

PROFESSIONAL TRAINING REPORT-II
entitled
PREDICTING BITCOIN PRICES USING DEEPLARNING

Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering
With specialization in Blockchain Technology

By
L.NAGA VENKATA ESWARA SASTRY
REG. NO: 41613025



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)
CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A++" by NAAC
(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119
www.sathyabamauniversity.ac.in



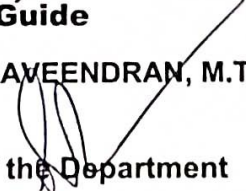
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **L.NAGA VENKATA ESWARA SASTRY**(Reg.No.41613025) who carried out the project entitled "Predicting Bitcoin Prices Using Deep Learning" under my supervision from JAN 2024 to MAY 2024.

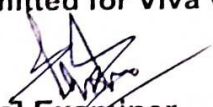

Guide

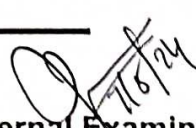
MS.LEKSHMI S RAVEENDRAN, M.TECH.,


Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 07/05/2024


Internal Examiner

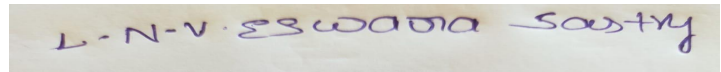

External Examiner

DECLARATION

I **L.NAGA VENKATA ESWARA SASTRY (41613025)** hereby declare that the Project Report entitled **Predicting Bitcoin Prices using Deep Learning** Done by me under the guidance of **MS. LEKSHMI S RAVEENDRAN, M.TECH.**, is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:07/05/2024

PLACE: Chennai

A rectangular box containing a handwritten signature in blue ink. The signature appears to be 'L.N.V. Eswara Sasthy' written in a cursive style.

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.Vigneshwari M.E., Ph.D., Head 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 **Ms.LEKSHMI S RAVEENDRAN,M.TECH.**, 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.

TRAINING CERTIFICATE



ABSTRACT

Cryptocurrency, especially Bitcoin, is one of the most volatile markets today and has gained a lot of attention from investors across the globe. Cryptocurrency, being a novel technique for transaction systems, has led to a lot of confusion among investors and any rumors or news on social media has been claimed to significantly affect the prices of cryptocurrencies. Implemented linear regression and Long Short-Term Memory (LSTM) models for bitcoin price prediction. The goal of this proposed work is to predict prices for Bitcoin using Machine learning techniques for the next day and prepare a strategy to maximize gains for investors. We also aim to find out if there is a correlation between fluctuating Bitcoin prices and related news. The proposed method compared to other Machine Learning models from relevant studies, it was revealed that the linear regression model's accuracy rate is quite high; it was shown to be 99.87 percent correct. On the other hand, the LSTM model has a minor error rate of 0.08 percent. The neural network model is thus shown to be more optimal than the machine learning model as a result of this. Ensemble learning and other eminent deep learning techniques may improve the bitcoin price prediction accuracy. The research purpose of this paper is to obtain an algorithm model with high prediction accuracy for the price of Bitcoin on the next day through random forest regression and LSTM, and to explain which variables have influence on the price of Bitcoin. There is much prior literature on Bitcoin price prediction research, and the research methods mainly revolve around the ARMA model of time series and the LSTM algorithm of deep learning. Although it cannot be proved by the Diebold–Mariano test that the prediction accuracy of random forest regression is significantly better than that of LSTM, the prediction errors RMSE and MAPE of random forest regression are better than those of LSTM. The changes in the variables that determine the price of Bitcoin in each period are also obtained through random forest regression. From 2015 to 2018, three US stock market indexes, NASDAQ, DJI, and S&P500 and oil price, and ETH price have impact on Bitcoin prices.

TABLE OF CONTENTS

VII

CHAPTER No.	TITLE	PAGE NO.
	ABSTRACT	VI
	LIST OF FIGURES	IX
	LIST OF ABBREVIATIONS	X
1	INTRODUCTION	1
	1.1 Introduction to deep learning	2
	1.2 Types of Deep learning	3
	1.3 Long Short Term Memory	4
	1.4 Significance Of Deep learning in bitcoin price prediction	6
	1.5 Advantages and disadvantages of LSTM model	9
	1.6 Feature selection with LSTM and Deep learning	11
	1.7 The Packages to implement the project	13

	1.8 Methodology	13
2	Literature Survey	23
	2.1 Survey	23
3	Requirement Analysis	25
	3.1 Objective	25
		26
	3.2 Hardware requirements	
	3.3 Software requirements	27
4	Design description of proposed product	29
	4.1 Ideation Map	29
	4.2 Various stages	30
	4.3 Working Principles	31
	4.4 Novelty of the project	31
5	Conclusion	33
	5.1 Reference	34
	5.2 Appendix	36

LIST OF FIGURES

IX

FIGURE NO.	FIGURE NAME	PAGE NO.
-----------------------	--------------------	---------------------

1.3	Fig.1.3 Pictorial representation of Deep Learning	5
1.4	Fig.1.4Pictorial representation of Deep learning in Bitcoin price prediction workflow	7
1.8.1	Fig.1.8.1 Random Forest	14
1.8.2	Fig.1.8.2 LSTM	16
1.8.3	Fig.1.8.3Training and Validation loss of LSTM	16
1.8.4	Fig.1.8.4 Correlation heatmap of explanatory variables	19
1.8.5	Fig 1.8.5 Bitcoin price chart	20
1.8.6	Fig 1.8.6 Interval division of training and testing samples	21
1.8.7	Fig 1.8.7 Pictorial representation of entire experiment	22
	Fig 4.1.1 Ideation Map	29
4.1.1	Fig 4.1.2 Various stages	30

LIST OF ABBREVIATIONS

ABBREVIATION

EXPANSION

LSTM

Long short term memory

CNN

Convolutional neural
network

CHAPTER 1

INTRODUCTION

In a multidisciplinary education context, project based learning appears one of the most interesting instructional strategies which tries to engage students in authentic real world tasks to enhance learning . In project based learning students typically engage individually or in groups with an instructor or coach or mentor. Each of the project designs and implements an approach to understand practical professional environment in the field of computer science engineering. In this paper, our students have applied project based learning to develop a bit coin pricing algorithm; in addition to technical knowledge they also learned to manage resources and time execution and work in teams. Building algorithms and models to predict prices and future events has been given significant amount of attention in the past decade. With user data being collected through various forms of paths, there has never been an abundance in raw data like there is now. Any model capable of predicting a future event whether it be to find out what the next big trend is or to predict the next behavior of a customer, most predictive models possess great potential to change opportunity into revenue. The price prediction category is no different. For years analysts and researches have been studying and trying to improve algorithms to help predict future prices. Unfortunately, the predictions cannot be based purely on just previous prices, though it should still considered to have the most contribution to the model, other things such as economic growth, social and popularity of the commodity also play a significant role price predictions. Bitcoin is the first cryptocurrency that was created back in 2009. But, it did not gain much popularity till 2012. In general, cryptocurrencies are just lines of code that can have some financial value. These codes are generated by highperformance computers. Since the creation of bitcoin, many other cryptocurrencies have been created and these are known as Altcoins. Since bitcoins are not managed by any bank and are generally open to the public to do transactions via Blockchain methodologies, it has gained popularity due to the fact of no middleman or examination by tax authorities. Bitcoins can be thought of like gold in the early 19th century. The banks and traders were ready to exchange it for money but did not have complete control over it. Similarly, bitcoins can be exchanged for

money, used to purchase goods and even complete transactions. Each coin who have to be mined and only twenty one million coins will ever exist out of which 11 million have already been mined. One of the problems that analysts and researchers faced was to implement a system capable of accurately predicting the prices. In this innovative project, our students have implemented machine-learning algorithm, for predicting changes in Bitcoin prices in the short run, from historical time series data of quantitative factors that affect Bitcoin prices. The concept was to implement a system capable to analyze real time data and give a sense of direction to investors to help in decision making. The application will take in real world data and will go through a series of data reshaping which will get the data ready to be fed into the machine learning algorithms. Using this data, we are able to predict the Bitcoin price of tomorrow. The rest of the paper is organized as follows. Section II discusses the related work done in the area of bitcoin price prediction. Section III presents the prediction model with the features. Section IV gives the software requirements of the proposed project work. Section V discusses the project design and implementation.

1.1 INTRODUCTION TO DEEP LEARNING

Profound learning is a part of AI which depends on fake brain organizations. It is fit for learning complex examples and connections inside information. In profound learning, we don't have to program everything unequivocally. It has become progressively famous lately because of the advances in handling power and the accessibility of huge datasets. Since it depends on counterfeit brain organizations (ANNs) otherwise called profound brain organizations (DNNs). These neural networks are made to learn from a lot of data and are based on the structure and function of biological neurons in the human brain.

Neural networks are used in Deep Learning, a subfield of Machine Learning, to model and solve complex problems. Brain networks are designed according to the construction and capability of the human mind and comprise of layers of interconnected hubs that cycle and change information.

The use of deep neural networks, which have multiple layers of interconnected nodes,

is the main feature of deep learning. By locating hierarchical patterns and features in the data, these networks are able to learn complex representations of the data. Profound Gaining calculations can naturally gain and improve from information without the requirement for manual component designing.

Profound Learning has made huge progress in different fields, including picture acknowledgment, regular language handling, discourse acknowledgment, and proposal frameworks. A portion of the famous Profound Learning designs incorporate Convolutional Brain Organizations (CNNs), Repetitive Brain Organizations (RNNs), and Profound Conviction Organizations (DBNs).

Preparing profound brain networks regularly requires a lot of information and computational assets. However, deep neural network training has become simpler due to the availability of cloud computing and the development of specialized hardware like GPUs.

1.2 Types of Deep Learning

Deep learning can be used for supervised, unsupervised as well as reinforcement machine learning. it uses a variety of ways to process these.

Supervised Machine Learning: Supervised machine learning is the machine learning technique in which the neural network learns to make predictions or classify data based on the labeled datasets. Here we input both input features along with the target variables. the neural network learns to make predictions based on the cost or error that comes from the difference between the predicted and the actual target, this process is known as back propagation. Deep learning algorithms like Convolutional neural networks, Recurrent neural networks are used for many supervised tasks like image classifications and recognition, sentiment analysis, language translations, etc.

Unsupervised Machine Learning: unsupervised machine learning is the machine learning technique in which the neural network learns to discover the patterns or to cluster the dataset based on unlabeled datasets. Here there are no target variables. while the machine has to self-determined the hidden patterns or relationships within the datasets. Deep learning algorithms like auto encoders and generative models are

used for unsupervised tasks like clustering, dimensionality reduction, and anomaly detection.

Reinforcement Machine Learning: Reinforcement Machine Learning is the machine learning technique in which an agent learns to make decisions in an environment to maximize a reward signal. The agent interacts with the environment by taking action and observing the resulting rewards. Deep learning can be used to learn policies, or a set of actions, that maximizes the cumulative reward over time. Deep reinforcement learning algorithms like Deep Q networks and Deep Deterministic Policy Gradient (DDPG) are used to reinforce tasks like robotics and game playing etc.

1.3 Long Short-Term Memory [LSTM]

Hochreiter and Schmidhuber's Long Short-Term Memory is an improved version of their recurrent neural network. LSTM excels at capturing dependencies over time and is well-suited for sequence prediction tasks. Its applications stretch out to assignments including time series and groupings. The ability of LSTM to comprehend order dependence, which is essential for resolving intricate problems like speech recognition and machine translation, is its strength. The LSTM model, architecture, working principles, and the crucial roles they play in various applications are covered in depth in this article, which provides an in-depth introduction to LSTM.

A traditional RNN only has one hidden state that changes over time, it can be hard for the network to learn dependencies over time. LSTMs address this issue by presenting a memory cell, which is a compartment that can hold data for a drawn out period. Because LSTM networks can learn long-term dependencies in sequential data, they are ideal for tasks like translating languages, recognizing speech, and forecasting time series. LSTMs can likewise be utilized in blend with other brain network models, like Convolutional Brain Organizations (CNNs) for picture and video examination.

The memory cell is constrained by three entryways: the information entryway, the neglect door, and the result entryway.

These entryways choose what data to add to eliminate from, and yield from the memory cell. The information entryway controls what data is added to the memory cell. The neglect entryway controls what data is taken out from the memory cell. Also, the result door controls what data is yield from the memory cell. This permits LSTM organizations to specifically hold or dispose of data as it moves through the organization, which permits them to learn long haul controls

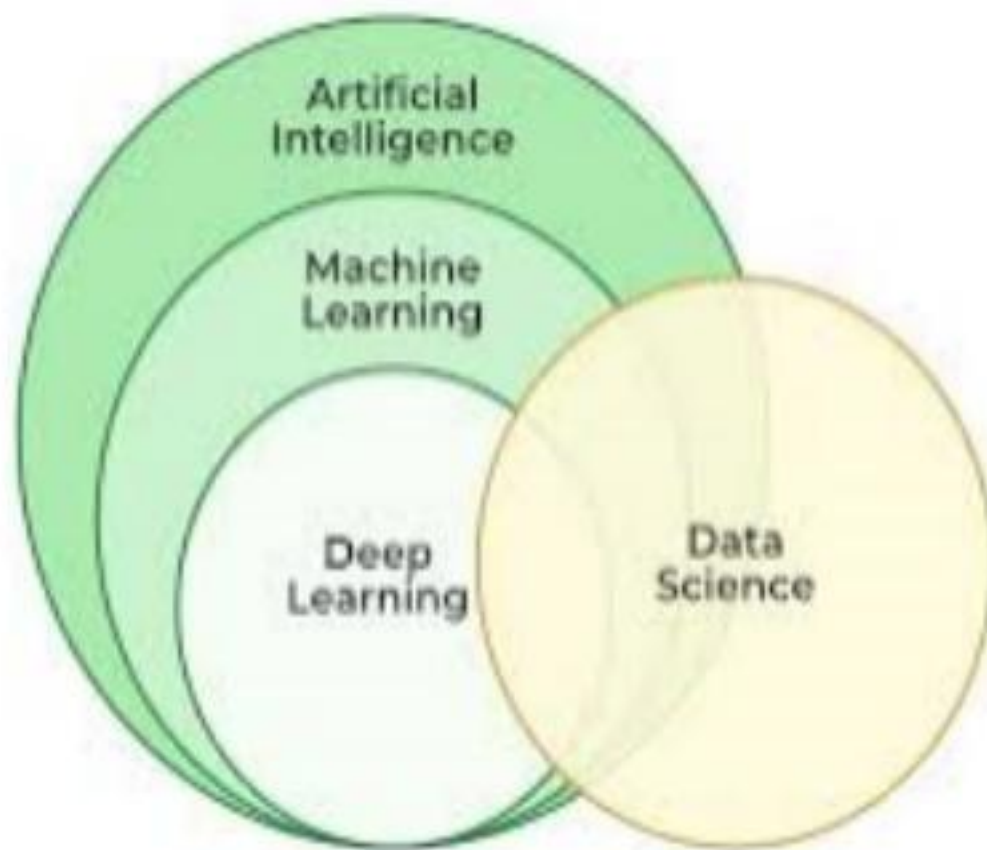


Fig 1.3-pictorial representation of Deep learning

1.4 Significance Of Deep Learning in Bitcoin price prediction

What is a LSTM?

There are not enough studies using deep learning models to predict cryptocurrencies using machine learning. As indicated by the exploration of 2016, more than 600 papers have been distributed on this subject. Our writing overview covers work done on bitcoin (BTC) cost expectation utilizing various methods, the need, and evaluation of intermittent brain organization (RNN) and its framework engineering. Dennyset al. utilized machine learning techniques like k-means clustering and artificial neural networks (ANN), support vector machines (SVM), and recurrent neural networks (RNN), as well as various attribute selection mechanisms, in the bitcoin price prediction. In any case, one limitation of this study is just centered around the financial backers. Strategy creators ought to be considered as a significant accomplice of the framework since digital currency can change the elements of world economy. Sean McNally and others predicted the direction of the USD price of Bitcoin by employing a Bayesian optimized recurrent neural network and LSTM. They also used ARIMA model to think about the profound learning strategies. In Atsalakis and colleagues this research centers around computational knowledge technique particularly half and half neuro-fluffy regulator to anticipate the conversion scale of bitcoin. The neuro-fuzzy method and artificial neural networks were utilized in this model. Good fellow et al. proposed a profound direct support learning system for financial signal representation and exchanging. They consolidated the support learning (RL), profound learning (DL), and their ongoing profound brain organization (NN) to produce exact forecast results. They approve the proposed approach involving product future business sectors too as stock market information. Madan et al. attempted to anticipate the cost of bitcoin utilizing machine learning and examine the patterns of BTC encompassing. They utilized 25 attributes relating to bitcoin to conjecture the day to day cost variety. In Lahmiri et al. they used machine learning algorithms to predict the daily price of cryptocurrencies with high data availability like bitcoin, ripple, and digital cash. They used RNN and GRNN (Generalized Regression Neural Network) to accurately predict

cryptocurrencies with high liquidity. Saxena and others Using LSTM and the ARIMA model, investigated the minimum bitcoin price accuracy

LSTM calculation gives efficient and exact outcomes.reassured that bitcoin is the newest and most widely used virtual currency, despite its uncertain security and volatility rate. This study makes it functional for the distributed exchange of bitcoin through the organization and the blockchain technology. Phaladisailoed al. accurately predicted the price of bitcoin using a variety of machine learning algorithm

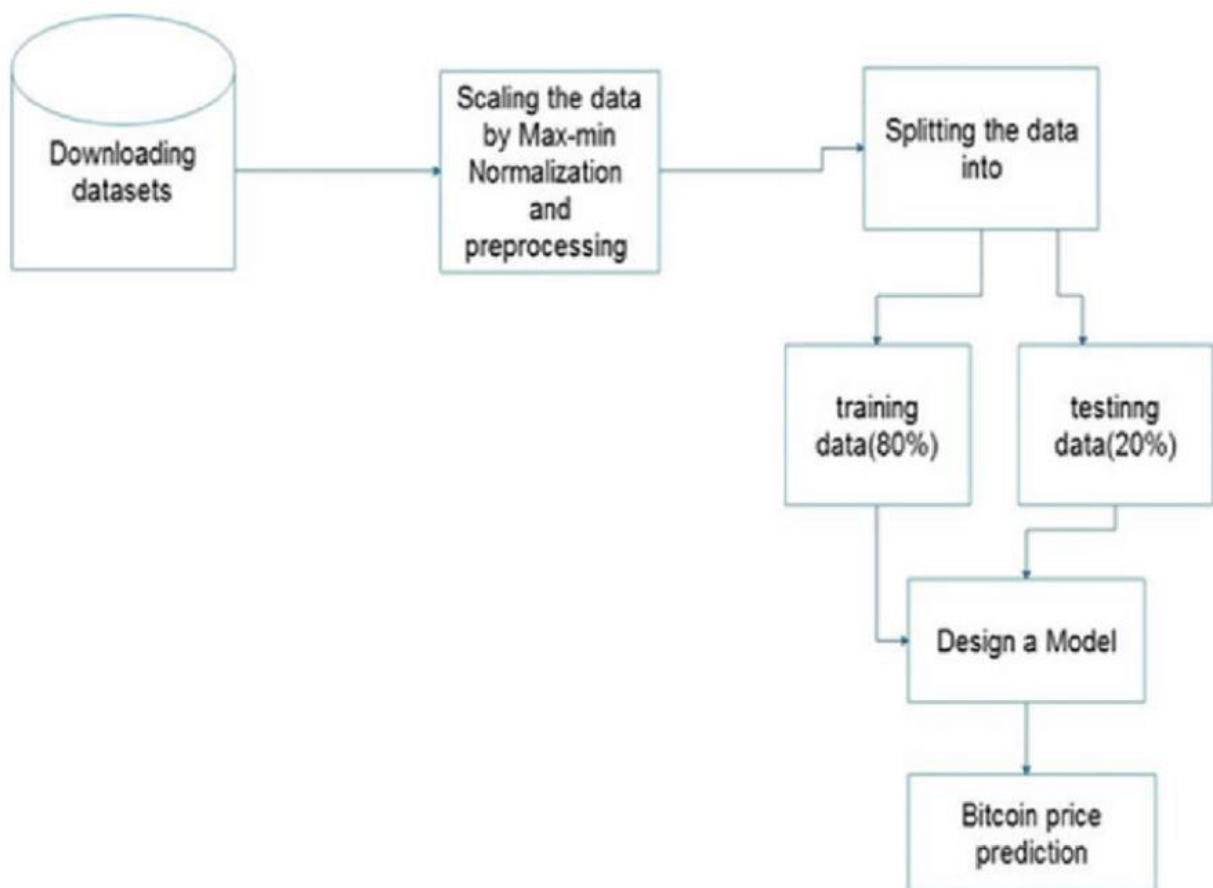


Fig:1.4 pictorial representation of deep learning in bitcoin price prediction workflow

The first building block of the LSTM unit determines the percentage of the Long-Term Memory (cell state) that will be retained. Technically it's called the forget gate. This block combines the input with the short-term memory and then runs the resulting computations through a Sigmoid activation function. It returns a number between 0 and 1 representing the percentage of the long-term memory that will be used in the following steps.

The second building block of the LSTM unit consists of 2 sub-blocks. The second sub-block combines the same input with the short-term memory (hidden state) to create a potential Long term memory using different trainable weights. This combination takes the form of a weighted sum of the input and the short-term memory and then runs the result through a Tanh activation function that returns a number between -1 and 1.

But, how much of this potential long-term memory will be added to the actual long-term memory value is controlled by the first sub-block. The first sub-block performs the same mathematical combination between the input and the short memory term but the resulting output will pass through a Sigmoid activation function. The final result represents the percentage of how much of the future potential long-term memory value, will be added to the actual cell state.

After updating the long-term memory during the first 2 blocks, the final block is responsible for refreshing the short-term memory. It gets the new value from the forget gate and passes it to the tanh activation function to produce the potential short-term memory. Then, following the same steps as in the second block, a second sub-block will combine the input and the old short-memory term to produce the percentage of how much the unit will retain from the potential value. This block results in a new hidden state and a new short-term memory representing the final output of the LSTM unit also called the output gate.

To wrap, a single LSTM contains three fundamental parts also called gates. The first gate or the forget gate and the input gate update the cell state with respect to the

current input. The output gate returns the new hidden state based on the newly calculated long-term memory or the cell state.

1.5 Advantages and Disadvantages of LSTM Model

Advantages:

1. **Handling Long Sequences:** LSTMs are well-suited for processing sequences of data with long-range dependencies. They can capture information from earlier time steps and remember it for a more extended period, making them effective for tasks like natural language processing (NLP) and time series analysis.
2. **Avoiding Vanishing Gradient Problem:** LSTMs address the vanishing gradient problem, which is a common issue in training deep networks, particularly RNNs. The architecture of LSTMs includes gating mechanisms (such as the forget gate) that allow them to control the flow of information and gradients through the network, preventing the gradients from becoming too small during training.
3. **Handling Variable-Length Sequences:** LSTMs can handle variable-length input sequences by dynamically adjusting their internal state. This is useful in many real-world applications where the length of the input data varies.
4. **Memory Cell:** LSTMs have a memory cell that can store and retrieve information over long sequences. This memory cell allows LSTMs to maintain important information while discarding irrelevant information, making them suitable for tasks that involve remembering past context.
5. **Gradient Flow Control:** LSTMs are equipped with mechanisms that allow them to control the flow of gradients during back propagation. The forget gate, for example, can prevent gradients from vanishing when they need to be propagated back in time. This enables LSTMs to capture information from earlier time steps effectively.

Disadvantages:

1. **Computational Complexity:** LSTMs are computationally more intensive compared to other neural network architectures like feed forward networks or simple RNNs. Training LSTMs can be slower and may require more resources.
2. **Overfitting:** Like other deep learning models, LSTMs are susceptible to overfitting when there is insufficient training data. Regularization techniques like dropout can help mitigate this issue.
3. **Hyperparameter Tuning:** LSTMs have several hyper parameters to tune, such as the number of LSTM units, the learning rate, and the sequence length. Finding the right set of hyper parameters for a specific problem can be a challenging and time-consuming process.
4. **Limited Interpretability:** LSTMs are often considered as “black-box” models, making it challenging to interpret how they arrive at a particular decision. This can be a drawback in applications where interpretability is crucial
5. **Long Training Times:** Training deep LSTM models on large datasets can be time-consuming and may require powerful hardware, such as GPUs or TPUs.
6. Despite their disadvantages, LSTMs have been instrumental in many fields, including NLP, speech recognition, and time series forecasting, due to their ability to capture complex sequential patterns.

1.6 Feature selection with LSTM and Deep Learning

Long Short-Term Memory—LSTM

LSTM is an updated version of RNN. They are specifically designed to avoid long-term dependence problems, whilst solving the vanishing gradient problem with an added mechanism, for regulating information, allowing it to be retained for long periods of time

. In short, the LSTM architecture is made up of a number of memory blocks that are recurrently connected sub networks. The network's memory blocks serve the dual functions of maintaining the network's state over time and regulating the flow of information between the cells. shows the LSTM block architecture, with input signal and the activation function. The input gate step is responsible for determining the information which should be kept in the cell state while the output is responsible for computation of the information that should be sent out from the cell state.

Gated Recurrent Unit—GRU

Gated Recurrent Units (GRUs) are a type of RNN that were introduced by in 2014 as an improvement over the traditional LSTM networks. Like LSTMs, GRUs are designed to be able to process input sequences of arbitrary length and maintain a state that encodes information about the past. However, unlike LSTMs, which use multiple gates and an internal memory cell to control the flow of information, GRUs use a single update gate to decide which information to retain and a reset gate to decide which information to discard. This makes GRUs simpler and easier to train than LSTMs, while still being able to achieve similar performance on many tasks .

In a study by , GRUs were shown to outperform LSTMs on the task of language modeling on the Penn Treebank dataset. In a comparison of NLP models by GRUs were found to be competitive with LSTMs and CNNs on several benchmarks. One advantage of GRUs is that they are able to capture long-range dependencies in sequential data more effectively than simple RNNs. This is because the update and reset gates in a GRU allow it to selectively retain or forget information from the past,

depending on the current input and the state of the network. This makes GRUs particularly well-suited for tasks that require the ability to remember and use information from long sequences, such as language translation.

Bi-Directional LSTM

A Bi-LSTM or bidirectional GRU (Bi-GRU) represented by, is a type of RNN that processes sequential data in both forward and backward directions. This allows the network to use information from both the past and future when making predictions or classifications. This can be particularly useful for tasks where the context of the current time depends on both past and future events. In a Bi-LSTM or Bi-GRU, two layers of LSTM or GRU cells are stacked together, and one layer processes the data in the forward direction while the other processes the data in the backward direction.

Hyperparameter Tuning

Hyper parameter optimization is a fundamental aspect that has a considerable effect on the efficacy of a machine learning algorithm. By selecting optimal hyper parameters, the algorithm's performance can be notably improved, leading to more precise predictions. The process of tuning the hyper parameters before the final run of the deep learning algorithm is crucial for ensuring optimal results. In the current study, the number of neurons in each layer, epoch size, and batch size were considered as the hyper parameters to be optimized. An epoch refers to a complete forward and backward pass of the entire dataset during the model's execution, while the batch size refers to the number of samples used in one forward/backward pass. It determines the number of samples that will be propagated through the network and updated the weights in a single iteration. Batch size is a hyper parameter that can affect model performance, and it can also affect the training time. A smaller batch size will result in more frequent weight updates but may lead to slower convergence, while a larger batch size may converge faster but may be computationally more demanding. For the following batch size 16, 32, 64, and 120 were used in the experiments. However, 120 was selected as the best hyper parameter, as it produced more accurate results for all prediction models used in this study.

1.7 The packages to implement the project

- .Pandas – This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.
- Numpy – Numpy arrays are very fast and can perform large computations in a very short time.
- Matplotlib/Seaborn – This library is used to draw visualizations.
- Sklearn – This module contains multiple libraries having pre-implemented functions to perform tasks from data preprocessing to model development and evaluation.
- XGBoost – This contains the eXtreme Gradient Boosting machine learning algorithm which is one of the algorithms which helps us to achieve high accuracy on predictions.

1.8 Methodology

Machine learning is an important branch of artificial intelligence (AI). According to whether there is a target variable, it can be divided into supervised learning, unsupervised learning, and reinforcement learning. The purpose of this study is to predict future Bitcoin prices, so a regression function with supervised learning is used. The unified execution logic of machine learning is that after the algorithm is preset, a learner is generated, and a high-precision learner is obtained by repeated training of the learner through training data and the process of validation. Finally, the test data is substituted into the trained learner for evaluation and application.

Both random forest regression and LSTM model training in this paper are implemented through the open-source library of python's machine learning. The library used by random forest regression is sklearn, and LSTM uses keras for research. The pre-processing and collation of the data are done by pandas.

Random Forest

Random forest is an ensemble form of multiple regression trees. Its advantage is high explicability, but the predicted results are limited by the training samples. The principle theof the regression tree is to divide the parent group into subgroups using a certain

$$\frac{1}{n_1} \sum_{i=1}^{n_1} (y_i - \overline{y_{(1:n_1)}})^2 + \frac{1}{n_2 - n_1} \sum_{j=n_1+1}^{n_2} (y_j - \overline{y_{(n_1+1:n_2)}})^2 \rightarrow \min$$

variable, and the classification is based on making the average sum squared residuals of each group the smallest.

Regarding parameter settings, the maximum depth of a single sub-regression tree is 10, and the number of sub-regression trees in the random forest is 500 . I tested the maximum depth of the interval [min = 3, max = 20] and the number of sub-regression trees of the interval [min = 200, max = 1000], respectively. My further experiments show that after the maximum depth is greater than 10 or the number of sub-regression trees is greater than 500, the training data and the prediction error no longer changes.

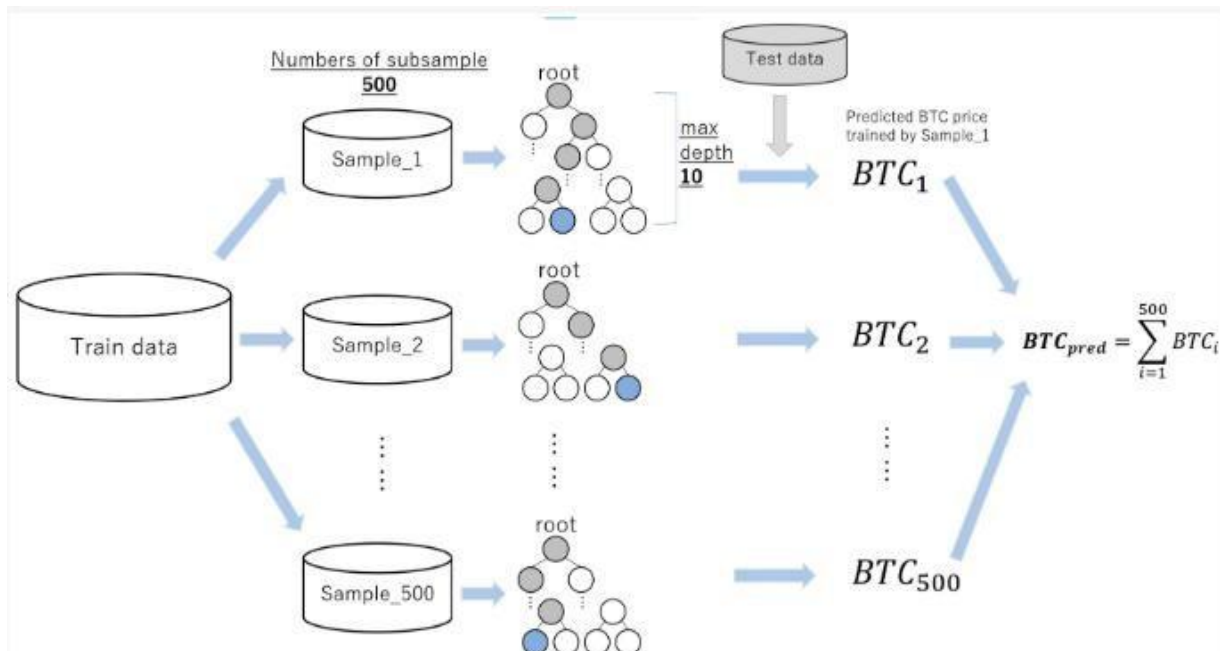


Fig:1.8.1-Random Forest

LSTM

The RNN algorithm is different from the normal DNN algorithm. When data is substituted into the model, it will not only generate an output value, but also modify the parameters of the model. RNN algorithm has the function of retaining the previous input data information in the model. This paper uses the LSTM model that makes up for the short memory defect of RNN. Data changes are made to the RNN model and the memory model through the paths of the three activation functions of Forget Gate, Input Gate, and Output Gate.

Based on the characteristic that the output value of the LSTM model can be resubstituted into another layer of the LSTM model, and the application of the dropout layer mentioned in the literature, the LSTM model structure of this experiment is as follows. Regarding the parameter setting of the dropout layer, I tested [min = 10%, max = 50%] for each dropout layer. It turns out that when the overall value of dropout is small, there is an over learning phenomenon in which the training data performs well but the prediction error of validation data is large. When the overall value of dropout is set too large, the errors of the training data and the validation data are both large. In addition, the experiment also found that the prediction accuracy of the dropout value with descending order is worse than ascending order. The number of layers of LSTM [min = 2, max = 6] and the parameter setting of each layer of units in [32, 64, 128, 256, 512] are tested. After balancing the accuracy and the risk of over learning. The activation function of each layer is set to “ReLU”, which has better performance than “sigmoid” and “tanh”. The specific value and framework of LSTM The last 10% of the training data is set as validation data.

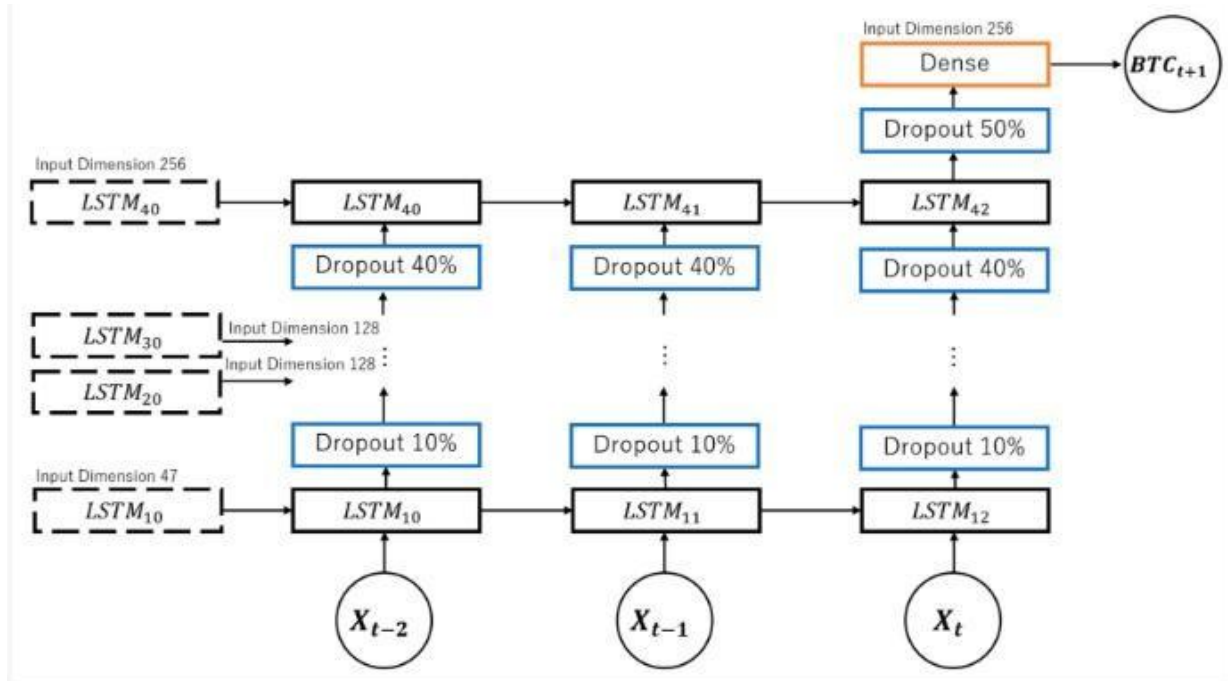


Fig:1.8.2-LSTM

In addition to the framework setting of the model, another important hyper parameter of deep learning is epochs. The value of epochs reflects the number of passes to learn the train data. The larger the epochs are, the smaller the prediction error of the training data will be. However, when the epochs are too large, it leads to over learning problems. Therefore, through the training and validation loss diagrams of Period 1 and Period 2 in **Figure 3** below, the epochs of Period 1 are set to 250, and the epochs of Period 1 are set to 75.

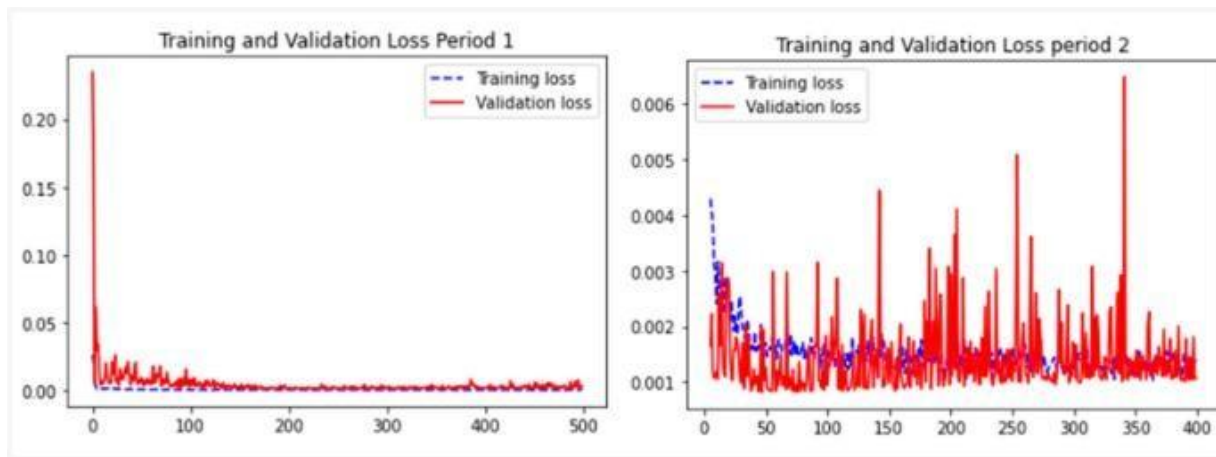


Fig:1.8.3-Training and Validation loss of LSTM

Errors and Evaluation Criteria

As an important criterion to evaluate the prediction accuracy of machine learning, this study quantifies the prediction performance of the model by using three errors, MAPE (mean absolute percentage error, and RMSE (root mean squared error, and DA (decision accuracy. However, due to the rising average Bitcoin price, RMSE can only be compared for model results based on the same sample. There is no meaningful comparison between the experimental results of different data samples,.

$$MAPE = \frac{1}{m} \sum_{t=1}^m \left| \frac{y(t) - \hat{y}(t)}{y(t)} \right|$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{t=1}^m (y(t) - \hat{y}(t))^2}$$

$$DA = \frac{1}{m} \sum_{t=1}^m a(t) \times 100\%$$

In addition to comparing the prediction accuracy of various models to obtain the performance of each model in predicting the future price of Bitcoin, this study also expects to compare the prediction errors under different lags of explanatory variables to analyze the memory length characteristics of the Bitcoin market.

In addition to the MAPE, RMSE, and DA errors of each prediction result, this paper also conducts a hypothesis test on the significant difference between the two different algorithms through the Diebold–Mariano test and the Clark–West test.

Data and Preprocessing

The sample data are the daily data from 31 March 2015 to 1 April 2022. The data of the study were collected from yahoo finance, Coinmarketcap.com, investing.com, bitinfocharts.com, and coinmatrics.io.

The target variable in the experiment is the price of Bitcoin in USD. A total of 47 variables are used as explanatory variables to predict the price of Bitcoin in the future, which are divided into eight categories: Bitcoin price variables, the specific technical features of Bitcoin, other cryptocurrencies, commodities, market index, foreign

exchange, public attention, and dummy variables of the week.

Each explanatory variable and its corresponding definition .

Explanatory Variables Analysis

The statistical features for each explanatory variable used to predict Bitcoin's future price. It is worth noting that the standard deviations of the variables related to the cryptocurrency market (five for Bitcoin, five for other cryptocurrencies, and Google search volume for Bitcoin) are all large. Among them, the ratio of the deviation to the mean value, except for the LTC of 0.99, all the others exceed 1. It reflects the high volatility of the cryptocurrency market since 2015. Except for the variables mentioned above, which are related to cryptocurrency, the value of standard deviation/mean ratio of the traditional market is not greater than 0.4.

In addition, differences between the explanatory variables of the cryptocurrency market and the traditional market were observed in terms of the ratio of the minimum and maximum values. Except for 194 times the Russian ruble in traditional markets, the max/min ratio is not greater than 7 (Regardless of the extremely negative price of -37.63 for crude oil on 20 April 2020). However, in the cryptocurrency market, the ratios are all greater than 300, and the highest is 11,067 times that of ETH. Both the Bitcoin market and the Russian ruble in the traditional market have shown high volatility.

The correlation heat map shows the correlation between Bitcoin and other explanatory variables except for the week dummy variables. Bitcoin has a positive correlation with other cryptocurrencies, commodity prices, stock market indexes, and public attention variables. The only exception is that the price of Bitcoin is inversely correlated with the 10-year U.S. Treasury yield in the commodities category. The price of Bitcoin and the exchange rate generally show a negative correlation. It seems understandable that the stronger the US dollar, the lower the price of Bitcoin. Interestingly, the Russian ruble exchange rate has a positive correlation with the Bitcoin price, and the correlation coefficient is high.

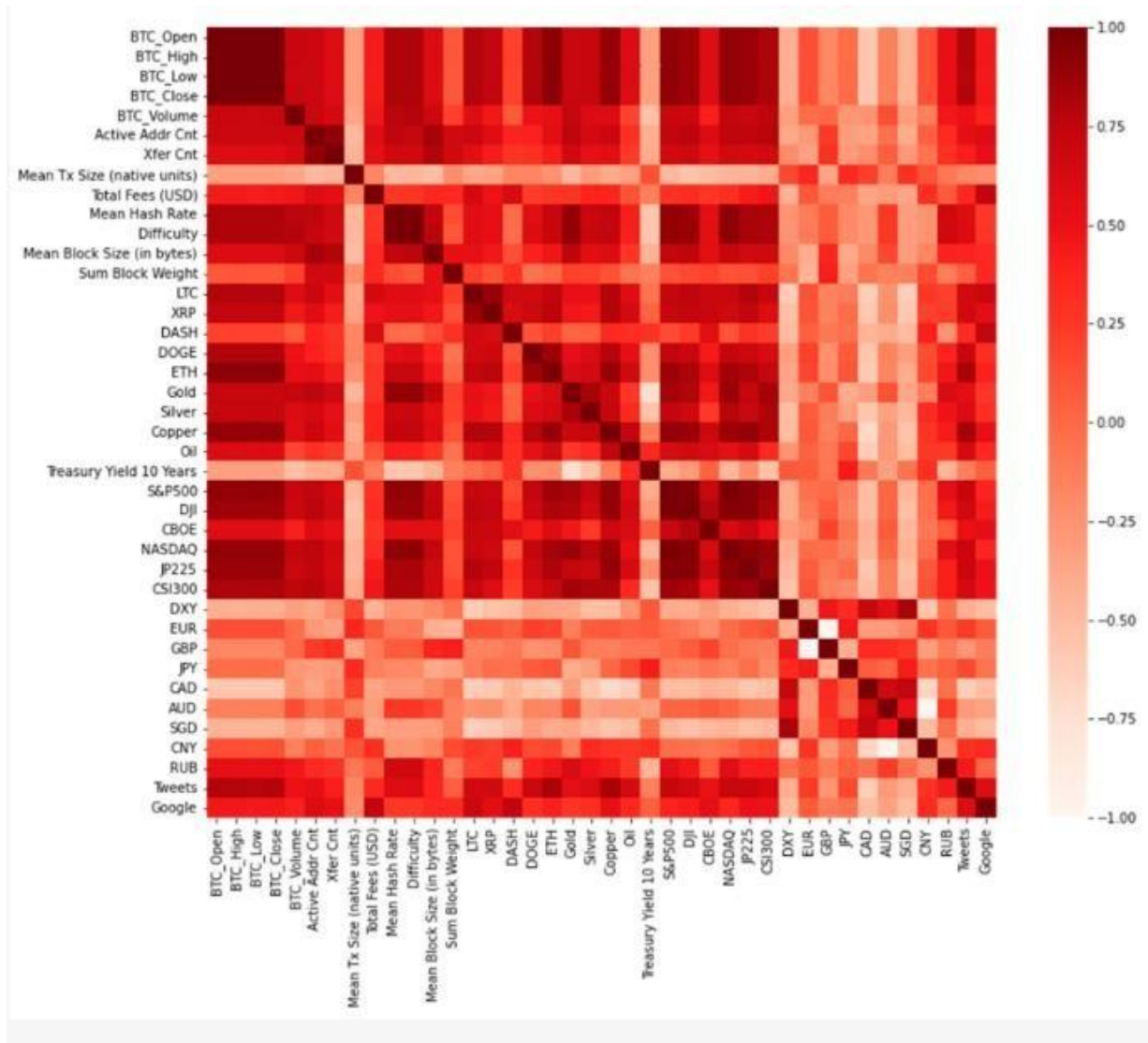


Fig:1.8.4-Correlation heatmap of explanatory variables

There is a brief explanation of the relationship between Bitcoin price and weekday variables. The extreme floats are mostly found on Wednesdays. The largest yield variance was seen on Wednesday and the largest daily gains and daily losses over the 7 years both occurred on Wednesday. The variance of yields is smaller on weekends than on weekdays, and yield fluctuations are more stable. The average daily return for Bitcoin is 0.28% with a 95% confidence interval of [0.13%, 0.43%]. The average return is highest on Mondays and smallest on Sundays. Monday's return is statistically greater than Sunday's ($\alpha = 95\%$). The daily probability of rising is 54.57% with a 95% confidence interval of [52.64%, 56.50%]. Saturday and Friday have the highest probability of rising. The probability of rising on Saturday is statistically greater

than on Sunday ($\alpha = 90\%$).

Regarding the two public attention variables, two conclusions can be drawn from the comparison with the Bitcoin price. First, the spike in Google Trends and daily Tweets came during a time when Bitcoin broke its all-time high price. Secondly, the highest Google Trend occurred at the end of 2017. After that, even with over USD 60,000 in 2021, the search volume did not surpass what it was at the end of 2017.

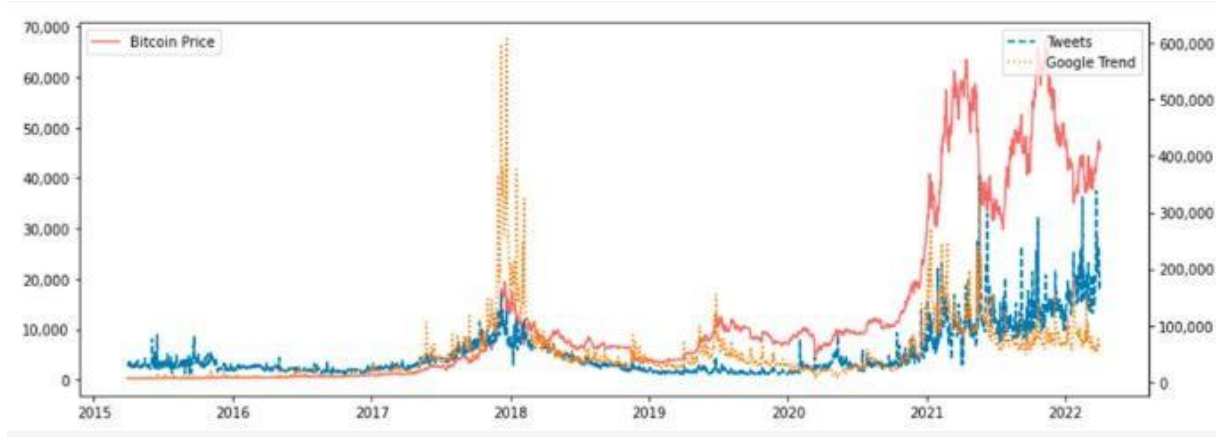


Fig:1.8.5-Bitcoin Price Chart

Preprocessing

The data research sample collected data from a total of 7 natural years from 31 March 2015 to 1 April 2022. However, due to the particularity of Bitcoin having two price bubbles at the end of 2017 and 2021, and the longest span of a single sample in past studies is no longer than 4 years. Based on the above two reasons, to improve the price prediction accuracy of the model, the total sample is divided into Period 1 (from 31 March 2015 to 30 September 2018) and Period 2 (1 October 2018 with 1 October 2018). Conduct independent research on two sub-samples, train models for their respective periods and predict respectively. Machine learning is the process of training initial samples through training samples and then substituting them into test samples for evaluation. Usually, training samples occupy 75% to 90% of the samples. The specific division of training and testing samples in this study is The last 10% of the training data is set as validation data.

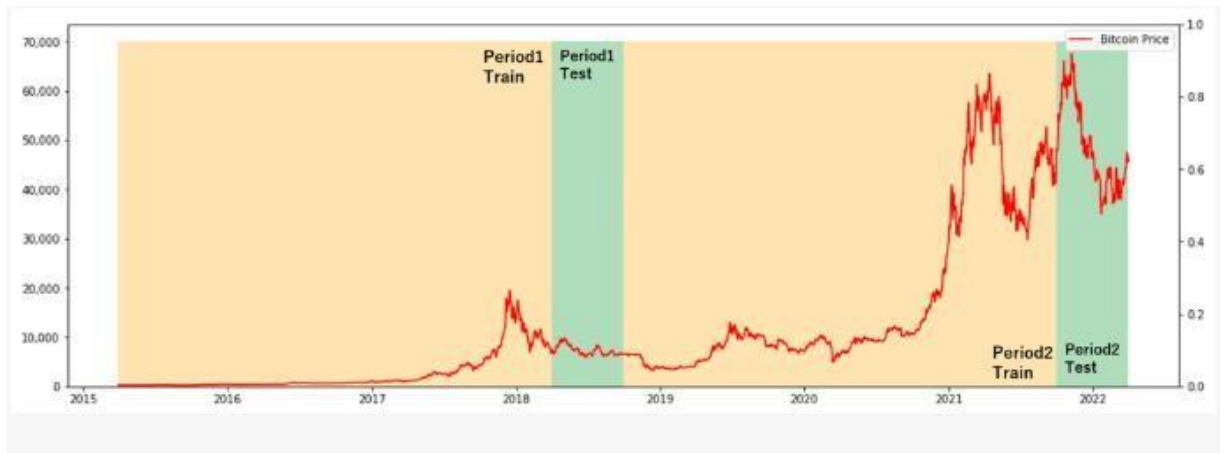


Fig:1.8.6-Interval division of training and testing samples

Among all the explanatory samples, only ETH has the problem of missing sample data because it came out (7 August 2015) later than April 2015, so the training samples used for ETH in the Period 1 model all start from 7 August 2015, not 31 March 2015.

Bitcoin is available for trading 24 h a day and 365 days a year, while the variables such as stock market indices, exchange rates, and commodity price indices are not traded during weekends and holidays, so there is missing data. There are two ways to deal with samples with these missing data, one is to delete the data with missing data before training, and the second method is to fill in the missing data. Considering that the research object of this study is time-series in nature, direct deletion of the samples affects the analysis of the period relationship. Therefore, filling in the missing data is chosen by replacing the value of the missing data with the value of the previous period. For example, in the case of gold prices, there is no price data for the weekend, and the value of the Friday gold price from the previous day is used to define the price for both days of the weekend.

The min/max preprocessing is important for LSTM because the activation function is not sensitive to values above 1. All variables are unified to [0, 1], eliminating the effect of metric units.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

The flow of the whole experiment is shown in **Figure 7**.

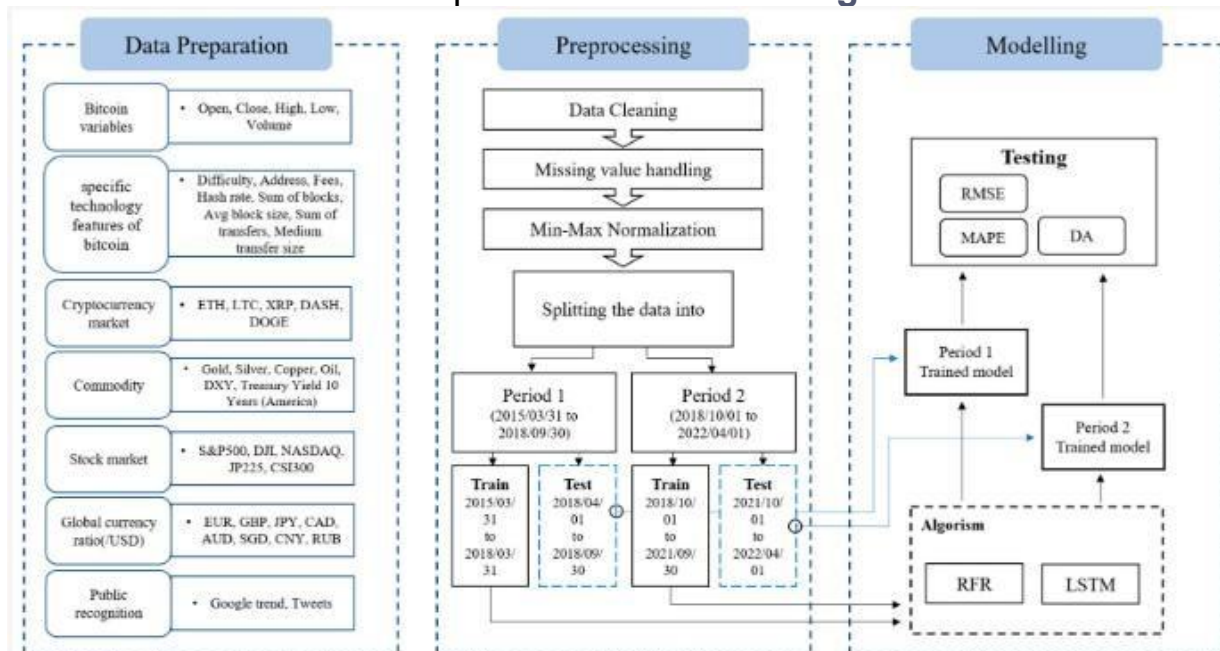


Fig:1.8.7-pictorial representation of entire experiment

CHAPTER 2

Literature survey

2.1 Survey

The features of the quick expansion in the interest in involving AI methods at foreseeing cryptographic money costs . When it comes to modeling the complex and volatile nature of cryptocurrency markets, the authors point out that conventional approaches like time-series analysis and econometric models have limitations. To capture nonlinear relationships and patterns in cryptocurrency price data, researchers have turned to machine learning algorithms like decision trees and Support Vector Machines (SVM). The creators bring up that while these strategies have shown promising outcomes, they additionally have limits. Neural networks, for instance, are susceptible to overfitting and may not adapt well to new data. Also, conventional brain networks expect that the info information is deterministic and commotion free, which may not be the situation in digital currency markets. The authors suggest employing stochastic neural networks, which incorporate randomness into the neural network architecture, to circumvent these limitations. This haphazardness permits the organization to catch the intrinsic vulnerability and instability of cryptographic money markets, prompting more exact forecasts. The creators refer to a few past works that have involved stochastic brain networks for monetary time-series gauging and feature their progress in catching the unpredictability and vulnerability of monetary business sectors. The correlation of the exhibition of Bitcoin and gold as speculation resources . The writing overview surveys the current writing on the subject, featuring the likenesses and contrasts among Bitcoin and gold as speculation resources. The creators note that both Bitcoin and gold are decentralized, have restricted supply, and are not expose to government intercession or control. Be that as it may, Bitcoin enjoys a few upper hands over gold, including its simplicity of move, lower capacity expenses, and straightforwardness. Gold, on the other hand, is less volatile than Bitcoin and has a long history of being used as a store of value. The creators additionally talk about past examination on the relationship among Bitcoin and other monetary resources, like stocks and bonds. They note that while Bitcoin is for the most part uncorrelated

with customary resources, there have been examples where it has displayed huge relationship during times of market pressure. It presents an experimental examination of the presentation of Bitcoin and gold in a portfolio setting. The creators use information from 2011 to 2017 to look at the connection, instability, and portfolio execution of Bitcoin and gold. They find that Bitcoin has higher unpredictability and relationship than gold, yet in addition better yields. Furthermore, the creators find that a portfolio comprising of both Bitcoin and gold beats a portfolio comprising of either resource alone, with lower unpredictability and better yields. The audit of blockchain innovation and its likely applications and difficulties in the monetary area. The writing overview in the paper features the current situation with research on blockchain innovation and its different use cases. It takes note of that blockchain innovation is a disseminated, and record framework that is decentralized and empowers exchanges that are straightforward, and secure without the requirement for go between. It surveys the different parts of blockchain innovation, like agreement instruments, shrewd agreements, and cryptography, and talks about how they add to the security and unwavering quality of the framework. In the methodology which we are following, the LSTM will utilize the past information to anticipate bitcoin costs 30 days in front of its end cost. In the methodology utilized by us, we carry out Bayesian improved Repetitive Brain Organization (RNN) and a Long Momentary Memory (LSTM) organization. With a RMSE of 8% and an accuracy of 52 percent, LSTM has the highest classification accuracy. By and by we execute the popular Auto in reverse consolidated moving typical (ARIMA) model for time plan checking as a relationship with the significant learning models. The ARIMA estimate is outperformed by the nonlinear profound learning strategies which performed much better. Finally, CPU and GPU benchmarks are used to evaluate the two profound learning models. The preparation time on the computer processor is defeated by the GPU execution by 67.7%. In the base papers chosen by us, the creator gathered an informational index of north of 25 highlights connecting with the bitcoin cost and installment network over a time of five years, recorded consistently had the option to foresee the indication of the day to day bitcoin cost change with an extraordinary precision of 98.7%.

CHAPTER 3

Requirements analysis

3.1 Objective

Bitcoin trade is Mt Go. A digital wallet, which is similar to a virtual bank account, is where bitcoins are kept. The record of all the exchanges, the timestamp information is put away in a spot called Blockchain. Each record in a blockchain is called a block. Each block contains a pointer to a past block of information. Blockchain data is encrypted. During exchanges the client's name isn't uncovered, yet just their wallet ID is unveiled. Bitcoin (BTC) is an original computerized cash frame work what capabilities without focal administering authority. Instead, a peer-to-peer network of Internet-connected users handles payment processing. Bitcoin clients declare new exchanges on this network, which are confirmed by network hubs and kept in a public conveyed record called the blockchain. In terms of total market capitalization, Bitcoin is the largest of its kind. They are made as a prize in a rivalry in which clients offer their processing ability to confirm and record exchanges into the blockchain. Bitcoins can also be exchanged for goods, services, and other currencies. The trade of the Bitcoins with different monetary standards is finished on the trade office, where "purchase" or "sell" orders are put away on the request book. " Purchase" or "bid" offers address an expectation to purchase specific measure of Bitcoins at some cost while "sell" or "inquire" offers address an expectation to sell specific measure of Bitcoins at some cost. The trade is finished by matching requests by cost from request book into a substantial exchange between purchaser

3.2 Hardware requirements

- Computer or Pc
- Knowledge about deep learning models
- Good internet connection

3.3 Software requirements

A portion of the structures utilized in this venture are talked about beneath. This task assembled fundamentally on Python. Python is a high level programming language, which is extremely productive when attempting to fabricate AI calculations. It has a lot of open source libraries created by third parties like Google, for example, because it is an open source language. These libraries can make it easier to build complicated programs and algorithms. Complex projects can be written in more limited lines of code in python when looked at to Java or other article arranged programs because of python's particular elements. Additionally, you can code on a wide range of platforms with it.- One more language utilized in this venture is Django. Django is the off the rack, go-to stage for creating python based web applications. It is a system worked for python web improvement. Django is built on Python as well, so it can do most things if the right libraries are installed. Python software engineers think that it is more adaptable and agreeable to code in Django as the coding climate continues as before for them most part. Anaconda Navigator is a Python distributor that is open source. It centers around giving

- IDE's and programming conditions for information science and AI. Boa constrictor is broadly utilized in view of the custom bundles that have been constructed. It is viable with Windows,
- Linux also, MacOS. Anaconda also provides support for R programming development and has a large community of R and Python developers.- Spyder is one of the stage incorporated advance mentclimate (IDE) accessible on Boa constrictor. It facilitates the creation of machine learning algorithms and gathers the robust libraries needed for data analysis.- The brain networks based on in this undertaking were finished utilizing the Keras libraries. Keras provides an API for neural networks that can be used
- Tensorflow or Theano. Keras was chosen because it has APIs that are easy to use and can support multiple CPUs and GPUs. Keras makes prototyping go smoothly.
- Keras, like all python libraries, uses the modularity idea to provide users with independent modules that can be changed. These modules are additionally

adaptable permitting the designers to make new and more powerful model to suit their necessities. Python developers don't have any trouble debugging or running complex modified code because all of the code is written entirely in Python.

CHAPTER -4

Design and description of proposed product

4.1 Ideation map

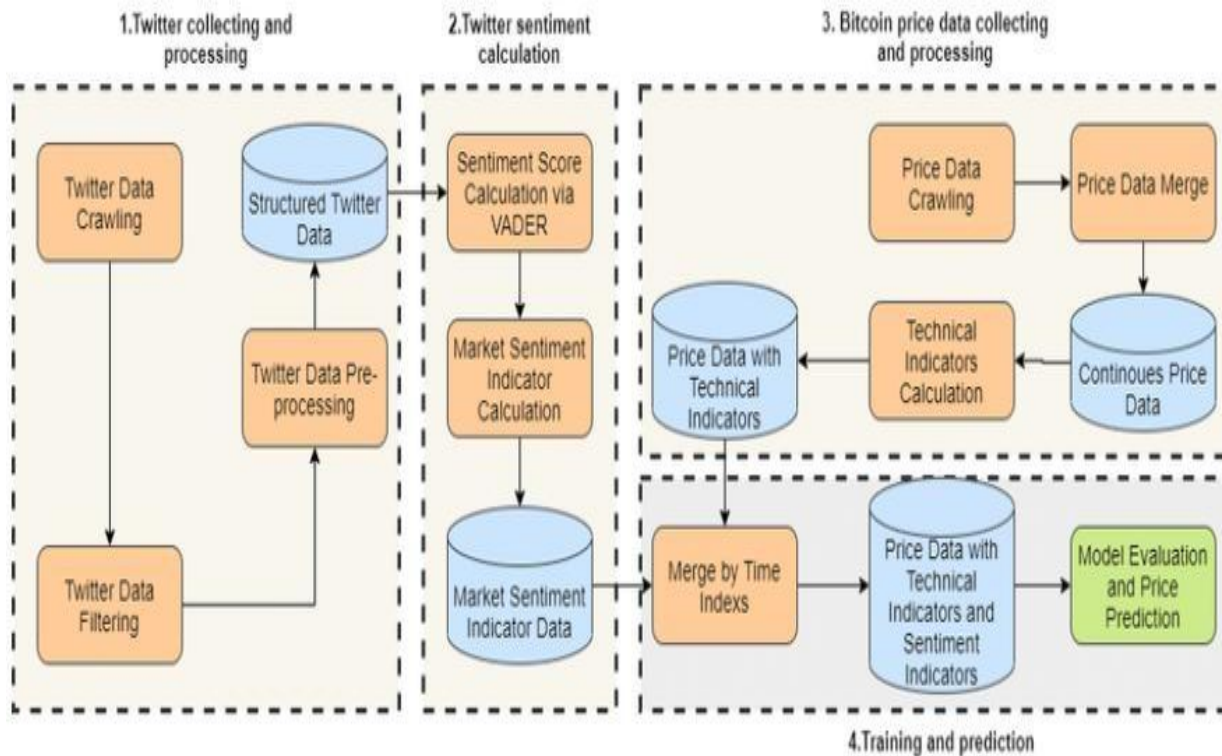


Fig:4.1.1-Ideation Map

We select and give mass sneak peaks or information sensors as understood passages. Basically, the more we bring the improved outcomes we hope to get. The model generates a graph with a date on the x-axis and a stock price on the y-axis to illustrate the trend. It can pull out a graph the following day and see the stock's estimated value with a planned graph. Subsequently, finding the closest cost of a resource is very essential to get a decent data set that has been endlessly handled utilizing different methods. Consequently the preparation information we will give to our model is vital in deciding the degree of exactness of the proposed framework.

Accordingly, to construct a precise model, we want to download the right data set as preparing information to the model

4.2 various stages

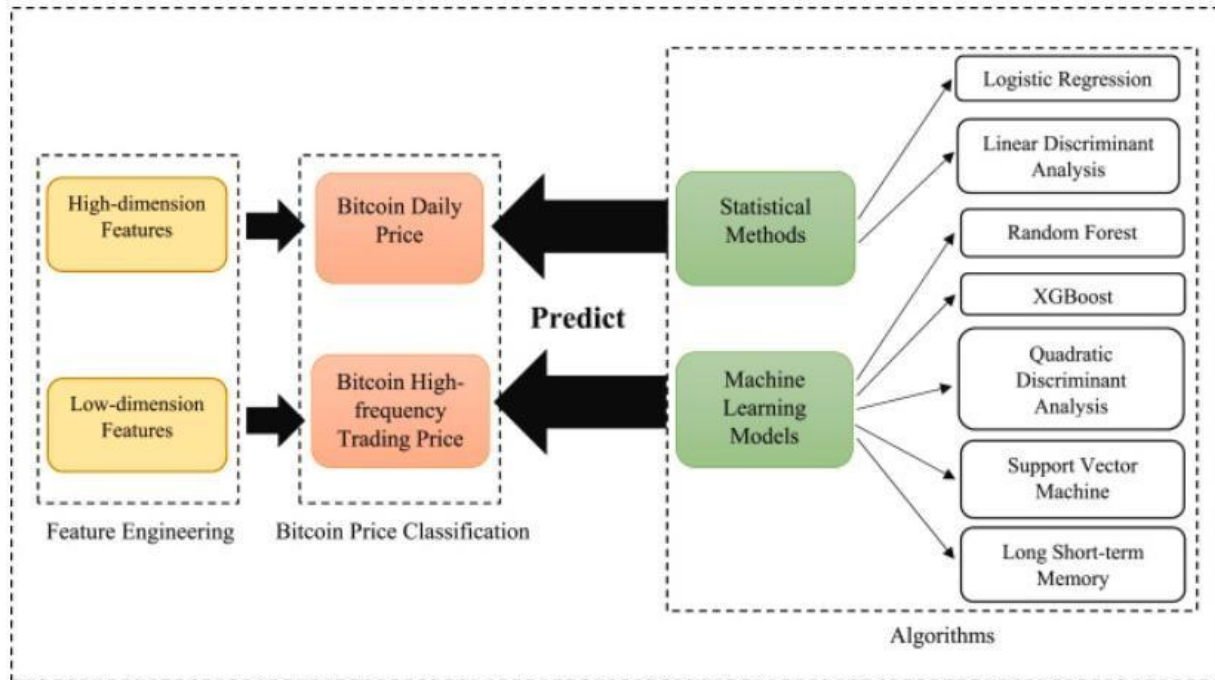


Fig:4.1.2-Various Stages

The proposed strategy considers two unique profound learning-based prediction models to figure day to day cost of bitcoin by recognizing and assessing relevant features by the actual model. We can determine which model is significantly more accurate for the future fulfillment of our target by applying both models to bitcoin prediction and selecting the appropriate parameters to improve performance. In this work, we have proposed profound learning systems, for example, LSTM and GRU which are the most recent and efficient strategies for the determining of bitcoin cost. As bitcoin is the most famous digital currency, the cost unpredictability issue ought to be taken care of with in a brief time frame.

4.3 Working principles

- First, collect the data set using the Rest-API to collect the historic of the Bitcoin prices from the online database.
- Arrange the data into the data frame according to the problem definition, to get the analysis correct and produce the results which are efficient to meet the goalsof the system.
- Then the rows of the dataset which are outdated for analysis/prediction to build a model and in order to feed the relevant data to the model extra columns are removed and stored in a CSV file.
- Then data-preprocessing is performed to missing values for the attributes, this is done to reduce the noise and inconsistency in the data.
- Then we build the model for the data-set using the LSTM (RNN) algorithm to predict the values of bit-coin on daily basis.
- Test the predictions with different layers of the RNN Model.

4.4 Novelty of the project

Since bitcoin investments are rising on a daily basis and the price of bitcoin fluctuates frequently, traders require a method to anticipate its price in order to mitigate risks and increase capital gains. The different existing chips away at cost expectation have low precision and foresee transient cost as it were. Because of the trouble of deciding the specific presence of a Period Series model, creating suitable forecasts is troublesome. The Proposed strategy utilizes profound figuring out how to anticipate bitcoin costs with Intermittent Brain Organization model utilizing the time series information to give the better precision. The oddity of work is to get a drawn out expectation, the repetitive brain network model is prepared and tried on the accessible dataset.

The LSTM model is a duplicate of a neural network that is able to learn long-term dependence. In the same way that we use previous data to inform future results, LSTM models use renewable gateways and forget gates to remember from time to time and forget pieces of historical information to inform their predictions. The

database used is different from previous versions. Details are timed and recorded at intervals (1 day or so). Such a sequence of data is called the Time Series. Temporary data sets are common to performance. Energy you use and your expenses (calories in, calories expired), climate change, stock market, statistics collected from users of your product / app and even your heart (probably in love) that produces the Time Series. You may be interested in a lot of architecture in relation to your Time Series - posture, time of year and automotive automation are some of the most popular features. Auto correlation is the interaction of data points separated by a period of time (known as lag). Seasonal means the discovery of a particular circular pattern at a certain time. The time series is called stationarity if it always has meaning and diversity. Also, the covariance is independent of time. One obvious question you might ask yourself when looking at Time Series data is: "Does the current action value affect the next one?" eg. Time Series Forecast. There are many ways you can use this purpose. But we will build networks for short-term memory that we predict and use to predict future Bitcoin prices. Included CNN also.

CHAPTER -5

CONCLUSION

Predicting the future will always be on the top of the list of uses for machine learning algorithms. Here in this project we have attempted to predict the prices of Bitcoins using two deep learning methodologies. This work focuses on the development of project based learning in the field of computer science engineering, by taking into account the problem definition, progression, student assessment and use of hands on activities based on use of deep learning algorithm to develop application which can predict bitcoin prices. Django was able to support the deep learning model and bring to life a graphical web app. The convolutional Neural Network though was primarily designed to study and classify images, was customized to predict a sequence of numbers. Though it fell short by having only 5% buffer amount, it was still able to compete with the LSTM model. Through this project, students were able to learn and understand the entire life cycle of App development. They also gained experience in building machine learning models and Web development using Django.

5.1 Reference

[1]

G. Solomon, "Project-Based learning: A Primer," Technology Learning, volume 23, Jan. 2003.

[2]

M. Hedley, "An undergraduate micro controller systems laboratory," IEEE Transactions in Education, vol. 41(4), pp. 345–353, Nov. 1998.

[3]

H. Markkanen, G. Donzellini, and D. Ponta, "NetPro: Methodologies and tools for project based learning in internet," in Proceedins of World Conference on Educational Multimedia, pp. 1230–1235.

[4]

D. Ponta, G. Donzellini, and H. Markkanen, "NetPro: Network based project learning in internet," in Proceedings of European Symposium of Intelligent Technologies, pp. 703–708, 2002.

[5]

P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar, and M. Alazab, "Stochastic neural networks for cryptocurrency price prediction," IEEE Access, vol. 8, pp. 82804–82818, (2020)

[6]

L. Herskind, P. Katsikouli, and N. Dragoni, "Privacy and cryptocurrencies—A systematic literature review," IEEE Access, vol. 8, pp. 54044–54059, (2020)

[7]

M. KubÆt, "Virtual currency bitcoin in the scope of money definition and store of value," Proc. Econ. Finance, vol. 30, pp. 409–416, (2015)

[8]

T A Muniye, Minakshi Rout, M Lipika, S Suresh," Bitcoin Price Prediction and Analysis Using Deep Learning Models", (2020)

[9]

B Agarwal, H priyanka, S Upkar, L Chouhan," Prediction of dogecoin price using deep learning and social media trends", (2021)

[10]

S. Nakamoto." Bitcoin: A Peer-to-Peer Electronic Cash System. [Online]", (2007)

[11]

Akila V, Sriharshini K, Sravani P, Sravanthi D, Gopi R, Sheela T., International Journal of Online and Biomedical Engineering, 17(1), 120-128, (2021).

[12]

M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," Bus. Inf. Syst. Eng., vol. 59, Vol. 3, pp. 183–187, (2017)

[13]

Prasanna Lakshmi K, Reddy CRK. ,"Int Conf. on Networking and Information Technology", (2010)

[14]

Sateesh. N, Sampath Rao.P, Ravishankar.D, V. Satyanarayana.K., "Materials Today: Proceedings." (2015)

[15]

ang, H. & Lee, J. [An Empirical Study on Modelling and Prediction of Bitcoin Prices with Bayesian Neural Networks based on Blockchain Information](#). An Empirical Study on Modelling and Prediction of Bitcoin Prices with BaNeural Networks based on Blockchain Information .

5.2 Appendix

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

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

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

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GRU, Conv1D, MaxPooling1D, Flatten
```

```
bitcoindf = pd.read_csv("/content/BNB-USD (1).csv")
bitcoindf = bitcoindf.rename(columns={'Date': 'date', 'Open': 'open', 'High': 'high', 'Low': 'low', 'Close': 'close',
                                     'Adj Close': 'adj_close', 'Volume': 'volume'})
bitcoindf.head()
```

```
bitcoindf = bitcoindf.fillna(method = 'ffill')
```

```
bitcoindf['date'] = pd.to_datetime(bitcoindf.date)
bitcoindf.head().style.set_properties(subset=['date', 'close'], **{'background-color': 'skyblue'})
```

```
fig = plt.figure(figsize = (15,10))
```

```
plt.subplot(2, 2, 1)
plt.plot(bitcoindf['date'], bitcoindf['close'], color="red")
plt.title('Bitcoin Close Price')
```

```
last1year_bitcoindf = bitcoindf[bitcoindf['date'] > '09-2020']
```

```
fig = plt.figure(figsize = (15,10))
fig.suptitle("Last 1 year close prices of Bitcoin", fontsize=16)
```

```
plt.subplot(4, 1, 1)
plt.plot(last1year_bitcoindf['date'], last1year_bitcoindf['close'], color="red")
plt.legend("B")
```

```
fig = plt.figure(figsize = (15,7))
```

```
plt.plot(last1year_bitcoindf['date'] , last1year_bitcoindf['volume'])
```

```
last1month_bitcoindf = bitcoindf[bitcoindf['date'] > '08-2021']
```

```
fig = plt.figure(figsize = (15,15))
fig.suptitle("Last month comparision of close and open prices of Bitcoin, Cardano, Dogecoin, Ethereum", fontsize=16)
fig.tight_layout()
```

```
plt.subplot(4, 1, 1)
plt.plot(last1month_bitcoindf['date'], last1month_bitcoindf['close'])
```

```
fig = plt.figure(figsize = (15,10))
```

```
plt.subplot(2, 2, 1)
plt.plot(bitcoindf['date'], bitcoindf['close'].rolling(50).mean())
plt.plot(bitcoindf['date'], bitcoindf['close'].rolling(200).mean())
plt.title('Bitcoin Close Price moving average')
```

```
fig = plt.figure(figsize = (15,12))
```

```
fig.tight_layout()
```

```
plt.subplot(4, 1, 1)
sns.histplot(bitcoindf['close'],color='darkred', kde=True)
plt.axvline(bitcoindf['close'].mean(), color='k', linestyle='dashed', linewidth=2)
plt.text(50,40,'Bitcoin Close Price', fontsize=16)
```

```
bitcoin_numeric = bitcoindf.select_dtypes(exclude = ["bool"])
bitcoin_numeric.hist(figsize=(18,12))
plt.show()
```

```
closedf = bitcoindf[['date','close']]
print("Shape of close dataframe:", closedf.shape)
```

```
closedf = closedf[closedf['date'] > '2020-09-13']
close_stock = closedf.copy()
print("Total data for prediction: ",closedf.shape[0])
```

```
del closedf['date']
scaler=MinMaxScaler(feature_range=(0,1))
closedf=scaler.fit_transform(np.array(closedf).reshape(-1,1))
print(closedf.shape)
```

```
training_size=int(len(closedf)*0.70)
test_size=len(closedf)-training_size
train_data,test_data=closedf[0:training_size:],closedf[training_size:len(closedf),:1]
print("train_data: ", train_data.shape)
print("test_data: ", test_data.shape)
```

```
fig, ax = plt.subplots(figsize=(15, 6))
sns.lineplot(x = close_stock['date'][:255], y = close_stock['close'][:255], color = 'black')
sns.lineplot(x = close_stock['date'][255:], y = close_stock['close'][255:], color = 'red')
```

```
# Formatting
ax.set_title('Train & Test data', fontsize = 20, loc='center', fontdict=dict(weight='bold'))
ax.set_xlabel('Date', fontsize = 16, fontdict=dict(weight='bold'))
ax.set_ylabel('Weekly Sales', fontsize = 16, fontdict=dict(weight='bold'))
plt.tick_params(axis='y', which='major', labelsize=16)
plt.tick_params(axis='x', which='major', labelsize=16)
plt.legend(loc='upper right', labels = ('train', 'test'))
```

```
def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]    ###i=0, 0,1,2,3-----99    100
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return np.array(dataX), np.array(dataY)
```

```
time_step = 17
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)
```

```
print("X_train: ", X_train.shape)
print("y_train: ", y_train.shape)
print("X_test: ", X_test.shape)
print("y_test", y_test.shape)
```

```
# Reshaping the input to be [samples, time steps, features] which is required for CNN
X_train_cnn = X_train.reshape((X_train.shape[0], X_train.shape[1], 1))
X_test_cnn = X_test.reshape((X_test.shape[0], X_test.shape[1], 1))
```

```
model_cnn = Sequential()
model_cnn.add(Conv1D(filters=64, kernel_size=2, activation='relu', input_shape=(time_step, 1)))
model_cnn.add(MaxPooling1D(pool_size=2))
model_cnn.add(Flatten())
model_cnn.add(Dense(50, activation='relu'))
model_cnn.add(Dense(1))
model_cnn.compile(optimizer='adam', loss='mse')
```

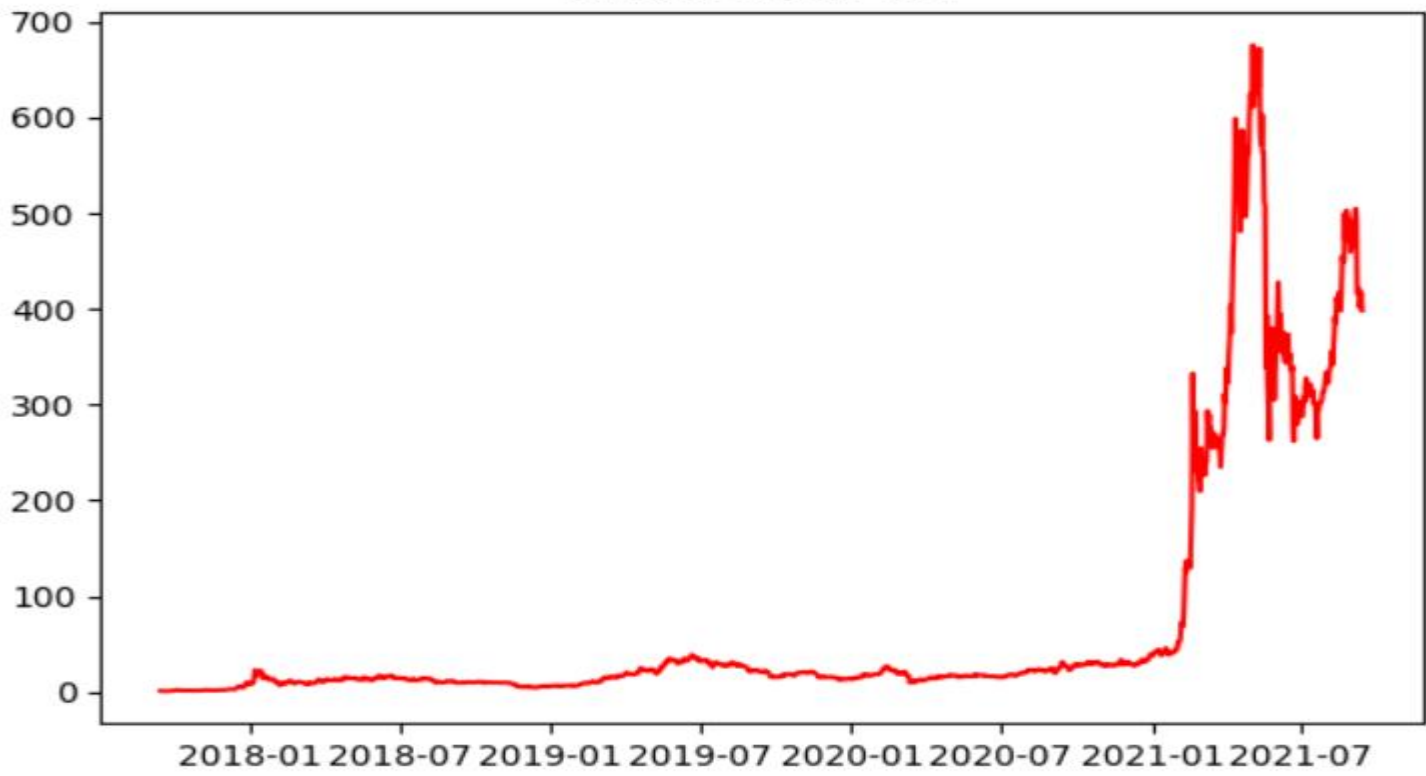
```
# Fit the CNN model
model_cnn.fit(X_train_cnn, y_train, epochs=100, verbose=0)
```

```
# Predictions
train_predict_cnn = model_cnn.predict(X_train_cnn)
test_predict_cnn = model_cnn.predict(X_test_cnn)
```

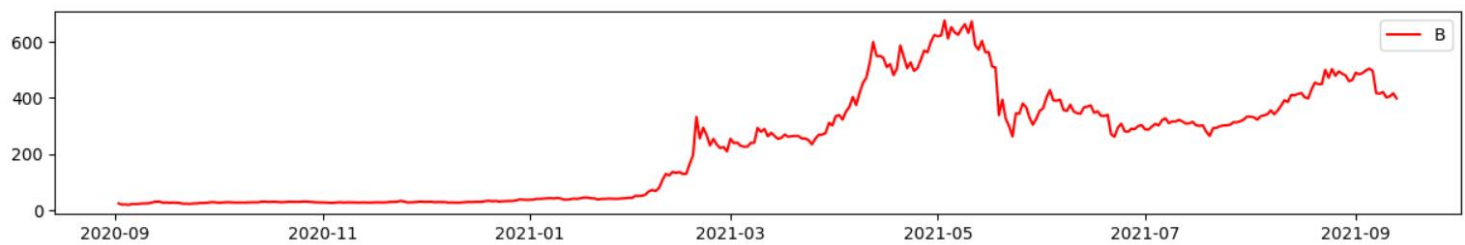
```
# Inverse transform predictions to actual values
train_predict_cnn = scaler.inverse_transform(train_predict_cnn)
```

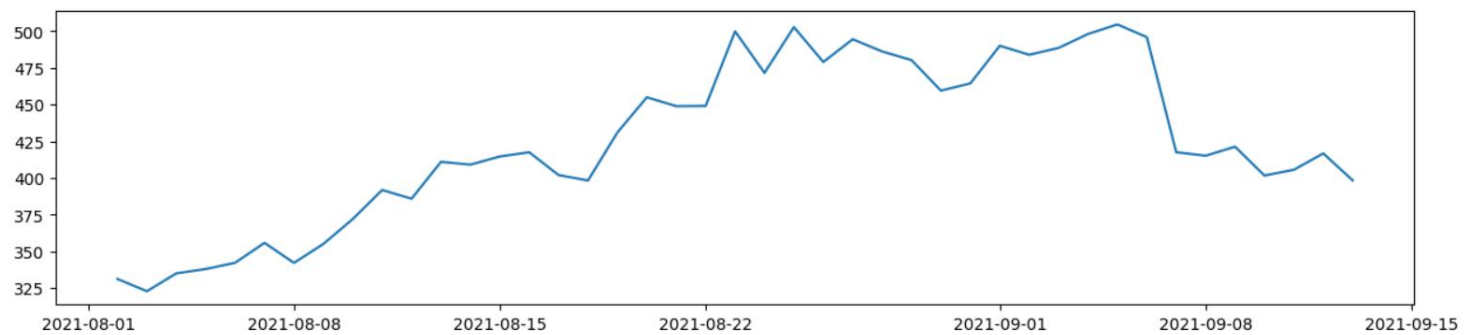
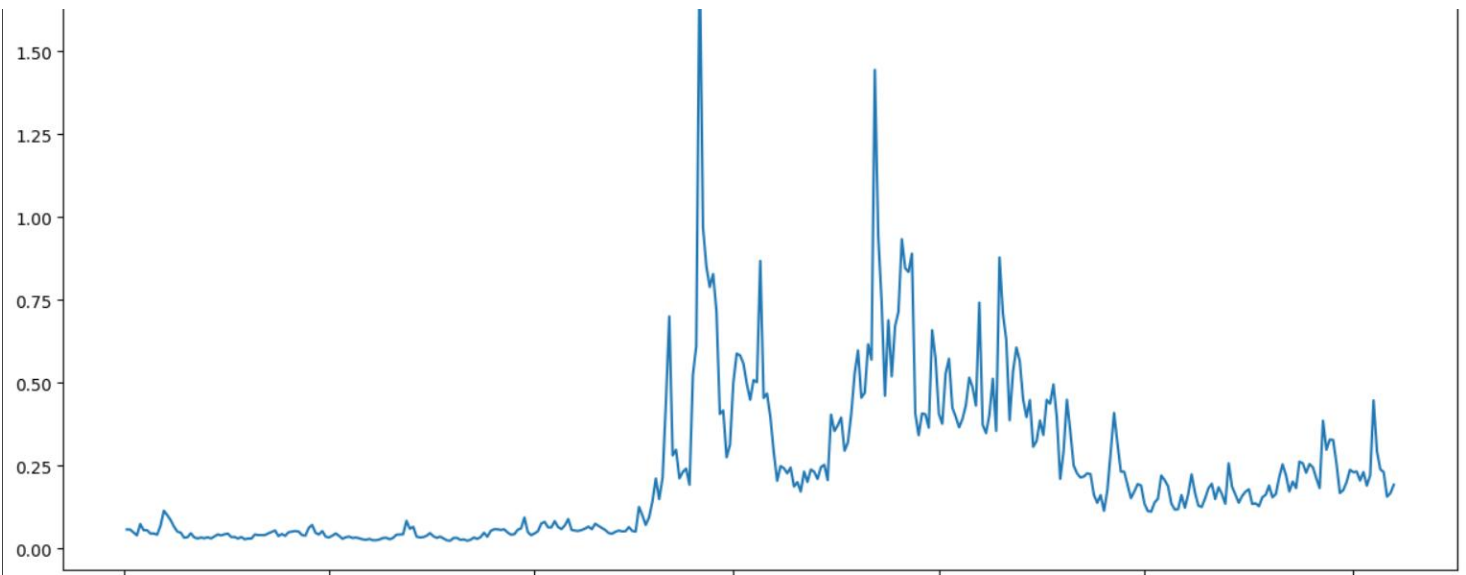
OUTPUT

Bitcoin Close Price

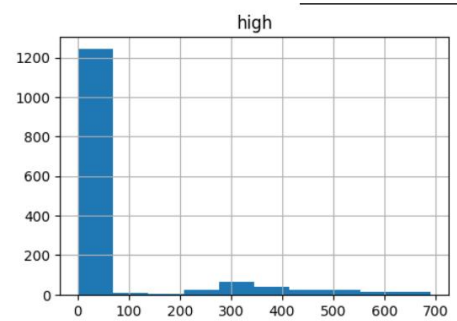
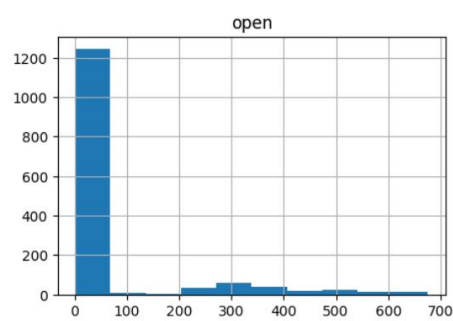
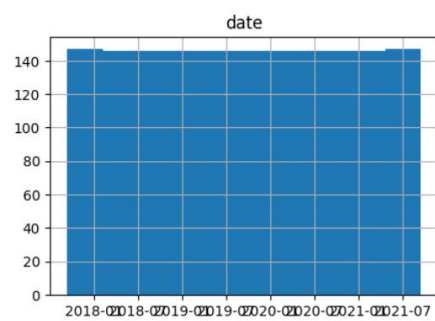
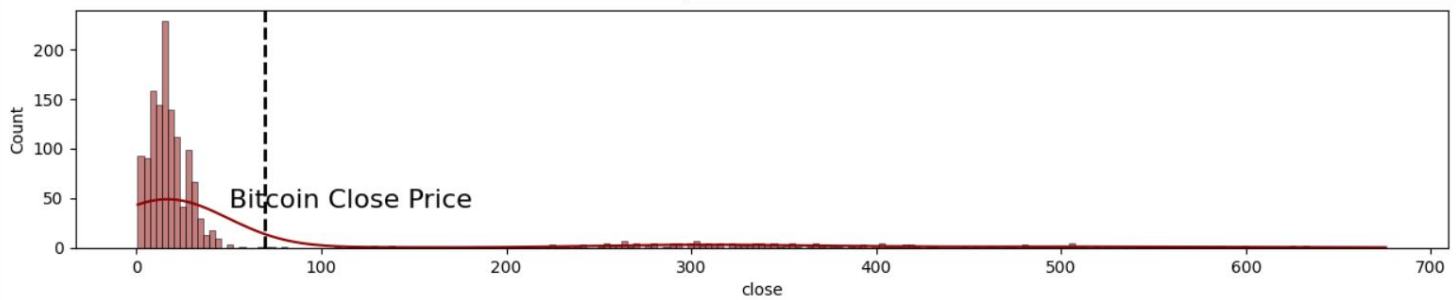
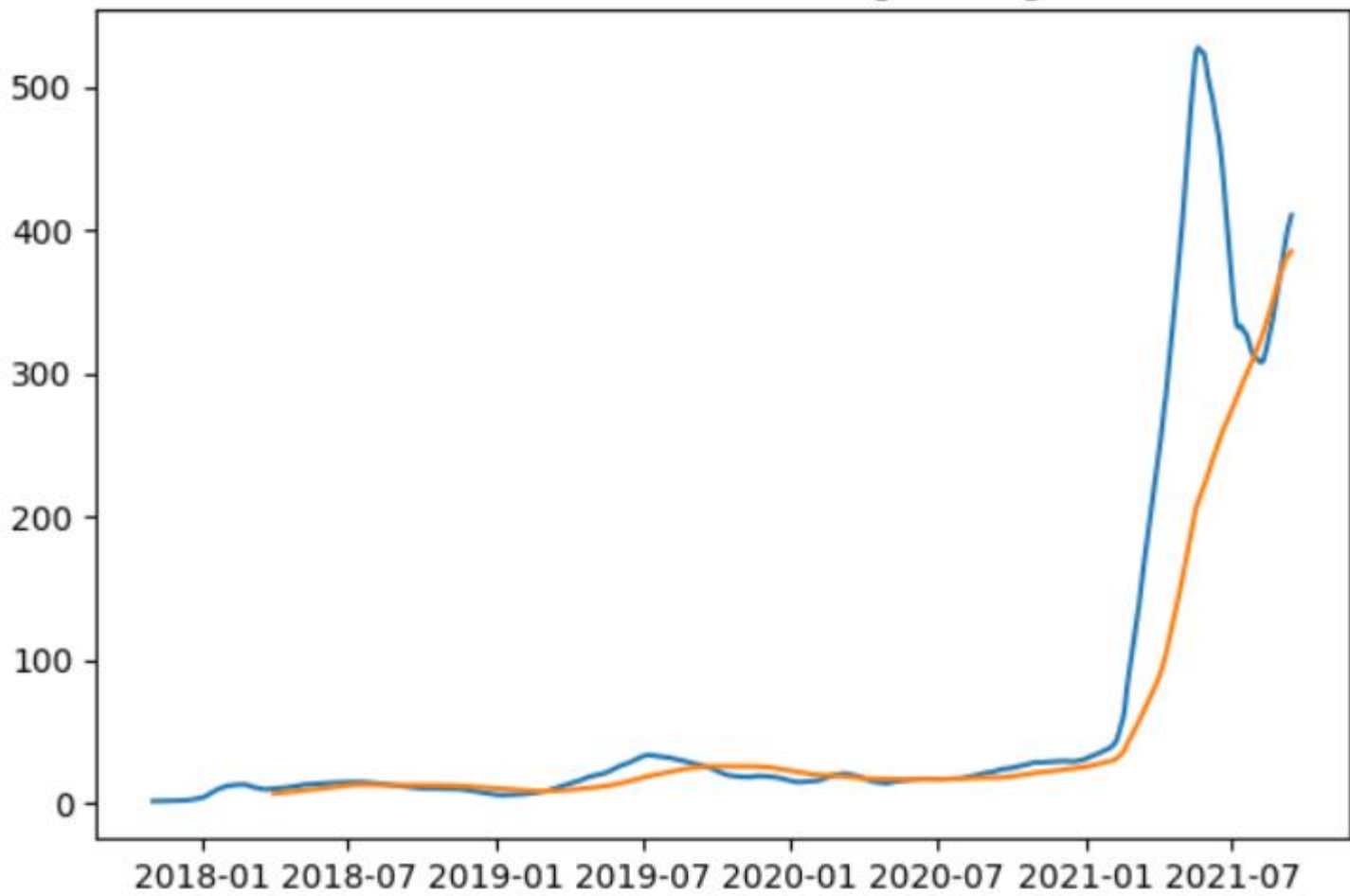


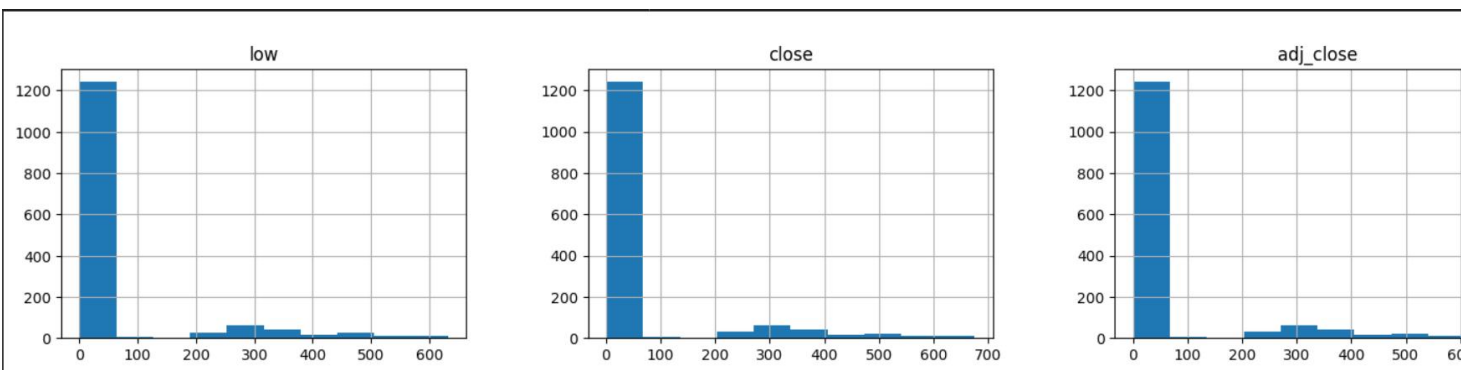
Last 1 year close prices of Bitcoin





Bitcoin Close Price moving average





```

Shape of close dataframe: (1462, 2)
Total data for prediction: 365
(365, 1)
train_data: (255, 1)
test_data: (110, 1)
X_train: (237, 17)
y_train: (237,)
X_test: (92, 17)
y_test (92,)
8/8 [=====] - 0s 3ms/step
3/3 [=====] - 0s 4ms/step

```

