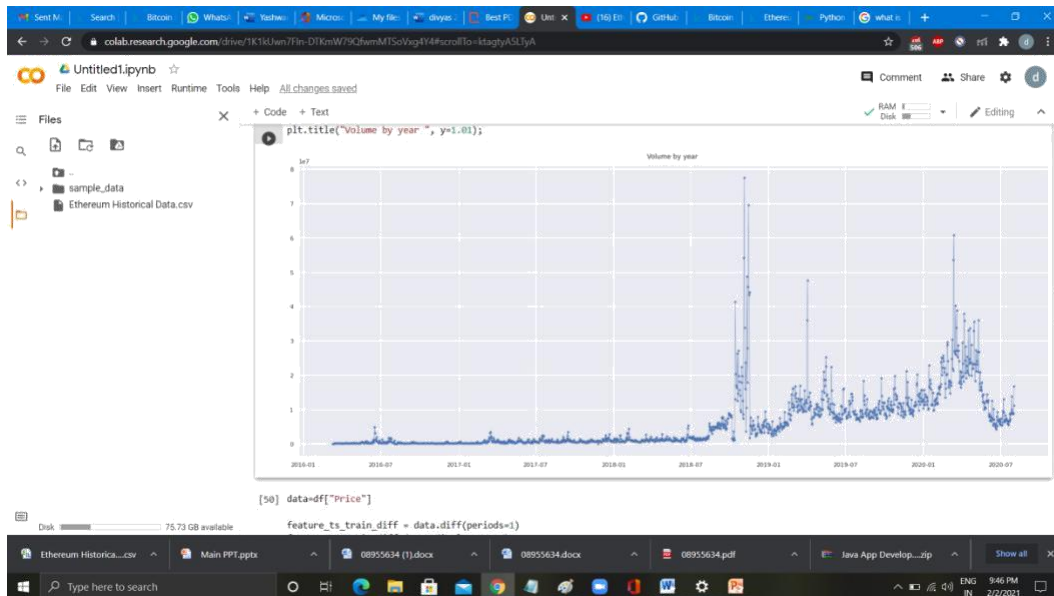


Figure 4.8 : Volume by Year

4.3 Data Normalization

Data normalization is also called as data standardization, Which is used to transform the data to fit into the model to train and test. Before doing the standardization, need to calculate statistics and P- value and graphs for autocorrelation and partial autocorrelation.

4.3.1 Auto-correlation and Partial-correlation

Autocorrelation and partial autocorrelation plots are heavily used in time series analysis and forecasting.

These are plots that graphically summarize the strength of a relationship with an observation in a time series with observations at prior time steps.

The Pearson's correlation coefficient is a number between -1 and 1 that describes a negative or positive correlation respectively. A value of zero indicates no correlation.

We can calculate the correlation for time series observations with observations with previous time steps, called lags. Because the correlation of the time series observations is calculated with values of the same series at previous times, this is called a serial correlation, or an autocorrelation.

A plot of the autocorrelation of a time series by lag is called the **AutoCorrelation Function**, or the acronym **ACF**. This plot is sometimes called a correlogram or an autocorrelation plot.

A partial autocorrelation is a summary of the relationship between an observation in a time series with observations at prior time steps with the relationships of intervening observations removed.

The autocorrelation for an observation and an observation at a prior time step is comprised of both the direct correlation and indirect correlations. These indirect correlations are a linear function of the correlation of the observation, with observations at intervening time steps.

It is these indirect correlations that the partial autocorrelation function seeks to remove. Without going into the math, this is the intuition for the partial autocorrelation.

In below figure 4.9 shows the Auto-correlation and Partial-correlation

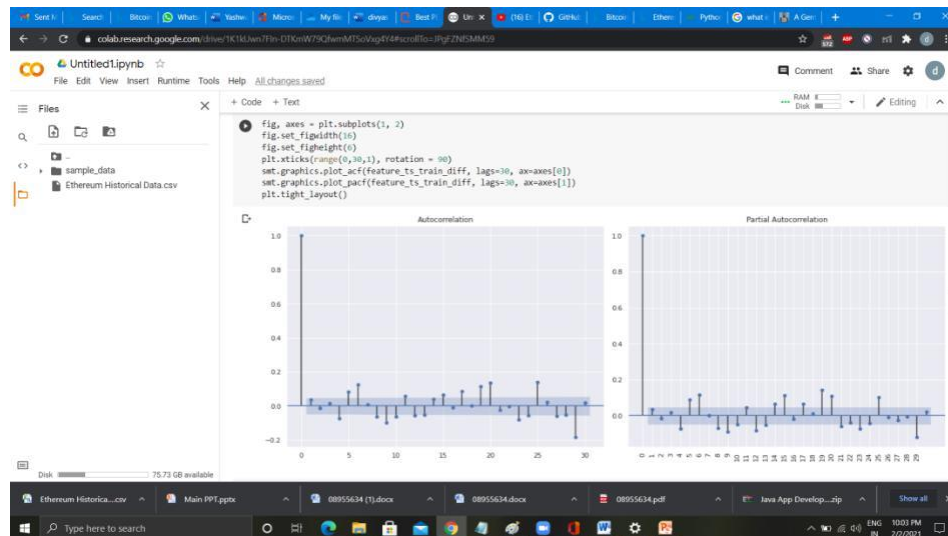


Figure 4.9 : Auto-correlation and Partial-correlation

4.3.2 Statistics and P-value

The null hypothesis states that there is no relationship between the two variables

being studied (one variable does not affect the other). It states the results are due to chance and are not significant in terms of supporting the idea being investigated. Thus, the null hypothesis assumes that whatever you are trying to prove did not happen.

The alternative hypothesis is the one you would believe if the null hypothesis is concluded to be untrue.

The alternative hypothesis states that the independent variable did affect the dependent variable, and the results are significant in terms of supporting the theory being investigated (i.e. not due to chance).

The level of statistical significance is often expressed as a p -value between 0 and 1. The smaller the p -value, the stronger the evidence that you should reject the null hypothesis.

- A p -value less than 0.05 (typically ≤ 0.05) is statistically significant. It indicates strong evidence against the null hypothesis, as there is less than a 5% probability the null is correct (and the results are random). Therefore, we reject the null hypothesis, and accept the alternative hypothesis.

However, this does not mean that there is a 95% probability that the research hypothesis is true. The p -value is conditional upon the null hypothesis being true is unrelated to the truth or falsity of the research hypothesis.

- A p -value higher than 0.05 (> 0.05) is not statistically significant and indicates strong evidence for the null hypothesis. This means we retain the null hypothesis and reject the alternative hypothesis. You should note that you cannot accept the null hypothesis, we can only reject the null or fail to reject it.

A statistically significant result cannot prove that a research hypothesis is correct (as this implies 100% certainty).

Instead, we may state our results —provide support for or —give evidence for our research hypothesis (as there is still a slight probability that the results

occurred by chance and the null hypothesis was correct – e.g. less than 5%).

In figure 4.10 shows the indicator of p-value

Statistics -2.4447651910986785

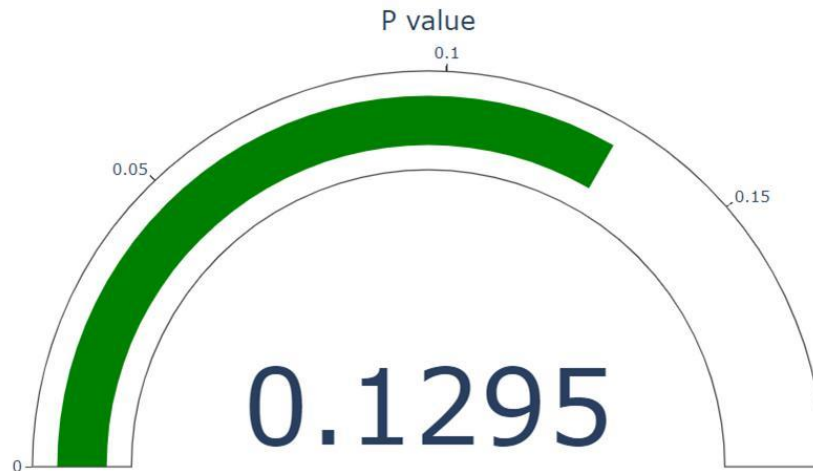
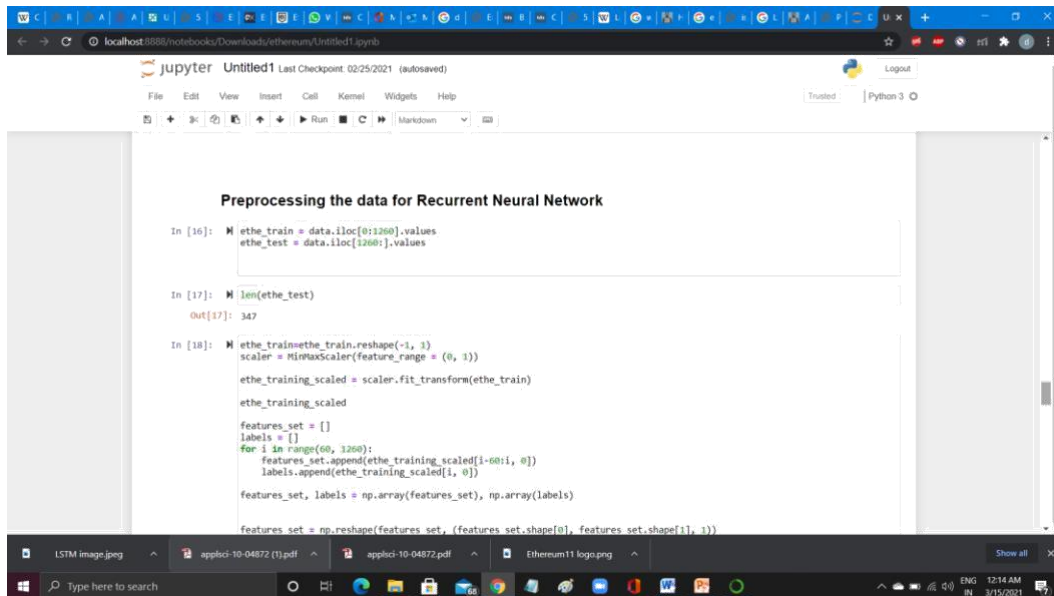


Figure 4.10 : P-Value

4.4 Building LSTM model

Now, the data is ready to feed into the model. Before that we have to divide the dataset into two part, as training data and testing data. After that we have create the Long Short-Term Memory recurrent neural network layers to predict the prices based on time series.

In figure 4.11, we can see that data is divided into testing data and training data.



```
Preprocessing the data for Recurrent Neural Network

In [16]: ethe_train = data.iloc[0:1200].values
         ethe_test = data.iloc[1200:].values

In [17]: len(ethe_test)
Out[17]: 347

In [18]: ethe_train = ethe_train.reshape(-1, 1)
         scaler = MinMaxScaler(feature_range = (0, 1))
         ethe_training_scaled = scaler.fit_transform(ethe_train)
         ethe_training_scaled

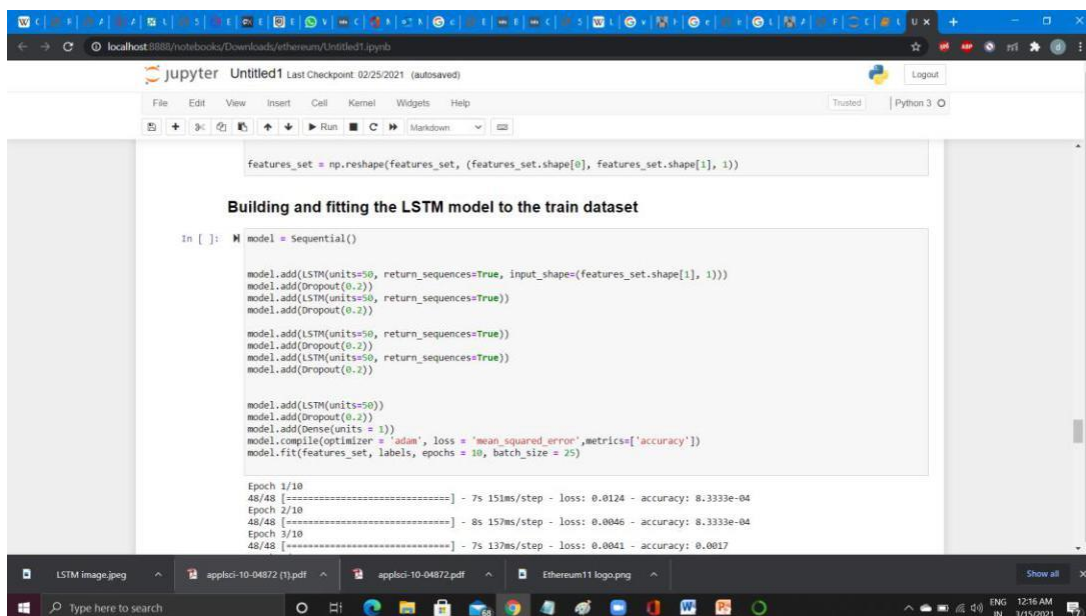
         features_set = []
         labels = []
         for i in range(60, 1200):
             features_set.append(ethe_training_scaled[i-60:i, 0])
             labels.append(ethe_training_scaled[i, 0])
         features_set, labels = np.array(features_set), np.array(labels)

         features_set = np.reshape(features_set, (features_set.shape[0], features_set.shape[1], 1))
```

Figure 4.11 dividing dataset

In figure 4.12 we can see that building and fitting the LSTM model for training data.

In this we have given 1200 epochs to train the model effectively.



```
Building and fitting the LSTM model to the train dataset

In [ ]: model = Sequential()

         model.add(LSTM(units=50, return_sequences=True, input_shape=(features_set.shape[1], 1)))
         model.add(Dropout(0.2))
         model.add(LSTM(units=50, return_sequences=True))
         model.add(Dropout(0.2))
         model.add(LSTM(units=50, return_sequences=True))
         model.add(Dropout(0.2))
         model.add(LSTM(units=50, return_sequences=True))
         model.add(Dropout(0.2))

         model.add(LSTM(units=50))
         model.add(Dropout(0.2))
         model.add(Dense(units = 1))
         model.compile(optimizer = 'adam', loss = 'mean_squared_error', metrics=['accuracy'])
         model.fit(features_set, labels, epochs = 1200, batch_size = 25)

Epoch 1/10
48/48 [=====] - 7s 151ms/step - loss: 0.0124 - accuracy: 8.3333e-04
Epoch 2/10
48/48 [=====] - 8s 157ms/step - loss: 0.0046 - accuracy: 8.3333e-04
Epoch 3/10
48/48 [=====] - 7s 137ms/step - loss: 0.0041 - accuracy: 0.0017
```

Figure 4.12 : building and fitting the LSTM model

For the last module result analysis, we see it in the next chapter.

Chapter – 5

Results and Discussion, Performance Analysis

5.1 Result

As we seen in the 4th chapter, we have built the LSTM time series model and trained it with 1200 epochs. Now, we are going to feed the test data to get the prediction.

In the below figure 5.1, we can see that testing data is feed to the model by using fit() method.

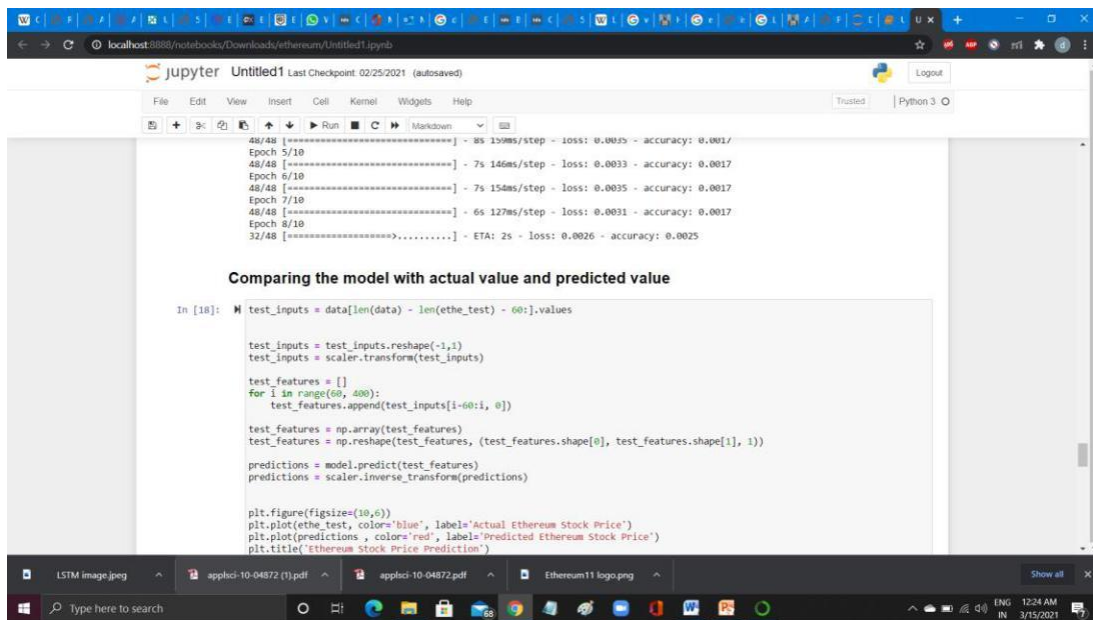


Figure 5.1 : making predictions

As a result, we are plotting the graph by comparing the both the testing data predictions.

In the below figure 5.2, we can see the graph by comparing model with actual and predicted values.

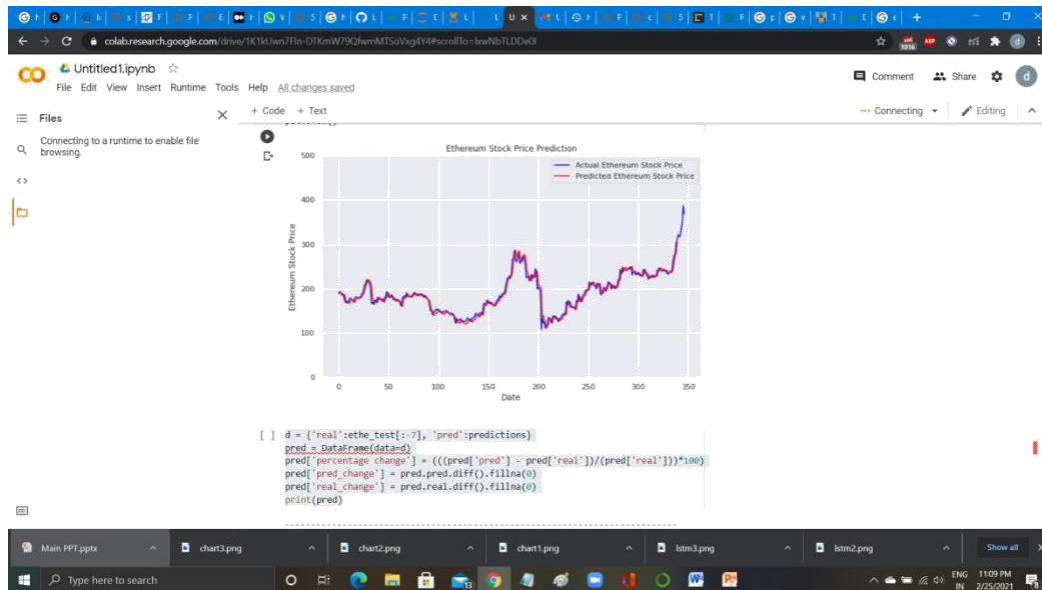


Figure 5.2 : Comparing

Result 5.2 Discussion and performance Analysis

In the LSTM model we have used the dense of one unit which means the number of hidden layers in the neural network is implies with the dense. And 20% of data is used in dropout. Dropout is used to minimize the over fitting occurs in the RNN models. Dropout is the value which we got as p-value. So that the 20 percent od data is removed automatically when the data over fit by the models when it is feed to the neural network layers.

A huge step in constructing up each and every Deep Network is to structure our statistics which can be used by neural nets for training and perhaps forecast activities This approach is termed as Pre-Processing, that may include steps for counting on the type of processing that we utilize.

We're going to do the following tasks for our pre-processing.

Data cleaning, filling out missing data points. Remove any unnecessary columns. Sort our data in order based on the dates. Split the training and testing data. By creating input samples, normalized between 0 and 1. Create and normalize target outputs for training and test sets between 0 and 1.

Convert our data to a NumPy array to be used by our model.

The algorithm giving the best performance with the accuracy if 70 percent.

Ether's price is not fixed; like many cryptocurrencies, it's volatile, and it's also upward

or downward. In this case, prices are on the downward trend and the integrative part of the algorithm, which converts the features into differences accounts for the trend and enables the rest of the model to learn about stationary features.

And then building LSTM-model. Throughout this approach, we applied seven LSTM layers with 1200 neurons per layer and 0.2 dropouts to every LSTM layer to protect from over-fitting and consequently the Dense layer to deliver our outcomes.

Chapter – 6

Summary and Conclusion

6.1 Summary

In this ethereum price prediction project we have built the very accurate ethereum price predicting model by going through the step by step process. First we have imported the important libraries and read the dataset. After that we have explored the dataset and made visualization graphs to find the relations between the data variables which are date, high, low, open, close, volume, %change. High indicates the highest price recorded and low indicates the lowest price. Open indicates the opening price in the stock exchange session. Volume indicates the number of transaction are happening using ethereum. %change indicates the change in prices. After that we have normalized our dataset to make ready for building model. After that we have created LSTM model and its layers and trained it with training dataset with large number of epochs. After that we have done the predictions for test dataset. We have compared the actual and predicted values in a graph which is our output.

6.2 Conclusion

Predicting cryptocurrency prices in real-time using the LSTM neural network. We have gone through a four-step process of obtaining real-time cryptocurrency data, preparing training and testing data, predicting prices using the LSTM neural network, and visualizing the results of predictions. The task of classifying price signal change is non-trivial, as demonstrated by the lack of clear class separation. While all methods achieved above 69 percent accuracy, the best performance was achieved by the LSTM and ADAM optimizers, which are attributed to their features and suitability for time-series data. Other methods were short-lived due to lack of data, assumptions made by models about data, and lack of convergence of models.

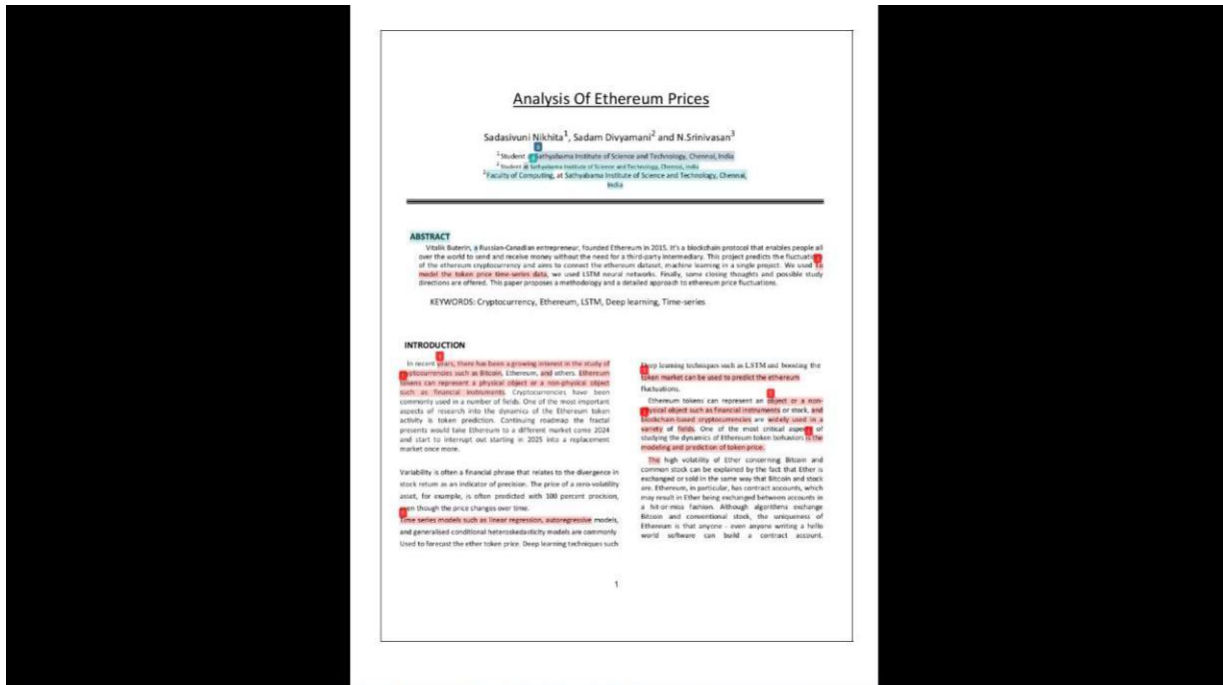
References

- [1] Franco Valencia, Alfonso Gómez-Espinosa, d Benjamín Valdés-Aguirre,|| Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning||, in entropi.
- [2] Gerlof Bremmer, || Predicting tomorrow's cryptocurrency price using a LSTM model, historical prices and Reddit comments||, ||Thesis code u743346||, July 2018
- [3] GUSTAV SEGERSTEDT THEODOR UHMEIER,|| How accuracy of time-series prediction for cryptocurrency pricing is affected by the sampling period|| EXAMENSARBETE INOM TEKNIK, GRUNDNIVÅ, 15 HP STOCKHOLM, SVERIGE 2018
- [4] Jaysing Bhosale, Sushil Mavale, Volatility of select Crypto-currencies: A comparison of Bitcoin, Ethereum and Litecoin, Annual Research Journal of SCMS, Pune Vol. 6, March 2018
- [5] Rituparna Ghosh, Khondoker Haider, Pedro Kim, Bitcoin or Ethereum, 2020
- [6] Yang Li, Zibin Zheng ,Hong-Ning Dai, Enhancing Bitcoin Fluctuation Prediction Using Attentive LSTM and Embedding Network, article, Applied Science,2018
- [7] Tensorflow Tutorials
<https://www.tensorflow.org/tutorials>
- [8] Understanding about algorithm <https://towardsdatascience.com/understanding>

Appendix

PAPER REPORTS

The paper on Ethereum Price Prediction System named as Analysis of Ethereum Prices obtained a plagiarism of 11% below are screenshots of the report.



LITERATURE SURVEY

In [1] According to the author, machine learning and statistical analysis, which were previously used for Bitcoin, can be used to predict the course of all price changes in the evolving cryptocurrency market. Using Twitter data, market data, or both, we analyzed and compared the performance of three prediction models: MLP, SVM, and RF for Bitcoin, Ethereum, Ripple, and Litecoin. [2] The observations of Geralt Berner's studies have found such set of abilities for Litecoin or even Ripple, and not just for Ethereum, as per their review. Conversely, the survey had shown that linkage could also be used to assess today's price or rates position with enough specificity also for similar cryptocurrencies. However, results still do not show a high enough degree of ability to be used for trading. Change in price of cryptocurrency is a daily process. [3] GUSTAV SÖGERSFELDT et al. Says that the accuracy of predictions can be improved by decreasing the sampling period of data. However, there doesn't seem to be any clear trend on how changing the neurons per LSTM layer affects predictive accuracy.

[4] According to Wang Shoukai et al., they compare three cryptocurrencies—Bitcoin, Ethereum, and Litecoin—in terms of their volatility and stability in recent times, as well as their patterns. In [5] Bhaganya Ghosh et al. did research on which is a better cryptocurrency among Bitcoin and Ethereum. They compare the two blockchain technologies and build a portfolio of Bitcoin and Ether that will deliver the highest return with the lowest risk over a five-year investment of a million dollars. In [6] Tong Li, Zhen Zheng et al. indicated in their research as Assess the situation of Cryptocurrency dispersion prediction in the sector of virtual currency. The results show that the designed features are the best fit for our presented problem statement and our proposed LSTM model which achieves the state of superior performance among every baseline. Our study may pave the way towards a real trading environment for the investors within the future.

METHODOLOGY

LSTM Model

This is achievable that now the Long-Short-Term Memory is indeed a RNN. The previous state's output is used as input in the current time step. The LSTM was produced by Hochreiter & Schmidhuber. It addressed the issue of RNN long-term dependencies, in which the RNN does not retain the kind stored in memory but can make more accurate predictions using more recent data. As the length of the gap increases, RNN does not provide efficient performance. By default, LSTM can retain data for a long time; it has been used to handle, estimate and characterize the understanding of historical data.

LSTM structure

LSTM has a chain structure as shown in Figure 1.0. It includes multiple neural networks as well as many memory blocks known as cells.

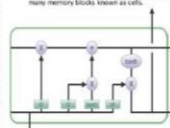
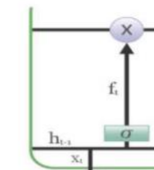


Figure 1.0

The information is stored in the cells and therefore the memory manipulations are performed by the gates.

There are three doors:-

Forget Gate: Just like in Figure 1.1, Forget Gate estimates results which is no longer convenient inside a convolution layer. Two variables x_t data at the real-time and h_{t-1} providing throughout of the neuron are granted to the gate as well as multiplied besides weight matrix accompanied by the sigmoid addition. The final value is equipped as a deep neural network providing a binary number. If the output is 0, the piece of data is forgotten for a particular cell state and the output x_t knowledge is retained for longer-term use.



Input Gate: Figure 1.2. Attaching relevant information to just the convolution layer is accomplished via the input gate. Then, learning is governed more by sigmoid function as well as screens the results to be retained much like a forget gate by using h_{t-1} as well as x_t interface. However a vector will be created by using tanh component that further computes from -1 to $+1$, that also compares of all unique combinations from h_{t-1} as well as x_t . And consequently, the vector measurements or the measured measurements are multiplied to make more informed decisions.

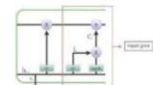


Figure 1.2

Output gate: Each activity of retrieving insight into the current convolution layer to be set there just like result is concluded by its output layer. Therefore, a model is created by employing tanh only to cell. The statistics is again generated also by sigmoid function as well as jumps the value to only be noted using the h_{t-1} as well as x_t inputs. And in the later part, the vector results as well as the regulated value systems are multiplied to always be multiplied as shown in Figure 1.3.

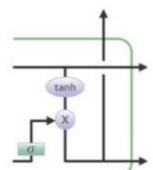


Figure 1.3

DATA ANALYSIS

The data is entering as a .csv file. Explore the data to find any missing value and treat their missing value with traditional techniques such as filling the null numerical variable with the

mean of the entire column and null value of the column with the median value after preprocessing the data to make it useful for data visualization.

In the Low Variable is defined as the lowest price of ethereum when the stock exchange session is started. High defines the highest price of the ethereum in the stock exchange session. Open defines the opening price of ethereum.

In data visualization, we can draw the graph between data variables to find the patterns between data variables in the ethereum historical dataset. In that dataset, we have variables of open, close, high, low, volume, change in the data visualization. We are going to visualize the relation between the different data variables.

In Figure 2.0, we can see that the graph plotted for the data variables concerning years. It has represented the graph for each data variable such as low, high, open and, price. In the graph, we can see that all data variables over the years are having the same kind of patterns by plotting the data points.

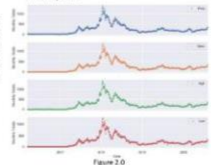
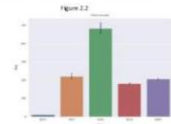


Figure 2.0

In Figure 2.1, it is seen how below has drawn a graph to see the patterns between the volume and year. Share in the ethereum entire dataset determines the amount of accounts in the ethereum trade. We can see that the graph is busy with a lot of points. This means that there has been a spike in the ethereum trade and even a lot of customers are exhibiting their passion in trading again with cryptocurrencies. Such a figure claims that there could be a shift in demand for cryptocurrency.



Figure 2.2: Figure 2.2. This figure illustrates the market modifications. In 2015, the price of ethereum was very low. But in subsequent years, the price is getting increased. After that from the year 2019, the price are decreasing with a lot of difference. Then the prices are slightly increased in the year of 2020.



RESULTS AND DISCUSSION

A huge step in constructing each and every Deep Network is to structure our statistics which can be used by neural nets for training and perhaps forecast activities. This approach is termed as Pre-Processing, that may include steps for counting on the type of processing that we utilize. We're going to do the following tasks for our pre-processing.

Data cleaning, filling out missing data points, Remove any unnecessary columns, Sort our data in order based on the dates, Split the training and testing data. By creating input samples, normalized between 0 and 1. Create and normalize target outputs for training and test sets between 0 and 1. Convert our data to a NumPy array to be used by our model.

The algorithm was of the best performance. Ether's price is not fixed, like many cryptocurrencies, it's volatile, and it's also upward or downward. In this case, prices are on the downward trend and the integrative part of the algorithm, which converts the features into difference accuracy for the model and enables the rest of the model to learn about stationary features.

And then building LSTM model. Throughout this approach, we applied seven LSTM layers with 340 neurons per layer and 0.2 dropout. In every LSTM layer, we performed both mean fitting and consequently the Dense layer to deliver our outcomes.



It is then possible to use the Model to predict the longer term, prices of Ethereum for the subsequent hours. There are indeed a plethora of ways to accurately predict. Substantial predictions at hourly predictions. I used heuristic predictions to evaluate for the next few hours. The study indicates most all at a duration of 1 hour each moment.

So we used the anticipated variables and added them as input variables within the our window by shifting by one step every time. Such a process is significant for several more predictions since that multiplies expected error for every instance, that often can get a huge impact on the quality of its predictions throughout a prolonged period.

Nevertheless, it is crucial to perform more remarkable results of more sample points and then by altering the input variables of the LSTM.

CONCLUSION

Predicting cryptocurrency prices in real-time using the LSTM neural network. We have gone through a four-step process of obtaining real-time cryptocurrency data, preprocessing and testing data, predicting prices using the LSTM neural network, and evaluating the results of predictions. The task of classifying price signal change is non-trivial, as demonstrated by the lack of clear class separation. While all methods achieved above 50 percent accuracy, the best performance was achieved by the LSTM and ACNN optimizers, which are attributed to their features and suitability for time-series data. Other methods were short-lived due to lack of data, assumptions made by models about data, and lack of emergence of models. We obtained the output with 70% accuracy.

REFERENCES

- [1] Franco Valencia, Alfonso Gómez-Espinoza, & Benjamin Valdés-Aguilera, "Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning", in *entropy*.
- [2] Gerolf Breuer, "Predicting tomorrow's cryptocurrency price using a LSTM model, historical prices and Reddit comments", "This code is 78340", July 2018.
- [3] GUSTAV SEGERSTEDT THEODOR LINDBER, "How accuracy of time series prediction for cryptocurrency pricing is affected by the sampling window", CAMDENARRETE ACM TEXAS, GRANDVIEW, 15 SEP STOCKHOLM, SWEDEN 2018.
- [4] Jayraj Bhosale, Saad Mawale, Viability of select Cryptocurrencies: A comparison of Bitcoin, Ethereum and Litecoin, Annual Research Journal of SCMS, Pune Vol. 5, March 2018.
- [5] Rituparna Ghosh, Rhondaker Haider, Pedro Kim, Bitcoin or Ethereum, 2020.
- [6] Yang Li, Jian Zheng, Jiong Ning Dai, Following Bitcoin Fluctuation Prediction Using Attention LSTM and Embedding Network, article, Applied Science, 2018.
- [7] TensorFlow Tutorials <https://www.tensorflow.org/tutorials>
- [8] Understanding about algorithm https://www.dailymotion.com/video/understanding-lstm-and-its-quick-implementations-in-python-for-sentiment-analysis-video_407480442

ANALYSIS OF ETHEREUM PRICES-PAPER1.doc	
ORIGINALITY REPORT	
11%	4% 7% 4%
SIMILARITY INDEX	INTERNET SOURCES PUBLICATIONS STUDENT PAPERS
PRIMARY SOURCES	
1	Yuzhou Chen, Hon Keung Tony Ng. "Deep Learning Ethereum Token Price Prediction with Network Motif Analysis", 2019 International Conference on Data Mining Workshops (ICDMW), 2019 Publication 4%
2	Submitted to National Institute of Technology Jamshedpur Student Paper 2%
3	res.mdpi.com Internet Source 1%
4	www.inderscience.com Internet Source 1%
5	arno.uvt.nl Internet Source 1%
6	www.scmspune.ac.in Internet Source 1%
7	www.slideshare.net Internet Source <1%
8	Submitted to UT, Dallas Student Paper <1%
9	Franco Valencia, Alfonso Gómez-Espinosa, Benjamín Valdés-Aguirre. "Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning", Entropy, 2019 Publication <1%
Exclude quotes Off Exclude matches Off	
Exclude bibliography Off	

Source code

Ethereum Price Analysis

Importing Modules

```
from pandas import read_csv, Grouper, DataFrame, concat
import matplotlib.pyplot as plt
```

```

from statsmodels.tsa.stattools import
adfuller import statsmodels.tsa.api as smt
import numpy as np
from sklearn.metrics import
mean_squared_error import seaborn as sns
from datetime import datetime
import pandas as pd
from pandas_profiling import ProfileReport
from sklearn.preprocessing import MinMaxScaler
from keras.preprocessing import sequence from
keras.models import Sequential
from keras.layers import Dense, Embedding
,Dropout from keras.layers import LSTM
from plotly.offline import plot, iplot, init_notebook_mode
import plotly.graph_objs as go
init_notebook_mode(connected=True)

```

Importing Dataset

```
df=pd.read_csv("Ethereum Historical Data.csv")#1062
```

Exploring Dataset

```
df.head()
```

```
df.info
```

```
df.describe
```

```
df.isnull().sum()
```

Preprocessing the data

```

df["Vol."]=df["Vol."].str.replace("-", "0")
df["Vol."] = (df["Vol."].replace(r'[KMB]+$', "", regex=True).astype(float) *
df["Vol."].str.extract(r'[d\.]+' + ([KMB]+)', expand=False).fillna(1).replace(['K','M', 'B'],

```

```
[10**3, 10**6, 10**9]).astype(int))
df["Price"]=df["Price"].str.replace(",","")
df["Price"]=df["Price"].astype("float")
df["Open"]=df["Open"].str.replace(",","")
df["Open"]=df["Open"].astype("float")
df["High"]=df["High"].str.replace(",","")
df["High"]=df["High"].astype("float")
df["Low"]=df["Low"].str.replace(",","")
df["Low"]=df["Low"].astype("float")
df["Change %"]=df["Change %"].str.replace("%","")
df["Change %"]=df["Change %"].astype("float")
```

Data Visualisation

Ethereum Price by year wise

```
df['Date']= pd.to_datetime(df['Date'], format="%b %d, %Y")
df = df.sort_values(by='Date', ascending=True)
#df.head()
df.index = df.Date
del df["Date"]
sns.set(rc={'figure.figsize':(10,9)})
sns.barplot(x=df.index.year, y=df["Price"], data=df)
plt.xlabel("Year", labelpad=17)
plt.ylabel("Price", labelpad=14)

plt.title("Price by year ", y=1.01);
```

Montly total for stock values

```
cols_plot = ["Price","Open","High","Low"]
axes = df[cols_plot].plot(marker='.', alpha=0.5, linestyle='None', figsize=(11, 9),
subplots=True)
for ax in axes:
    ax.set_ylabel('Monthly Totals')
```

Year wise report on Change %

```
plt.figure(figsize=(10,6))
sns.boxplot(data=df, x=df.index.year, y=df["Change %"])
plt.title("Year wise report on Change %")
Text(0.5, 1.0, 'Year wise report on Change %')
```

Volume flow by year wise

```
sns.set(rc={'figure.figsize':(25,10)})
start, end = '2016', '2020'
plt.plot(df.loc[start:end, 'Vol.'],
marker='.', linestyle='-', linewidth=0.5, label='Daily')
plt.title("Volume by year ", y=1.01);
```

Auto-correlation and Partial-correlation

```
data=df["Price"]
feature_ts_train_diff = data.diff(periods=1)
feature_ts_train_diff.dropna(inplace=True)
feature_ts_train_diff
fig, axes = plt.subplots(1, 2)
fig.set_figwidth(16)
fig.set_figheight(6)
plt.xticks(range(0,30,1), rotation = 90)
smt.graphics.plot_acf(feature_ts_train_diff, lags=30, ax=axes[0])
smt.graphics.plot_pacf(feature_ts_train_diff, lags=30, ax=axes[1])
plt.tight_layout()
```

Statistic and P-value

```
dfctest = adfuller(data)
print("Statistics",dfctest[0])
fig = go.Figure(go.Indicator(
    mode = "gauge+number",
    value = dfctest[1],
```

```

        title = {'text': "P value"},
        domain = {'x': [0, 1], 'y': [0, 1]}
    ))
fig.show()

```

Statistics -2.444765191098683

Preprocessing the data for Recurrent Neural Network

```

ethe_train = data.iloc[0:1260].values
ethe_test = data.iloc[1260:].values
len(ethe_test)
ethe_train=ethe_train.reshape(-1, 1)
scaler = MinMaxScaler(feature_range = (0, 1))
ethe_training_scaled = scaler.fit_transform(ethe_train)
ethe_training_scaled
features_set = []
labels = []
for i in range(60, 1260):
    features_set.append(ethe_training_scaled[i-60:i, 0])
    labels.append(ethe_training_scaled[i, 0])
features_set, labels = np.array(features_set), np.array(labels)
features_set = np.reshape(features_set,
(features_set.shape[0], features_set.shape[1], 1))

```

Building and fitting the LSTM model to the train dataset

```

model = Sequential()
model.add(LSTM(units=50, return_sequences=True,
input_shape=(features_set.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=True))

```

```

model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units = 1))
model.compile(optimizer = 'adam', loss = 'mean_squared_error', metrics=['accuracy'])
model.fit(features_set, labels, epochs = 10, batch_size = 25)
test_inputs = test_inputs.reshape(-1,1)
test_inputs = scaler.transform(test_inputs)
test_features = []
for i in range(60, 400):
    test_features.append(test_inputs[i-60:i, 0])
test_features = np.array(test_features)
test_features = np.reshape(test_features,
(test_features.shape[0], test_features.shape[1], 1))
predictions = model.predict(test_features)
predictions = scaler.inverse_transform(predictions)
plt.figure(figsize=(10,6))
plt.plot(ethe_test, color='blue', label='Actual Ethereum Stock Price')
plt.plot(predictions , color='red', label='Predicted Ethereum Stock Price')
plt.title('Ethereum Stock Price Prediction')
plt.xlabel('Date')
plt.ylabel('Ethereum Stock Price')
plt.ylim(0,500)
plt.legend()
plt.show()
predictions=predictions.flatten()
len(predictions)
340
len(ethe_test)
347
d = {'real':ethe_test[:-7], 'pred':predictions}

```

```

pred = DataFrame(data=d)
pred['percentage change'] = (((pred['pred'] - pred['real'])/(pred['real']))*100)
pred['pred_change'] = pred.pred.diff().fillna(0)
pred['real_change'] = pred.real.diff().fillna(0)
def accuracy(df):
    count = 0
    for index, row in pred.iterrows():
        if row['pred_change'] < 0 and row['real_change'] < 0 or row['pred_change'] >
0 and row['real_change'] > 0:
            count += 1
    return (count/len(df)*100)
def variance(df):
    count = 0
    percentage = 0
    for index, row in pred.iterrows():
        if row['pred_change'] < 0 and row['real_change'] > 0 or row['pred_change'] >
0 and row['real_change'] < 0:
            count += 1
            percentage += abs(row['percentage change'])
    return percentage/count

print('Variance of incorrect prediction how far off (up/down): ', variance(pred))

```

