UNIVERSITI TEKNOLOGI MARA

FORECASTING ANALYSIS OF ETHEREUM USING MACHINE LEARNING: LINEAR REGRESSION AND ARTIFICIAL NEURAL NETWORK

SYED ANAND NAJMI BIN SAYED ABU BASHAR
MUHAMMAD NAZMAN BIN MOHD NASIR
MUHAMMAD FIKRI BIN ABDULLAH
IZWANA KAMILIA BINTI AHMAD NAJMI KAMIL

FACULTY OF COMPUTER AND

MATHEMATICAL SCIENCES

AUGUST 2021

UNIVERSITI TEKNOLOGI MARA

FORECASTING ANALYSIS OF ETHEREUM USING MACHINE LEARNING: LINEAR REGRESSION AND ARTIFICIAL NEURAL NETWORK

SYED ANAND NAJMI BIN SAYED ABU BASHAR 2018439602
MUHAMMAD NAZMAN BIN MOHD NASIR 2018249406
MUHAMMAD FIKRI BIN ABDULLAH 2018201142
IZWANA KAMILIA BINTI AHMAD NAJMI KAMIL 2018439572

Thesis submitted in fulfilment
of the requirements for the degree of
Bachelor of Science (Hons.) Actuarial Science

Faculty of Computer and Mathematical Sciences

AUGUST 2021

Approved by

…………………………………
Supervisor
Mr. Zahrul Azmir Kamarul Adzhar, A.S.A
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA, Shah Alam, Malaysia

Approved by

…………………………………

Co-Supervisor

Mrs. Suraya Fadilah Ramli
Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA, Shah Alam, Malaysia

CANDIDATE'S DECLARATION

I hereby declare that this project was carried out in accordance with regulations of Universiti Teknologi MARA. It is original and is the result of our work, unless otherwise indicated or acknowledged as referenced work. This topic has not been submitted to any other academic institution or non-academic institution for any other degree or qualification.

We acknowledge that we have been supplied with the Academic Rules and Regulations for undergraduate Universiti Teknologi MARA (UiTM), regulating the conduct of my study and research.

Name of Candidate	:1. Syed Anand Najmi bin Sayed Abu Bashar	(2018439602)
	2. Muhammad Nazman Bin Mohd Nasir	(2018249406)
	3. Muhammad Fikri Bin Abdullah	(2018201142)
	4. Izwana Kamilia Binti Ahmad Najmi Kamil	(2018439572)
Programme	: Bachelor of Science (Hons.) Actuarial Science	
Faculty	: Faculty of Computer and Mathematical Sciences	
Project Title	: Forecasting Analysis of Ethereum using Machine Learning: Linear Regression and Artificial Neural Network.	

Anand

Signature of Student : …………… …………… …………… ……………

Date : August 2021

ABSTRACT

Cryptocurrency is one of the most advanced types of funds and it is popular among investors as the cryptocurrencies are not bound by a central bank or government interference. In this study, we seek the effectiveness of using machine learning to forecast Ethereum as well as to study the effects of the volatility of other cryptocurrencies. Using Linear Regression model and Artificial Neural Network model, the Ethereum Close price can be forecasted either in short-term or long-term. To study the volatility of the cryptocurrencies, GARCH (1,1) is used. The data consist of the daily data of Ethereum and Bitcoin which consist of around 1500 observations. The result of this study shows that the significant variables to forecast Ethereum are Open price, High price, and Low price. This study also proves that Linear Regression model and Artificial Neural Network model can be used to forecast Ethereum. However, the Artificial Neural Network can only forecast Ethereum in the short-term and failed in the long-term while Linear Regression model performed greatly in the short-term and in the long-term. Linear Regression model has an overall better performance than Artificial Neural Network model in forecasting Ethereum in both the short-term and the long-term forecasting. This study proves that the volatility of Ethereum is more persistent in the long run as compared to Bitcoin meanwhile in the short run, Ethereum's volatility is less persistent as compared to Bitcoin. Although machine learning succeeds in forecasting Ethereum, it does not have the ability to predict spikes and dips of cryptocurrencies. Thus, to gain a more accurate result, a longer period of study and the addition of social economy are advised to be included.

ACKNOWLEDGEMENT

First and foremost, we wish to thank God for giving us the opportunity, the strength and the spirit needed to finish our Final Year Project (FYP). Without His blessings, this project may not be finished.

We would like to express our deepest gratitude to our supervisors, Mr. Zahrul Azmir Kamarul Adzhar and Mrs. Suraya Fadilah Ramli for their patience, invaluable guidance, advice and the critiques on this project. Without their guidance, the project will not be as accomplished as it can be.

We would also like to dedicate this appreciation to our group members who have sacrificed their blood, sweat and tears in completing this project. Without them, the project will never be finished in time.

Finally, we would like to express our gratitude to our dearest family and friends for their never-ending encouragement and support during our study. Thank you so much for your undying support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations

CA	Contracts Act
GARCH	Generalized Auto Regressive Conditional Heteroskedasticity
LR	Linear Regression
ANN	Artificial Neural Network
NN	Neural Network
MLR	Multiple Linear Regression
RBF	Radial Basis Function
ETH	Ethereum
LTC	Litecoin
BTC	Bitcoin
GBM	Gradient Boosting Machine
GDBT	Gradient Boosting Decision Tree
LSTM	Long-Short-Term Memory
MLP	Multi-layer Perceptron
USD	United States Dollar
UiTM	Universiti Teknologi MARA
VBA	Visual Basic for Applications
ACF	Autocorrelation Factor
NNAR	Autoregressive Neural Network

CHAPTER ONE
INTRODUCTION

1.1 Background of Research

Cryptocurrencies is a virtual currency which acts as a digital asset that can be traded in the form of cryptography (Nakamoto,2008 cited in Ngunyi et al., 2019). Design as a digital asset assists the popularity of cryptocurrency among investors as the cryptocurrency is not tied by the central banking system and government interference (Fantazzini et al., 2016 cited in Naimy et al., 2021). Furthermore, decentralized characteristics of cryptocurrency is an appeal for those who prefer anonymous transactions (Fantazzini et al., 2016 cited in Naimy et al., 2021). However, being digital assets makes the cryptocurrency more susceptible to the cyberattack (Bouoiyour et al., 2015 cited in Kyariazis et al., 2019). Even with those risks, the popularity of cryptocurrency did not decline at all. In fact, increased people jumped on the boats to invest into the cryptocurrency and made their price rise (Bhosale and Mavele, 2018).

Ethereum also known as Ether is a cryptocurrency traded in Ethereum platform (Bhosale and Mavele,2018). Ethereum was created by Vitalik Buterin in 2013 as inspiration from Bitcoin (Bhosale et al., 2018). According to Zhang et al. (2019), Ethereum platform is a decentralized network which is an open source for blockchain based computing allowing anyone to join in to contribute computing to the network. Since it was introduced in 2013 and its development system began in 2015, the prices of Ethereum have gradually risen (Bhosale and Mavele, 2018). In addition, the characteristics of Ethereum are similar to Bitcoin, making the investors of Bitcoin quickly catch up their eyes as compared to when Bitcoin first lived(Zhang et al. ,2019). The familiar nature of Ethereum helped it establish quickly and become the second most popular traded cryptocurrency in today's market (Chen et al., 2017). In fact, the popularity of Ethereum only rose and has no sign of decline as for now. Zhang et al. (2019) predicted the rise of Ethereum and its potential to surpass Bitcoin to be a market leader in the cryptocurrency market.

In Malaysia, cryptocurrency is not illegal to be traded. Malaysia ruled the trade of cryptocurrency falls within Section 73 of the Contracts Act 1950. Section 73 of the CA 1950 stated that, “A person to whom money has been paid, or anything delivered, by mistake or under coercion, must repay or return it". The dispute regarding trade of cryptocurrency will be held in Shah Alam Sessions Court. The legal entity, Securities Commission Malaysia has approved the establishment of 3 cryptocurrency trading companies which are Luno Malaysia Sdn Bhd, SINEGY Technologies (M) Sdn Bhd, and Tokenize Technology (M) Sdn Bhd.

Even with the rise and growth of stability of Ethereum for the past couple of years, the volatility of Ethereum still became a huge debate even in today's study. However, the major take for investors is the volatility of Ethereum, still in the realm of predictable. Naimy et al. (2021) and Ngunyi et al. (2019) have proposed several GARCH-type models to help understand the volatilities of Ethereum. However, their work still leaves a much more desirable result. Even with that, the result of their works is still debatable since those results might be conditional to the period of study and the very nature of cryptocurrency.

1.2 Problem Statement

Ethereum had a healthy growth in prices as it rose as much as 13,000% from 2014 to 2017. As Ethereum's popularity grew, it became particularly crucial for investors to understand the currency's volatility before investing or trading. The issue is that, unlike conventional financial assets, cryptocurrency markets are considered to be extremely volatile and unpredictable, even though they share many of the same characteristics. The high volatility of an asset in financial markets is commonly seen as a negative factor. It also shows that risk in investing cryptocurrencies is higher compared to financial assets. Indeed, these markets are decentralized and unregulated, and subject to manipulation. Cryptocurrencies, according to Chowdhury et al. (2020), are open source and decentralised currencies that operate on a peer-to-peer basis.

Volatilities of cryptocurrencies are different from each other. Andrade and Brandalise (2019) stated that from the end of 2017 to early 2018, Ethereum and Bitcoin have shown higher volatilities. However, Bhosale and Mavale (2018) believe Ethereum and Litecoin are less consistent compared to Bitcoin. Short-term trading, on the other hand, will enforce or earn high profits if traders close and open the correct positions. Due to the high volatility of cryptocurrencies, especially Bitcoin, cryptocurrency trading has made huge profits in recent years. Nevertheless, high volatility may lead to huge loss on investment. So, the investors should find out the volatility of the cryptocurrency before moving to the next investment plan.

The problem also rises in terms of model application in forecasting future price of cryptocurrency. Traditional forecasting methods, especially time series prediction methods or models, rely solely on linear assumptions, which necessitate data that can be broken down into trends, seasonality, and noise to be accurate. These models and methodologies are only useful for activities that have seasonal implications, such as sales forecasting. Due to the complexity of the task, deep learning makes for an interesting technology to be considered based on its performance in similar areas and especially artificial neural network models in deep learning. This is because there is no definitive algorithm when it comes to artificial neural networks, the model builds one as it keeps learning from the data which helps immensely with volatile data.

Due to the increase in computing capacity and data availability, machine learning has become a popular technique (Laxmi et al., 2020). The author also suggested that testing such a power to see whether it could estimate cryptocurrency prices was the best choice. Hence, finding the best predictive model for cryptocurrency forecasting is crucial. Algorithm that can be used to predict the price flow of Ethereum is the Artificial Neural Network (ANN) and Linear Regression model.

1.3 Research Questions

1. What variables are significant in forecasting Ethereum?
2. Which models, Linear Regression or Artificial Neural Network, is the best model to forecast Ethereum?
3. Is Linear Regression and Artificial Neural Network able to forecast Ethereum?
4. What is the volatility effect on Ethereum and compared it to other cryptocurrencies?

1.4 Research Objective

1. To determine variables that are significant in the forecasting of Ethereum.
2. To find the best model between Linear Regression and Artificial Neural Network in forecasting Ethereum price value.
3. To determine the ability of Linear Regression and Artificial Neural Network to forecast Ethereum market price in the short-term and in the long-term.
4. To determine the effect of the volatility of Ethereum and compare it with other cryptocurrencies' volatilities.

1.5 Scope and limitation

We will only use the Ethereum data from 2015 to 2021 as Ethereum has only been introduced in 2015. By limiting to a single cryptocurrency, the results gained will only be viable to Ethereum only. Thus, making the study invalid or inaccurate for others' cryptocurrency.

Another limitation that we faced is that the price has become more volatile as the pandemic is still ongoing. This means that the forecasting will be affected by the pandemic and thus may make it invalid in the future when the pandemics have subsided.

1.6 Significance of study

The findings of this study will be beneficial for the future investor or the people who are considering buying the cryptocurrency, as they will have the ability to forecast the cryptocurrency. In addition, this study will also help the potential investor to reduce their potential losses while maximizing their overall profit. With this study, we are hoping that it will provide investors and other parties with more trust in investing in cryptocurrency.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

Cryptocurrency is a synonym known as digital currency spreads throughout the world without centralizing the central banking system. Ethereum is an open-source programmable virtual currency (Bhosale et al., 2018). Ethereum's popularity as a cryptocurrency is only second to more established Bitcoin. Ethereum had a healthy growth in prices as it rose as much as 13,000% from 2014 to 2017. Cryptocurrency such as Ethereum enjoys its popularity comes from anonymous currency secured against the prying eyes of the central banking system and government (Fantazzini et al., 2016 cited in Naimy et al., 2021). As the Ethereum popularity trajectory escalated, it became more important for investors to gain knowledge about its volatility before jumping to bandwagons. There is now comprehensive research on various aspects of cryptocurrencies, especially for Bitcoin. But currently, Ethereum is the second most advanced market capital and supports much more features than Bitcoin (Chen et al., 2017).

First of all, cryptocurrency such as Ethereum is fundamentally different from traditional currency like Dollar or Ringgit as it is backed by any legal entities or government. A study conducted by Naimy et al. (2018) regarding volatilities for cryptocurrencies against fiat currencies which are Canadian Dollar, British Pound, Japanese Yen, Euro, Swiss Franc and Australian Dollar shows that fiat currencies are more stable in exchange rates and less volatile. The authors also stated that cryptocurrency is more perceptive to changes of markets. Therefore, numerous studies were conducted to predict the market price of Ethereum to help investors closely track cryptocurrency exchange rate fluctuations in achieving higher returns and avoiding losses in fund selection.

2.2 Methods Used by Past Researchers

2.2.1 Linear Regression Model

According to Yatsyk (2020), a methodology for modelling a relationship between a dependent variable and several independent variables is known as multiple linear regression. The author used Least Squares Regression model for the regression analysis and resulted in having positive coefficients for variables indicating price and the search interest are directly proportional.

Cohen (2020) stated that linear regression is a supervised learning method usually used in predicting, forecasting, and finding quantitative data relationships. It is among the earliest learning methods now commonly used. In the method, a continuous line that best suits the prices between two points is referred to as a linear regression line. The term “best fit” applies to the process of constructing a line with the smallest space between the price points and the Linear Regression line. Some traders may perceive a buying opportunity when the prices fall below the linear regression line, whereas a trader may offer for sale when prices rise above the linear regression line.

Linear and multiple regression models were chosen for a variety of reasons, including the simplicity to write, use and understand, their speed of computation, their popularity, and their ability to adapt well to datasets with few features (Uras et al., 2020). In the case of Ethereum price prediction, the components of the independent variable are the values of the closing price of the previous days while the dependent variable represents the closing price to be predicted for the linear regression model. The authors attempted to predict using several independent variables, by supplementing the price function with findings of several other features using the Multiple Linear Regression model (MLR). Regarding the deployed algorithms, both regression models and the neural network model found the best result. However, linear regression models performed better than neural networks in terms of execution speed.

Poongodi et al. (2020) defined the linear regression (LR) method as a method that is used to evaluate the relationship between two continuous variables. The indicator, or independent variable, is one of the variables, while the answer, or dependent variable, is the other. To find a statistical association, the LR algorithm is employed. The main goal of a machine learning algorithm is to find the best fit line for the results. The line that ideally suits the results is the one with the smallest cumulative prediction error (across all data points). The interval between the point and the regression line is known as the defect.

A statistical diagnosis by Nashirah & Sofian (2018) is conducted to determine the relationship between Bitcoin and Ethereum's volatile exchange rate movement. To detect the covariance shift of the exchange rate, the authors used a statistical method of correlation test to investigate the bivariate correlation and regression analysis between both currencies. For the study, Pearson correlation analysis is used to analyse the bivariate correlation as for the aim of linear regression analysis is to evaluate the relationship between dependent and independent variables. The result via a scatterplot shows that linear regression calculates the best-fit axis, minimizing the squared residual. The line with the smallest squared residual error is the optimal linear equipped line. Simultaneously, the better linear fitting line showed the line with the smallest error variance.

By using a linear regression model, we can determine the association between the complex activity of the Bitcoin and Ethereum exchange rates (Bakar & Rosbi, 2018). This model also helps to study the prediction of ether cryptocurrency prices and figure out the kind of patterns that are found over time in Ethereum prices (Poongodi et al., 2020). As stated by Yatsyk (2020), multiple linear regression is proposed to be used for the cryptocurrency prediction and for understanding the correlation between the cryptocurrency price and the analysed variable.

Contradicted to Kumar (2018), he said that due to the non-random, autocorrelated, and heteroscedastic nature of the results, linear regression is ruled out. Since the data does not meet the assumption of linearity, some statistical analyses are performed to rule out linear regression models. He also proposed that a quadratic regression model based on transaction volume can be used jointly to forecast the prices of cryptocurrencies after analysing each of the variables. The author stated that the polynomial regression results and visual display of the model fit show explicitly that a second-degree polynomial regression model is one of the best fits to forecast cryptocurrencies.

Ali Alahmari (2020) attempted a future prediction of cryptocurrency price using linear, polynomial and radial basis function (RBF) but somehow results demonstrated that RBF outperforms most other methods in predicting the parameters needed for the model to forecast the price of cryptocurrencies. The results show in comparison to linear and polynomial methods, it can be inferred that the RBF kernel method can fix the issue of dataset variances and high dimensional data.

2.2.2 Artificial Neural Network (ANN) Model

According to Nor and Amelia (2018), Ethereum (ETH) is a block-chain based technology that uses Turing-complete language programming to construct and perform circulated systems. It is also one of the common types of cryptocurrencies. Since the Ethereum popularity trajectory rose, it is crucial for investors and capitalists to gain knowledge on predicting its future market price to gain more profits. Thus, the Artificial Neural Network (ANN) model may be used to forecast Ethereum price movement.

Artificial Neural Network (ANN) is a statistical model inspired by biological neural networks that constitutes the human brain which has an interconnected group of artificial neurons. It is a remarkably similar system containing interacting processing neurons or nodes which function like the human brain and process the information by interconnecting with a lot of simple processing features (Nor & Amelia, 2018). These networks can gain knowledge and model the nonlinear and complex relationships between inputs and outputs. Several types of ANN models such as Recurrent Neural Networks, Convolutional Neural Network and Multilayer perceptron.

Reaz et al. (2020) conducted a research of forecasting cryptocurrency using neural network models and 500 training cycles are trained using the model hence the results were compared with other algorithm models such as Light Gradient Boosting Machine (Light GBM), Gradient Boosting Decision Tree (GBDT) algorithm, it showed that neural network, Long-Short-Term Memory (LSTM) gave better accuracy. It implies that neural networks are good tools to help investors in predicting the price of cryptocurrency, especially Ethereum. Researchers nowadays are interested in constructing complex networks with multiple layers which can handle lots of problems. According to Laxmi et al. (2020), the main benefit of the LSTM model is that it allows each LSTM cell to recognize patterns for a certain period in which they essentially can remember crucial information and forget or remove useless information.

Neural Networks can solve multiple complex problems compared to other models. Multi-layer perceptron (MLP) which is a type of neural network is different from logistic regression because it can support one or more non-lineal layers compared to logistic regression (Franco et al., 2019). This model also helps to find out the prediction of ether cryptocurrency prices and figure out patterns or trends that are found over time in Ethereum prices. Mohil et al. (2020) stated that Traditional time-series models, for example ARIMA (Auto Regressive Integrated Moving Average), ARCH (Autoregressive Conditional Heteroskedasticity), and GARCH are used frequently for various financial schemes analysis. These are often used for time series prediction, however, have disadvantages due to assumptions. Long Short Term Memory (LSTM) has proven to be the best for now because of their ability to remember, recognize and extract the temporal features of data (Mohil et al., 2020). Hence, it can be concluded that Neural Networks models are one of the best models to forecast price movement of Ethereum.

2.2.3 GARCH (1,1)

GARCH (1,1) is one of the models that can determine the volatility of cryptocurrency and it is often used by past researchers. According to a study done by Andrade and Brandralise (2019), GARCH (1,1) models determine the volatility of Bitcoin and Ethereum. The authors found that Bitcoin has more volatility as compared to Ethereum in 2016. This study is important as Bitcoin can be traded to most cryptocurrency and often used as standardised cryptocurrency. Shaza et al. (2019), conducted a study to compare Ethereum and other cryptocurrency using GARCH (1,1) and more sophisticated models such as EGARCH, PGARCH and TGARCH. The best models to explain them are different for every cryptocurrency. In parallel, Nyungi et al. (2019) conducted a study using more sophisticated models of GARCH to determine the volatility of cryptocurrency. Naimy et al. (2021), also used sophisticated models of GARCH to model the volatility of cryptocurrency. Both of them reach the same conclusion that supports the finding of Shaza et al. Since GARCH (1,1) models are basic models to model the volatility, it is commonly used by past researchers.

2.3 The Long-Term and Short-Term Forecasting of Cryptocurrency.

Cryptocurrency is the new form of digital asset in this new decade of development. As the cryptocurrency is a virtual currency, it is secured by cryptography which makes it almost invulnerable to counterfeit. This makes the cryptocurrency a super secured currency that can be used in this era of technology. As technology advances, the significance of cryptocurrency has been skyrocketing for the past few years. Just like

Bitcoin value has been increasing exponentially for the past few years, it is worthy to say that this market is worth investing in. However, in the rapid increase of the Bitcoin value, it also experienced a market crash just a few months after reaching its peaks in December of 2017 which dropped at 250% of the highest value. The fluctuations of the cryptocurrency prices are radical. Therefore, the need to predict the cryptocurrency prices is crucial to avoid such heavy losses. There are multiple methods and techniques that can be used to analyse and predict the cryptocurrency whether it is short-term or long-term. Data mining is the best example to highlight the ability to predict cryptocurrency as it can deduce a lot of information from a large batch of data. However, most of the research regarding cryptocurrency has only involved the short-term part while neglecting the long-term prediction.

There are a lot of techniques and methods that can be used to analyse cryptocurrency whether it is long-term or short-term. We can see how the technology advancement has given us abundant simplicity as we see in machine learning and Artificial Intelligence. However, we must keep on researching new ways that can analyse cryptocurrency. This is because researching new methods can be ground-breaking to the future of forecasting. It also may enable us to do forecasting in cryptocurrency or other types of investment with near zero risk.

2.4 Volatility of Ethereum

Volatilities of cryptocurrencies also vary from each other. Andrade and Brandalise, 2019 shows that Bitcoin and Ethereum show higher volatilities at the end of 2017 to early 2018. However, Bhosale and Mavale, 2018 believes Bitcoin is more consistent as compared to Ethereum and Litecoin. The contracted result is believed due to the still developed nature of cryptocurrencies contributing to different results due to the period of study (Brauneis and Mestel, 2018 cited in Charfeddine and Maouchi, 2019).

Furthermore, the cryptocurrency also shows different signs of persistence of volatilities. Shaza et al. (2019) using GARCH-type models shows that the Ethereum does not show persistence to volatility as compared to Bitcoin. The result from the GARCH model from Ngunyi et al. (2019) also acts as supplement evidence that Ethereum does not have persistence behavior regarding volatilities meanwhile Bitcoin has.

2.5 Impact of Volatilities of Cryptocurrency

Research done by Malholtra and Gupta, 2019 about cryptocurrencies and Asian stocks to study volatiles split over and correlations indicates positive relationship between strong risk taking in cryptocurrency market. The authors believed volatilities cryptocurrency used as a leverage in trading in stock markets is possible to gain profits. This hypothesis is backed by study done by Leverik,2019 as he believed there is a positive relationship between volatility of cryptocurrency with S&P 500 index. Moreover, the positive relationship between volatilities of cryptocurrency with other financial assets like American indexes is also observed by Ghobel et al, 2021.

CHAPTER THREE
METHODOLOGY

3.1 Introduction

In this chapter, we explain the description and collection of the data, also the methodology that was used in the research. Initially all the data collected were analysed accordingly to fulfil the research objectives. Furthermore, every detail of formulas, models and methods used to find the best solution will be further explained in this chapter as well.

3.2 Data Description

The data used in our research are the historical daily data of cryptocurrencies which are Bitcoin (BTC), Ethereum (ETH), Litecoin(LTC) and Dash. Each has 1459 observations with 7 variables. The variables are date, opening price, closing price, low price, high price, adjusted closing price and volume. We also divided the data in terms of daily and monthly as the daily is used for short-term forecasting while the latter will be used for long-term forecasting.

3.3 Data Collection

Data for Bitcoin, Ethereum, Litecoin and Dash prices are directly extracted from finance.yahoo.com. All data from finance.yahoo.com are publicly available for view, download and used by the public. Data from 31 March 2017 to 1 April 2021 are used for this research. The data used are ETH-USD and BTC-USD which are respectively based on the United States Dollar (USD).

Date	Open	High	Low	Close	Adjusted Close	Volume
3/31/2017	51.7537	51.7682	47.3802	50.0373	50.0373	151416000
4/1/2017	50.0336	51.9282	48.8753	50.6995	50.6995	92461904
4/2/2017	50.7384	51.2745	45.4258	48.7486	48.7486	134604000
4/3/2017	48.8212	48.8212	43.4096	44.3564	44.3564	190512992
4/4/2017	43.9217	45.7396	41.7218	44.6418	44.6418	157568000

4/5/2017	44.6610	47.8433	44.5381	45.3040	45.3040	119733000
4/6/2017	45.2205	45.6418	40.9011	43.2421	43.2421	147120000
4/7/2017	42.8651	44.0374	41.6460	42.1626	42.1626	81474704
4/8/2017	41.7984	45.2052	41.7308	44.3070	44.3070	74138800
4/9/2017	44.2803	44.4905	42.8927	43.2669	43.2669	55143000

Table 3.3.1 First 10 Observations of Ethereum Daily Data

Date	Open	High	Low	Close	Adjusted Close	Volume
1/4/2017	50.034	79.021	40.901	79.021	79.021	3154319672
1/5/2017	79.322	236.965	73.087	230.669	230.669	14679954720
1/6/2017	230.886	414.756	214.480	294.916	294.916	34410801184
1/7/2017	293.353	295.509	133.723	203.871	203.871	31034420096
1/8/2017	204.688	388.748	204.688	383.042	383.042	33197513792
1/9/2017	383.467	390.044	195.035	301.465	301.465	24310299968
1/10/2017	301.547	349.345	277.575	305.879	305.879	13427328896
1/11/2017	305.762	522.307	281.172	447.114	447.114	31904819744
1/12/2017	445.209	881.944	414.411	756.733	756.733	79862276544

Figure 3.3.2 First 10 Observations of Ethereum Monthly Data

3.4 Framework

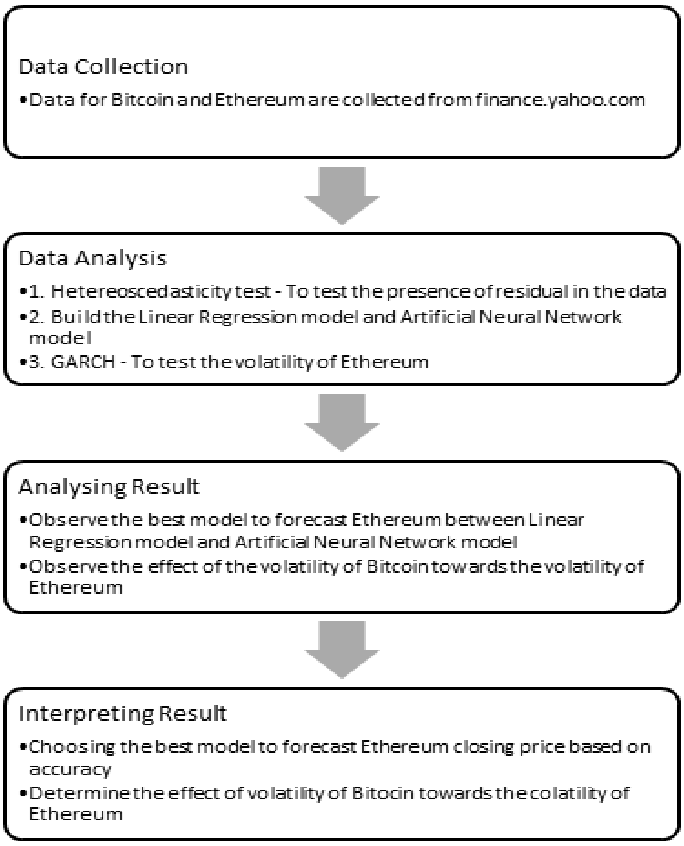


Figure 3.4.1 Theoretical Framework

3.5 Machine Learning

Mastering analysis is necessary for trading success. Technical analysis and fundamental analysis are two ways of determining future value. Technical analysis predicts future prices using trading data from the market, such as price and trading volume, whereas fundamental analysis forecasts price using data from outside the market, such as the economy, interest rates, and geopolitical issues. Many investors concentrate on technical analysis, while others focus on fundamental analysis. Some investors, on the other hand, are interested in the overlaps between fundamental and technical analysis. The purpose of this project is to give technical analysis using machine learning algorithms. For more than two decades, machine learning has been established as a credible model in classical statistics in the forecasting field.

3.5.1 Steps in Machine Learning

The first step is the data collection. Data collection is a crucial part of machine learning. The accuracy of the model is highly determined by the quantity and the quality of the collected data. Thus, a reliable source of data is important to build an accurate machine learning model. The data was collected using yahoo.com.finance.

The next step is to prepare the data. We gather information regarding the data and prepare it for training. Any missing data values, error values and data conversions will be dealt during this process. Visualisation of data can aid in the detection of significant correlations between variables or class imbalances and bias or conduct other exploratory research. We then split the data into training dataset, validation data set and test set.

The third step is to choose the model. Different models have different algorithms and each can be used to model the new data set. The algorithms that are used in the project are Linear Regression and Artificial Neural Network.

Next, we will train the data set. The goal of training the data set is to predict the price of future Ethereum as accurately as possible. Initially, the model is assigned with random values and input given. The achieved output is compared to the actual output, and the disparity is reduced by experimenting with different weight and bias levels. The iterations are continued until the model achieves the desired degree of accuracy utilising different entries from our training data set.

We then evaluate the model after the model has been trained, it must be tested to see if it will perform well in real-world scenarios. The model is tested within previously unseen data. This unseen data is intended to be indicative of model performance in the real world, yet it nevertheless aids in model optimization.

Then we will do some parameter tuning. This process aims to improve on the positive outcomes of the previous step (evaluation). There are diverse ways to improve the prediction accuracy of the model. One of these is to refer to the training stage and train the model using several sweeps of the training data set. This could lead to increased accuracy because the longer the training period gives the model more exposure and increases its quality.

Lastly, the models will perform the prediction. This is the point at which the model is ready for use in practical applications. The model will be used to forecast the future price of Ethereum using the data set provided. The model will also forecast the price in short-term and long-term investment.

3.6 Data Analysis

3.6.1 Heteroscedasticity Test

Heteroscedastic is a systematic change in the spread of the residuals over the range of measured values which becomes a problem because ordinary least square (OLS) regression assumes that all residuals are drawn from a population that has a constant variance (Kumar, 2018). Therefore, we need to do heteroscedasticity tests on our data by using Breusch-Pagan test.

For a regression model,

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + U \tag{1}$$

If the data is homoscedastic then

$$Var(U / X) = \sigma^2 \tag{2}$$

If the data is heteroscedastic then

$$Var(U / X) = \sigma^2 f(x) \tag{3}$$

Thus,

$$\widehat{U}^2 = \delta_0 + \delta_1 X_1 + \delta_2 X_2 + \cdots + \delta_p X_p \tag{4}$$

$$H_0 : \delta_1 = \delta_2 = \delta_3 = \cdots = \delta_p = 0$$

H_1 : At least one of them is not equal to zero.

If the chosen delta is not equal to zero, we will then reject the null hypothesis. Therefore, there is evidence that heteroscedasticity is present in the data.

3.6.2 Assumptions

Our objective is to predict the price of Ethereum using machine learning. The machine learning that we use here is R. By using R, we will build a linear regression model using the opening price of Ethereum as the independent variable and the closing price of Ethereum as the dependent variable. But before we could build the linear regression model using the data collected, the data must be checked if it can be analysed using the linear regression method. The data must be checked with assumption as follows:

The first assumption for the data to be able to be analysed with a linear regression method is that the two variables should be measured at the continuous level. Both the opening and closing price are at the continuous level as they are interval variables. The second assumption is that there must be a linear relationship between the two variables. Referring to diagram (1), the opening price and closing price of Ethereum has a linear relationship. The third assumption is there should be no significant outlier. Referring to diagram (1), the data has no significant outlier.

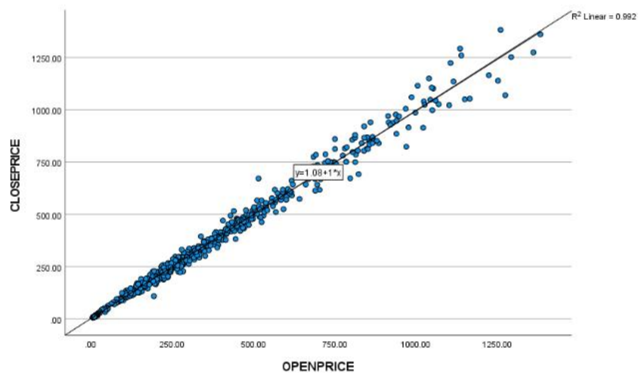


Figure 3.6.2.1 Scatterplot of Opening Price and The Closing Price of Ethereum

The fourth assumption is that the data should have independence of observations by checking the correlation matrix in R. The fifth assumption is that the data needs to show homoscedasticity. Based on what we explain earlier in the data analysis, the data is required to do the heteroscedasticity test. The test can be done using the R. The data is homoscedasticity if the variances along the line of best fit remain similar. The data is heteroscedasticity if the variances dispersed from the line of best fit. If the data is heteroscedasticity, the best way to continue the analysis is to do the Breusch-Pagan test. To do the Breusch-Pagan test we try to fit a quadratic model to see if we can solve the heteroscedasticity problem, there are several options: compute (i) the square term of independent variable, (ii) logarithmic transformation of dependent variable, (iii) square term of dependent variable

Finally, the sixth assumption is to check the residuals of the regression line are approximately normally distributed. After the data has followed all the assumptions stated above, then we can proceed to build the linear regression model using the data.

3.7 Linear Regression

The objective of this research is to predict the closing price of Ethereum. According (Uras et al., 2020), linear regression model represented by the main equation:

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 \tag{5}$$

Where:

y: closing price to be predicted

b_0: intercept

b_n: slope coefficients

x_1, x_2, x_3: values of the open, high and low price

The algorithm aims to find the best fit for the data, which best describes the relations between dependent and independent variables.

3.8 Linear Regression in R Studio

Linear regression is used to predict the value of a continuous variable Y based on one or more input predictor variables X. The aim is to establish a mathematical formula between the response variable (Y) and the predictor variable (X).

For this analysis, we will use the Ethereum daily and monthly dataset that is obtained from (YAHOO) and save it in format (.xlsx). The data consists of 1459 observations (rows) and 7 variables (columns) Date, Open, High, Low, Close, Adjusted Close and Volume.

The goal is to establish a mathematical equation for close price as a function of open price, so we can use it to predict the close price when only the open price of Ethereum is known. Before we begin building the regression model, we need to analyse the variables.

Typically, for each of the predictors, plots will help to visualize the patterns. Scatter plots can help visualise linear relationships between the response and the predictor.

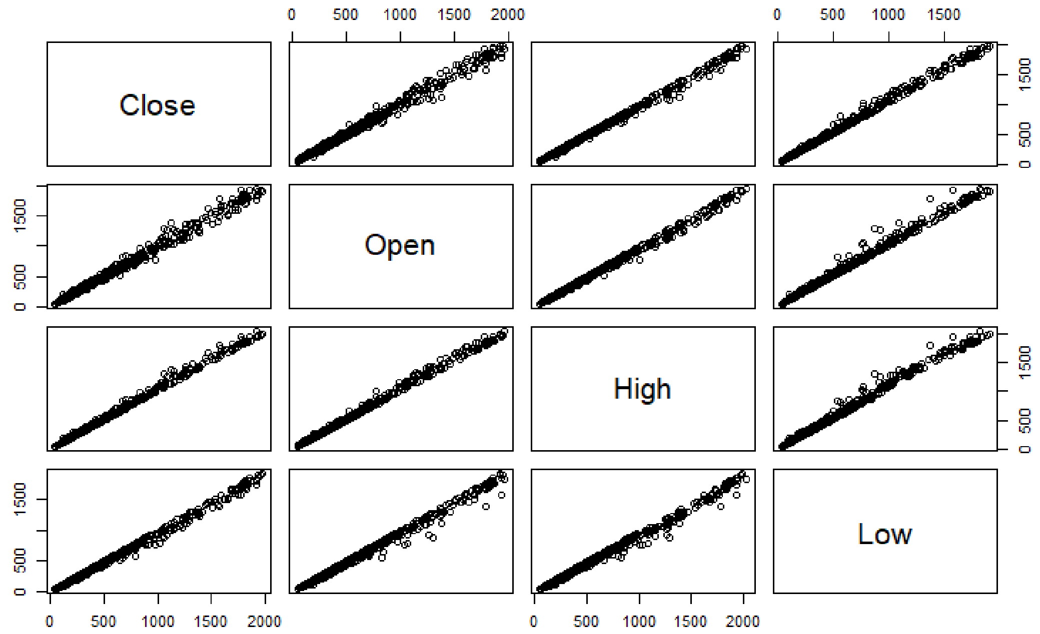


Figure 3.8.1 Scatterplot of Close, Open, High and Low Price of Ethereum

The scatter plot along with the smoothing line above suggests that the opening, high, low and closing price of Ethereum are related in a linear and positive relationship. Other than that, we can use the density plot to check if the response variable is close to normal. To apply the density plot, we must install packages `e1071` for the skewness function. From the result below, when we run the summary we can see that the data is skewed to the left

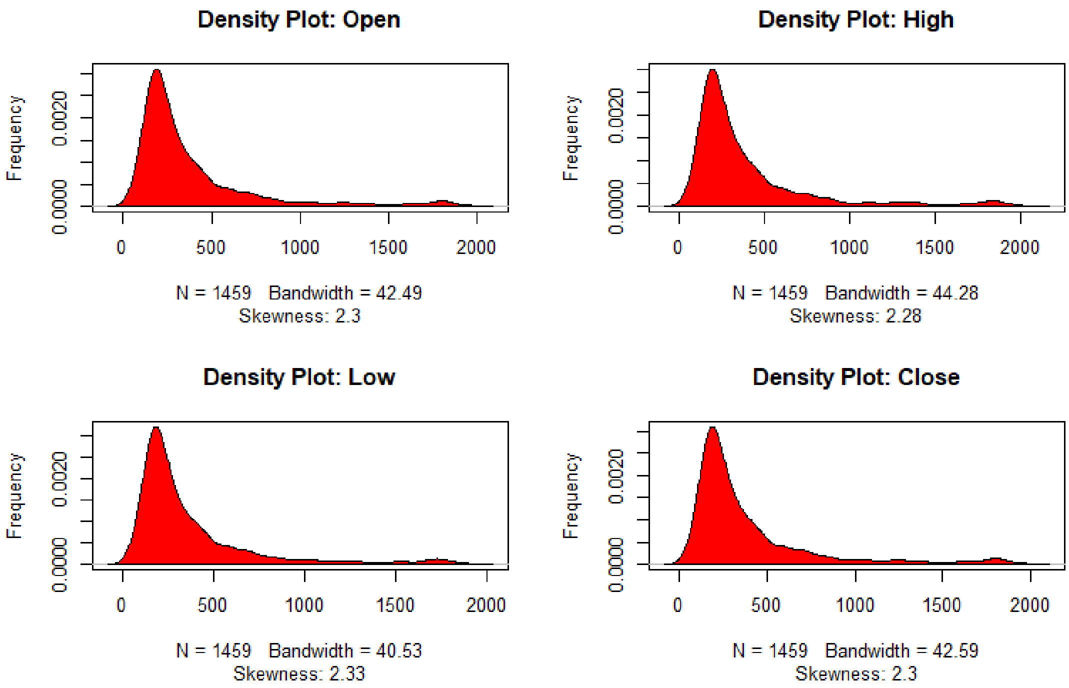


Figure 3.8.2 Density Plot

Then, to investigate the strength of the association between two continuous variables, we perform a correlation analysis. It entails calculating the coefficient of correlation between two variables. The two variables must be present in pairs in order to compute correlation. The correlation coefficients might range from -1 to +1. They have a strong positive correlation (value near to +1) if one variable consistently grows with the rising value of the other, and a strong negative correlation (value close to -1) if one variable consistently declines with the increasing value of the other (value close to -1). A number near 0 indicates that the variables have a weak association. We just use the cor() function with the four numeric variables as parameters to compute correlation.

3.8.1 Build the Linear Regression Model

```
#LINEAR REGRESSION
model_1 <- lm(Close~Open+High+Low, data=eth.day)
plot(Close~Open+High+Low, data=eth.day)
abline (model_1)
summary(model_1)
```

Figure 3.8.1.1 Create a linear model

The lm() function is used to create a linear model. We have created the relationship between the predictors and response in the form of a mathematical formula by developing the linear regression model.

The Akaike's information criterion (AIC) and the Bayesian information criterion (BIC) are measures of the goodness of fit of the linear regression model and can also be used for model selection.

Thus far, we have constructed a linear regression model from the entire dataset. Next, the preferred method is to split the dataset into an 80:20 sample (training: test), then create the model on 80% sample, and then apply the model to predict the response variable on test data. As a result of doing this, we will have the model predicted values for the 20% of data and the actual values from the original dataset.

```
# CREATE TRAINING AND TEST DATA
set.seed(100)  # setting seed to reproduce results of random sampling
trainingRowIndex <- sample(1:nrow(eth.day), 0.8*nrow(eth.day))  # row indices for training data
trainingData <- eth.day[trainingRowIndex, ]  # model training data
testData  <- eth.day[-trainingRowIndex, ]   # test data
```

Figure 3.8.1.2 Splitting the dataset

To create the training and test data, it can be simply done using the sample()function and set.seed() so the samples can be recreated for future use. After that, fit the model on training data and predict Close on test data.

Training Set	1167 observations of 5 variables
Testing Set	292 observations of 5 variables

Table 3.8.1.3 Data Partition for Daily Dataset

Training Set	39 observations of 5 variables
Testing Set	10 observations of 5 variables

Table 3.8.1.4 Data Partition for Monthly Dataset

```
# Min-Max Accuracy Calculation
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))

# MAPE Calculation
mape <- mean(abs((actuals_preds$predicteds - actuals_preds$actuals))/actuals_preds$actuals)
```

Table 3.8.1.5 Prediction accuracy

The prediction accuracy of the model can be determined by computing accuracy measures such as min max accuracy and error rates (such as MSE or MAPE)

To calculate the Min Max accuracy and MAPE is such follows:

$$MinMaxAccuracy = mean\left(\frac{min(actuals, predicteds)}{max(actuals, predicteds)}\right) \quad (6)$$

$$MeanAbsolutePercentageError(MAPE) = mean\left(\frac{abs(predicteds ? actuals)}{actuals}\right) \quad (7)$$

3.9 Artificial Neural Network (ANN)

The next machine learning is the Artificial Neural Network (ANN) model or commonly known as Neural Network. According to Nor and Amelia (2018), it is a computing system that operates similarly to a human brain, consisting of interconnected and communicating processing nodes or neurons, and processes information by interacting with a variety of simple processing features. Each layer of a neural network is made up of multiple perceptrons. Since inputs are only interpreted in the forward direction, ANN is also known as a Feed-Forward Neural Network. Neural networks may be used to model complex input-output relationships or to find trends in data to forecast cryptocurrency prices. The model will be applied to forecast future Ethereum prices.

The Neural Network consists of 3 types of layers:

1. Input Layer: Initial data for the neural network.
2. Hidden Layer: layer between input and output, where all computation is done.
3. Output Layer: Generate results based on inputs given.
4. Artificial neuron: pathway which connects between the layers

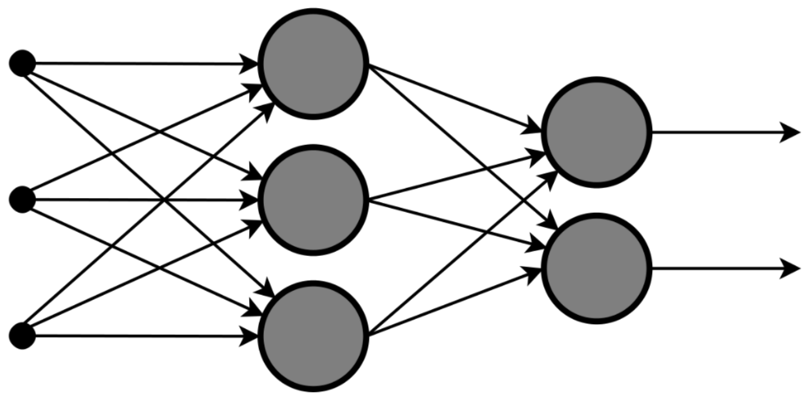


Figure 3.9.1 Artificial Neural Network

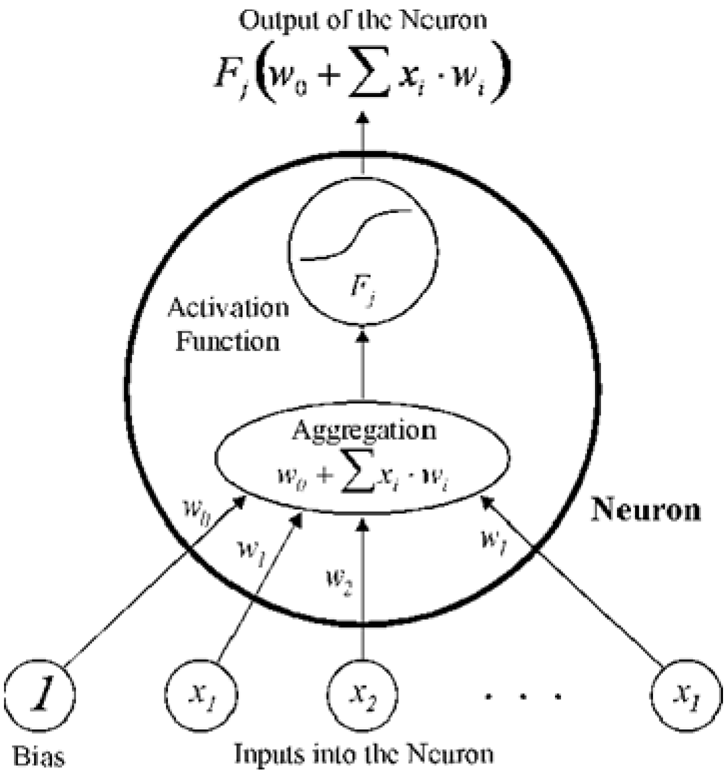
An artificial neuron receives inputs and generates a result. The overall activity is described as a node's activation function. The reason ANN is an excellent choice to predict price flow of cryptocurrencies is because of the ability of ANN to learn any nonlinear function. As a result, these networks are often referred to as Universal Function Approximators. ANN can learn weights that map any input to a desired output. With the help of activation functions, the network can learn any complex relationship easily.

Therefore, this research will be focusing on building predictive models using Neural Networks that can accurately forecast the future price of Ethereum. The model can be carried out using several statistical software such as SPSS (Statistical Package for the Social Sciences) programming, R studio and Excel VBA programming.

3.10 Artificial Neural Network in R Studio

Neural network model to predict the price of Ethereum will be carried out in RStudio. Neural networks (also known as artificial neural networks) may learn from examples. A neural network is made up of interconnected information processing units, like the human nervous system, which is made up of interconnected neurons. The data processing units do not operate linearly. In fact, the strength of a neural network comes from the simultaneous processing of data, which allows it to deal with non-linearity. The use of a neural network can help infer meaning and find patterns from large data sets.

3.10.1 Build the ANN Model



(8)

Figure 3.10.1.1 Artificial Neural Network Process

Where:

x_i = Input Variables

w_i = Weights of respective inputs

l = Bias

F = Activation function

Based on the figure 3.9.1, the middle part is the activation function process. The Activation Function is a mechanism for mapping inputs to outputs. The process splits into two parts which are the combination function (aggregation) and transfer function. The combination function combines the inputs into a single value, which then passes to the transfer function to create output. The combination function apply formula from below equation:

$$G = W_0 + \sum \left(X_i \times W_i\right) \tag{9}$$

Where G is the weighted sum of outcome before activation function is applied, w_0 is the bias weight, x_i is the input values and w_i is the weight of respective inputs. The neuron's bias and weights are both adjustable factors. Each input has different weights. Then, the value is passed to the transfer function. Transfer function is a mathematical representation of the relationship between inputs (“Open”, “High” and “Low”) and output (“Close”).

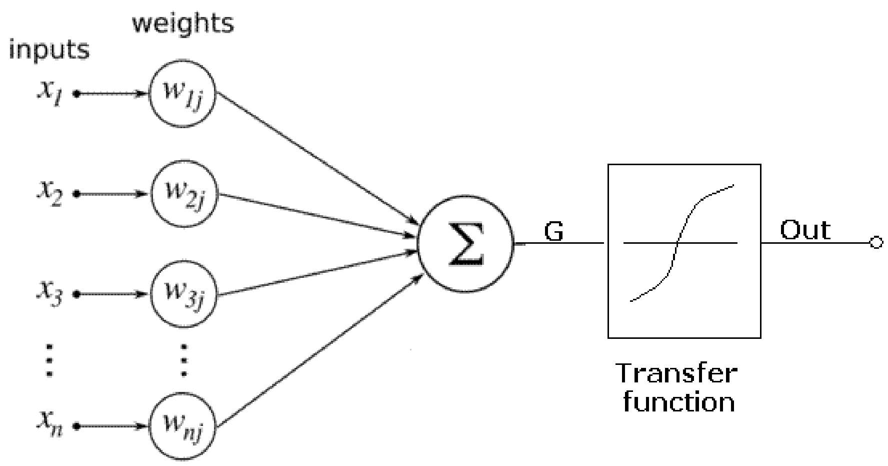


Figure 3.10.1.2 Transfer Function in ANN

There are several types of transfer functions that can be used in ANN models. Based on the figure above, the G (weighted sum) value is passed to the transfer function. The function used in the model is the sigmoid function known as logistic function. The following equation is as below:

$$sigmoid\ function = f(G) = \frac{1}{1 + e^{(-G)}} \tag{10}$$

Where f(G) is the output, the value is displayed as the outcome at the end. Some learning rules are used to alter the parameters. The outcomes of neurons can be negative or positive and the neurons do not know boundaries. As a result, a mapping mechanism (activation function) between the neuron's input and output is required.

Autoregressive Neural Network

Lagged values of the time series can be utilised as inputs to a neural network with time series data, similar to linear autoregression model. Thus, it is called an autoregressive neural network or NNAR model. The NNAR(*p*,*k*) function indicates that there are *p* lagged inputs and *k* nodes in the hidden layer. For example, a NNAR(3,2) model is a neural network with the last three observations (y_t-1, y_t-2, y_t-3) used as inputs for forecasting the output y_t, and two neurons in the hidden layer.

The nnetar () function fits an NNAR (*p*, *P*, *k*) model. The *p* value is the number of non-seasonal lags while *P* value is the number of seasonal lags used as inputs. If these two values are not specified, they are selected automatically. For the seasonal time series, *p* is determined from the optimal linear model fitted to the seasonally adjusted data and the default value is *P* = 1.The default *k* value is *k* = (*p* + *P* + 1) / 2 (rounded to the nearest integer).

3.10.2 Gathering Data

Neural network is used to forecast the future close price of Ethereum (Y) using more than one variable (X). The aim is to study and analyse the complex interactions between the input variables (X) and the target variable (Y) with the goal of predicting final component properties. The obtained data will be imported into RStudio.

For this analysis, we used the Ethereum daily and monthly dataset obtained from Yahoo. The daily data consists of 1459 observations (rows) and 7 variables (columns) Date, Open, High, Low, Close, Adjusted Close and Volume. Meanwhile, monthly data consists of 49 observations (rows) with the same number of variables. Both data start from April 2017 to April 2021.

In R studio, Ethereum daily and monthly data are imported. The date, close price, open price, low price and high price variables are used in building the model. The adjusted close price and volume of Ethereum are not included for ANN forecasting model.

3.10.3 Data Preparation

```
CloseTS <- ts(data$Close, start=c(2017, 1), frequency=365)
time <- time(CloseTS)
CloseTS
```

Figure 3.10.3.1 Fitting Dataset into Time-series Object

The above figure shows daily and monthly datasets are stored in an R object called time-series object under variables CloseTS . Using ts() function, the close price is the target variable and the data is stored starting from year 2017 with frequency = 365 for daily data. For monthly data, the frequency = 12 and the others remain the same.

```
validLength <- 292
trainLength <- length(CloseTS) - validLength
CloseTrain <- window(CloseTS, end=time[trainLength])
CloseValid <- window(CloseTS, start=time[trainLength+1])
```

Figure 3.10.3.2 Data Partition

The data is partitioned into two sections which are training data set and testing data set. Most of the dataset will be used to train our model in the first section. The second section will be utilised to assess the performance of the trained model.In R studio, the data is partitioned with Close price of Ethereum as target variable and Open, High, Low price of Ethereum as input variables.

In this project, 80% of the data will be the training set and 20% of the data is the testing set. Based on the figure above, the daily dataset is partitioned where 292 observations are the testing set and 1167 observations are the training set. Most of the training set data is from year 2017 to 2020 and the others are the testing set.Monthly dataset is also partitioned using the same ratio.

Training Set	1167 observations of 5 variables
Testing Set	292 observations of 5 variables

Table 3.10.3.1 Data Partition for Daily Dataset

Training Set	39 observations of 5 variables
Testing Set	10 observations of 5 variables

Table 3.10.3.2 Data Partition for Monthly Dataset

The result above shows that the data is resampled and partitioned. Training set and testing set are obtained. Data partitioning can improve query performance and manageability of predictive models.

3.10.4 Building Neural Network Model Function

After building the linear regression model, we proceed with building the ANN model. In order to test the model for different datasets, we need to understand the flow of the neural network before forecasting the close price of Ethereum.

```
nn <- neuralnet(Close ~ Open + High + Low , data = TrainingSet,
                hidden = c(3,2), linear.output= FALSE,
                threshold = 0.01 )
nn$result.matrix
plot(nn)
```

Figure 3.10.4.1 Plotting Neural Network

Using R studio, “Neuralnet” library is loaded into the R console. Neuralnet is used to regress the dependent variables, Close price (Close) against the other independent variables (Open, High, Low). Given that the impact of the independent factors on the dependent variable (dividend) is anticipated to be non-linear, the linear.output variable is set to FALSE. The threshold is set to 0.01, which means that if the change in error during an iteration is less than 1%, the model will not do any more optimization.

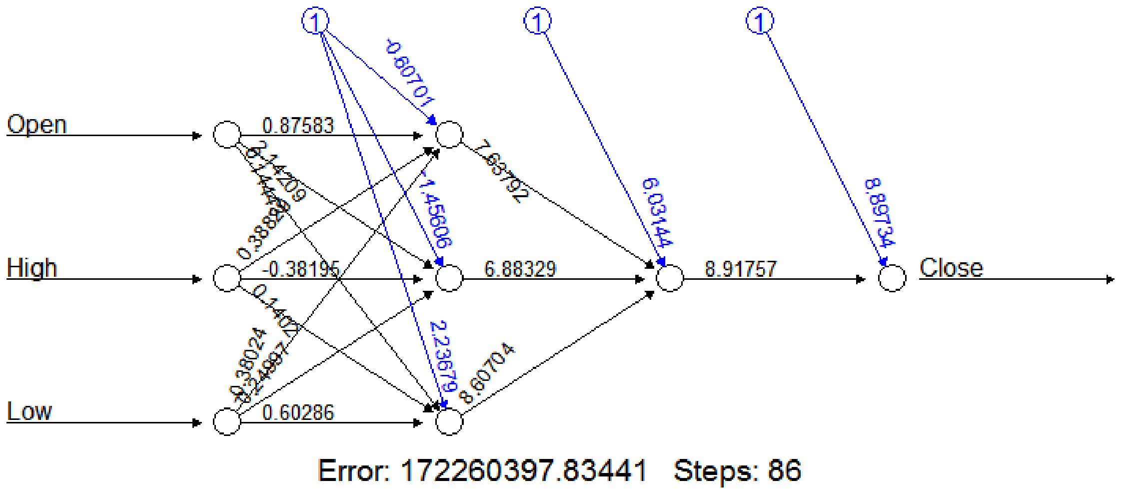


Figure 3.10.4.2 ANN plot in R Studio

```
set.seed(227)
CloseNN <- nnetar(CloseTrain, repeats = 100, P=1,size = 3)
CloseNN
```

Figure 3.10.4.3 Building ANN model

Using the nnetar() function in R from the above figure, the neural network model is fitted in the time series model. The nnetar() function in R's forecast package fits a neural network model to a time series using the time series' lagged values as inputs (and some other exogenous inputs). For the parameter,the ANN model is built using the training data set and repeats = 100 means there are 100 of networks to fit with different random starting weights. The number of nodes in the hidden layer are three. Figure 3.10.4.3 shows only the daily dataset, however the monthly dataset also uses the same method.

3.10.5 Trained Model Evaluation for Daily and Monthly Dataset

The next step is to evaluate and compare the ANN model performance with two different datasets. It is important to identify which dataset is suitable to apply ANN model for forecasting purposes. First of all, we calculated the accuracy of the trained model for both datasets.

Actual and Predicted Close Price Graph Plot

```
CloseNN.pred <- forecast(CloseNN, h = validLength)
accuracy(CloseNN.pred, Closevalid)
CloseNN.pred
```

Figure 3.10.5.1 Validating Trained ANN Model

Figure above shows the R coding of validating trained ANN model. The trained model is used to forecast and then compare the predicted close price with actual price from the test dataset. The accuracy is taken into account. Furthermore, the actual and predicted close price graph is plotted.

Check Residual of Model

```
checkresiduals(CloseNN.pred)
```

Figure 3.10.5.2 Residuals of the ANN model

Then, checkresiduals() function is used to check the residuals, histogram distribution and the ACF (Autocorrelation factor) plot of the ANN model. The residual plots are used for performance comparison.

Coefficient of Determination, R^2

```
r <- cor(fitted(fit_nnetar)[14:length(eth_data)],
        eth_data[14:length(eth_data)])
r2 <- cor(fitted(fit_nnetar)[14:length(eth_data)],
        eth_data[14:length(eth_data)])^2
print(r2)
```

Figure 3.10.5.3 Coefficient of Determination, R^2

Next, We compare the performance of trained models by referring to the coefficient of determination, R^2 . It is used to represent that proportion of the variance for target variables (close price) that is explained by independent variables (open, high, and low price). The higher the R^2 value, the higher the accuracy of the model. The R^2 can also be calculated manually using formula.

$$R^2 = 1 - \frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2} \quad (11)$$

Where:

y = actual value of close price

$\hat{y}$ = predicted value of close price

$\bar{y}$ = mean value of close price

Measurement and Error Analysis

Then, there are also several values that are tabulated to check the accuracy of the ANN model. The accuracy of the model can be measured using mean absolute percentage error (MAPE), mean absolute error (MAE), root mean square error (RMSE), and mean square error (MSE).The lower the error measures, the higher the accuracy of the model. Below is the formula for MAPE and MAE.

$$MAPE = \frac{100\%}{n} \sum \left| \frac{y - \hat{y}}{y} \right| \tag{12}$$

$$MAE = \frac{1}{n} \sum |y - \hat{y}| \tag{13}$$

$$RMSE = \sqrt{\frac{1}{n}(y - \hat{y})^2} \tag{14}$$

$$MSE = RMSE^2 = \frac{1}{n}(y - \hat{y})^2 \tag{15}$$

Where:

n = number of observations

y = actual value of close price

$\hat{y}$ = predicted value of close price

$\bar{y}$ = mean value of close price

After considering all the residuals, R^2 values and error measures, we could conclude which dataset ANN model performed well.

3.10.6 Trained Model Performance between LR and ANN model

Proceed to the next step which is comparing the model performance with another type of model, linear regression. In this section, both models are compared using coefficient of determination, R^2 and error values (MAE, MAPE, RMSE and MSE) and are tabulated. Model with the lowest error values and the higher R^2 value is considered as the best forecasting model.

3.10.7 ANN Model performance in Short-term and Long-term Period Forecasting.

We continue with assessing performance of ANN model in predicting future close price of Ethereum in short-term and long-term period using daily dataset and monthly dataset..

```
forecast_ <- forecast:::forecast.nnetar(fit_nnetar, h = 25,
                                        level = c(75,95), PI = TRUE)
autoplot(forecast_)
summary (forecast_)
```

Figure 3.10.7 Forecasting Future Close Price

From the figure above, the model predicted the future close price of Ethereum using the forecast() function in R. Since we forecasted with ANN model, forecast.nnetar() function is used. h = 25 is the number of periods for forecasting. For monthly dataset, h = 25 while for daily dataset, h = 100.The model is plotted, and the value of future daily and monthly close price are tabulated for reference. Summary() function is used to analyse the predicted model.

Lastly, the accuracy of the predictive models are measured using error values and are tabulated. Then, we could infer whether ANN model is able to predict future close prices in short-term and long-term periods.

3.11 GARCH (1,1)

Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) is a model developed in 1986 by Dr. Tim Bollersev (Shazia et al., 2019). In this research, GARCH (1,1) is used to determine the volatility of Ethereum and Bitcoin. The formulations of GARCH (1,1) are as follow:

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2 \tag{15}$$

Where:

$$\sigma_t^2 = the\ conditional\ variance$$
$$\omega\ = a\ constant$$
$$\alpha\ = the\ parameter\ that\ captures\ the\ short-run\ persistence\ of\ volatility$$
$$\beta\ = the\ parameter\ that\ captures\ the\ long-run\ persistence\ of\ volatility$$
$$\varepsilon_{t-1}^2 = the\ residual$$

Limitation:

$$\omega > 0$$
$$\alpha, \beta > 0$$

The αrepresent the value of time series at time t affected by the residual and the volatility at time t, σ_t^2. σ_t^2 is affected by value of time series α_{t-1} at time t-1 and also the volatility at time t -1, σ_{t-1}^2. It means that the volatility at time t is affected by time series and volatility at time t-1.

3.11.1 Volatilities of Ethereum, Bitcoin, Litecoin and Dash

The volatilities of Ethereum, Bitcoin, Litecoin and Dash are calculated using machine learning. Machine learning used to calculate both volatility is R-programming. The packages used are ggplot2, caTools, dygraphs, xts, forecast, fGarch and tseries.

The ggplot2 package is a system for declaratively creating graphics.The ggplot2 is used to map the variables to aesthetics and graphical primitives. The caTools includes a moving function and is extremely useful in moving data. dygraphs is used to explore and interpret dense data sets. The xts package is used to manipulate the data into extensible time-series objects. The forecast package is used to forecast the result from the data. The fGarch is a a collection of functions to analyze and model heteroskedastic behavior in financial time series models and tseries is a time series analysis and computational finance

First of all, the data was extracted from excel in the form of comma separated value (csv) into R using the read.csv function. Then the data was manipulated using xts function to create xts data.

Then, the graphs of the closing price of Ethereum, Bitcoin, Litecoin and Dash are plotted using the ggplot function in R. The volatility of the closing price of both cryptocurrencies also plotted using the same function.

The GARCH (1,1) model for both Ethereum, Bitcoin, Litecoin and Dash are created using the garchfit function in R. The library of fGarch is needed for the garchfit function to be used. The packages and coding of both cryptocurrencies are available in the appendix.

CHAPTER FOUR
DATA ANALYSIS

4.1 Introduction

This chapter will present the data analysis and findings of the data used in performing the study. All the data and the results are related to the research questions that act as our guide. All the findings and data are a result of using R studio.

4.2 Data Exploration

The data set for the Ethereum price and Bitcoin price each contains 1459 observations with 7 variables. The variables are date, opening price, closing price, low price, high price, adjusted closing price and volume.

4.3 Linear Regression Model Analysis

Before we could use the model, we ensure that it is statistically significant. Here are the summary statistics for the linear model.

Min	First quartile	Median	Third quartile	Max
-98.102	-2.155	0.737	2.797	95.058

Table 4.3.1 Residual

	Estimate	Std. Error	t-value	Pr(>|t|)
(Intercept)	-0.65370	0.47675	-1.371	0.171
Open	-0.45871	0.01737	-26.406	<2e-16***
High	0.87594	0.01678	52.202	<2e-16***
Low	0.57844	0.01261	45.862	<2e-16***

Table 4.3.2 Coefficient

LR model	AIC	BIC
Daily	11482.32	11508.74
Monthly	575.6607	585.1198

Table 4.3.3 AIC and BIC

P-values are critical because a linear model is considered statistically significant when both of these p-values are less than the specified statistical significance criterion of 0.05. However, the importance of stars at the end of the row against the x variable can be interpreted visually. From the above result, we can see that the variables are statistically significant.

So now, we have created a training data set with 1,167 observations and the test data set with 292 observations. Review diagnostics measures of the model on test data are shown below.

Min	First quartile	Median	Third quartile	Max
-100.070	-2.210	0.790	2.868	88.113

Table 4.3.4 New Residual

	Estimate	Std.Error	t-value	Pr(>|t|)
(Intercept)	-0.15567	0.54357	-0.286	0.775
Open	-0.46327	0.01940	-23.884	<2e-16 ***
High	0.91094	0.01916	47.545	<2e-16 ***
Low	0.54402	0.01375	39.569	<2e-16 ***

Table 4.3.5 New Coefficient

LR model with test data set	AIC	BIC
Daily	9203.407	9228.718
Monthly	453.0139	461.3317

Table 4.3.6 New AIC and BIC

The model's p value and the p values of the predictors are both smaller than the significance level, indicating that we have a statistically significant model. Additionally, the R-Squared and Adjusted R-Squared are comparable to the original model constructed using all available data.

A simple correlation between the actual and predicted values may be used to determine the accuracy of a prediction model. Having a higher degree of correlation accuracy indicates that the actuals and predicted values move in the same direction. Thus, the significant variables are Open price, High price and Low price. Then, the equation we can derive is from (5):

$$\begin{aligned} Closing\ Price \ = & -0.15567 \ - \ 0.46327 \cdot Open\ Price \ + 0.91094 \cdot High\ Price \\ & + \ 0.54402 \cdot \ Low\ Price \end{aligned} \tag{15}$$

Using the equation (15), we then predict the result for the first 10 days and the equation for monthly price for 10 months and compare it to the actual data.

	Actuals	Predicted
1	44.46190	43.86058
2	43.24210	42.72345
3	42.16260	42.75836
4	44.30710	44.36246
5	50.21940	49.96374
6	48.22440	48.27462

Table 4.3.7 The First 6 results of Actuals and Predicted Values for Daily Data

	Actuals	Predicted
1	294.9160	334.5365
2	383.0420	321.6495
3	283.0040	310.7334
4	141.5141	154.5162
5	218.6541	247.6605
6	152.5397	168.7609

Table 4.3.8 The First 6 Results of Actuals and Predicted Values for Monthly Data

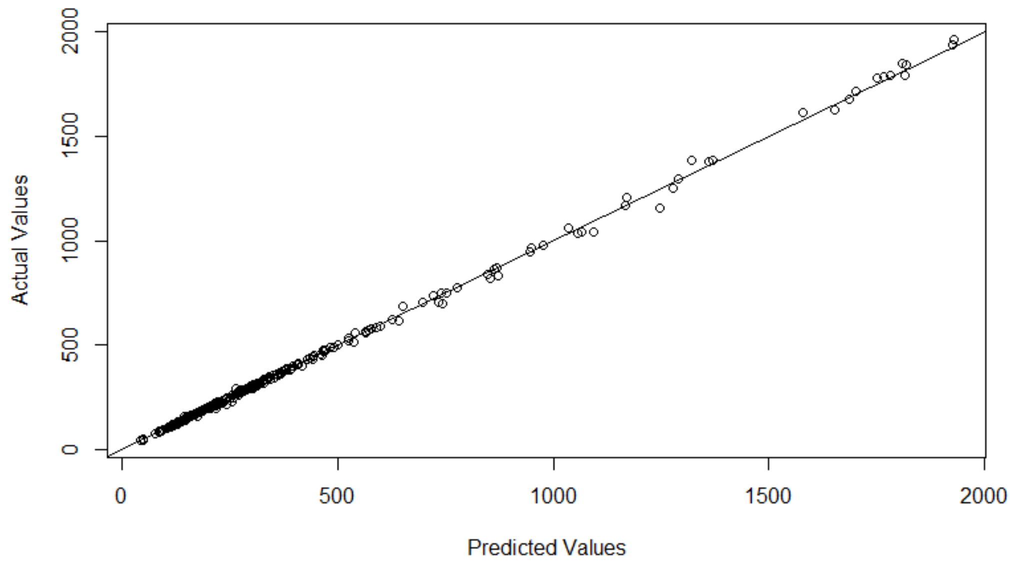


Figure 4.3.1 Scatter Plot of Actual vs Predicted Values of Daily Close Price

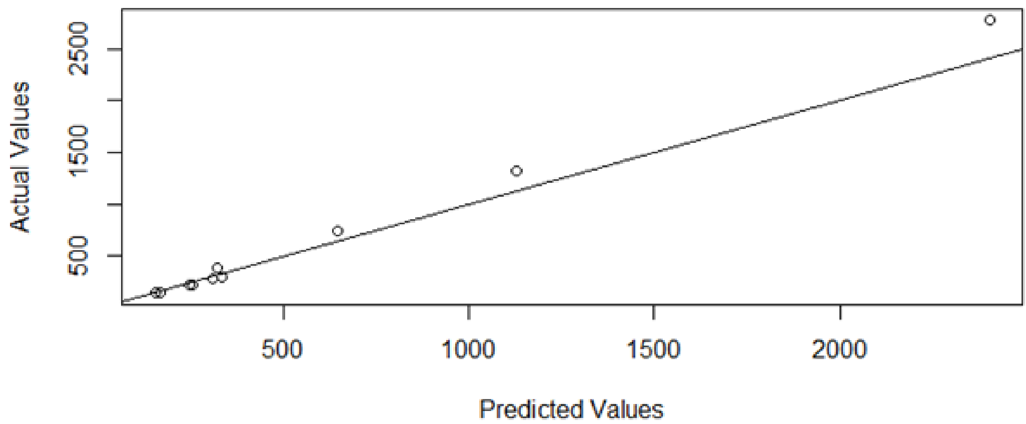


Figure 4.3.2 Scatter Plot of Actual vs Predicted Values of Monthly Close Price

The predicted values from the model are represented on the x-axis, while the actual values from the dataset are represented on the y-axis. The calculated regression line is shown by the diagonal line in the centre of the figure.

It is reasonable to conclude that the regression model performs an adequate job of fitting the data because each of the data points is quite near to the predicted regression line, as seen in the graph above and predicted almost near to the value of actual in the table above.

LR model	Min Max Accuracy	MAPE Accuracy
Daily	98.52%	1.52%
Monthly	88.24%	12.47%

Table 4.3.9 Accuracy for Linear Regression Model

From the result shown, we can determine that our model can predict 96.43% accurately for daily closing price and 88.24% for monthly closing price according to the min max accuracy.

4.3.1 Summary on LR models

The models can only predict when there are the predictor variables which are the opening, high, and low price of Ethereum price. And the data that we build is based on daily and monthly opening, high and low price data of Ethereum. So, we are only able to predict the values for the daily closing price of Ethereum.

4.4 Artificial Neural Network Model Analysis

4.4.1 Plotting graph Close Price vs Time

The figure 4.4.4.1 and 4.4.4.2 shows the closing price of Ethereum throughout the study period.

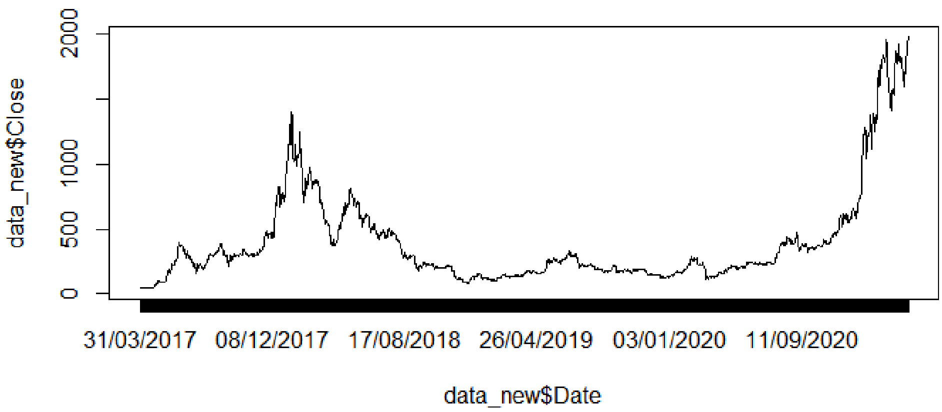


Figure 4.4.1.1 Daily Data Plot of Close Price Ethereum vs Time

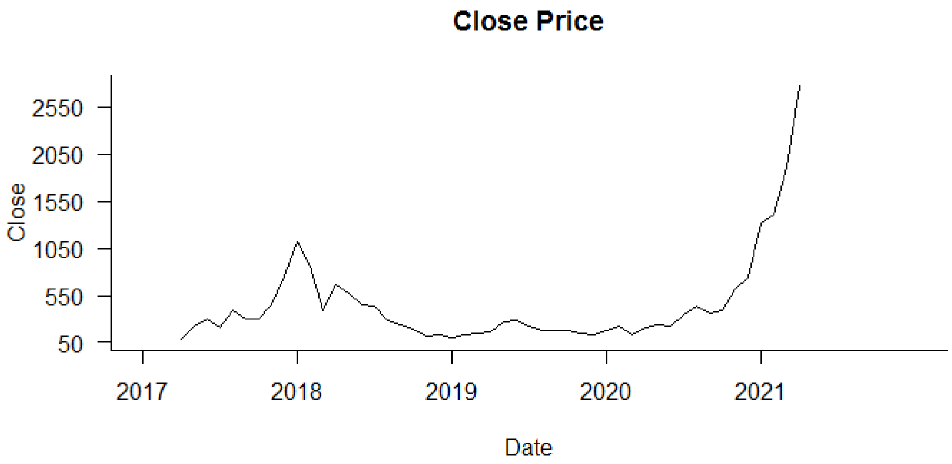


Figure 4.4.1.2 Monthly Data Plot of Close Price Ethereum vs Time

4.4.2 Actual and Predicted Close Price Graph Plot Before Proceed to Data partition

First and foremost, we conducted a simple ANN model forecasting using the whole dataset (daily and monthly) before data partitioning to see how well the predicted close price fit with the actual data.

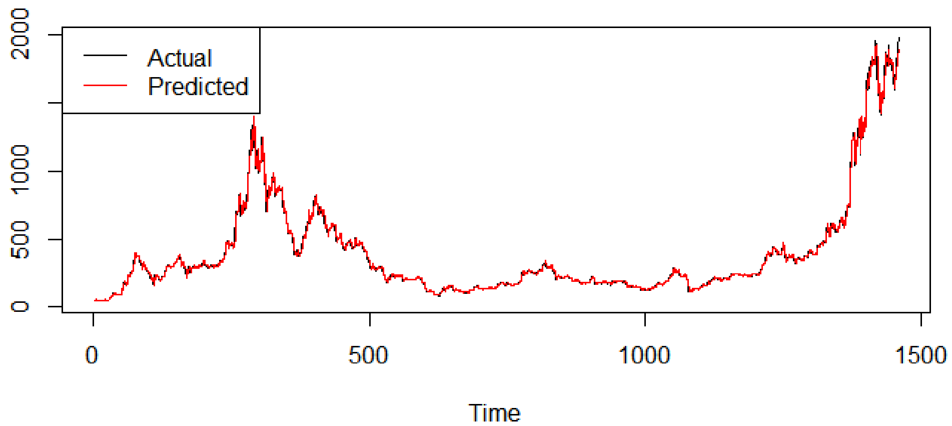


Figure 4.4.2.1 Graph Plot of Actual and Predicted Close Price on Daily Dataset

Figure 4.4.2.1 shows the plot of actual close price and predicted close price of Ethereum from daily data. From the graph, the forecasted predicted close price displays a precise prediction to actual close price. We can conclude that the neural network is able to predict the close price of Ethereum in the short-term.

Date	Actual Close Price	Predicted Close Price
23/3/2021	1678.650146	1704.42
24/3/2021	1593.413452	1692.143
25/3/2021	1595.359253	1606.051
26/3/2021	1702.842041	1608.072
27/3/2021	1716.494629	1715.414
28/3/2021	1691.355957	1728.273
29/3/2021	1819.684937	1704.441
30/3/2021	1846.033691	1818.191
31/3/2021	1918.362061	1838.94
1/4/2021	1977.276855	1891.141

Table 4.4.2.1 The Last 10 Results of Actual and Predicted Close Price from Daily Data

The table above shows the comparison between actual close price and the predicted close price of Ethereum using the fitted neural network model for daily data. The predicted price and the actual price do not have significant difference thus we could deduce that the neural network model has high accuracy in forecasting close prices in the short term.

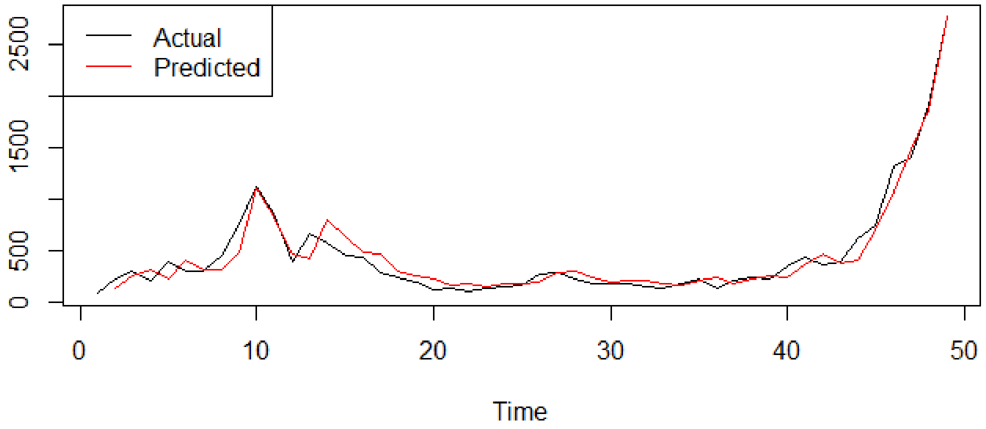


Figure 4.4.2.2 Graph Plot of Actual and Predicted Close Price on Monthly Dataset

Figure 4.4.2.2 shows the plot of actual close price and predicted close price of Ethereum from monthly daily data. From the graph, the forecasted predicted close price displays a lower accuracy at the beginning of study period. At time 0 to time 20, the predicted close price and actual close price have significant difference, then the predicted price portrays a better after time 20. We can conclude that the neural network model has difficulties in forecasting in the long-term.

Date	Actual Close Price	Predicted Close Price
1/5/2017	230.669006	135.4496
1/6/2017	294.915985	247.2185
1/7/2017	203.871002	307.9977
1/8/2017	383.041992	224.6551
1/9/2017	301.464996	403.8543
1/10/2017	305.878998	314.7235
1/11/2017	447.114014	319.3048
1/12/2017	756.732971	478.3008
1/1/2018	1118.310059	1090.7584
1/2/2018	855.198975	819.4992

Table 4.4.2.2 First 10 Results of Actual and Predicted Close Price from Monthly Data

The table above shows the comparison between actual close price and the predicted close price of Ethereum using a fitted neural network model for monthly data. The predicted price and the actual price have significant differences. At 1/12/2017, the model predicted the close price was 756.73 while the actual price was 478.3 which shows a substantial difference. Hence, we could deduce that the neural network model has less accuracy to forecast the close price of Ethereum in long-term forecasting.

4.4.3 Data Partition

Daily Dataset	Split Data	Stored into Time-series
Training Set	1167 observation	Time-series [1:1167] from 2017 to 2020
Testing Set	292 observations	Time-series [1:292] from 2020 to 2021

Monthly Dataset	Split Data	Stored into Time-series
Training Set	39 observations	Time-series [1:39] from 2017 to 2020
Testing Set	10 observations	Time-series [1:10] from 2020 to 2021

Table 4.4.3 Result of Data Partition

Table 4.4.3 shows the result of data partitioning both daily and monthly datasets. For the daily data, 1167 observations are put into the training set while the others are placed in the testing set. Using ts() function, the data is store into a time-series object and it also shows the date of the data. Meanwhile, the same method is used for monthly data in which 39 observations are partitioned into a training set and the others are in the testing set. These training and testing sets are used until the end of the project.

4.4.4 Result of Trained Model Performance for Both Datasets

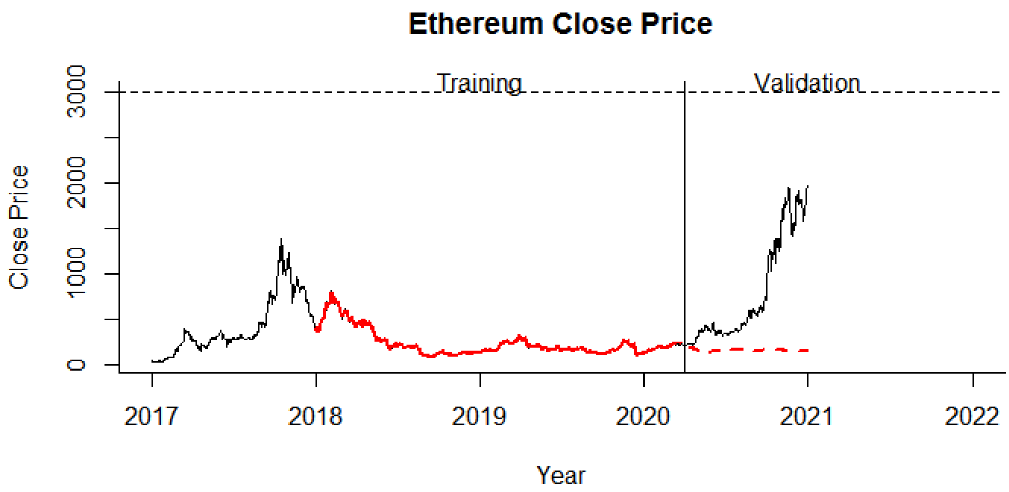


Figure 4.4.4.1 Training and Validation Test for Daily Dataset

Figure above the predicted close price of trained ANN model using training set. The red line indicates the forecasting of the trained model.from 2018 to 2019, the predicted close price of Ethereum fits well with the actual close price. After 2020, the predicted close price and the actual price have significant differences. The model fits very well with the training set but low accuracy when facing unseen data. We could infer that the model may be over-fitting.

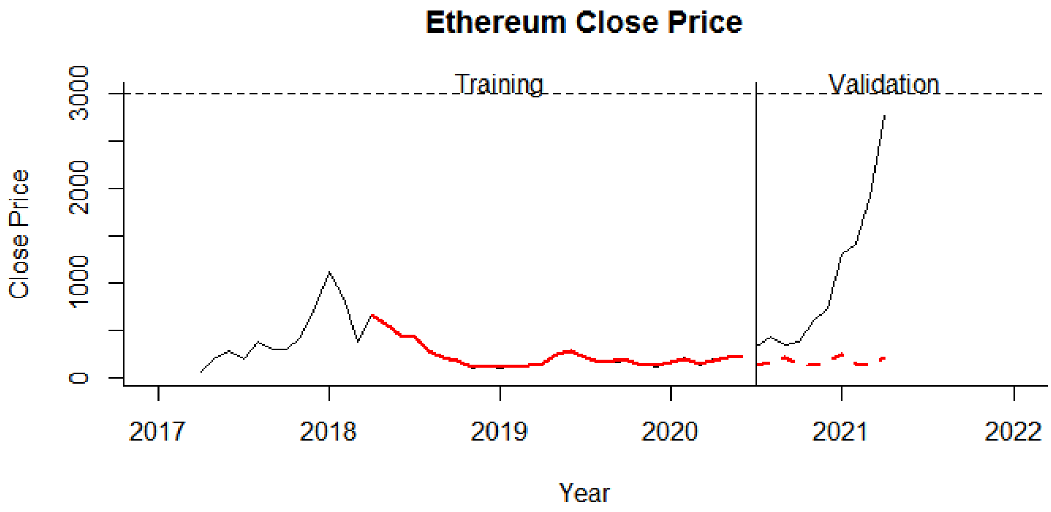


Figure 4.4.4.2 Training and Validation Test for Monthly Dataset

Figure above the predicted close price of trained ANN model using training set. The red line indicates the forecasting of the trained model.from 2018 to 2019, the predicted close price of Ethereum fits well with the actual close price. After 2020, the predicted close price and the actual price have a large gap of differences. The model fits well with the training set but low accuracy when facing unseen data. We could infer that the model may be over-fitting.

Check Residuals

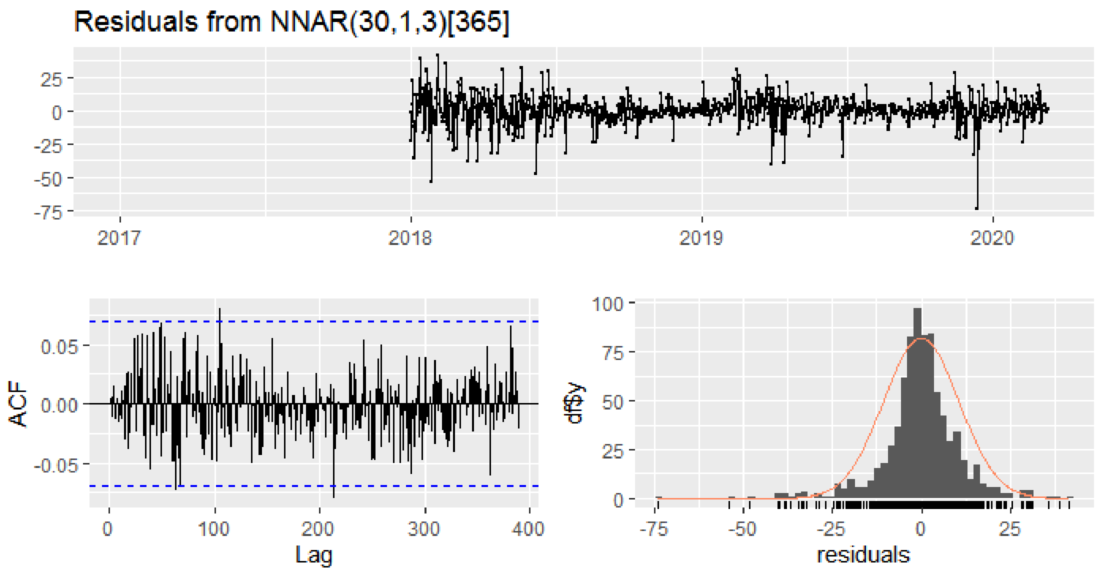


Figure 4.4.4.3 Daily Data Residuals of Ethereum Price

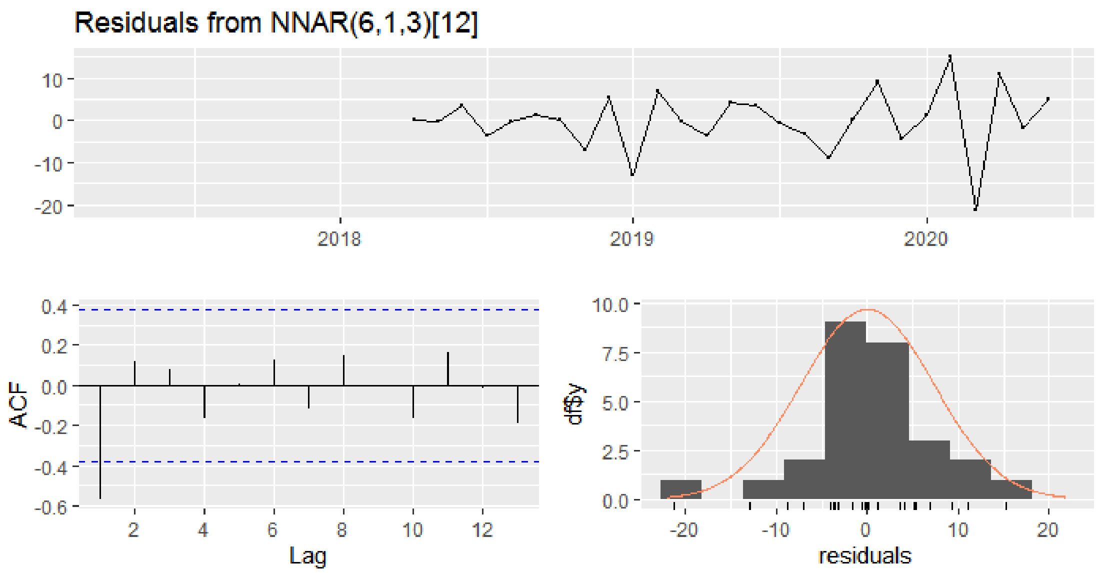


Figure 4.4.4.4 Monthly Data Residuals of Ethereum Price

Figure 4.4.4.3 shows the residual of Ethereum daily data. At times 2018, there is a spike of residuals because of the significant increase in Ethereum close price. There is also a large positive residual at the end of study period. The mean of the residuals is near to zero and there is no significant correlation in the residuals series. The histogram also shows the distribution of the data, and it appears to be a normal distribution. The ACF (Autocorrelation Factor) of the residual shows the model is statistically significant.

Figure 4.4.4.4 shows the residual of Ethereum monthly data. At times 2019 and 2020, there is a big spike residual because of the significant increase in Ethereum close price. There is also a large positive residual at the end of study period. The histogram also shows the distribution of the data, and it appears not normal since the right tail is a bit long. The ACF (Autocorrelation Factor) of the residual portrays the model is less significant. This graph shows the mean of the residual is far from zero which may have difficulties for the next forecasting step.

Coefficients of Determination,R^2

R	0.99663746531866
R^2	0.993286237276804

Table 4.4.4.1 R^2 Values for Daily Data of Ethereum Price

Based on the table, the R^2 value is 0.9933. 99.33% of the variation in Ethereum close price (Y) can be explained by all independent variables; Open price, High price, and Low price of Ethereum. But the other 0.67% could not explain the changes in Close price.

R	0.986497537636671
R^2	0.973177391763216

Table 4.4.4.2 R^2 Values for Monthly Data of Ethereum Price

Based on the table, the R^2 value is 0.9732. 97.32% of the variation in Ethereum close price (Y) can be explained by all independent variables; Open price, High price, and Low price of Ethereum. But the other 2.68% could not explain the changes in Close price. Based on both coefficients, all input variables have a relationship between output which is close price.

Accuracy Comparison of ANN model Between Daily and Monthly Dataset

ANN Model	RMSE			MSE		
	Train	Valid	GAP= Valid-Train	Train	Valid	GAP
Daily Dataset	10.60504	819.49895	808.89391	112.46687	671578.5291	671466
Monthly Dataset	7.17588	1143.3118	1136.13593	51.4932538	1307161.895	1307110

ANN Model	MAE			MAPE		
	Train	Valid	GAP	Train	Valid	GAP
Daily Dataset	7.103587	587.926197	580.82261	3.18052	62.69509	59.51457
Monthly Dataset	4.972517	850.859402	845.886885	3.069529	72.443359	69.37383

ANN Model	MASE		
	Train	Valid	GAP
Daily Dataset	0.02925553	2.42132524	2.39206971
Monthly Dataset	0.02139447	3.66086025	3.63946578

Table 4.4.4.3 Error Measures from Training and Validation of Both Dataset

Table 4.4.4.3 shows the error values of ANN model from daily and monthly dataset. From the table, ANN model performs very well in the training data. There is a small difference between daily dataset and monthly dataset values for each error value. However, ANN model did not perform well on the validation data. There are huge differences and gap between train set and data set for both daily and monthly dataset.

From the table above, ANN model performs better in Ethereum daily dataset compared to monthly dataset. The RMSE gap for daily data is 808.894 which is lower than the RMSE gap for monthly data, 1136.136. The MSE gap also shows that the daily dataset is better compared to the monthly dataset.

To conclude, ANN model using daily dataset has higher accuracy as compared to ANN model using monthly dataset. This can be proven by referring to the error values gap of RMSE, MSE, MAE, MAPE and MASE. Although ANN model did not perform well for both dataset, ANN model performed better on daily dataset compared to monthly dataset.Thus, Neural Network does have the ability to forecast only for short-term. However, ANN model does not perform well in long-term period forecasting.

4.4.5 Short-term and Long-term Forecasting of Ethereum Close Price

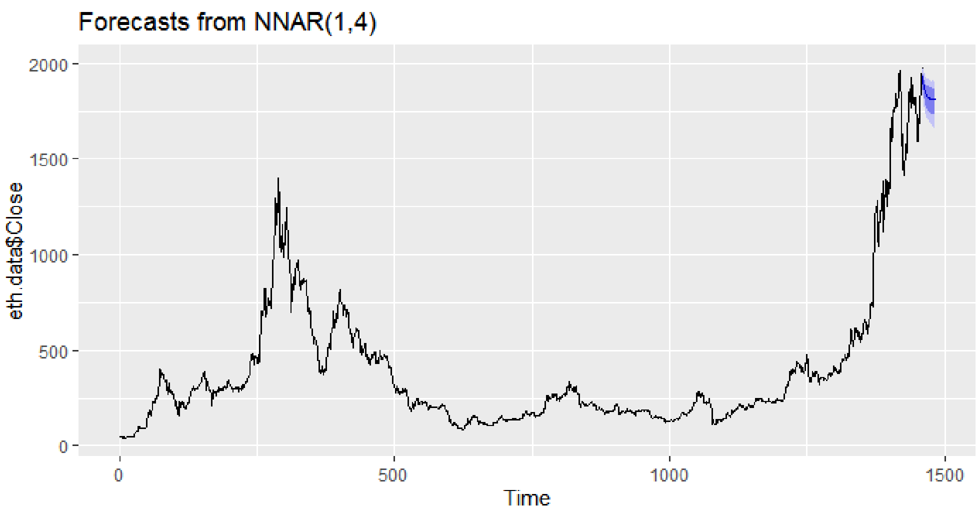


Figure 4.4.5.1 Graph Plot of Forecasted Close Price Using Daily Data

Date	Future Close Price of Ethereum
2/4/2021	1928.648
3/4/2021	1898.006
4/4/2021	1877.149
5/4/2021	1862.248
6/4/2021	1851.248
7/4/2021	1842.936
8/4/2021	1836.549
9/4/2021	1831.579
10/4/2021	1827.672
11/4/2021	1824.579

Table 4.4.5.1 Table of Future Close Price Using Daily Data

Figure 4.4.5.1 shows the graph of close price of Ethereum, and the blue line represents the forecasted close price for the next few days. It shows that the close price of Ethereum plummeted for the next few days. We can refer to the table 4.4.5.1 which shows the future close price of Ethereum from 2 April to 11 April 2021. On 2 April, the close price was 1928.65 and it began to fall the next day until the price was 1824.58 on 11 April.

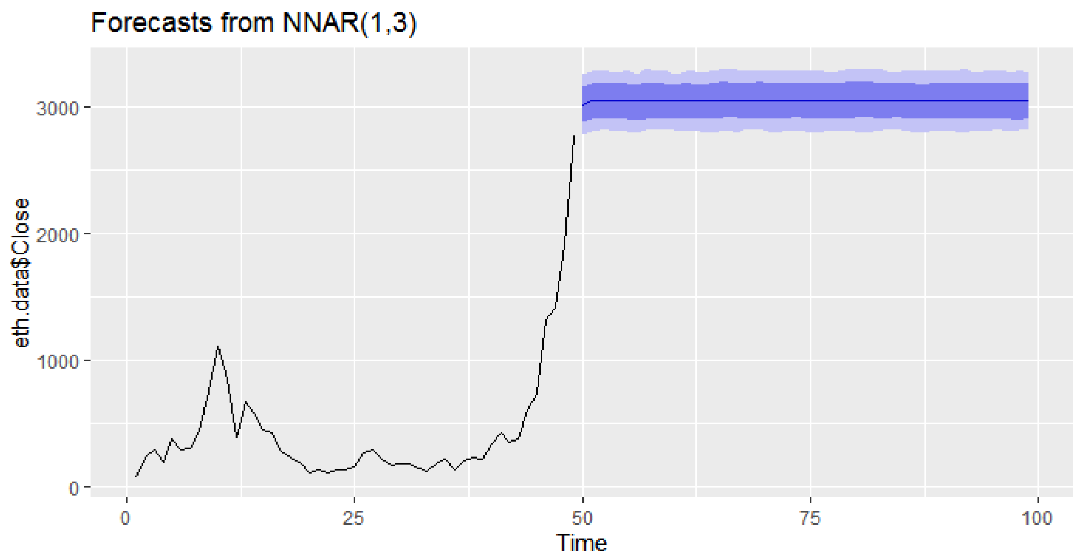


Figure 4.4.5.2 Graph Plot of Forecasted Close Price Using Monthly Data

Date	Future Close Price of Ethereum
1/5/2021	3012.941
1/6/2021	3041.012
1/7/2021	3043.883
1/8/2021	3044.173
1/9/2021	3044.202
1/10/2021	3044.205
1/11/2021	3044.205
1/12/2021	3044.205
1/1/2022	3044.205
1/2/2022	3044.205

Table 4.4.5.2 Table of Future Close Price Using Monthly Data

Figure 4.4.5.2 shows the graph of close price of Ethereum, and the blue line represents the forecasted close price for the next months. It shows that the future close price increase will remain the same for the next month. We can refer to Table 4.4.5.2 which shows the future close price of Ethereum on 1 May is 3012.941. The next month, the close price increased to 3041 on 1 June and 3043.88 on 1 July. From 1 August to 1 September, the close price has a slight increase. Then, the close price is 3044.305 and is constant from 1 October until next year.

ANN Model	RMSE	MSE	MAE	MAPE	MASE
Daily Dataset	104.415	10902.49223	78.45935	25.14157	0.5952941
Monthly Dataset	119.5968	14303.39457	86.49864	26.47179	0.6562906

Table 4.4.5.3 Error Values for Daily and Monthly Dataset

From the table above, error values on the daily dataset are lower than the error values on the monthly dataset. RMSE value for daily dataset is 104.415 lower than monthly dataset, 119.5968. MSE value for daily dataset is 10902.49223 while for monthly dataset is 14303.39457. The MAE, MAPE and MASE values on daily data are also less than on monthly data. This indicates that the accuracy of ANN model on daily data is higher compared to accuracy of ANN model on monthly data. We concluded that ANN model has better forecasting in the short-term period as compared to long-term period. ANN model has the ability to forecast close prices in a short-term period.

4.4.6 Summary on ANN models

It can be concluded that ANN model can be used to forecast the close price of Ethereum. R^2 values for daily dataset and monthly dataset are acceptable. Both ANN models can predict the close price of Ethereum better in short-term compared to long-term forecasting supported by the error measures (MAE, MAPE, MSE, RMSE and MASE) of daily data are lower compared to error measures of monthly data. It shows that the accuracy of ANN model for short-term forecasting is higher than ANN model for long-term.

4.5 Comparison Between Models

	Linear Regression Model		Artificial Neural Network Model	
	Daily	Monthly	Daily	Monthly
R^2	99.89%	96.09%	99.32%	97.32%
MAPE Accuracy	1.52%	12.47%	3.77%	25.39%

Table 4.5.1 Models Comparison

R^2 represents the ability of the model to predict the closing price using the given variables which are open price, low price and high price. The lower the MAPE accuracy, the higher the accuracy of the model.

For short-term forecasting, the Linear Regression model displays a higher value of R^2 compared to Artificial Neural Network. This implies that the Linear Regression has a better ability in using the variables to determine the closing price of Ethereum. Linear Regression also displays extreme accuracy through MAPE with a whopping 1.52% meanwhile Artificial Neural Network does not reach such accuracies. Therefore, in the short-term forecasting, Linear Regression model excels in forecasting than Artificial Neural Network.

For long-term forecasting, both models show a good value of R^2. However, Artificial Neural Network MAPE accuracy is 25.39% meanwhile Linear Regression has a MAPE accuracy of 12.47%. Although both models' performance deteriorates in the long-term forecasting compared to the short-term forecasting, Linear Regression displays better performance than Artificial Neural Network.

Linear Regression proves to be a more reliable model in the short-term and long-term forecasting. Artificial Neural Network seems to only be reliable in forecasting short-term, but unreliable in long-term forecasting. To conclude, Linear Regression performs better than Artificial Neural Network in the short-term and long-term forecasting of Ethereum.

4.6 Volatility of Ethereum, Bitcoin, Litecoin, Ripple and Dogecoin

The figure 4.6.1, 4.6.2, 4.6.3 and 4.6.4 shows the closing price of Ethereum, Bitcoin, Litecoin and Dash throughout the study period.

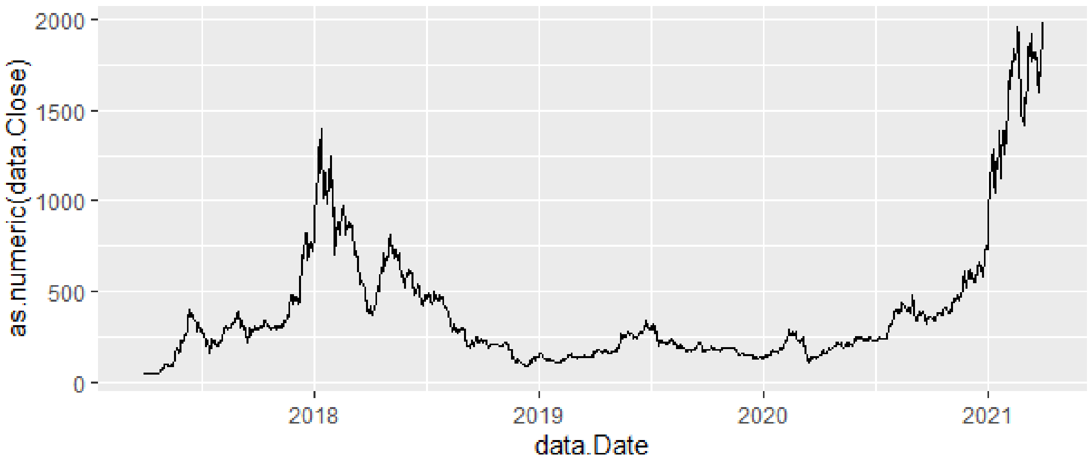


Figure 4.6.1 Graph of Close Price of Ethereum

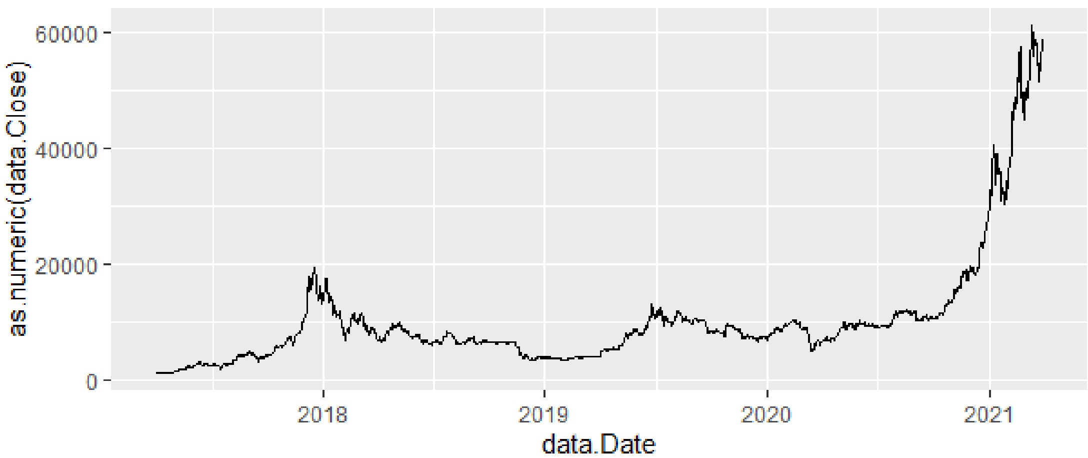


Figure 4.6.2 Graph of Close Price of Bitcoin

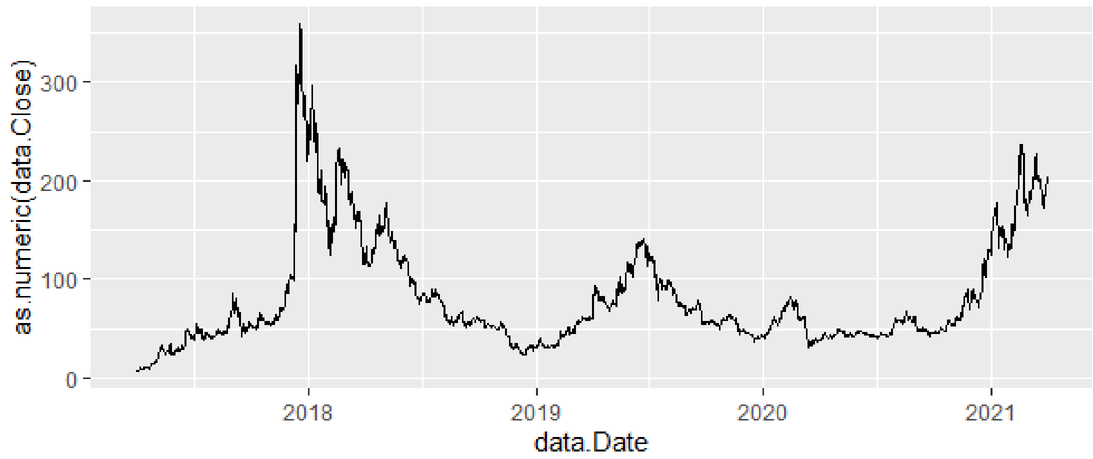


Figure 4.6.3 Graph of Close Price of Litecoin

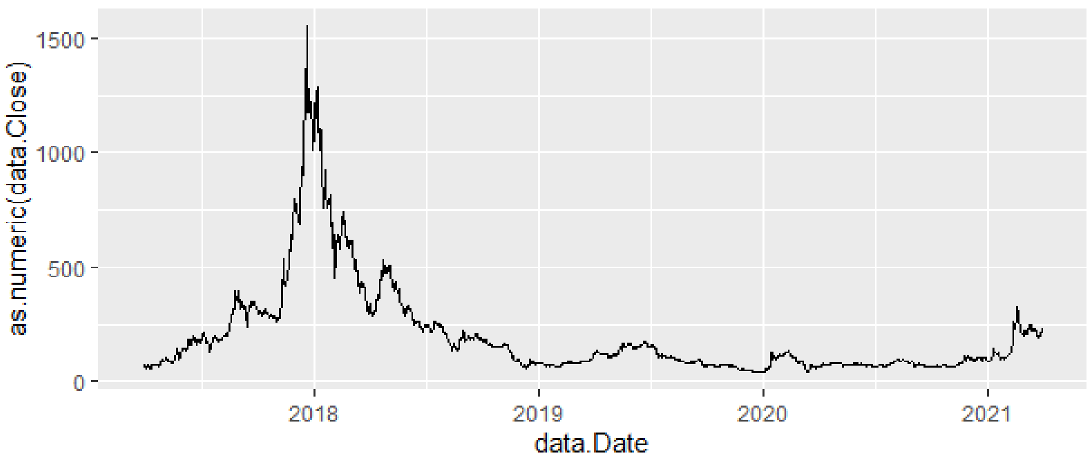


Figure 4.6.4 Graph of Close Price of Dash

The figure 4.6.5, 4.6.6, 4.6.7 and 4.6.8 shows the closing price volatility of Ethereum, Bitcoin, Litecoin and Dash.

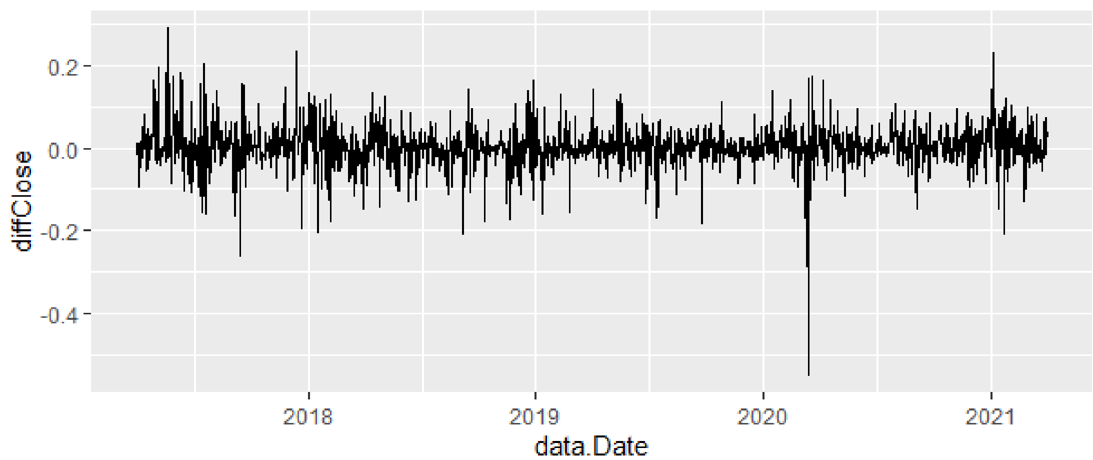


Figure 4.6.5 Graph of Closing Price Volatility of Ethereum

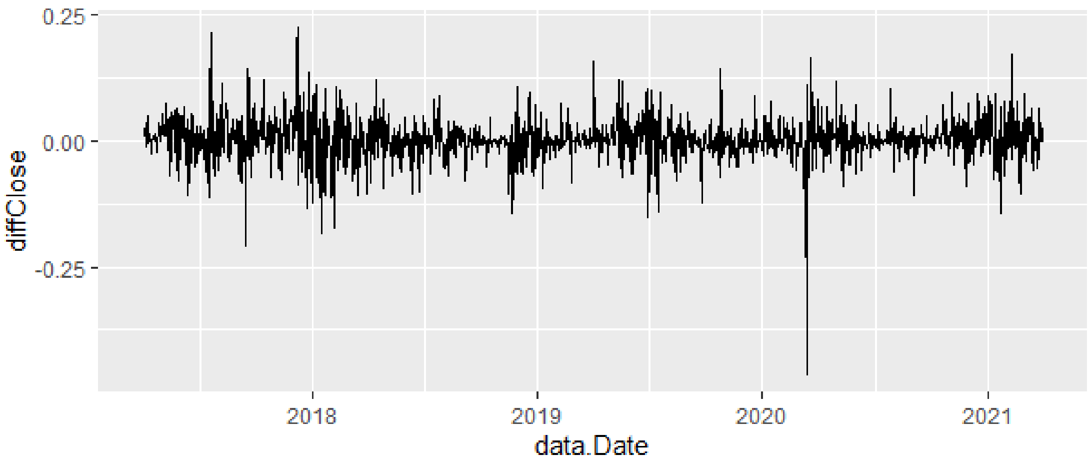


Figure 4.6.6 Graph of Volatility of Bitcoin

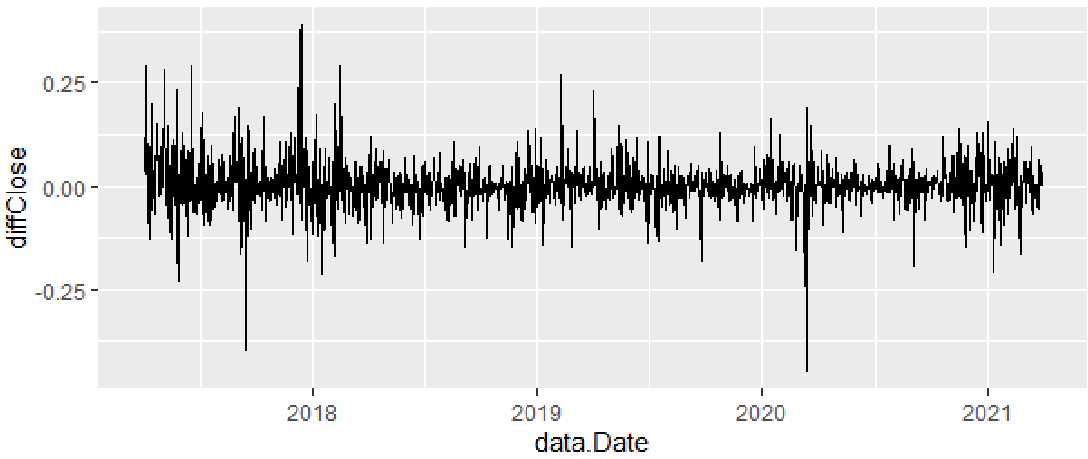


Figure 4.6.7 Graph of Volatility of Litecoin

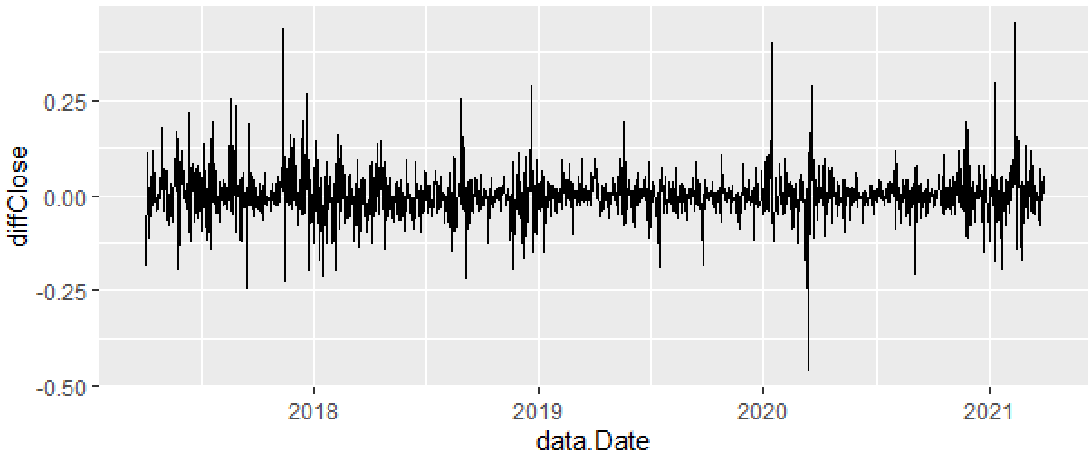


Figure 4.6.8 Graph of Volatility of Dash

The table below shows the Parameters GARCH (1,1).

Parameters	Estimate	Std. Error	t value	Pr(>|t|)
mu	0.0004193	0.0002776	1.511	0.13090
ω	0.00001015	0.000003204	3.168	0.00153
α	0.1057	0.02276	4.644	0.00000342
β	0.8266	0.03947	20.94	< 2e-16

Table 4.6.1 Parameters GARCH (1,1) for Ethereum

Parameters	Estimate	Std Error	t value	Pr(>|t|)
mu	0.0005207	0.0001632	3.190	0.00142
ω	0.000003410	0.0000006649	5.128	0.000000293
α	0.1254	0.02097	5.982	0.00000000221
β	0.8200	0.02489	32.942	< 2e-16

Table 4.6.2 Parameters GARCH (1,1) for Bitcoin

Parameters	Estimate	Std Error	t value	Pr(>|t|)
mu	0.0002720	0.0003416	0.796	0.425867
ω	0.000007781	0.000002207	3.525	0.000423
α	0.07471	0.01460	5.117	0.000000311
β	0.8910	0.02089	42.660	< 2e-16

Table 4.6.3 Parameters GARCH (1,1) for Litecoin

Parameters	Estimate	Std Error	t value	Pr(>|t|)
mu	-0.0004635	0.0002916	-1.589	0.112
ω	0.00001503	0.000002956	5.086	0.000000367
α	0.2240	0.03297	6.794	0.0000000000109
β	0.7346	0.03031	24.233	< 2e-16

Table 4.6.4 Parameters GARCH (1,1) for Dash

Parameters	Ethereum	Bitcoin	Litecoin	Dash
mu	0.0004193	0.0005207	0.0002720	-0.0004635
ω	0.00001015	0.000003410	0.000007781	0.00001503
α	0.1057	0.1254	0.07471	0.2240
β	0.8266	0.8200	0.8910	0.7346

Table 4.6.5 Summary GARCH (1,1)

From Figure 4.6.1, Ethereum price increased and quickly declined in the second and the third quarter of 2017. The price of Ethereum climbed in late 2017 and continued until early 2018. During the second quarter of 2018, a dip occurs in the Ethereum market, and its price continues to fall. Ethereum market recovers from the dip in the second quarter of 2018 and spikes for a moment before falling again. The decrease in price continues from the third quarter until the fourth quarter of 2018. From the fourth quarter of 2018 until early first quarter of 2020, the price of Ethereum remained stable with no major spike nor dip. From the late first quarter of 2020, Ethereum price begins to increase slowly until the end of 2020. The beginning of first quarter 2021, the Ethereum price surge and a massive spike occurred in the market.

Based on Figure 4.6.2, the price of Bitcoin is stable for the second and third quarter of 2017 and its price spikes in the fourth quarter of 2017. In early 2018, Bitcoin price temporarily crashed and quickly recovered. However, in the first quarter of 2018, a dip occurred in the Bitcoin market and its price sharply declined. In the second quarter of 2018, the Bitcoin market price recovered and remained stable until the fourth quarter of 2018. In the last quarter of 2018, the bitcoin price decreased and remained constant from the last quarter of 2018 to the second quarter of 2019. From the second quarter of 2019, Bitcoin price gradually increased. In the third quarter of 2019 until the first quarter of 2020, there is neither a major spike nor dip in the Bitcoin market. In the second quarter of 2020, Bitcoin price sharply declined and managed to recover back after a short while and continue to maintain its value until the third quarter of 2020. From the fourth quarter of 2020 until the first quarter of 2021, the Bitcoin price spiked and continued to escalate.

Based on Figure 4.6.3, the price of Litecoin is increasing from second quarter 2017 to fourth quarter 2017. At the end of 2017, the price of Litecoin escalated and dripped in the first quarter of 2018. The price of Litecoin jumped up in the first quarter and down in the second quarter. From the second quarter 2018, the price decreased over time until the beginning of 2019. Beginning 2019, the price of Litecoin increased until the second quarter 2019. From the third quarter to the end of 2019, the price decreased. At the beginning 2020, the price jumped a little bit before falling again and stabilized until the fourth quarter 2020. At the fourth quarter 2020 until the second quarter 2021, the price of Litecoin jumped up with a few drops between them.

Based on Figure 4.6.4, the price of Dash increased from second quarter 2017 to fourth quarter 2017. In the fourth quarter 2017 ,the price of Dash jumped up and dripped in the first quarter of 2018. In the first quarter 2108, the price jumped a bit before slowly decreasing. In the second quarter also the price jumped a little bit before dropping and slowly decreased until 2019. From the first quarter of 2019 to the end of 2020, the price of Dash stabilized. In the first quarter of 2021, the price of Dash increased before stabiled.

Based on Figure 4.6.5, Ethereum volatility in the second quarter and the third quarter of 2017 was high. In the fourth quarter of 2017, the volatility of Ethereum remained stable. In the beginning of 2018, Ethereum volatility began to be unstable for an abbreviated period of time and became stable again. The second quarter and third quarter of 2018, its volatilities are stable with no major swings. The fourth quarter of 2018, volatility of Ethereum becomes high and calms down in the first quarter of 2019. From the second quarter of 2019 to the first quarter of 2020, Ethereum volatility is stable and no major increase in volatility is observed. However, in the second quarter of 2020, the volatility of Ethereum increased and quickly regained stability for the rest of the second quarter. From the third quarter to the end of 2020, the volatility of Ethereum is low and stable. In the beginning of 2021, the volatility of Ethereum spikes and remains unstable for the rest of the study period.

Based on Figure 4.6.6, the volatility of Bitcoin in the second quarter of 2017 is low. In the third quarter, its volatility was relatively stable. During the fourth quarter, Bitcoin volatility was quite unstable as compared to previous quarter and continues until the first quarter of 2018. From the second quarter until the third quarter of 2018, the volatility remained stable throughout the period. In the fourth quarter of 2018, the volatility picks up until the beginning of 2019. From the first quarter of 2019 to the first quarter of 2020, the volatility of Bitcoin was moderate with a few short jumps in between them. In the second quarter of 2020, a huge decline occurs, and its volatility remains unstable throughout the period. From the third quarter of 2020 until the fourth quarter of 2021, the volatility of Bitcoin remains low and stable. In the first quarter of 2021, the volatility of Bitcoin increased and became unstable.

Based on Figure 4.6.7, the volatility of Litecoin is moderate from second quarter 2017 to fourth quarter 2017. In the fourth quarter 2017 , the volatility was high before decreasing at the end of 2017. The volatility became high again in the beginning of 2018 before slowly stabilizing. The volatility of Litecoin is low from second quarter 2018 to first quarter 2020. In the first quarter of 2021, the volatility jumped before stabilizing for the rest of 2020. The volatility of 2021 is high from the first quarter to second quarter.

Based on Figure 4.6.8, the volatility of Dash is moderate from second quarter 2017 to fourth quarter 2017. The volatility increased at the end of fourth quarter 2017 to second quarter 2018. The volatility of Dash is low from the second quarter until the end of 2019 with few short increases in volatility. In the first quarter of 2020, the volatility of Dash increased and was high before decreasing in the second quarter. From second quarter 2020 to third quarter 2020, the volatility is low. In the fourth quarter 2020, the volatility increased and was high again until the second quarter 2021.

From Table 4.6.1, Table 4.6.2, Table 4.6.3 and 4.6.4, short run persistence is evident as the estimated coefficient on the α term is statistically significant, and in each case the short run persistence is less than the long run persistence (β). The estimated coefficient on β is statistically significant for Ethereum, Bitcoin, Litecoin and Dash indicating the importance of long run persistence.

From Table 4.6.5 ,Ethereum's α, 0.1057 is smaller than Bitcoin's α, 0.1254 and Dash's α, 0.1254 while bigger than Litecoin's α,0.07471. This shows that volatility in the short run of Ethereum is less persistent than volatility in the short run of Bitcoin and Dash while more persistent than Litecoin. Its means that the price of Ethereum is less likely to make small jumps as compared to Bitcoin and Dash .

For the β, Ethereum's β, 0. 8266 is larger than Bitcoin, 0.8200 and Dash,0.7346 while smaller than Litecoin,0.8910. From the β, it shows that persistence of volatility in the long run for Ethereum is higher than Bitcoin and Dash meanwhile less persistence than Litecoin. It means that the price of Ethereum is likely to follow the changes of previous day's price with a smaller increase or decrease compared to Bitcoin and Dash. However, it also means that when the changes of the previous day's price are high, the changes in today's price are also predicted to be high.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

For the first objective, the variables that we have are open, high, low, close, adjusted close and volume. As we are forecasting the closing price of Ethereum, the variable close is chosen to be the responding variable. By using Linear Regression, we have reduced the number of significant variables into 3 variables which are open, high and low price. All of these variables have shown P-values that are lower than the specified statistical criterion which is below 0.05. Thus, we used these variables throughout the whole study so that the models are consistent with each other.

Next objective is to find the best model between Linear Regression and Artificial Neural Network that can forecast the Ethereum closing price. We build both models in R studio as it is a platform that is widely used to perform machine learning. The result proves that both Linear Regression and Artificial Neural Network models can be used to forecast the Ethereum closing price. Both models show excellent results in short-term forecasting of Ethereum. As both models have an accuracy of over 95%, Linear Regression has a slightly more accuracy than Artificial Neural Network in forecasting short-term. However, for long-term forecasting of Ethereum, Linear Regression outperforms Artificial Neural Network by a large margin. Linear Regression has almost 90% accuracy while Artificial Neural Network only has about 70% accuracy. Thus, we conclude that Linear Regression is the best model to forecast Ethereum closing price.

For the third objective, using the equation we got from Linear Regression, we then proceed to forecast Ethereum for the first 6 days and 6 months. The predicted result was then compared to the actual data. For short-term forecasting, the predicted result shows that it can forecast the price close to the actual data with minimal margin of error. This is consistent with the accuracy of the short-term forecasting using Linear Regression model. For long-term forecasting, although the result is close to the actual data, the margin of error is slightly higher than the short-term forecasting. This corresponds to the accuracy result for the long-term forecasting using Linear Regression. Thus, it is proven that the Linear Regression model does have the ability to forecast Ethereum in the short-term and in the long-term.

Finally, for the last objective, the volatility of Ethereum market price was analysed and compared with another cryptocurrency which is Bitcoin, Litecoin and Dash. Based on the GARCH (1,1) model, when comparing the volatility of Ethereum with Bitcoin, Litecoin and Dash, Ethereum's volatility in the short run is less persisted as compared to Bitcoin and Dash while it is high compared to Litecoin. Ethereum's price in the short run is more volatile and has wide price fluctuations when a larger volume of purchases occurs than Bitcoin and Dash. As for the persistence in the long run, GARCH (1,1) model shows the volatility of Ethereum is more persistent as compared to Bitcoin and Dash and less persistent to Litecoin. Ethereum is more likely to sustain the price increase when the spike occurs as compared to Bitcoin and Dash.

5.2 Recommendation

The research paper aims to predict the price of Ethereum in the short run and long run. During the research for the paper, some ideas or recommendations had arisen and might come handy for future research on Ethereum or other cryptocurrencies. With the recommendation provided, the future researchers can obtain better results and less noise in the study.

The first recommendation is to extend the period of study. The data used in this study only from 31 March 2017 to 1 April 2021. Since cryptocurrency was introduced in 2008 and still in 2021 as this paper is written, the larger amount of data can help the future researcher draw more conclusive conclusions based on data collected.

Next, we would like to recommend involving the social economy into the research. Although our research proves to be accurate, it does not have the ability to predict the crashes in the market such as the Bitcoin crash in December 2017. Thus, a social economy which considers the public opinions and expectations towards the cryptocurrencies can further improve the accuracy of forecasting cryptocurrency.

REFERENCES

Ngunyi, A., Mundia, S., & Omari, C. (2019). Modelling Volatility Dynamics of Cryptocurrencies Using GARCH Models. *Journal of Mathematical Finance*, *09*(04), 591–615. https://doi.org/10.4236/jmf.2019.94030

Shazia Salamat, Lixia, N., Sobia Naseem, Muhammad Mohsin, Muhammad Zia-ur-Rehman, & Sajjad Ahmad Baig. (2019). Modeling cryptocurrencies volatility using GARCH models: a comparison based on Normal and Student's T-Error distribution. *Journal of Entrepreneurship and Sustainability Issues*, *7*(3), 1580–1597. https://www.researchgate.net/publication/340281131_Modeling_cryptocurrencies_volatility_using_GARCH_models_a_comparison_based_on_Normal_and_Student%27s_T-Error_distribution

Andrade, J. G., & Brandalise, N. (2019). Volatility behavior of virtual currencies. *Independent Journal of Management & Production*, *10*(2), 537. https://doi.org/10.14807/ijmp.v10i2.852

Leirvik, T. (2021). Cryptocurrency returns and the volatility of liquidity. *Finance Research Letters*, *February*, 102031. https://doi.org/10.1016/j.frl.2021.102031

Malhotra, N., & Gupta, S. (2019). Volatility Spillovers and Correlation Between Cryptocurrencies and Asian Equity Market. *International Journal of Economics and Financial Issues*, *9*(6), 208–215. https://doi.org/10.32479/ijefi.8624

Naimy, V., Haddad, O., Fernández-Avilés, G., & El Khoury, R. (2021). The predictive capacity of GARCH-type models in measuring the volatility of crypto and world currencies. *PLoS ONE*, *16*(1 January), 1–18. https://doi.org/10.1371/journal.pone.0245904

Bhosale, J., & Mavale, S. (2018). Volatility of select Crypto-currencies: A comparison of Bitcoin, Ethereum and Litecoin. *Annual Research Journal of SCMS, Pune*, *6*(March), 132–141. https://www.scmspune.ac.in/journal/pdf/current/Paper 10 - Jaysing Bhosale.pdf

Ghorbel, A., & Jeribi, A. (2021). Investigating the relationship between volatilities of cryptocurrencies and other financial assets. *Decisions in Economics and Finance*. https://doi.org/10.1007/s10203-020-00312-9

Alessandretti, L., Elbahrawy, A., Aiello, L. M., & Baronchelli, A. (2018). Anticipating Cryptocurrency Prices Using Machine Learning. *Complexity*, *2018*, 1–24. https://doi.org/10.1155/2018/8983590

Fang, T., Su, Z., & Yin, L. (2020). Economic fundamentals or investor perceptions? The role of uncertainty in predicting long-term cryptocurrency volatility. *International Review of Financial Analysis*, *71*(39), 101566. https://doi.org/10.1016/j.irfa.2020.101566

Conrad, C., Custovic, A., & Ghysels, E. (2018). Long- and Short-Term Cryptocurrency Volatility Components: A GARCH-MIDAS Analysis. *Journal of Risk and Financial Management*, *11*(2), 23. https://doi.org/10.3390/jrfm11020023

Brauneis, A., Mestel, R., & Theissen, E. (2021). What drives the liquidity of cryptocurrencies? A long-term analysis. *Finance Research Letters*, *39*(March 2020), 101537. https://doi.org/10.1016/j.frl.2020.101537

Kenny, M. (2020). *Investigating the predictability of a Chaotic time-series data using Reservoir computing , Deep-Learning and Machine- Learning on the short- , medium- and long-term pricing of Bitcoin and Ethereum* . https://doi.org/10.21427/42ks-q868

Tzouvanas, P., Kizys, R., & Tsend-Ayush, B. (2020). Momentum trading in cryptocurrencies: Short-term returns and diversification benefits. *Economics Letters*, *191*, 108728. https://doi.org/10.1016/j.econlet.2019.108728

Meshcheryakov, A., & Ivanov, S. (2020). Ethereum as a hedge: The intraday analysis. *Economics Bulletin*, *40*(1), 101–108.

Zhang, Y., Chu, J., Chan, S., & Chan, B. (2019). The generalised hyperbolic distribution and its subclass in the analysis of a new era of cryptocurrencies: Ethereum and its financial risk. *Physica A: Statistical Mechanics and Its Applications*, *526*, 120900. https://doi.org/10.1016/j.physa.2019.04.136

Kyriazis, N. A., Daskalou, K., Arampatzis, M., Prassa, P., & Papaioannou, E. (2019). Estimating the volatility of cryptocurrencies during bearish markets by employing GARCH models. *Heliyon*, *5*(8). https://doi.org/10.1016/j.heliyon.2019.e02239

Agha Salman Khan, P. A. (2019). *Predictive Analytics in Cryptocurrency Using Neural networks: A Comparative Study*. Retrieved from https://www.researchgate.net/profile/Peter_Augustin3/publication/336640118_Predictive_Analytics_in_Cryptocurrency_Using_Neural_networks_A_Comparative_Study/links/5da95aac4585155e27f64c69/Predictive-Analytics-in-Cryptocurrency-Using-Neural-networks-A-Compa

Franco Valencia, A. G.-E.-A. (2019). *Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning*. Retrieved from https://www.mdpi.com/1099-4300/21/6/589

K. Ramya Laxmi, M. A. (2020). *Cryptocurrency Price Prediction Using Machine Learning*. Retrieved from https://smsjournals.com/index.php/SAMRIDDHI/article/view/2205/1062

Luisanna Cocco, R. T. (2021). *Predictions of bitcoin prices through machine learning based frameworks*. Retrieved from https://peerj.com/articles/cs-413/

Mohil Maheshkumar Patel, S. T. (2020). *A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions*. Retrieved from https://www.sciencedirect.com/science/article/pii/S2214212620307535

Nor Azizah Hitam, A. R. (2018). *Comparative Performance of Machine Learning Algorithms for Cryptocurrency Forecasting*. Retrieved from https://www.researchgate.net/publication/327415267

Reaz Chowdhury, M. A. (2020). *Predicting and Forecasting the Price of Constituents and Index of Cryptocurrency Using Machine Learning*. Retrieved from https://arxiv.org/abs/1905.08444

Steve Hyun, J. L.-M. (2019). *What Coins Lead in the Cryptocurrency Market: Using Copula and Neural Networks Models*. Retrieved from https://www.mdpi.com/1911-8074/12/3/132

Ali Alahmari, S. (2020). Predicting the Price of Cryptocurrency Using Support Vector Regression Methods. *Journal of Mechanics of Continua and Mathematical Sciences*, *15*(4). https://doi.org/10.26782/jmcms.2020.04.00023

Chen, M., Narwal, N., & Schultz, M. (2017). Predicting Price Changes in Ethereum. *Cs229.Stanford.Edu*, *2016*, 1–6. http://cs229.stanford.edu/proj2017/final-reports/5244039.pdf

Cohen, G. (2020). *Learning Systems. i*. https://doi.org/10.1038/219309c0

Kumar, S. A. (2018). *A regression model for crypto-currency price*. *May*.

Nashirah Abu Bakar, & Sofian Rosbi. (2018). Statistical Diagnostics for Bivariate Correlation and Regression Analysis Between Cryptocurrency Exchange Rates of Bitcoin and Ethereum. *International Journal of Economics , Commerce and Management United Kingdom*, *VI*(4), 455–465. https://www.researchgate.net/profile/Nashirah_Abu_Bakar/publication/324529181_STATISTICAL_DIAGNOSTICS_FOR_BIVARIATE_CORRELATION_AND_REGRESSION_ANALYSIS_BETWEEN_CRYPTOCURRENCY_EXCHANGE_RATES_OF_BITCOIN_AND_ETHEREUM/links/5ad317e1aca272fdaf79fad0/STATISTICA

Poongodi, M., Sharma, A., Vijayakumar, V., Bhardwaj, V., Sharma, A. P., Iqbal, R., & Kumar, R. (2020). Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system. *Computers and Electrical Engineering*, *81*, 106527. https://doi.org/10.1016/j.compeleceng.2019.106527

Uras, N., Marchesi, L., Marchesi, M., & Tonelli, R. (2020). Forecasting Bitcoin closing price series using linear regression and neural networks models. *ArXiv*. https://doi.org/10.7717/peerj-cs.279

Yatsyk, T. (2020). a Multiple Linear Regression Model for Cryptocurrency Price in the Financial Analysis and Accounting. *European Journal of Economics and Management*, *6*(5), 100–107. https://doi.org/10.46340/eujem.2020.6.5.13

APPENDICES

APPENDIX A
R Coding for Linear Regression

```
eth.day <- ETH_USD_Daily
View(eth.day)
summary (eth.day)

#RUN SUMMARY FOR X VARIABLE
summary(eth.day$Open)
summary (eth.day$High)
summary (eth.day$Low)

#CORRELATION MATRIX
round(cor(eth.day),2) #to round off the number
eth.day <- eth.day[, -1]
eth.day<- eth.day[,-5:-6] #buang variable that is uncorrelated
cor(eth.day)
pairs(~Close+Open+High+Low, data=eth.day)

#LINEAR REGRESSION
model_1 <- lm(Close~Open+High+Low, data=eth.day)
plot(Close~Open+High+Low, data=eth.day)
abline (model_1)
summary(model_1)

# CREATE TRAINING AND TEST DATA
set.seed(100)  # setting seed to reproduce results of random sampling
trainingRowIndex <- sample(1:nrow(eth.day), 0.8*nrow(eth.day))  # row indices for
training data
trainingData <- eth.day[trainingRowIndex, ]  # model training data
testData  <- eth.day[-trainingRowIndex, ]   # test data

# FIT THE MODEL ON TRAINING DATA AND PREDICT CLOSE ON TEST
DATA
lmMod <- lm(Close~Open+High+Low, data=trainingData)  # build the model
distPred <- predict(lmMod, testData)  # predict close price


#REVIEW DIAGNOSTICS MEASURES
summary (lmMod)  # model summary

# CALCULATE PREDICTION ACCURACY AND ERROR RATES
```

```
actuals_preds <- data.frame(cbind(actuals=testData$Close, predicteds=distPred))  # 
make actuals_predicteds dataframe.
correlation_accuracy <- cor(actuals_preds)
head(actuals_preds)

#PLOT OF PREDICTED VS ACTUAL VALUES
plot(x=distPred, y=testData$Close, xlab = 'Predicted Values', ylab = 'Actual Values')
abline(a=0,b=1)

# Min-Max Accuracy Calculation
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, 
max))

# MAPE Calculation
mape <- mean(abs((actuals_preds$predicteds - 
actuals_preds$actuals))/actuals_preds$actuals)

AIC(model_1)
BIC (model_1)

AIC(lmMod)
BIC(lmMod)
```

APPENDIX B
R Coding for Artificial Neural Network

```
#CODING USING DAILY DATA
library(DBI)
library(corrgram)
library(caret)
library(gridExtra)
library(ggpubr)
library(ggplot2)
library(forecast)
library(xts)
library(dygraphs)
library(caTools)
library(tseries)
library(zoom)
library(doParallel)

cl <- makeCluster(detectCores(), type= 'PSOCK')
registerDoParallel(cl)

setwd("C:/Users/Personal/Desktop")
getwd()

data <- read.csv("ETH-USD Monthly.csv", header = TRUE, sep = ',')
colnames(data)

head(data, 10)
tail(data, 10)
dim(data)
table(unlist(lapply(data, class)))

data <- data[,-7]
data <- data[,-6]
plot.ts(data[,c(4)])
plot.ts(data["Close"])

data_new <- data
data_new$Date <- as.Date(data_new$Date, format = "%Y-%m-%d")
data_new <- data_new[order(data_new$Date), ]
data_new
plot(data_new$Date,
    data_new$Close,
    type = "l",
    xaxt = "n")
axis(1,
    data_new$Date,
    format(data_new$Date, "%d/%m/%Y"))
```

```
eth_data <- as.numeric(data_new[,c("Close")])
eth.data <- data_new

fit_nnetar <- nnetar(eth.data$Close, repeats = 100 , size = 3)
print(fit_nnetar)
checkresiduals(fit_nnetar)

r <- cor(fitted(fit_nnetar)[14:length(eth_data)],
        eth_data[14:length(eth_data)])
r2 <- cor(fitted(fit_nnetar)[14:length(eth_data)],
         eth_data[14:length(eth_data)])^2
print(r2)
print(r)

x <- eth_data
y <- fitted(fit_nnetar)
ts.plot(x,y,
       gpars = list(col = c("black", "red")))
legend("topleft", legend = c("Actual", "Predicted"), col = c("black", "red"), lty = 1)

fitted(fit_nnetar)
#plotting Observed and Predicted with CI
accuracy(fitted(fit_nnetar),eth_data)

forecast_ <- forecast:::forecast.nnetar(fit_nnetar, h = 50,
                                       level = c(75,95), PI = TRUE)
autoplot(forecast_)

par(mfrow = c(1,1))
plot(forecast_)
summary (forecast_)

head(fitted(fit_nnetar),11

#TESTING AND VALIDATION
data <- read.csv("ETH_DATA.csv", header = TRUE, sep = ',')
colnames(data)
data$Date <- as.Date(data$Date, format = "%d/%m/%Y")
str(data)
featurePlot(x=data[,2:4],y=data$Close,plot='pairs')
data <- data[,-c(7)]
data <- data[,-c(6)]

CloseTS <- ts(data$Close, start=c(2017, 1), frequency=365.25)
time <- time(CloseTS)
CloseTS
plot(data$Date,
    CloseTS,
```

```
    type = "l",main = "Close Price", xlab="Date",  ylab="Close", bty="l", xaxt="n",
yaxt="n")
axis(1,
    data$Date,
    format(data$Date, "%d/%m/%Y"))

validLength <- 292
trainLength <- length(CloseTS) - validLength
CloseTrain <- window(CloseTS, end=time[trainLength])
CloseValid <- window(CloseTS, start=time[trainLength+1])
CloseTrain
CloseValid

# Use nnetar to fit the neural network.
set.seed(227)
CloseNN <- nnetar(CloseTrain, repeats = 100, P=1,size = 3)
CloseNN
CloseNN.pred <- forecast(CloseNN, h = validLength)
accuracy(CloseNN.pred, CloseValid)
CloseNN.pred

# Set up the plot
plot(CloseTrain, ylim = c(50, 3000), main = "Ethereum Close Price", ylab = "Close
Price", xlab = "Year", bty = "l", xaxt = "n", xlim =c(2017,2022), lty = 1)
axis(1, at = seq(2017,2022,1), labels = format(seq(2017,2022,1)))

lines(CloseNN.pred$fitted, lwd = 2, col = "red")
lines(CloseNN.pred$mean, lwd = 2, col = "red", lty = 2)
lines(CloseValid)
abline(v = 2020.25, col = "black", lty = 1, lwd = 1)
abline(h = 3000, col = "black", lty = 2, lwd = 1)
mtext("Training", line = -.5, at = c(2019,3200))
mtext("Validation", line = -.5, at = c(2021,3200))

# Plot the errors for the training period
plot(CloseNN.pred$residuals,
    main = "Residual Plot for Training Period")
CloseNN.pred
checkresiduals(CloseNN.pred)
fitted(CloseNN)
plot(fitted(CloseNN))



#CODING USING MONTHLY DATA
cl <- makeCluster(detectCores(), type= 'PSOCK')
registerDoParallel(cl)
setwd("C:/Users/Personal/Desktop")
getwd()
data <- read.csv("ETH-USD Monthly.csv", header = TRUE, sep = ',')
```

```
colnames(data)

head(data, 10)
tail(data, 10)
dim(data)
table(unlist(lapply(data, class)))

plot.ts(data[,c(4)])
plot.ts(data["Close"])

data <- data[,-c(7)]
data <- data[,-c(6)]

data_new <- data
data_new$Date <- as.Date(data_new$Date, format = "%Y-%m-%d") #For daily data,
format = "%d/%m/%Y
data_new <- data_new[order(data_new$Date), ]
data_new
plot(data_new$Date,
    data_new$Close,
    type = "l",
    xaxt = "n")
axis(1,
    data_new$Date,
    format(data_new$Date, "%d/%m/%Y"))

eth_data <- as.numeric(data_new[,c("Close")])
eth.data <- data_new
fit_nnetar <- nnetar(eth.data$Close, repeats = 100 ,P = 1, size = 3)
print(fit_nnetar)

checkresiduals(fit_nnetar)

r <- cor(fitted(fit_nnetar)[14:length(eth_data)],
        eth_data[14:length(eth_data)])
r2 <- cor(fitted(fit_nnetar)[14:length(eth_data)],
         eth_data[14:length(eth_data)])^2
print(r2)

x <- eth_data
y <- fitted(fit_nnetar)
ts.plot(x,y,
       gpars = list(col = c("black", "red")))
legend("topleft", legend = c("Actual", "Predicted"), col = c("black", "red"), lty = 1)

accuracy(eth_data,fitted(fit_nnetar))

forecast_ <- forecast:::forecast.nnetar(fit_nnetar, h = 25,
                                       level = c(75,95), PI = TRUE)
autoplot(forecast_)
```

```
summary (forecast_)
forecast_

#TRAINING VALIDATION
data <- read.csv("ETH-USD Monthly.csv", header = TRUE, sep = ',')
colnames(data)
data$Date <- as.Date(data$Date, format = "%Y-%m-%d")
data <- data[,-c(7)]
data <- data[,-c(6)]

CloseTS <- ts(data$Close, start=c(2017, 4), frequency=12)
yrange = range(CloseTS)
time <- time(CloseTS)
CloseTS

plot(c(2017, 2022), yrange, type="n", main = "Close Price", xlab="Date",
ylab="Close", bty="l", xaxt="n", yaxt="n")
lines(CloseTS, bty ="l")
axis(1, at=seq(2016,2021,1), labels=format(seq(2016,2021,1)))
axis(2, at=seq(50,3000,500), las=2)

validLength <- 10
trainLength <- length(CloseTS) - validLength
CloseTrain <- window(CloseTS, end=time[trainLength])
CloseValid <- window(CloseTS, start=time[trainLength+1])
CloseTrain
CloseValid
set.seed(227)
CloseNN <- nnetar(CloseTrain, P=1,size = 3)
CloseNN.pred <- forecast(CloseNN, h = validLength)
accuracy(CloseNN.pred, CloseValid)

plot(CloseTrain, ylim = c(50, 3000), main = "Ethereum Close Price", ylab = "Close
Price", xlab = "Year", bty = "l", xaxt = "n", xlim =c(2017,2022), lty = 1)
axis(1, at = seq(2017,2022,1), labels = format(seq(2017,2022,1)))
lines(CloseNN.pred$fitted, lwd = 2, col = "red")
lines(CloseNN.pred$mean, lwd = 2, col = "red", lty = 2)
lines(CloseValid)
abline(v = 2020.5, col = "black", lty = 1, lwd = 1)
abline(h = 3000, col = "black", lty = 2, lwd = 1)
mtext("Training", line = -.5, at = c(2019,3200))
mtext("Validation", line = -.5, at = c(2021.25,3200))



plot(CloseNN.pred$residuals, main = "Residual Plot for Training Period")
checkresiduals(CloseNN.pred)
CloseNN.pred
```

APPENDIX C1
R Coding for GARCH (1,1) Ethereum

```
library(ggplot2)
library(caTools)
library(dygraphs)
library(xts)
library(forecast)
library(fGarch)
library(tseries)
getwd()
data=read.csv("ETH_DATA.csv",header= TRUE)
df=data
df=df[,-c(2,7)]
df=xts(df[,-1],order.by=as.Date(df[,1],"%d/%m/%Y"))
m <- head(df, n=1463)
dygraph(m) %>%
  dyCandlestick()
da=data.frame(data$Date,data$Close)
class(da$data.Close)
da$data.Date = as.Date(da$data.Date, '%d/%m/%Y')
dev.off()
ggplot(data=da, aes(x=data.Date,y=as.numeric(data.Close))) + geom_line()
da$logClose= log(as.numeric(da$data.Close))
ggplot(data=da, aes(data.Date,as.numeric(logClose))) + geom_line()
da$sqrt= sqrt(da$logClose)
ggplot(data=da, aes(data.Date,as.numeric(sqrt))) + geom_line()
diffClose=diff(da$logClose)
newFrame=da[-c(1),]
acf(diffClose)
ggplot(data=newFrame, aes(data.Date,diffClose)) + geom_line()
adf.test(diffClose)
fit1 = auto.arima(da$sqrt, trace = TRUE, test = "kpss", ic = "bic")
Box.test(fit1$residuals, lag = 12, type = "Ljung-Box")
acf(fit1$residuals^2)
tsdisplay(fit1$residuals)
tsdiag(fit1)
# garch effect is there
model=garchFit(~garch(1,1), data=diff(da$sqrt))
summary (model)
```

APPENDIX C2

R Coding for GARCH (1,1) Bitcoin

```
library(ggplot2)
library(caTools)
library(dygraphs)
library(xts)
library(forecast)
library(fGarch)
library(tseries)
getwd()
data=read.csv("BTC_DATA.csv",header= TRUE)
df=data
df=df[,-c(2,7)]
df=xts(df[,-1],order.by=as.Date(df[,1],"%d/%m/%Y"))
m <- head(df, n=1463)
dygraph(m) %>%
  dyCandlestick()
da=data.frame(data$Date,data$Close)
class(da$data.Close)
da$data.Date = as.Date(da$data.Date, '%d/%m/%Y')
dev.off()
ggplot(data=da, aes(x=data.Date,y=as.numeric(data.Close))) + geom_line()
da$logClose= log(as.numeric(da$data.Close))
ggplot(data=da, aes(data.Date,as.numeric(logClose))) + geom_line()
da$sqrt= sqrt(da$logClose)
ggplot(data=da, aes(data.Date,as.numeric(sqrt))) + geom_line()
diffClose=diff(da$logClose)
newFrame=da[-c(1),]
acf(diffClose)
ggplot(data=newFrame, aes(data.Date,diffClose)) + geom_line()
adf.test(diffClose)
fit1 = auto.arima(da$sqrt, trace = TRUE, test = "kpss", ic = "bic")
Box.test(fit1$residuals, lag = 12, type = "Ljung-Box")
acf(fit1$residuals^2)
tsdisplay(fit1$residuals)
tsdiag(fit1)
# garch effect is there
model=garchFit(~garch(1,1), data=diff(da$sqrt))
summary (model)
```

APPENDIX C3
Bitcoin and Ethereum Candlestick

Bitcoin Candlestick

Ethereum Candlestick

APPENDIX C4
Bitcoin Model Summary

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~garch(1, 1), data = diff(da$sqrt))

Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 0x000001ec8e4bd888>
 [data = diff(da$sqrt)]

Conditional Distribution:
 norm

Coefficient(s):
        mu       omega      alpha1       beta1
5.2068e-04  3.4097e-06  1.2544e-01  8.2001e-01

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     5.207e-04   1.632e-04    3.190  0.00142 **
omega  3.410e-06   6.649e-07    5.128 2.93e-07 ***
alpha1 1.254e-01   2.097e-02    5.982 2.21e-09 ***
beta1  8.200e-01   2.489e-02   32.942  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood:
 5228.521    normalized:  3.586091

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  17468.93  0
 Shapiro-Wilk Test  R    W      0.9000009 0
 Ljung-Box Test     R    Q(10)  18.71211  0.04407539
 Ljung-Box Test     R    Q(15)  21.63485  0.1177421
 Ljung-Box Test     R    Q(20)  28.65415  0.09479606
 Ljung-Box Test     R^2  Q(10)  2.848143  0.9847752
 Ljung-Box Test     R^2  Q(15)  3.489308  0.9989885
 Ljung-Box Test     R^2  Q(20)  4.003814  0.9999531
 LM Arch Test       R    TR^2   2.956828  0.9958412

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-7.166695 -7.152196 -7.166710 -7.161286
```

APPENDIX C4
Ethereum Model Summary

```
Title:
 GARCH Modelling

Call:
 garchFit(formula = ~garch(1, 1), data = diff(da$sqrt))

Mean and Variance Equation:
 data ~ garch(1, 1)
<environment: 0x000001ec8a15c580>
 [data = diff(da$sqrt)]

Conditional Distribution:
 norm

Coefficient(s):
        mu       omega      alpha1       beta1
4.1931e-04  1.0151e-05  1.0569e-01  8.2662e-01

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     4.193e-04   2.776e-04    1.511  0.13090
omega  1.015e-05   3.204e-06    3.168  0.00153 **
alpha1 1.057e-01   2.276e-02    4.644 3.42e-06 ***
beta1  8.266e-01   3.947e-02   20.940  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log Likelihood:
 4489.663    normalized:  3.07933

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  5070.929  0
 Shapiro-Wilk Test  R    W      0.9211445 0
 Ljung-Box Test     R    Q(10)  22.1513   0.01435212
 Ljung-Box Test     R    Q(15)  26.61989  0.03199037
 Ljung-Box Test     R    Q(20)  37.86591  0.009195569
 Ljung-Box Test     R^2  Q(10)  8.267552  0.6027207
 Ljung-Box Test     R^2  Q(15)  9.662376  0.8404229
 Ljung-Box Test     R^2  Q(20)  11.34703  0.9367218
 LM Arch Test       R    TR^2   8.292321  0.7618896

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC
-6.153174 -6.138675 -6.153189 -6.147765
```