University of Strathclyde

Department of Management Science

M.Sc. Financial Technology

“ Analytics of Cryptocurrency Data – Estimating Bitcoin Price by Machine Learning algorithms”

MS996 Management Science Project

WAN-TING TSENG

201984399

Submitted in partial fulfilment of the requirements for the degree of M.Sc. Financial Technology

August 2020

Abstract

This study first presents the development of cryptocurrency and narrow to the overview of Bitcoin. As the trend of cryptocurrency is getting upward, more and more investors are trading them. However, the volatility of Bitcoin is pretty fluctuated, it is quite difficult to predict the price by traditional methods such as Auto-Regressive Moving Average. This study aims to predict the Bitcoin price by secondary data. By using machine learning and deep learning models, it can show the performance of different algorithms. Multiple researches have been developed to use these algorithms to estimate the price of Bitcoin. In this study, different types of attributes are added as the input variables. The models used in this project include Linear Regression, Random Forest Regression and Long Short Term Model. The evaluation methods utilized in here are Root Mean Squared Error and Mean Absolute Error. This paper will compare the results of different models using these assessments. At the development of cryptocurrency is not completely mature, there are still some limitations for this study. In addition, adding more samples and applying more types of machine learning might help produce accurate results for the Bitcoin price.

Contents

Contents of Figure

Contents of Table

1. Introduction

Cryptocurrency can be seen as a popular electronic money, and its market capitalization grew over $100 billion in 2017 and kept increasing steadily (Elbahrawy, Alessandretti, Kandler, Pastor-Satorras, & Baronchelli, 2017). In 2008, Bitcoin was the earliest cryptocurrency to be developed to make transactions with decentralized system (Nakamoto, 2008). This system is called Decentralized Ledger Technology (DLT) and its main feature is that it does not need the exist of centralized system and has the ability to trace approved transactions between different parties for each ledger (Zargham, Zhang, & Preciado, 2018). The one used in Bitcoin is Blockchain. There are several places that can buy Bitcoin to make exchange for other currencies. For example, Bitstamp and Coinbase are the international place for Bitcoin transaction (Bitcoin.org, 2020). After the Bitcoin was launched, numbers of cryptocurrencies have been developed, such as Ripple, Ethereum, Litecoin and Cardano. However, much of them have no significant differences compared to Bitcoin, they just have the similar aspects which might borrow from it (Hileman & Rauchs, 2017).

In Figure 1.1, it is showed that the growth of market capitalization of cryptocurrency has increased rapidly since 2016 (Hileman & Rauchs, 2017). From same study, it is indicated that over 1,500 cryptocurrencies are bought and sold by investors worldwide. As Figure 1.2 shows the price of Bitcoin is fluctuated dramatically from almost zero to 20,000 (Blockchain.com, 2020b), it might be interesting to find out the variables that can have impacts on the price and how to predict it. Since the development of technology progress successfully, there are several deep machine learning tools to date can be used to make prediction on Bitcoin price. This study will mention the development of cryptocurrency and Bitcoin and make comparison on different models to forecast the Bitcoin price.

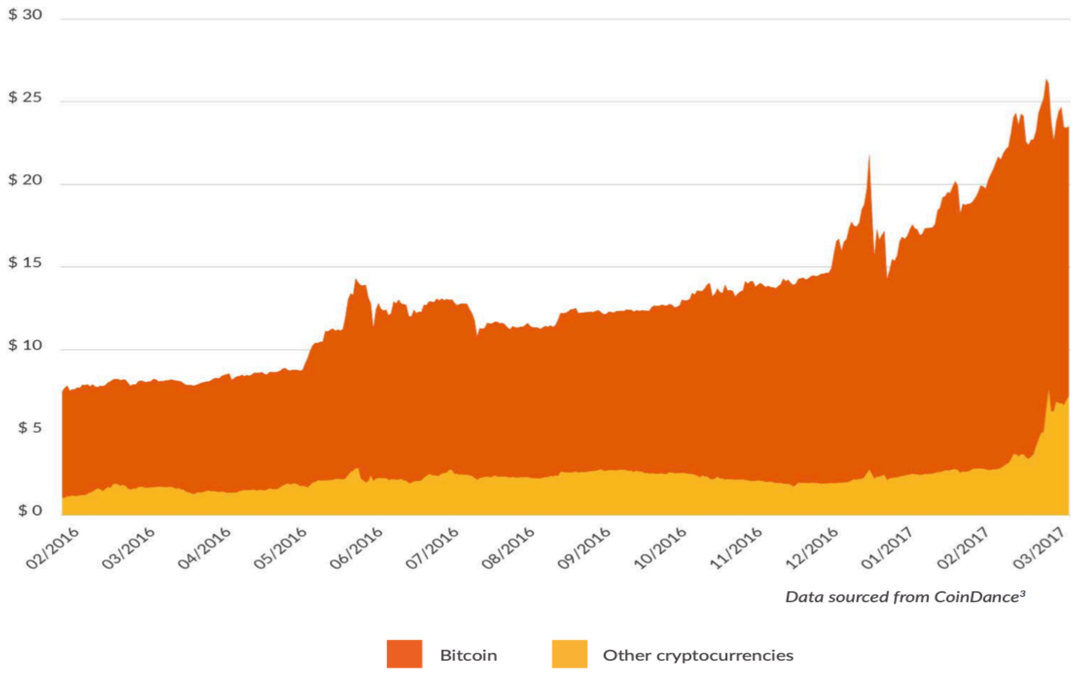


Figure 1.1 The market capitalization of cryptocurrency

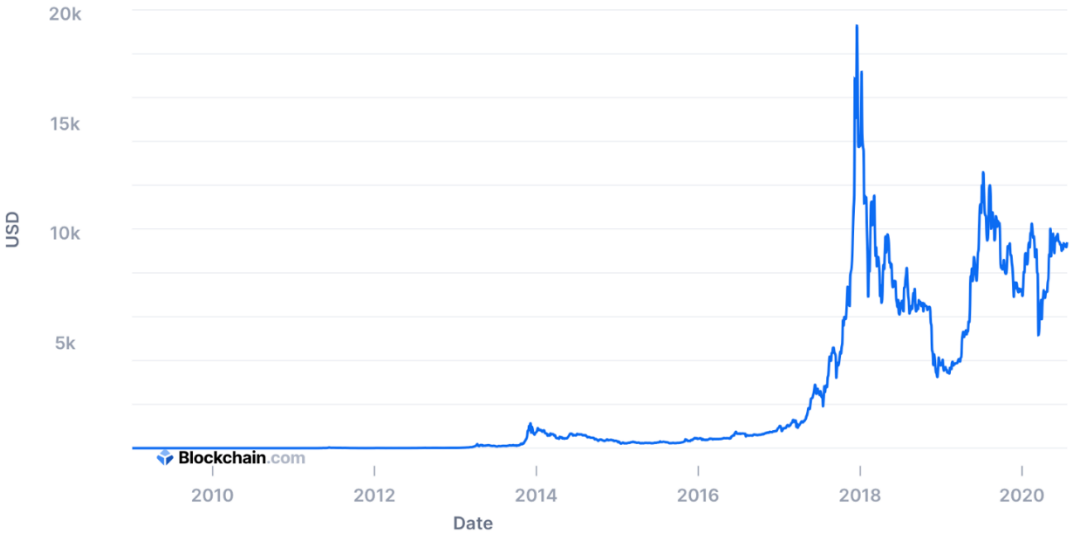


Figure 1.2 Market price of Bitcoin

2. Literature Review

2.1 Description of Cryptocurrency and Bitcoin

As the cryptocurrencies becomes more and more important and popular in the world, the price of them are hard to predict due to the dramatic fluctuations. As a result, it is essential to construct a model that can precisely forecast the price of cryptocurrencies for investors (Dutta, Kumar, & Basu, 2019). Satoshi Nakamoto launched Bitcoin which is an electronic currency and a system that can make exchange in 2008 and it was completely operated in 2019 (Ron & Shamir, 2013). Bitcoin took use of cryptography which is a smart incentive system to control the decentralized system. In the algorithm of Bitcoin, there is a mechanism that can recognize the fake currency, however there is still some risks when using the network (Gandal & Halaburda, 2016). Blockchain provides Bitcoin a network under decentralized system, however there are two disadvantages. Firstly, it takes almost an hour to complete the transactions. Secondly, miners need to spend huge computational workings to create a new block (Muzammal, Qu, & Nasrulin, 2019).

In some precious works for the prediction of the price of cryptocurrencies, they were aimed at price analysis and prediction by traditional methods that used for financial industry (Ciaian, Rajcaniova, & Kancs, 2016; Dutta, Kumar, & Basu, 2019). Factors like market beta, trading volume and volatility might have important influences for the price of cryptocurrencies in the short and long-term (Yhlas Sovbetov, 2018). Compared to traditional financial products, the forecast of cryptocurrency is different because of its volatility (Muzammal, Qu, & Nasrulin, 2019). The Bitcoin also shows a significant volatility (Lo & Wang, 2014). In prior research, it is indicated that Bitcoin is classified as a speculative asset and has no intrinsic and fundamental value, which are concerned by the public about the future of Bitcoin (Cheah & Fry, 2015; David Yermack, 2013). However, some people showed the opposite opinion. They thought Bitcoin is a money-like products and the fee of production model used to mine the Bitcoin can increase the fundamental value of Bitcoins (Hayes, 2017; Woo, 2013).

There are many variables that can be utilized to estimate Bitcoin price. Interestingly, it is found that the Google Search Engine has a high relationship to the Bitcoin price. In addition, the relationship is bidirectional, which means Bitcoin price is also strongly correlated to the query of the Google Search Engine (Kristoufek, 2013). Apart from the keywords searched for Bitcoin, the total transactions of Bitcoin also play an important role (Polasik, Piotrowska, Wisniewski, Kotkowski, & Lightfoot, 2015). In the long period, fundamental variables such as usage in transactions, supply and price level have significant effect on the price of Bitcoin (Kristoufek, 2015). Moreover, in terms of technical aspect, difficulty and hash rate seemed to have positive relationship with Bitcoin. However, it is not steady and showed less relative to Bitcoin (Kristoufek, 2015). Additionally, Bitcoin is regard as a safe haven just like the gold. In contrast, there is no similar dynamics between Bitcoin and gold (Kristoufek, 2015), and there is no sufficient evidence that Bitcoin is a safe haven investment (Bouoiyour & Selmi, 2015; Ciaian, Rajcaniova, & Kancs, 2016; David Yermack, 2013). In this project, besides from the variables mentioned above, more factors would be considered to forecast the Bitcoin price, including transaction fees, average Bitcoin block size, estimated transaction volume, Nasdaq Composite Index, Dow Jones Industrial Index and other cryptocurrency price (such as Ethereum, Litecoin).

2.2 Models Used to Predict Bitcoin Price

As Bitcoin becomes a well-known digital currency in recent years, there are many researches that are related to Bitcoin. It is found that Bitcoin has properties that combine the features of both commodity and currency, and it compares the difference between Bitcoin and other traditional financial products (Dyhrberg, 2016; Kinderis, Bezbradica, & Crane, 2018). Auto-Regressive Moving Average (ARMA) is a traditional tool used to predict time series data, and there are also some commonly used methods such as univariate Autoregressive (AR), Univariate Moving Average (MA) and Simple Exponential Smoothing (SES) (SIAMI-NAMIN & SIAMI-NAMIN, 2018). However, the traditional methods cannot be used to predict cryptocurrency because of its high volatility.

As machine learning and deep learning algorithms became popular, experts had started to predict the price of cryptocurrency or stock. In (Greaves & Au, 2015), it focused on variables that are related to network-based. The author applied Linear Regression and Support Vector Machine (SVM) algorithms and the Mean Squared Error (MSE) of them are 1.94 and 1.98 respectively. In addition, this paper also used classification models, and it showed that the Neural Network had the higher accuracy with about 55 percent. Another study used Support Vector Machine (SVM), Random Forests and Binomial GLM to forecast the Bitcoin price, and it came out with a high accuracy with more than 97 percent (Madan, Saluja, & Zhao, 2015). Although the accuracy is high, the results of this study might be overfit the data because the outcomes did not cross validated.

Deep learning tools seem to have effective results than machine learning methods (Pichl & Kaizoji, 2017). Long Short Term Memory (LSTM) network is a form of Recurrent Neural Network. LSTM can decide the data they want to remember and forget depends on the variables. In (Gers, Eck, & Schmidhuber, 2001), it is indicated that LSTM performed similarly as the RNN did in the mission. The only limitation in both algorithms is that they need amount of computation to run the models. Another study mentioned that LSTM and RNN can produce the best result for the prediction of price (McNally, Roche, & Caton, 2018).

The Bitcoin price seemed to have correlation with commodity market such as the exchange rate of Euro-Dollar and stock market index (such as Standard and Poor's 500). Although the alike pattern between Bitcoin price and S&P 500 just showed from 2014 to 2015, it still showed that they were correlated with each other at that time (Kinderis, Bezbradica, & Crane, 2018).

In this project, Nasdaq Composite Index and Dow Jones Industrial Index will be utilized as features for the Bitcoin price prediction. This paper compares three different prediction models, including Linear Regression, Random Forest Regression and LSTM, and put some features that have not been used to previous researches on the Bitcoin price forecast.

3. Research Purpose

How accurate the machine learning and deep learning models can achieve on the Bitcoin price prediction? The purpose of this project is to compare distinct algorithms to estimate the Bitcoin price and find out the error between their results and the actual price.

4. Methodology

4.1 Data Gathering

The data used in this project is a secondary data that is available on Myplace. The date of all variables started from February 18, 2013 to September 19, 2018. The total number of this data is 2000 with 15 features. Firstly, the daily price of Bitcoin is presented in the first column, which is the target variable in this project. Secondly, data related to Blockchain is collected, including Bitcoin network hash rate, the average Bitcoin Block size, difficulty of Bitcoin. Thirdly, regarding to the transaction of Bitcoin such as the number of Bitcoin transactions, the estimated transaction volume of Bitcoin and transaction fee of Bitcoin. Moreover, other cryptocurrencies are collected. Finally, gold and two important indices are put in to predict the Bitcoin price.

In Figure 4.1, it shown the history of the Bitcoin price from 2014 to 2018. It is indicated that the volatility of Bitcoin price is exactly high, from just over zero grew to almost near 20,000 in short period and then fell to about 7,000 USD dollars. From the original data, the Pearson correlation between each feature is presented in Figure 4.2. It is clearly to see that the price of Gold is less correlated to the Bitcoin price and all Bitcoin-related variables. In addition, the other cryptocurrencies (like Ethereum), the two indices, difficulty and estimated transaction volume have high correlation which are over 0.7 with the target variable. The correlation of the rest of the attributes like hash rate are between 0.5 to 0.6.

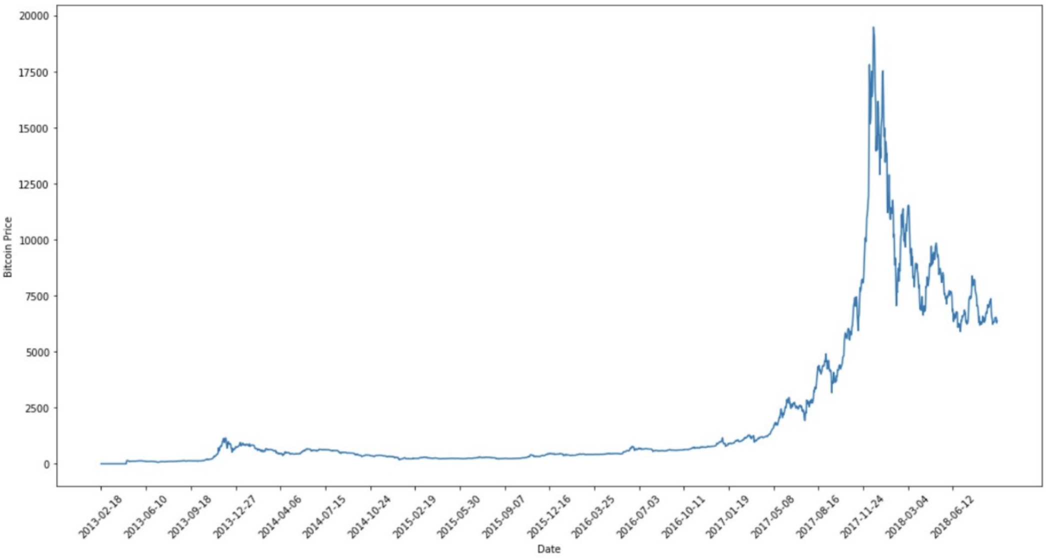


Figure 4.1 History of Bitcoin price

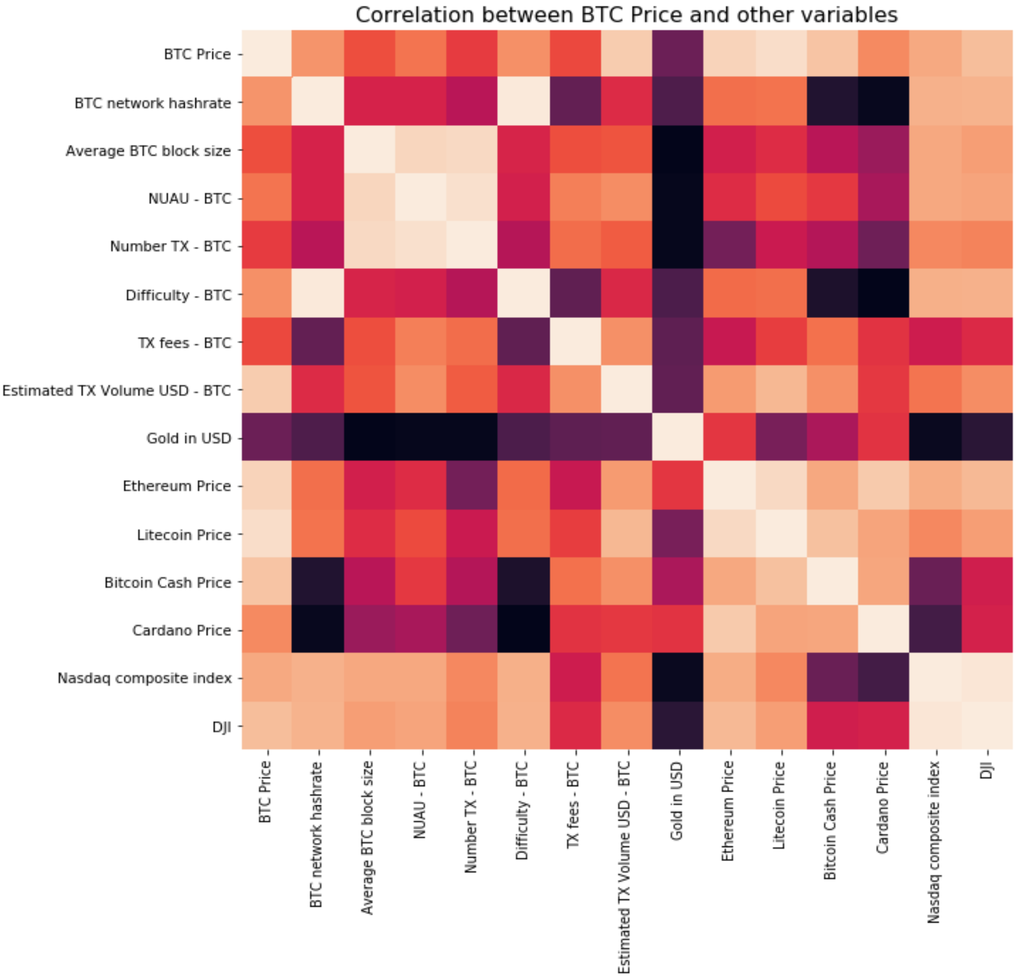
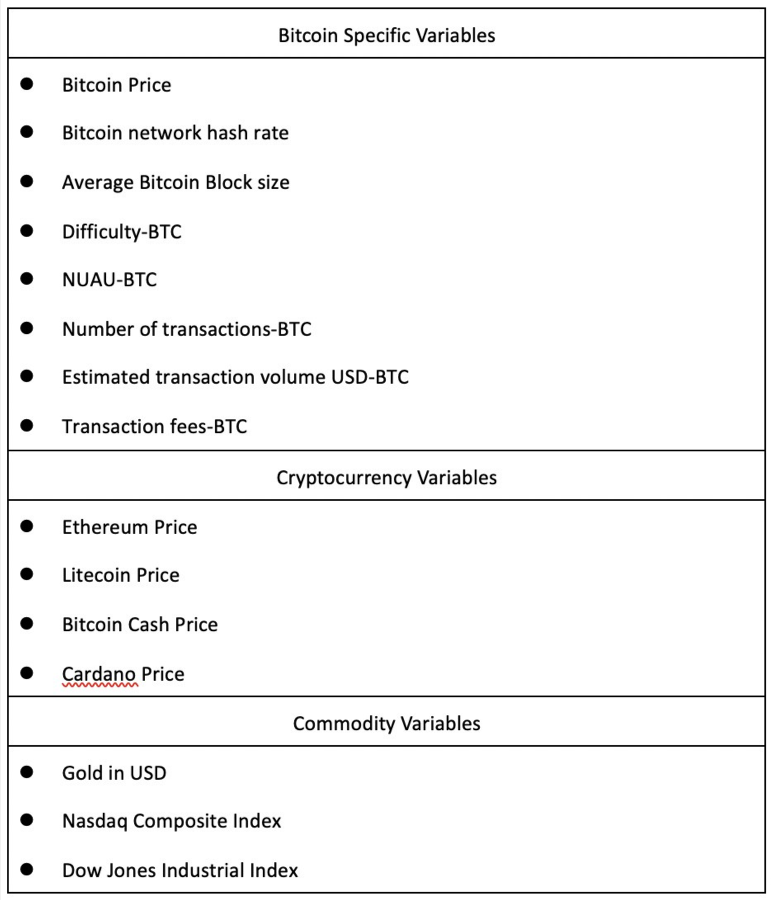


Figure 4.2 Pearson correlation (1.0 is the highest correlation)

4.2 Data Feature Extraction

The attributes used in this project can be classified into three types: a) Bitcoin specific variables, b) Cryptocurrency variables and c) Commodity variables. The aim of this paper is to use these features to predict the Bitcoin price. Figure 4.3shows the entire variables in different groups. The definitions of some attributes are shown in Figure 4.4(Blockchain.com, 2020a; Madan, Saluja, & Zhao, 2015). Almost all of these attributes have high correlation to the Bitcoin price. The reason to choose these three groups of attributes is to give a clear and general for the prediction of Bitcoin price. Not only includes the Bitcoin-related features, also contains the other cryptocurrencies price and the traditional financial products.



Bitcoin Specific Variables

- Bitcoin Price
- Bitcoin network hash rate
- Average Bitcoin Block size
- Difficulty-BTC
- NUAU-BTC
- Number of transactions-BTC
- Estimated transaction volume USD-BTC
- Transaction fees-BTC

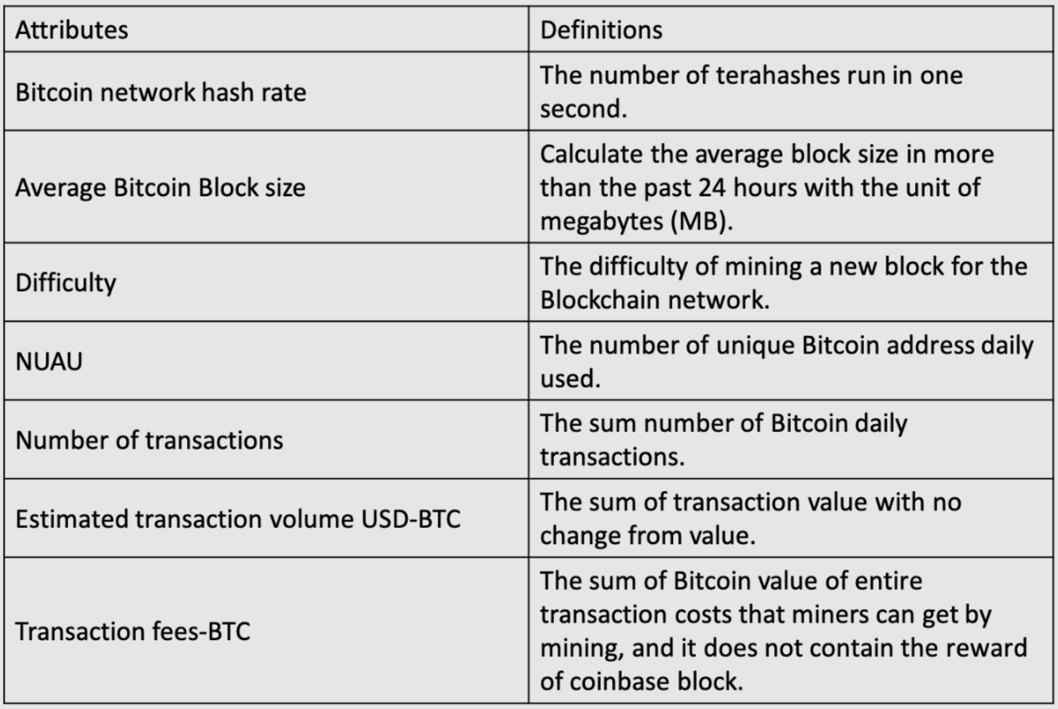
Cryptocurrency Variables

- Ethereum Price
- Litecoin Price
- Bitcoin Cash Price
- Cardano Price

Commodity Variables

- Gold in USD
- Nasdaq Composite Index
- Dow Jones Industrial Index

Figure 4.3 Categories of attributes



Attributes	Definitions
Bitcoin network hash rate	The number of terahashes run in one second.
Average Bitcoin Block size	Calculate the average block size in more than the past 24 hours with the unit of megabytes (MB).
Difficulty	The difficulty of mining a new block for the Blockchain network.
NUAU	The number of unique Bitcoin address daily used.
Number of transactions	The sum number of Bitcoin daily transactions.
Estimated transaction volume USD-BTC	The sum of transaction value with no change from value.
Transaction fees-BTC	The sum of Bitcoin value of entire transaction costs that miners can get by mining, and it does not contain the reward of coinbase block.

Figure 4.4 Definitions of attributes

4.3 Algorithms

4.3.1 Linear Regression

Linear Regression (LR) is a linear model that there is a linear relationship between various descriptive variables (x) and dependent variables (y). In more detail, y can be predicted by numbers of x and a constant also called intercept. In this project, it is a Multiple Linear Regression (MLR) because there are 12 input variables. In the package of Python, the scikit-learn library has a function called LinearRegression(). Reducing the Root Mean Squared Error (RMSE) between real values and forecasted results is the goal of linear regression.

The equation and parameters of linear regression is shown below (Géron, 2019):

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- $\hat{y}$ is the value predicted by the model.
- n means the total number of attributes.
- x_i is the value of each feature
- θ_j is the model parameter

The advantages for using Multiple Linear Regression are:

1. It is easy to understand and utilize.
2. The computation is fast.
3. It is widely used.

However, it only used in model that has linear relationship.

4.3.2 Random Forest Regression

Random Forest is an extension of Decision Trees and bagging method. In general, it is an ensemble algorithm that used to forecast the average prediction across the decision tree. There are two types of this algorithm, including classification and regression. In this paper, it will use Random Forest Regression to deal with the regression mission. Bagging is a method that every decision tree is fitted by a little distinct train data and performs slightly different with others. It can make the decision tree to be unsimilar and does not have high correlation and errors. Random Forest is quite different from bagging. It will choose a group of variables (x) at every split point in the decision trees. As it is an ensemble model, its predicted result is output from the average of multiple moderately good models (Görür, 2018). Random Forest can also find out which features are more important to the model. The formula used to check the result of Random Forest Regression is shown below:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

- N is the total sample of data
- f_i is the outcome of model
- y_i is the actual value for each data point

Mean Squared Error (MSE) can help to find out the better branch point from the model.

4.3.3 Long Short Term Model

As a special type of Recurrent Neural Network (RNN), it is used to deal with time series data and is able to learn long-term dependencies. For the Deep Learning Model, TensorFlow and Keras are used to build the backend as part of the neural network. In Python, Keras (Keras.io, 2017) is a deep learning Application Programming Interface (API) that builds on the machine learning tool called TensorFlow. Its advantages are that it can experiment fast results and it can deal with non-linear dataset. Layers and models are the main compositions of Keras. The model used in this project and also the simplest one is Sequential model. To stack more layers on the Sequential model, .add() function can help to make complicated construction. TensorFlow (TensorFlow.org, 2017) is tool that provides open source for building and training machine learning models like Neural Networks. LSTM is good at prediction by using several input variables than classic linear models.

One of the rules in LSTM model is that input attributes should be a three-dimensional array. There is a function called reshape() that can turn two-dimensional array to three-dimensional array. For example, for training data of Bitcoin Price in this project, the original shape of training input is: (904,1). After reshaping, the new shape is: (904,1,1). These three numbers mean samples, window size and features and are belong to the function input_shape.

In the layers of LSTM, there are lots of information carried by cells. Figure 4.5 shows the process of LSTM. The line from C_{t-1} to C_t is called the cell state and is the core part of LSTM because it holds both long and short-term memory. Gates in the LSTM can take out or add information to the cell state. There is a forget-gate that can decide when to remember and forget. There are three gates and other variables in LSTM and are shown as follow (Fan, Jiang, Xu, Zhu, Cheng, & Jiang, 2020; Olah, 2015):

1. $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
2. $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
3. $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
4. $\tilde{C} = \tanh\,(W_C \cdot [h_{t-1}, x_t] + b_C)$
5. $C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t$
6. $h_t = o_t \times \tanh\,(C_t)$

Where

- f_t is Forget gate, i_t is Input gate and o_t is Output gate.
- C_t is the cell state.
- h_{t-1} is the previous hidden gate and h_t is the new hidden state.
- x_t is the input variable.
- $\tilde{C}$ and o_t each is the vector composed of values between 0 and 1.

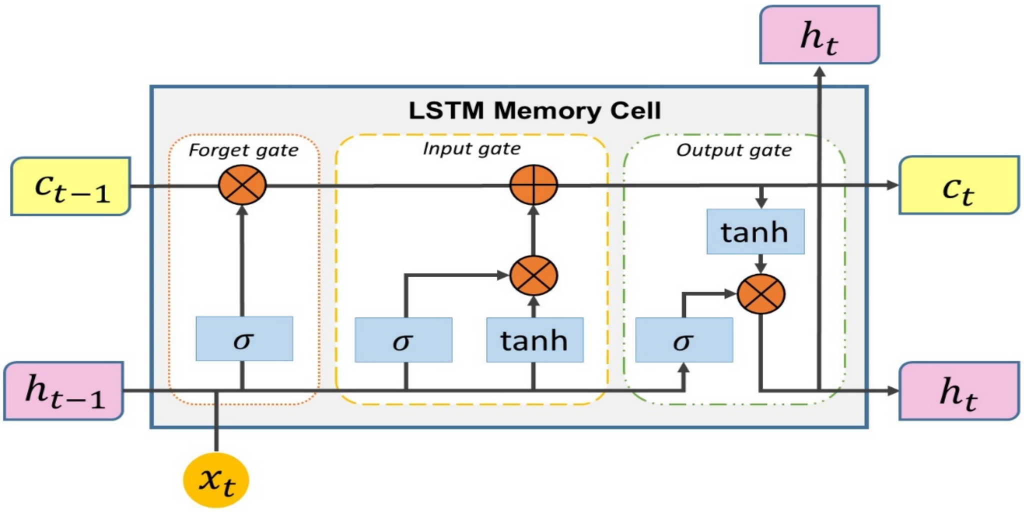


Figure 4.5 The concept of LSTM cell

The arguments of LSTM used in this project are shown below:

1. Units

The unit is the dimension of output state and hidden state. It is set to 12.

2. Activation function

There are three commonly used activation function, including sigmoid, tanh, and ReLU. Sigmoid controls the three gates function (forget, input and output) and produces values range from zero to one. However, it has vanishing gradient problem that needed to be solve. Tanh output values that are zero centered. ReLU is also commonly used. The default of activation function is tanh. Since the result of using tanh is the best, there is no need to change to default setting of this function.

3. Optimizer

There are multiple optimizers in LSTM algorithms, including Adam, Adagrad and RMSProp. In this project, Adam is selected due to its better performance than the others. All of these optimizers can be found in the Keras in Python.

4. Loss function

There are two ways to assess the performance of model. One is the RMSE which is the square root of MSE. Another is the MAE. RMSE is good at predicting data that are bell-shaped distribution, and MAE is better in handling dataset that has outliers. As a result, the MAE is chosen in LSTM to deal with Bitcoin price.

5. Dropout Rate

It is a regularization tool that can decrease overfitting and improve the performance of model. It is set to range from zero to one. It is set to 0.2 in this project due to better result.

6. Epochs

Epoch means the times that need to work through the whole dataset. After trying different numbers such as 100, 200 and 500, 500 epochs are selected to have a good performance.

7. Batch Size

Batch size means the size of data used in one iteration. This value is set to the number of train data divided by 5 (which is a guess number).

4.4 Software and Packages used

In this project, several libraries are used in the Python. Pandas (Pandas.org, 2020)is utilized for the missions related to dataset. Numpy (NumPy.org, 2020) is made use of changing DataFrame into array and also calculate the Mean Squared Error. Scikit-learn (Sklearn) (Scikit-learn.org, 2020) is used for the Min-Max Scaling, R Square Score and regression model such as Linear Regression. Matplotlib (Matplotlib.org, 2020) is utilized to visualize the data.

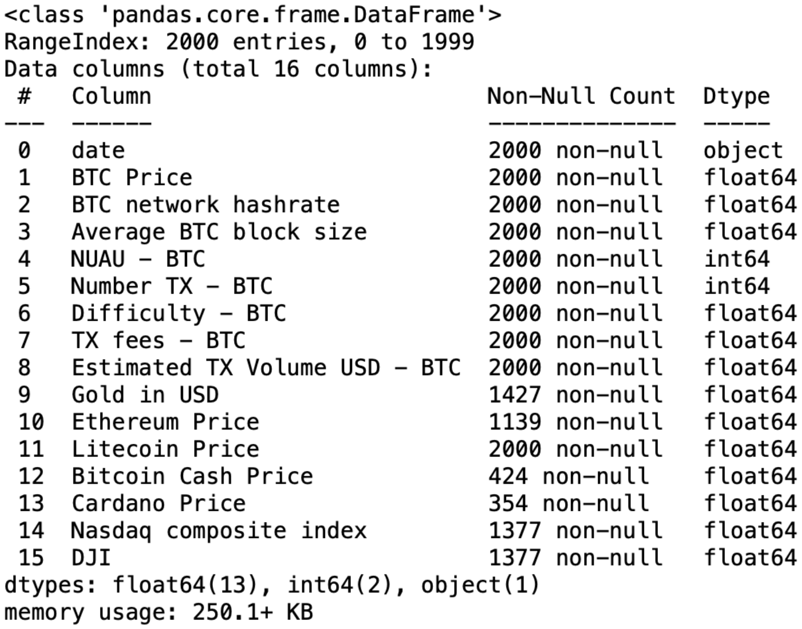
5. Analysis and Results

5.1 Exploratory Analysis

5.1.1 Descriptive Analysis

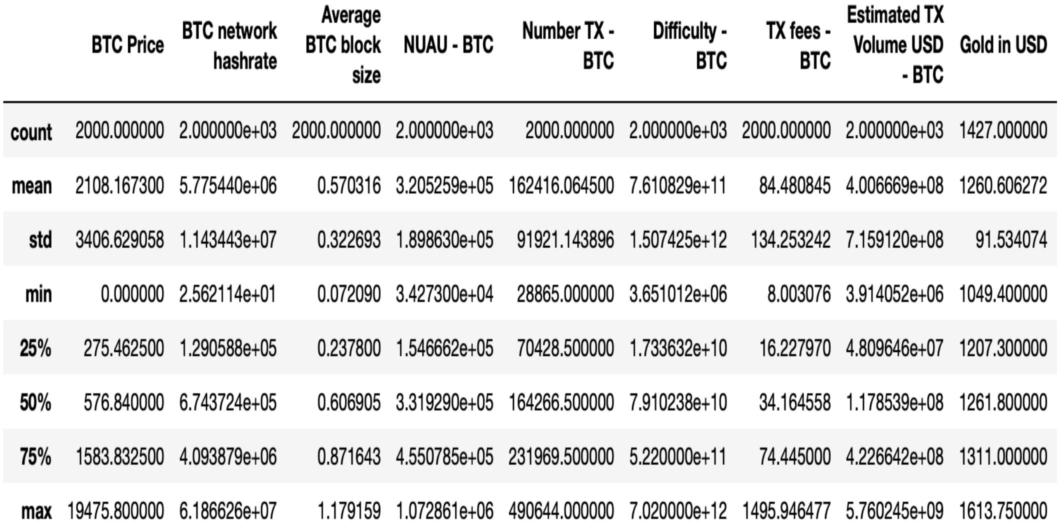
There are 16 attributes in this dataset. To overview the types of data, there is a function called info() in Python. This function can help to see if there is any string data or NaN values. In order to make plots for visualizing, it is necessary to convert string data into integer or float data. Figure 5.1 shown that all features are float or integer data expect date. The date column can be removed or set as index. There are seven variables that contain NaN values. These Nan values will be deal with in the data cleansing part. In Python, the describe() function can show the summary statistics for each attribute, including mean, minimum, maximum and so on (see Figure 5.2).

The next step is to see if there is any outlier in each column. By using boxplot() function, it is easy to find out the outliers. Many of the attributes in this dataset have numbers of outliers. However, this is a time series dataset. As the distribution of time series data usually is not Gaussian distribution, the tool of detecting outliers might not be useful for this type of data. As a result, although there are lots of outliers in this dataset, there is no need to change the values. Figure 5.3 showed the distribution of data.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 16 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   date                         2000 non-null   object 
 1   BTC Price                    2000 non-null   float64
 2   BTC network hashrate         2000 non-null   float64
 3   Average BTC block size       2000 non-null   float64
 4   NUAU - BTC                   2000 non-null   int64  
 5   Number TX - BTC              2000 non-null   int64  
 6   Difficulty - BTC             2000 non-null   float64
 7   TX fees - BTC                2000 non-null   float64
 8   Estimated TX Volume USD - BTC  2000 non-null   float64
 9   Gold in USD                  1427 non-null   float64
 10  Ethereum Price               1139 non-null   float64
 11  Litecoin Price               2000 non-null   float64
 12  Bitcoin Cash Price           424 non-null    float64
 13  Cardano Price                354 non-null    float64
 14  Nasdaq composite index       1377 non-null   float64
 15  DJI                          1377 non-null   float64
dtypes: float64(13), int64(2), object(1)
memory usage: 250.1+ KB
```

Figure 5.1 Description of dataset



	BTC Price	BTC network hashrate	Average BTC block size	NUAU - BTC	Number TX - BTC	Difficulty - BTC	TX fees - BTC	Estimated TX Volume USD - BTC	Gold in USD
count	2000.000000	2.000000e+03	2000.000000	2.000000e+03	2000.000000	2.000000e+03	2000.000000	2.000000e+03	1427.000000
mean	2108.167300	5.775440e+06	0.570316	3.205259e+05	162416.064500	7.610829e+11	84.480845	4.006669e+08	1260.606272
std	3406.629058	1.143443e+07	0.322693	1.898630e+05	91921.143896	1.507425e+12	134.253242	7.159120e+08	91.534074
min	0.000000	2.562114e+01	0.072090	3.427300e+04	28865.000000	3.651012e+06	8.003076	3.914052e+06	1049.400000
25%	275.462500	1.290588e+05	0.237800	1.546662e+05	70428.500000	1.733632e+10	16.227970	4.809646e+07	1207.300000
50%	576.840000	6.743724e+05	0.606905	3.319290e+05	164266.500000	7.910238e+10	34.164558	1.178539e+08	1261.800000
75%	1583.832500	4.093879e+06	0.871643	4.550785e+05	231969.500000	5.220000e+11	74.445000	4.226642e+08	1311.000000
max	19475.800000	6.186626e+07	1.179159	1.072861e+06	490644.000000	7.020000e+12	1495.946477	5.760245e+09	1613.750000

Figure 5.2 Statistic information of dataset

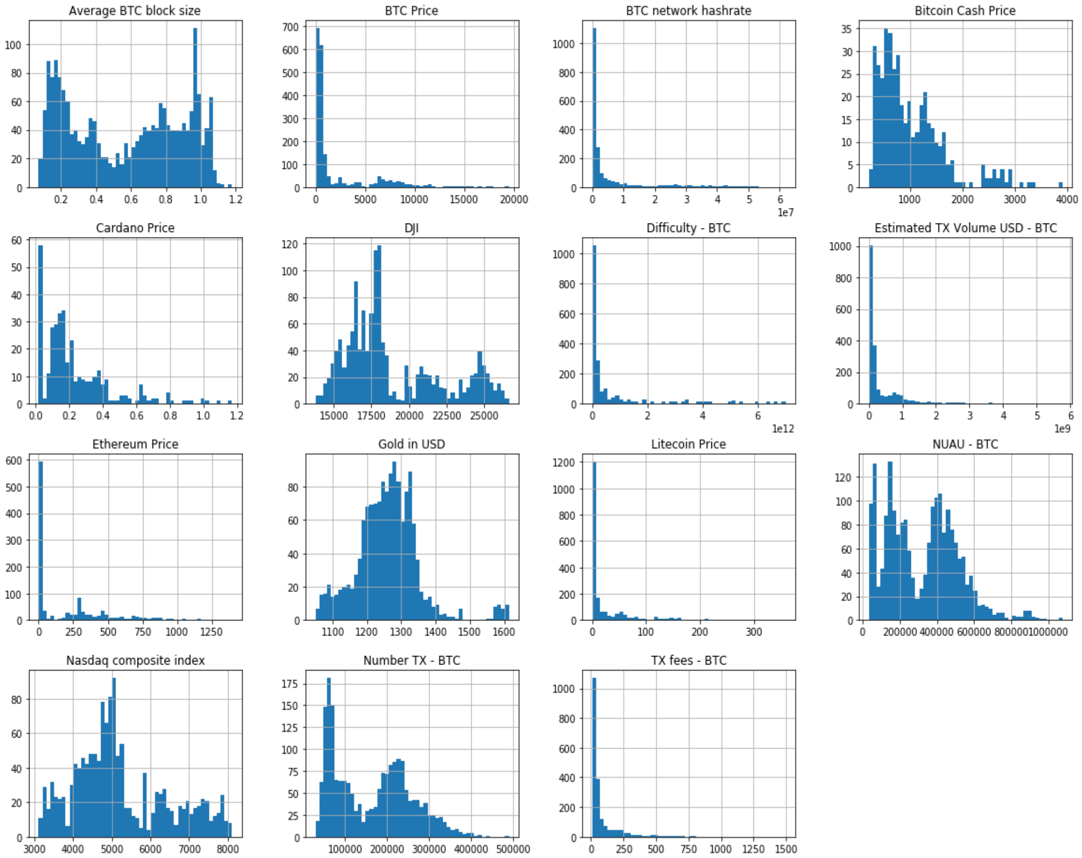
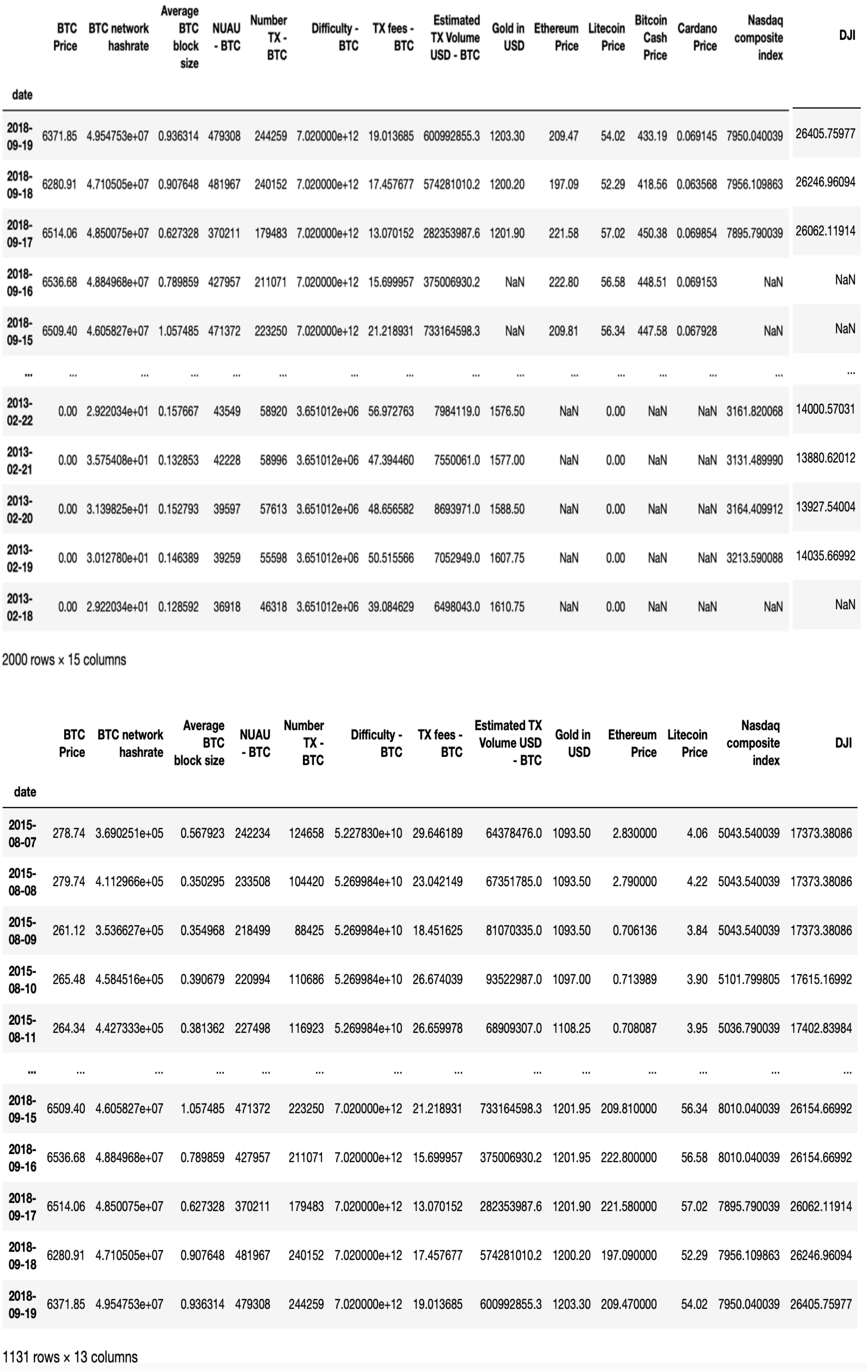


Figure 5.3 Distribution of dataset

5.1.2 Data Cleansing

To make accurate result through analysis process, it is necessary to detect and correct the wrong data in the given dataset. In the original dataset, there are some zero value in the Bitcoin price variable. To make the prediction more accurate, the Bitcoin price that are zero are deleted by using the Python function. The Ethereum price and Litecoin price also have the same problem, so those equal to zero are not used in this project. Bitcoin Cash price and Cardano price have lots of NaN data because they were launched lately than other cryptocurrencies. It is not reasonable to fill these NaN value with any value because they did not exist in the previous years. Bitcoin Cash price has 1576 NaN values and Cardano price has 1646 NaN values. If these NaN values are deleted, the total number of rows will be 354 which is not enough to make a good prediction. As a result, these two attributes are deleted from the datasets.

The gold price, Nasdaq Composite Index and Dow Jones Industrial Index also have NaN values which occurred on the weekend. As the financial market did not open on weekend, there are two ways to deal with these NaN values. Firstly, remove all NaN values that happened on weekend because it would not have much influence. Secondly, these NaN values can be filled by the mean of the variable, the backward or the forward values. In this project, the NaN values for these three features are filled in with forward values. For example, if there is a NaN value on 1 January 2015, it will be replaced with the value of the date of 31 December 2014. These processes are done by using Python. After these steps, there are now 1131 rows and 13 columns in this dataset (see Figure 5.4).



date	BTC Price	BTC network hashrate	Average BTC block size	NUAU - BTC	Number TX - BTC	Difficulty - BTC	TX fees - BTC	Estimated TX Volume USD - BTC	Gold in USD	Ethereum Price	Litecoin Price	Bitcoin Cash Price	Cardano Price	Nasdaq composite index	DJI
2018-09-19	6371.85	4.954753e+07	0.936314	479308	244259	7.020000e+12	19.013685	600992855.3	1203.30	209.47	54.02	433.19	0.069145	7950.040039	26405.75977
2018-09-18	6280.91	4.710505e+07	0.907648	481967	240152	7.020000e+12	17.457677	574281010.2	1200.20	197.09	52.29	418.56	0.063568	7956.109863	26246.96094
2018-09-17	6514.06	4.850075e+07	0.627328	370211	179483	7.020000e+12	13.070152	282353987.6	1201.90	221.58	57.02	450.38	0.069854	7895.790039	26062.11914
2018-09-16	6536.68	4.884968e+07	0.789859	427957	211071	7.020000e+12	15.699957	375006930.2	NaN	222.80	56.58	448.51	0.069153	NaN	NaN
2018-09-15	6509.40	4.605827e+07	1.057485	471372	223250	7.020000e+12	21.218931	733164598.3	NaN	209.81	56.34	447.58	0.067928	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2013-02-22	0.00	2.922034e+01	0.157667	43549	58920	3.651012e+06	56.972763	7984119.0	1576.50	NaN	0.00	NaN	NaN	3161.820068	14000.57031
2013-02-21	0.00	3.575408e+01	0.132853	42228	58996	3.651012e+06	47.394460	7550061.0	1577.00	NaN	0.00	NaN	NaN	3131.489990	13880.62012
2013-02-20	0.00	3.139825e+01	0.152793	39597	57613	3.651012e+06	48.656582	8693971.0	1588.50	NaN	0.00	NaN	NaN	3164.409912	13927.54004
2013-02-19	0.00	3.012780e+01	0.146389	39259	55598	3.651012e+06	50.515566	7052949.0	1607.75	NaN	0.00	NaN	NaN	3213.590088	14035.66992
2013-02-18	0.00	2.922034e+01	0.128592	36918	46318	3.651012e+06	39.084629	6498043.0	1610.75	NaN	0.00	NaN	NaN	NaN	NaN

2000 rows × 15 columns

date	BTC Price	BTC network hashrate	Average BTC block size	NUAU - BTC	Number TX - BTC	Difficulty - BTC	TX fees - BTC	Estimated TX Volume USD - BTC	Gold in USD	Ethereum Price	Litecoin Price	Nasdaq composite index	DJI
2015-08-07	278.74	3.690251e+05	0.567923	242234	124658	5.227830e+10	29.646189	64378476.0	1093.50	2.830000	4.06	5043.540039	17373.38086
2015-08-08	279.74	4.112966e+05	0.350295	233508	104420	5.269984e+10	23.042149	67351785.0	1093.50	2.790000	4.22	5043.540039	17373.38086
2015-08-09	261.12	3.536627e+05	0.354968	218499	88425	5.269984e+10	18.451625	81070335.0	1093.50	0.706136	3.84	5043.540039	17373.38086
2015-08-10	265.48	4.584516e+05	0.390679	220994	110686	5.269984e+10	26.674039	93522987.0	1097.00	0.713989	3.90	5101.799805	17615.16992
2015-08-11	264.34	4.427333e+05	0.381362	227498	116923	5.269984e+10	26.659978	68909307.0	1108.25	0.708087	3.95	5036.790039	17402.83984
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2018-09-15	6509.40	4.605827e+07	1.057485	471372	223250	7.020000e+12	21.218931	733164598.3	1201.95	209.810000	56.34	8010.040039	26154.66992
2018-09-16	6536.68	4.884968e+07	0.789859	427957	211071	7.020000e+12	15.699957	375006930.2	1201.95	222.800000	56.58	8010.040039	26154.66992
2018-09-17	6514.06	4.850075e+07	0.627328	370211	179483	7.020000e+12	13.070152	282353987.6	1201.90	221.580000	57.02	7895.790039	26062.11914
2018-09-18	6280.91	4.710505e+07	0.907648	481967	240152	7.020000e+12	17.457677	574281010.2	1200.20	197.090000	52.29	7956.109863	26246.96094
2018-09-19	6371.85	4.954753e+07	0.936314	479308	244259	7.020000e+12	19.013685	600992855.3	1203.30	209.470000	54.02	7950.040039	26405.75977

1131 rows × 13 columns

Figure 5.4 Overview of dataset (Before & After)

5.1.3 Split Data

After data cleansing, the history of Bitcoin price is shown in Figure 5.5. The dataset is split into train and test data. The train data is made up of 80 percent of the whole dataset, and test data takes up to 20 percent. As the dataset contains time series data, the values are not randomized. The train and test data must split by the date. The train data starts from 7 August 2015 to 4 February 2018, and the test data starts from 5 February 2018 to 19 September 2018.

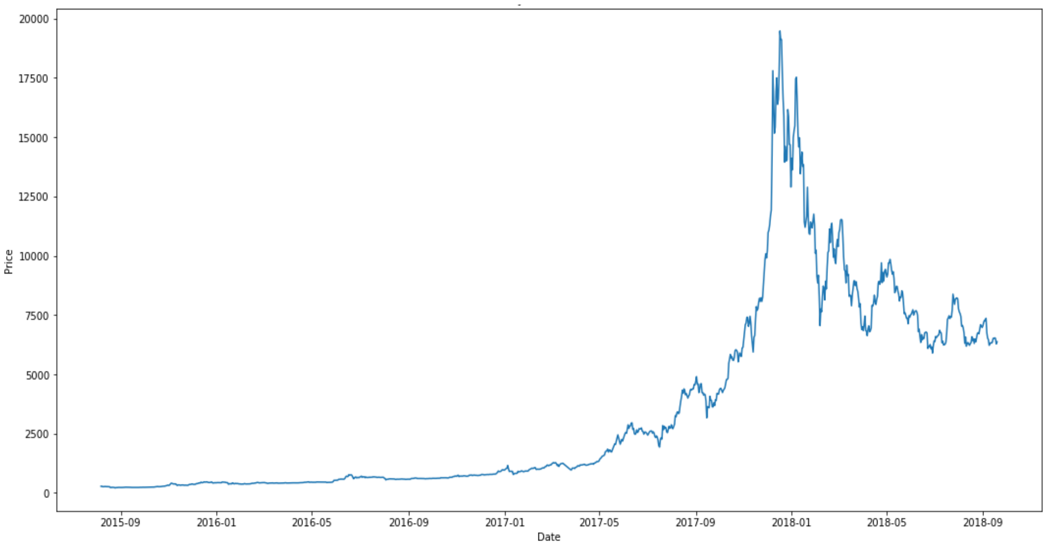


Figure 5.5 History of Bitcoin price (After data cleansing)

5.1.4 Data Scaling

Before putting data into the training model, it is important to check if the data need to be scaled. In general, the performance of machine learning model will not be good if the input data have different scales. Note that scaling the target values is generally not required. There are two features for the data that can make model learning easier (Géron, 2019). Firstly, the entire values in the dataset should be between 0 and 1. Secondly, every attribute should be homogeneous. As the principle of neural network mentioned that scaling is necessary for target feature (y), there are some common tools to transform the data.

1. Min-Max Scaling (also called normalization): the original values will be transformed, and they will be in the range between 0 and 1. The algorithm of this method is shown:

$$x' = \frac{x - \min\,(X)}{\max(X) - \min\,(X)}$$

2. Standardization (Z-Score): it does not shifted values into a particular range, which is unsimilar with the Min-Max Scaling. The algorithm of this method is shown:

$$z = \frac{x_i - \mu}{\sigma}$$

In this project, it will talk about the neural network. As neural network usually use data that are between 0 and 1, the standardization method will not be useful. As a result, Min-Max Scaling is applied to transform the data for this research. In Python, Scikit-Learn package provides a function called MinMaxScaler to transform the data.

5.2 Inferential Analysis

In this section, it will discuss three different outcomes for machine learning and deep learning model. Machine learning can help minimize the error of prediction model and make the results more accurate. Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are commonly used to assess the performance of each algorithm. RMSE is the square root of MSE. MAE reflects the real situation of the difference between actual values and forecastable values and ignore the direction. RMSE is the method that measures the difference between actual values and predicted values.

The equation of MAE, MSE and RMSE are shown below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2$$

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

The lower values of them, the better results they are. The distinction between RMSE and MAE is the way they treat outliers. As the value output by RMSE is squared before averaged, large errors will be weighted highly by RMSE. As a result, RMSE is relatively valuable when measuring errors with normal distribution (Chai & Draxler, 2014). As mentioned before, the dataset contains lots of outliers. It might be better to use MAE to evaluate the outcome of each model.

5.2.1 Linear Regression

The reason to choose Linear Regression is that it is easy to apply and has fast computation. Figure 5.6 shows the predicted values of Bitcoin price using Linear Regression compared to the actual price. It is indicated that from February 2018 to June 2018, the prediction is quite fitted to the true value. However, in the latter period, the difference between real and forecastable price becomes significant. The RMSE and MAE of Linear Regression is 913.46 and 27.44 respectively.

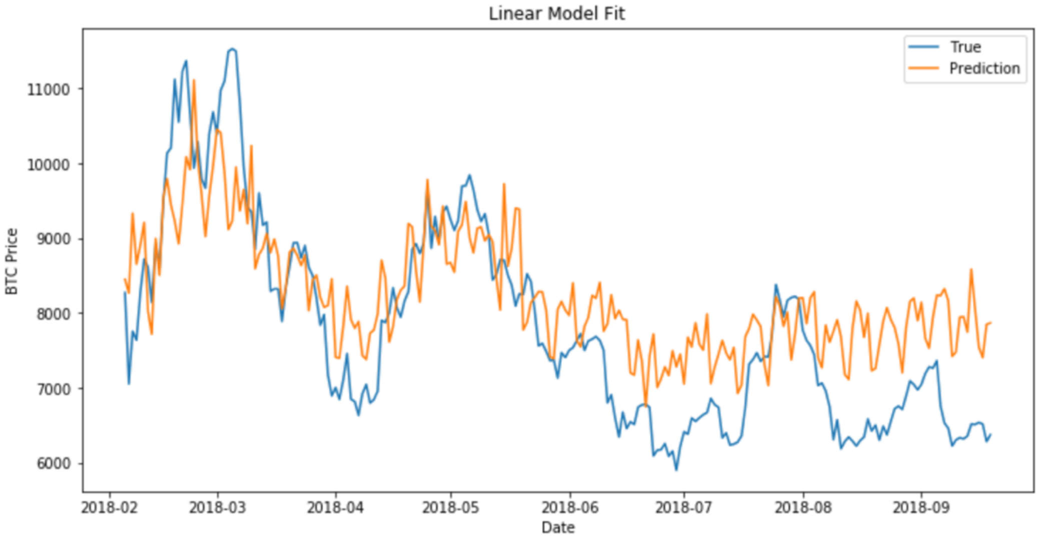


Figure 5.6 Linear Regression Model Fit

5.2.2 Random Forest Regression

Random Forest Regression can average the output from each node. Figure 5.7 presents the trend of true and predicted Bitcoin price. It seems that prediction is far from the actual values. The RMSE and MAE for Random Forest Regression is 2208.2 and 44.59 individually. Figure 5.8 shows the percentage of important attributes in this dataset. There is no significant feature that has strong impact on the performance of model. Only the importance of Nasdaq composite index is more than 0.2.

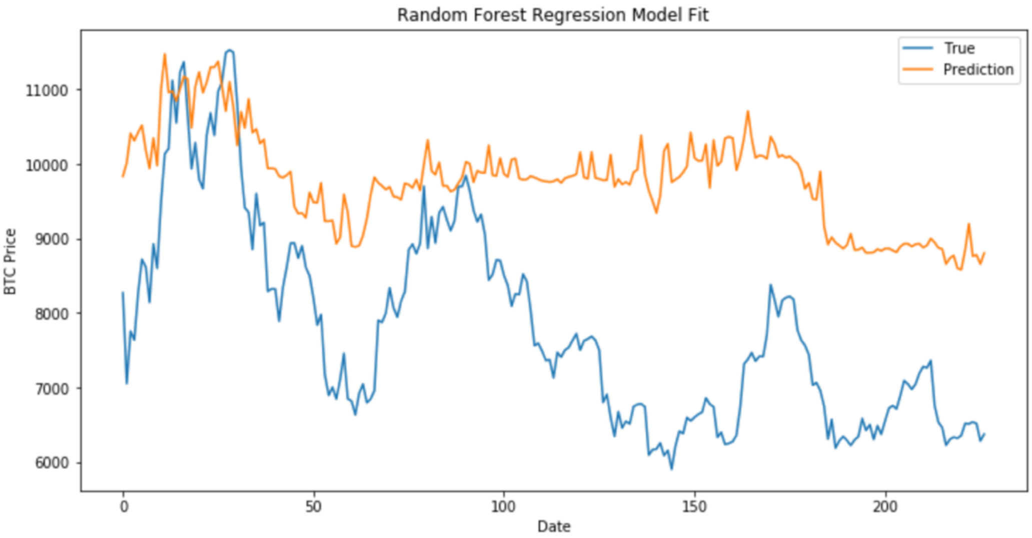
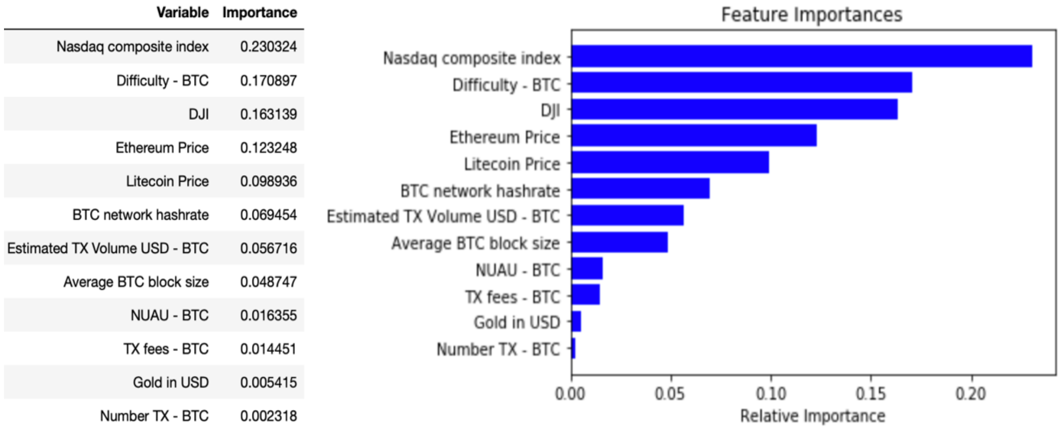


Figure 5.7 Random Forest Regression Model Fit



Variable	Importance
Nasdaq composite index	0.230324
Difficulty - BTC	0.170897
DJI	0.163139
Ethereum Price	0.123248
Litecoin Price	0.098936
BTC network hashrate	0.069454
Estimated TX Volume USD - BTC	0.056716
Average BTC block size	0.048747
NUAU - BTC	0.016355
TX fees - BTC	0.014451
Gold in USD	0.005415
Number TX - BTC	0.002318

Figure 5.8 Importance of attributes

5.2.3 Long Short Term Model

In this project, the loss function used in LSTM is MSE because MSE is better to deal with outliers. Figure 5.9 shows the visualization of loss in different scale of y-axis. The upper plot is linear scale and the lower is logarithmic scale. It is shown that the trend of loss is downward as the epochs increase. Figure 5.10 presents the LSTM model fit. From the plot, predicted values are quite fit with the actual prices. Although some of the estimated values are not close to the real prices, the trend is quite similar to the true data. The RMSE and MAE for LSTM is 593.86 and 21.95 separately.

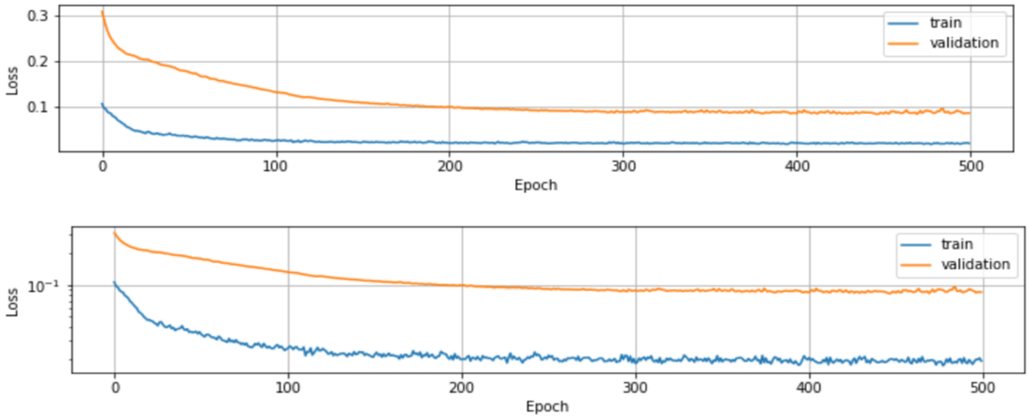


Figure 5.9 Loss of LSTM

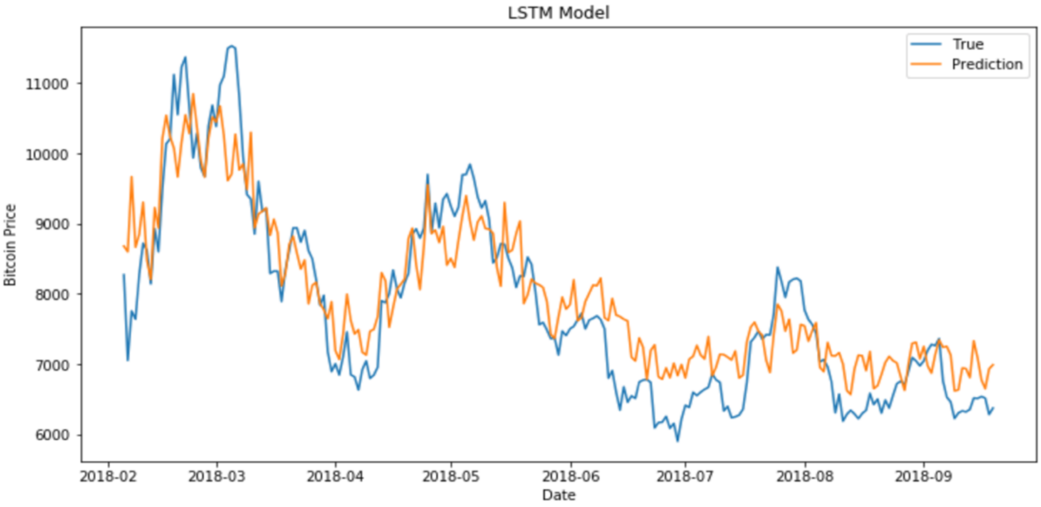


Figure 5.10 LSTM Model Fit

6. Findings

Table 6.1 summarizes the results of each model for the Bitcoin price forecast. The model performance is better when RMSE and MAE is close to zero, which indicates the lower the better. There are several perspectives for the results. Unsurprisingly, LSTM performs better than Linear Regression and Random Forest Regression model. Both of the RMSE and MAE of LSTM are smaller than the other models. Since LSTM is good at handling long-term dependencies, it is expected to have well performance on estimating Bitcoin price for a specific period. For machine learning models, the outcome of Linear Regression is better than Random Forest Regression. Unlike LSTM, both of them do not required much computations and GPU to complete training model. The RMSE and MAE of Random Forest Regression are both relatively high compared to those of the LR and LSTM.

Although the volatility of Bitcoin is pretty high, machine learning models still can approximately forecast its price. The accuracy of results can still be optimized to get the best results. From the outcomes, these attributes seem to have contributions to the Bitcoin price prediction. Machine learning algorithms can be a good tool to forecast cryptocurrency price, there is still some improvements need to make.

Table 6.1 Comparison of RMSE and MAE

Models	RMSE	MAE
Linear Regression	913.46	27.44
Random Forest Regression	2208.2	44.59
LSTM	593.86	21.95

7. Conclusion

Number of researches on predicting Bitcoin price have been written so far. Many of them used machine learning and deep learning to make prediction. However, most of these papers are depended on limited attributes and without multiple categories. For example, many of them used only Bitcoin related variables or other cryptocurrencies as features. In this study, three types of attributes are utilized, including Bitcoin specific variables, cryptocurrency variables and commodity variables. Lots of machine learning or deep learning algorithms have been applied to forecast the Bitcoin price. In this research, three regression models are applied, Linear Regression, Random Forest Regression and Long Short Term Model. Although number of studies have already discussed these methods on Bitcoin price prediction, different input variables will output distinct results. The measurements of evaluating models in this study are RMSE and MAE. These tools are widely used for comparing difference between real and estimated values. The outcomes show that the LSTM produces the best result than the other models. Since the dataset is a time series data, LSTM will be suitable for doing prediction.

However, there are some limitations in this study. First of all, the samples of data are not large enough to make accurate prediction. Since the historical data of cryptocurrencies is limited, the estimation might be biased. Secondly, the hyperparameter tuning is not done in this paper. Some of the arguments might not be the perfect one to produce the accurate results. For example, Grid Search can be used to optimize the hyperparameters (Brownlee, 2016). Thirdly, as two cryptocurrencies (Bitcoin Cash Price and Cardano) have little non-NaN values, they are removed in the process of data cleansing. In the future, if there is enough data for these two cryptocurrencies, they can be added into the input features to estimate the Bitcoin price. In addition, other features such as internet search result can be added into the dataset and make comparison between different categories attributes. This can find out which type of variables might have more impact on cryptocurrency price.

In conclusion, using RMSE and MAE to assess the prediction of these models, LSTM produces the best result. Nevertheless, outcome still can be optimized by multiple methods, including adding more samples to the dataset and hyperparameters tuning.

Bibliography

Bitcoin.org. (2020). *Bitcoin Exchanges*. https://bitcoin.org/en/exchanges#new-zealand

Blockchain.com. (2020a). *Blockchain Charts*. https://www.blockchain.com/charts

Blockchain.com. (2020b). *Market Price*. https://www.blockchain.com/charts/market-price

Bouoiyour, J., & Selmi, R. (2015). What does Bitcoin look like? *Annals of Economics and Finance*, *16*(2), pp.449–492.

Brownlee, J. (2016). *How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras*.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), pp.1247–1250.

Cheah, E. T., & Fry, J. (2015). Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin. *Economics Letters*, *130*, pp.32–36.

Ciaian, P., Rajcaniova, M., & Kancs, D. (2016). The economics of BitCoin price formation. *Applied Economics*, *48*(19), pp.1799–1815.

David Yermack. (2013). Is Bitcoin A Real Currency? An Economic Appraisal. In *Working Paper 19747*.

Dutta, A., Kumar, S., & Basu, M. (2019). *A Gated Recurrent Unit Approach to Bitcoin Price Prediction*. *13*(2), pp.23.

Dyhrberg, A. H. (2016). Hedging capabilities of bitcoin. Is it the virtual gold? *Finance Research Letters*, *16*, pp.139–144.

Elbahrawy, A., Alessandretti, L., Kandler, A., Pastor-Satorras, R., & Baronchelli, A. (2017). Evolutionary dynamics of the cryptocurrency market. *Royal Society Open Science*, *4*(11).

Fan, H., Jiang, M., Xu, L., Zhu, H., Cheng, J., & Jiang, J. (2020). Comparison of long short term memory networks and the hydrological model in runoff simulation. *Water (Switzerland)*, *12*(1), pp.1–15.

Gandal, N., & Halaburda, H. (2016). Can we predict the winner in a market with network effects? Competition in cryptocurrency market. *Games*, *7*(3), pp.1–21.

Géron, A. (2019). Hands on Machine Learning with Scikit Learn Keras and TensorFlow 2nd Edition-2019. In *Journal of Chemical Information and Modeling* (Vol. 53, Issue 9).

Gers, F. A., Eck, D., & Schmidhuber, J. (2001). *Applying LSTM to Time Series Predictable through Time-Window Approaches BT - Artificial Neural Networks — ICANN 2001* (G. Dorffner, H. Bischof, & K. Hornik (eds.); pp. 669–676). Springer Berlin Heidelberg.

Görür, Y. (2018). Bitcoin Price Detection With Pyspark Using Random Forest. In *Journal of Chemical Information and Modeling*.

Greaves, A., & Au, B. (2015). *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*.

Hayes, A. S. (2017). Cryptocurrency value formation: An empirical study leading to a cost of production model for valuing bitcoin. *Telematics and Informatics*, *34*(7), pp.1308–1321.

Hileman, G., & Rauchs, M. (2017). 2017 Global Cryptocurrency Benchmarking Study. *SSRN Electronic Journal*, *44*(0).

Keras.io. (2017). *Keras documentation*. https://keras.io/about/

Kinderis, M., Bezbradica, M., & Crane, M. (2018). *Bitcoin Currency Fluctuation*. *Complexis*, pp.31–41.

Kristoufek, L. (2013). BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era. *Scientific Reports*, *3*, pp.1–7.

Kristoufek, L. (2015). What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis. *PLoS ONE*, *10*(4), pp.1–15.

Lo, S., & Wang, J. C. (2014). Bitcoin as Money? *Federal Reserve Bank of Boston*, *14*(4), pp.1–28.

Madan, I., Saluja, S., & Zhao, A. (2015). *Automated Bitcoin Trading via Machine Learning Algorithms*. *20*, pp.1–5.

Matplotlib.org. (2020). *Matplotlib: Visualization with Python*. https://matplotlib.org

McNally, S., Roche, J., & Caton, S. (2018). Predicting the Price of Bitcoin Using

Machine Learning. *Proceedings - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018*, pp.339–343.

Muzammal, M., Qu, Q., & Nasrulin, B. (2019). Renovating blockchain with distributed databases: An open source system. *Future Generation Computer Systems*, *90*, pp.105–117.

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. pp.1–9.

NumPy.org. (2020). *NumPy*. https://numpy.org

Olah, C. (2015). *Understanding LSTM Networks*. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Pandas.org. (2020). *Pandas*. https://pandas.pydata.org

Pichl, L., & Kaizoji, T. (2017). Volatility Analysis of Bitcoin Price Time Series. *Quantitative Finance and Economics*, *1*(4), pp.474–485.

Polasik, M., Piotrowska, A. I., Wisniewski, T. P., Kotkowski, R., & Lightfoot, G. (2015). Price fluctuations and the use of bitcoin: An empirical inquiry. *International Journal of Electronic Commerce*, *20*(1), pp.9–49.

Ron, D., & Shamir, A. (2013). Quantitative analysis of the full Bitcoin transaction graph. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7859 LNCS*, pp.6–24.

Scikit-learn.org. (2020). *Scikit-learn*. https://scikit-learn.org/stable/

SIAMI-NAMIN, S., & SIAMI-NAMIN, A. (2018). *Forecasting Economic and Financial Time Series: ARIMA VS. LSTM*. pp.1–19.

TensorFlow.org. (2017). *TensorFlow*. https://www.tensorflow.org

Woo, D. (2013). *Bitcoin: a first assessment*. *December*, pp.1–14.

Yhlas Sovbetov. (2018). Factors Influencing Cryptocurrency Prices: Evidence from Bitcoin, Ethereum, Dash, Litcoin, and Monero. *Journal of Economics and Financial Analysis*, *2*(2), pp.1–27.

Zargham, M., Zhang, Z., & Preciado, V. (2018). *A State-Space Modeling Framework for Engineering Blockchain-Enabled Economic Systems*.