STAT 3799 Final Report

# Cryptocurrency Market Prediction Using A Neural Network: Mimicking Quantitative Strategies

Topic: Financial Data Analysis


**Department of Statistics and Actuarial Science,**

**The University of Hong Kong**

**Chong, Inbum** (3035553355)

**BSc, Major in Decision Analytics**

**Supervisor:**

**Dr. Ke, Zhu**

## Abstract

Along with the personal interests in the application of machine learning methodologies in the market prediction task, this project proposes a neural network using indexes from the discipline of quantitative finance as the input features. The objective of this project is to build a trading network (main model) based on the aforementioned neural network and to examine its effectiveness in trading cryptocurrencies. The main model is then compared with the extrapolation model, ARIMA-GARCH, modeling the residuals of the autoregressive moving average model with the generalized autoregressive conditional heteroskedasticity model. Both models' accuracies and performances in the mock trading are compared. As result, although the accuracy of the model has overperformed the ARIMA-GARCH model, the overall accuracy was lower than the expectation. On the other hand, the performance of the main model was outstanding during the mock trading.


# 1. Introduction

## 1.1 Market Prediction

A market prediction has long attracted researchers [13]. The final objective of all financial analysis and research is to increase profit gain through financial instruments. Ever since Louis Bachelier's doctoral thesis, *"Theory of Speculation"*, established in 1900, quantitative approaches in investments have been actively studied by numerous researchers. In recent years, along with the advancements of machine learning and artificial intelligence, there have been increasing attempts to build neural networks to forecast market movements [16, 17, 23, 26, 29]. Despite the level of inference accuracy using neural networks being high in the image and language domain, yet the domain of market prediction is underperforming [13]. The major reason is due to the complexity of the financial market. Through this capstone project, I have desired to understand how neural networks can be used in market prediction tasks.

## 1.2 Cryptocurrency

Among various asset classes, the volume of the stock market is undoubtedly the largest. On the other hand, the cryptocurrency market is emerging since digitalization is rapidly progressing [17]. On 8 November 2021, the overall market capitalization of cryptocurrencies exceeded 3 trillion U.S. dollars, which is a huge growth considering its value was 450 billion U.S [32]. dollars in November 2020. Major reasons that this capstone project is based on cryptocurrency market data are:

- **Crypto market system**: Unlike the stock market, the crypto market is always open. Hence, it is continuous, rarely has off-market risks.
- **Fewer regulations**: Since the crypto market is newly introduced compared to the stock market, fewer regulations are to consider.
- **Volatility**: Crypto has high volatility compared to other instruments, which enables neural networks to learn different circumstances.

### 1.3 Quantitative Finance

As mentioned above, quantitative approaches in investments have been developed over the past century [19]. Practitioners in financial markets actively use quantitative strategies. I have long considered how to collaborate classical quantitative trading strategies into the discipline of artificial intelligence. From the idea of trading strategies using quantitative indexes, which is explored thoroughly in part 5.2, I have incorporated those indexes as variables of the neural network.

### 1.4 Extrapolation and Interpolation

The majority of time series forecasting methodology is based on extrapolation. Given previous data, it predicts future values. Nevertheless, the leading problem is that the confidence interval of prediction is getting wider as further the predicting time step [19]. Moreover, in the particular task of market prediction, it often conveys non-quantitative variables in price changes.

On the other hand, the interpolation method makes inferences of target value given values of variables. In fact, it does not take the influence of time into account unless implicitly setting up time-related variables, whereas the price of financial instruments is dependent on previous prices. However, I have assumed that, as previously mentioned, with variables used in the discipline of quantitative finance, deep neural networks will successfully learn trading patterns at some points, which in fact mimicking existing trading strategies of quantitative traders.

Hence, this project additionally attempts to explore how the interpolation model is different from the extrapolation model in making inferences about market returns. In contrast to the neural network, I propose to implement the ARIMA-GARCH model, which estimates the residuals of the ARIMA model with the GARCH model. There exists a computational burden using multivariate model making inferences for each row of test data, the univariate model will be implemented in this project. Thus, the neural network model is likely to overperform the extrapolation model, but since the objective is to capture the difference, it is justifiable to use the light model.

### 1.5 Project Overview

This project mainly consists of three parts: exploratory data analysis, experiment using self-build neural network (main model), and comparison of the main model to extrapolation model, ARIMA-GARCH. The main objective of this project is to build a trading network based on deep learning architecture using quantitative indexes as features and to examine its effectiveness in trading cryptocurrencies. Conceivably, this research offers the direction of future researches. Also, auxiliary objectives are postulating appropriate data analysis procedures for the corresponding task and examining the competitiveness of using interpolation over extrapolation in market prediction.

## 2. Related Works

Although this project's direction is not to update the state of art model in market prediction tasks, rather it explores the competitiveness of approaches using quantitative indexes as features, it is beneficial to learn what works have been done by other researchers for formulating the directions of future research and adjusting the details of the project.

Already in 2003, Wang *et al*. [26] applied neural networks on stock market price prediction. Along with the recent advances in deep learning and machine learning frameworks, many researchers have proposed applications of deep learning models to market prediction tasks. Liu *et al*. [16] proposed applications of CNN and LSTM to the stock market prediction, it is a hybrid model in which CNN is used for the stock selection strategy and LSTM learns time-series features. Recently, there have been increasing attempts to amend the network structure of existing deep learning frameworks. Shen and Shafiq [23] have customized the layers in LSTM for its application towards stock market predictions. The model proposed in this project adds a concatenation technique to the basic feedforward neural network structure.

Apart from the stock market prediction using deep learning frameworks, there have been attempts to use those frameworks to predict the movements of the cryptocurrency market. Zhengyang *et al*. [29] applied ANN-LSTM model to predict the price of Bitcoin (BTC) and Livieris *et al*. [17] applied CNN-LSTM framework to the corresponding task. A distinction of the proposed model in this project from other relevant research is that it attempts to adopt indexes that are used in the discipline of quantitative finance as input features of the feedforward neural network.

## 3. Problem Formulation

This project mainly aims at predicting the return of a next tick $r$ of the target tickers (cryptocurrencies) $c$ in a selected trading universe $C$ on each tick $t$ of the test set (5 trading days: 1440 ticks) using a neural network. Each tick represents 5 minutes, which is justified in part 5.1. This implies that each row of data represents a single tick in a 5-minute candlestick, and the target variable of each row is a percentage in price change from the current tick to the next tick.

$$r_t = p^c_{t+1} - p^c_t$$

Where $p^c_t$ indicates the closing price at tick $t$. Although many relevant types of research often formulate the target variable as binary (i.e. stock price movement 1: Up or 0: down) [21], this project is proceeding with calculating the return explicitly, since the degree of change is significant in the postulation of trading strategies. Hence, it is a regression task, instead of classification.

As mentioned above, this project includes two auxiliary objectives. Hence, it aims at justifying the proposed main model throughout the exploratory data analysis. Lastly, implementing an extrapolation model to predict the return of the next tick, as the main model aims at.

## 4. Data Collection and Software

### 4.1 Data

Data is collected from the Korean cryptocurrency exchange platform, Upbit. Upbit provides a public API, which lets the public parse data at any length. Collected data is a five-minute candlestick of 13 selected tickers (Appendix B) and consists of six columns: open price, close price, high price, low price, volume, and value. The prices are in Korean Won (KRW). The length of data of each ticker is 12,000 rows, from 12 October 2021 to 20 November 2021. The length of the test set is 1,440 rows, from 16 November to 20 November.

### 4.2 Trading Universe

As mentioned above, I have selected 13 tickers, included in the trading universe. Since there are over 6,000 cryptocurrencies registered in the market [30], I have selected those that are available in Upbit and Binance, which is the largest exchange across the world. Also, I excluded tickers that have an extensively low market capitalization. Thereby, 13 different main models are built under the same algorithm and network architecture, instead of building a unified model. Since each ticker exhibits different characteristics in movement, discrete models were adopted in this project.

### 4.3 Software Environment

The whole experimentation was undergone in a Python 3.8 environment. The main model was built upon the PyTorch (1.10) framework, and the extrapolation model was built upon the pmdarima and arch library. Also, NumPy, pandas, statsmodel, and Scikit-learn libraries were used.

## 5. Exploratory Data Analysis

Considering the characteristics of financial data, it is essential to have an in-depth understanding of the dataset. The exploratory data analysis consists of three parts: frequency of trading, impulse responsive analysis of BTC and feature analysis.

### 5.1 Frequency of Trading

The frequency of trading is important when it comes to determining the trading strategies. It is assumed that the randomness in price movements has higher randomness with the longer frequency of trading [22].

The lagged return $r_t$ of the different datasets with four different frequencies are used in this analysis: 5-minutes, 15-minutes, 60-minutes, and the daily candlestick of BTC.

#### 5.1.1 Variance Analysis

Variance is one of the indicators of volatility in the discipline of finance [22]. The standard deviations of the four datasets are calculated and the kernel distribution estimation plots (Appendix D) are drawn.

**Standard deviations of lagged returns:**

- 5-minute: 0.0018
- 15-minute: 0.0021
- 60-minute: 0.0048
- Daily: 0.035

As there exists clear evidence from the results, a longer frequency of data results in higher standard deviations of lagged returns. This indicates that the prediction of the further data point is more volatile.

### 5.1.2 Time-series Analysis

In addition to the volatility analysis, this section analyzes the difference in the predictability of time-series models on datasets with different frequencies. In this analysis, the 5-minute and daily candlestick are used. The time-series forecasting model used in this analysis is ARIMA (Autoregressive Integrated Moving Average).

$$\hat{y}_t = c + \sum_{i=1}^{p} a_i y_{t-1} + \sum_{i=1}^{q} b_i \varepsilon_{t-1} + \varepsilon_t$$

$$c = \mu(1 - \sum_{i=1}^{p} a_i), \varepsilon_t = \sqrt{\sigma_t} z_t$$

In order to fit the ARIMA model to the dataset, the confirmation of stationarity of the datasets is compulsory. In order to check the stationarity, the Augmented Dickey-Fuller (ADF) test was conducted. The null hypothesis of the ADF test is that a unit root is present in a dataset. The results of the ADF test on the lagged returns of 5-minute and daily candlestick are:

- 5-min
  - ADF Statistic: -4.26
  - P-value: 0.0005
- Daily:
  - ADF Statistic: -13.722
  - P-value: 0.0000

In the following steps, the autocorrelation functions and partial autocorrelation functions were applied to both datasets, which in order for the determination of AR and MA coefficients of the ARIMA model (Appendix C.1, C.2). The ARIMA(5,0,1) model and ARIMA(4,0,2) model were fitted to the lagged returns of the 5-minute and daily candlesticks (Appendix C.2). The training data contains 190 ticks and the test data contains 10 ticks. The result of fitted ARIMA models to the datasets can be referred from the appendix C.4 and C.5. The following results indicate the inferences on the test data:

ARIMA (5,0,1) to 5-minute candlestick:

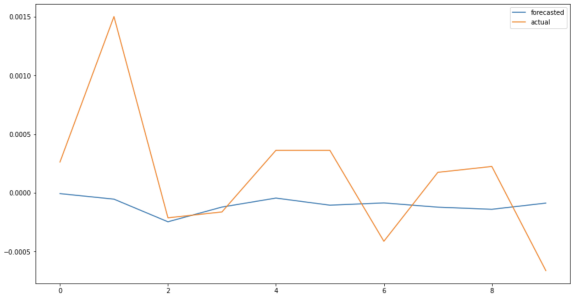



*Figure 1. The graph of inference made by the ARIMA model on a 5-minute candlestick (BTC) (blue) and actual movements of return (orange)*

R-squared: -12.58
Correlation: 0.351
RMSE: 0.001

ARIMA (5,0,1) to 5-minute candlestick:

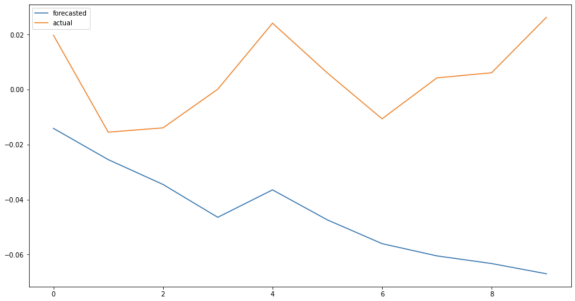



*Figure 2. The graph of inference made by the ARIMA model on a daily candlestick (BTC) (blue) and actual movements of return (orange)*

R-squared: -1345.26
Corr: -0.118

RMSE: 0.055

As shown from the graphs, there exist discrepancies between the prediction accuracy between the two data. The negative correlation between forecasted and actual data indicates that the model fitted to the daily candlestick has a lower performance compared to the model fitted to the 5-minute candlestick.

This results empirically conveys that making predictions using data with longer frequency is more difficult than using data with shorter frequency. Therefore, this project has chosen the 5-minute candlestick as an input to the main model.

### 5.2 Feature Analysis

This section introduces the features that were used in the project and analysis.

#### 5.2.1 Features

A distinctive characteristic of this project from other relevant research is that the neural network proposed in this report attempts to mimic quantitative trading strategies of the practitioners. Thus, this project assumes that if the model is learning the quantitative indexes as variables, the model would behave as quantitative traders. 15 features from 8 categories of quantitative indexes have been chosen:

- **Bollinger Band**: Width of Bollinger Band, Distance to Upper BB, Distance to Lower BB

Bollinger is one of the technical indicators, developed by John Bollinger, which is in order to supplement an indicator called envelop, unable to indicate an appropriate point of sale during the highly volatile market [9]. Bollinger band assumes that the majority of price movements occur within the band deviated by a certain proportion of standard deviations from the simple moving average [9]. Thus, the Bollinger band consists of three components:

- Simple Moving Average (*d*):

$$SMA(d) = \sum_{i=0}^{d-1} p_{t-i} \times d^{-1}$$

- Upper Bollinger Band (*d, k*):

$$UBB(d,k) = SMA(d) + k \times \sigma_d(p_t)$$

- Lower Bollinger Band (*d, k*):

$$LBB(d,k) = SMA(d) - k \times \sigma_d(p_t)$$

$\sigma_d(p_t)$indicates the standard deviation of $d$ ticks calculated from $p_t$. In this project, the value of $k = 2$ and $d = 10$.

One of the trading strategies using the Bollinger band is utilizing the characteristic of price movements near the upper and lower boundaries. When the current price is located near the upper or lower boundary, the following price has a tendency to return to the simple moving average [9].

The project utilizes another feature from this category, the width of the Bollinger band. The width of the Bollinger band determines the volatility of the market: a thicker band implies higher volatility and a narrower band implies lower volatility.

- **MACD**: MACD, MACD Oscillator

Moving average convergence and divergence (MACD), developed by Gerald Appel, supplements the disadvantage of

moving average. Appel noticed that short-term and long-term moving averages are repetitively moving away and closer to each other, thus they are continuously converging and diverging [1]. Then, he had assumed that the change of market trends occurs when the short-term and the long-term moving average is at the maximum of its divergence [1]. This project utilizes two features of MACD:

- MACD (p,q): Short term moving average (p days) - Long term moving average (q days)
- MACD Oscillator (p,q,r): MACD (p,q) - MACD Signal (r)

MACD Signal is the simple moving average of MACD of tick *r*. This project uses $p = 12$ , $q = 26$, and $r = 9$.

One trading strategy using MACD is to refer to the value of MACD. Increasing the trend of price often implies the positive MACD, and decreasing trend implies the negative MACD [5]. Moreover, MACD Oscillator indicates the change in trends of price movements. If MACD is 0 there is a high likelihood of the change in market trend occurring [5].

- **RSI**: RSI

Relative Strength Index (RSI), developed by Welles Wilder, is a technical indicator of appreciation pressure and depreciation pressure of a current price. RSI determines the degree of overbought and oversold.

$$RSI(n) = AU \times (AU + AD)^{-1} \times 100$$
$$AU(n) = Mean\ of\ upturns\ of\ closing\ prices\ of\ n\ ticks$$
$$AD(n) = Mean\ of\ downturns\ of\ closing\ prices\ of\ n\ ticks$$

Wilder asserts that if RSI is greater than 70 the market is in an overbought state, and if it is less than 30, the market is in an oversold situation [5]. This project uses $n = 14$.

- **Pivot**: Distance to second support, Distance to the second resist

Pivot points are actively used by investors in trading financial instruments [18]. Pivot points consist of five points: pivot, first support, second support, first resist, and second resist. The prices are making meaningful movements near those points [18]. This project adopts second support and second resist points since there exists a higher likelihood of the points indicating more meaningful information in relation to the following price movements. For every tick *t*:

$$pivot = (p_{high} + p_{low} + p_{close}) / 3$$
$$second\ support = pivot - p_{high} + p_{low}$$
$$second\ resist = pivot + p_{high} - p_{low}$$

- **Stochastic**: Fast k, Slow k

Stochastic oscillator (so-called stochastic), developed by George C. Lane, is one of the momentum indicators. It examines the relationship between the degree of price fluctuation for a certain period and the current price [20]. Similar to RSI, stochastic indicates whether the market is in an overbought or oversold state [20].

$$fast\ k\ (n) = (p_c - p_{highest,\ n}) \times (p_{highest,\ n} - p_{lowest,\ n}) \times 100$$

$$slow\ k\ (n,\ m)\ =\ MA(fast\ k\ (n),\ m)$$

$p_{highest,\,n}$ indicates the highest price in the period of $n$ ticks, and $p_{lowest,\,n}$ indicates the lowest price in the period of $n$ ticks. This project uses $n = m = 5$.

The stochastic is further divided into fast k and slow k. In practice, a chart of fast k is frequent and often includes false signals, practitioners often refer to the slow k [20]. However, this project uses both stochastics since the frequency of trading is shorter, thus the fast k may include some useful information.

- **Momentum**: Momentum

Momentum simply calculates the degree of change in prices for a certain period. There is a higher likelihood that the price will increase if it increased before [3]. However, this statement is not necessarily true since there exists a complex market mechanism applied [3]. Hence, this project adopts self-calculated momentum.

$$M = 0.5(p_t / p_{t-3} - 1) + 0.3(p_t / p_{t-6} - 1) + 0.2(p_t / p_{t-12} - 1)$$

The momentum calculates the price change in 3 ticks, 6 ticks, and 12 ticks, which represents the price change in 15, 30, 60 minutes. Then it weighted the values. This project assumes that, at a current price with a higher momentum, there exists a higher chance of the next price increase.

- **Ichimoku**: Distance between Tenkan Sen and Kijun Sen, Cloud Thickness, Distance to cloud

The Ichimoku chart, the so-called Ichimoku cloud, visualizes the balance in the trend of market prices [6]. The Ichimoku chart consists of four components:

$$tenkan\ =\ (p_{lowest,\,9} + p_{highest,\,9})\ \div\ 2$$
$$kijun\ =\ (p_{lowest,\,26} + p_{highest,\,26})\ \div\ 2$$
$$senkou\ span\ a\ =\ shift(\ [(kijun\ +\ tenkan)\ \div\ 2],\ 26)$$
$$senkou\ span\ b\ =\ shift(\ [(p_{lowest,\,52} + p_{lowest,\,52})\ \div\ 2],\ 26)$$

$p_{lowest,\,n}$ indicates the lowest price in $n$ ticks. *shift*(*X,n*) indicates the function that allocates X ahead of the data points by $n$.

This project derives three features calculated from those components of the Ichimoku:

- Distance between Tenkan and Kijun
  $= tenkan - kijun$
- Cloud thickness
  $= senkou\ span\ a\ -\ senkou\ span\ b$

If the distance between Tenkan Sen and Kijun Sen is positive, it implies that the market trend is positive, and the negative implies the negative market trend [6]. The cloud thickness implies the likelihood of change in trends of market movement [6]. Thicker cloud implies stronger resistance, which makes the trend harder to alter its trend. Lastly, if the clout is located under the current price, it indicates the increasing market trend and opposite [6].

- **Volume**: Volume

Higher volume often indicates the stronger price change in the next tick [14].

- **Target Variable:** Return in next tick

The predicting value of the model is the return next tick, the lagged return. The lagged return was multiplied by 100, which represents the value as a percentage. The reason is in order to scale up the value to see the losses during the training process more clearly.

### 5.2.2 Features at the Extreme Values

The appendix E is a correlogram of the ETH training data, which visualizes the one-to-one correlation between each feature. The noticeable result is that none of the features have a higher than 0.1 correlation coefficient between individual features and the target variable, which contradicts the assumption that each feature is having an influence on the target variable. However, it turns out that the features are not correlated unless they are at extreme values.

The experiment filtered the dataset with values that are away from its mean by its three standard deviations, which are considered as extreme values. Then, it re-calculated the correlation coefficient between the filtered data and the corresponding values of the target variable. The result of this experiment shows that there exist increasing values of correlation. Therefore, this implies that the features are having meaningful information, especially when the values of the features are extreme. In addition, this fits into the descriptions of the quantitative indexes, which indicates that the movements of the following prices act in certain ways when the value of indexes is not normal.

### 5.2.3 Impulse Response Analysis

The major reason to predict the return in the next tick, which means that it only predicts the one time step away from the present, is due to the impulse that has an effect on the target variable. Impulse response analysis examines the response in a variable when an impulse was imposed on another variable. For example, if an impulse in a size of one standard deviation, the value of the target variable could be influenced. Thus, this section analyzes the response of the target variable, return in the next tick, when an impulse is imposed on the other feature.

Impulse response analysis can be conducted from the vector autoregressive (VAR) model fitted to the data. In order to examine the one-to-one relationship between the individual feature and the target variable, the variables of the VAR model are the $x_i$ and $x_{target}$. $x_i$ is the individual introduced above. For example, in the VAR ($k$) model of $x_i$, i = distance to upper Bollinger band and $x_{target}$ is:

$$X_t = \mu + A_1 X_{t-1} + \ldots + A_k X_{k-1} + u_t$$

$$X = \left[x_i \; x_{target}\right]^T$$

The time lag $k$ determines the length of previous $k$ time steps that influence the model. This project has fixed $k = 1$ since the scope of the project is to take only a previous tick's inputs into account. Next, the model requires data to have stationarity, and it is checked from the appendix F.1. Once the VAR model is fitted (Appendix F.2), then it is ready to plot the impulse response function. The VAR framework provided by the statsmodel library was used to plot the impulse response function.

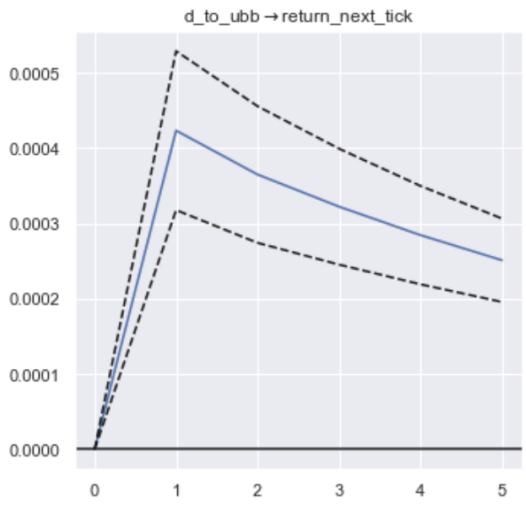



Fig 3.Impulse response function from d_to_ubb to return_next_tick by number of ticks away from the present tick

The figure 3 represents the response function of the amount of response gained by the target variable (return in next tick) when there was an impulse imposed on the feature (distance from current price to upper bollinger band). X-axis represents the number of ticks away from the present tick. When an impulse was imposed on the feature, it positively influences the target variable by 0.0004, which means that the 0.04 percent of return can be influenced by the impulse. The degree of response is maximum at one tick away from the present tick, and gradually decreases as more ticks away from the present. This implies that the return in the tick is the most influenced by the impulse of the feature. Therefore, the main model proposed by this project aims at predicting the values of returns in next ticks since the features have the most influence in the following price.

## 6. Main Model Overview

The model is in the form of deep feedforward neural networks (Fig 4.). The network consists of one input layer, four hidden layers, five linear layers, and the output layer. The input layer, $X_0$, consists of 15 inputs. Before entering the first linear layer, its batch is normalized, and 10 percent of the batch is dropped out. Once $X_0$ goes through the linear layer, it produces a hidden layer $X_1$ with 40 variables. Then, the model concatenates the $X_1$ and $X_0$ before passing another linear layer. Again, the batch normalization and dropout functions are executed. It repeats the same process until it finally outputs the predicted return $y$.

The network does not consist of a lot of layers as other networks do. The main reason is to prevent gradient vanishing problems. Since the length of the training, the dataset is not huge, and also the number of features is limited, going through more layers will lead to the model being overfitted. A number of researchers have pointed out that the gradient vanishing problem exists in neural networks [12], and recent researches have developed algorithms to prevent the problem. However, this project uses basic architecture with concatenation, instead of using developed algorithms, since the objective of the project is to examine the competitiveness of using interpolation models, I have decided to use the basic structure. Thus, if the model could show some meaningful result, continuing research using this approach will have competitiveness in the market prediction task.

### 6.1. Ensemble: Concatenation

Relevant researches have empirically illustrated that the ensemble of multiple models increases the performance of the final

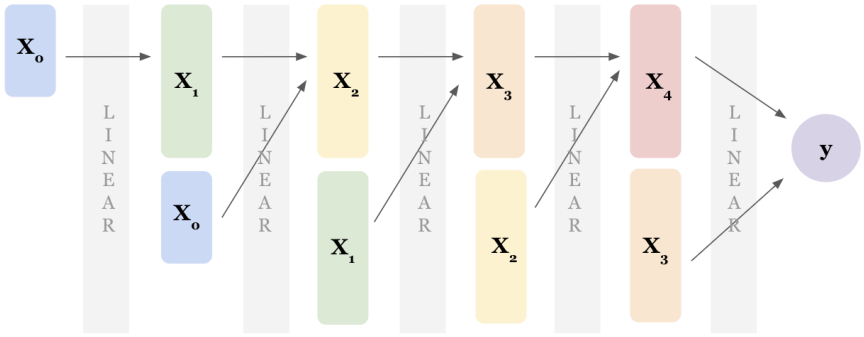



*Fig 4. The neural architecture of the main model*

model [11]. Among several techniques, the concatenation of layers prevents the gradient vanishing problem. This eventually reduces the chances of learning from the previous layers being diminished as it goes through the layer. In addition, it has a subsidiary effect in preventing the model from being overfitted.

### 6.2. Forward Phase

A forward phase of the training procedure goes through multiple linear layers. When the network first learns the weights *W*, when $X_0$ goes through the $X_1$ in the first epoch, the network randomly chooses $x_i$ and estimates the weight $w_i$.

$$X_1 = W_0 X_0$$

Then $X_1$ is concatenated with $X_0$ to go through another layer to produce $X_2$.

$$X_2 = W_1 concat(X_0, X_1)$$

Therefore, $X_{i+1} = W_i concat(X_{i-1}, X_i)$

### 6.2.1 Activation Function

Before entering the consequent linear layer, the activation function is executed. An activation function is crucial in composing the neural networks. It defines how the weighted summation of the input is transformed into nodes in a layer [15]. From a number of researches, the use of the activation function is effective in preventing the gradient vanishing problem [15]. It basically determines what nodes to activate or deactivate.

This project uses the leaky ReLU (Leaky Rectified Linear Unit). While a typical ReLU determines whether the node is activated or deactivated, the leaky ReLU does not completely deactivate the node, rather it preserves the value for a certain portion α.

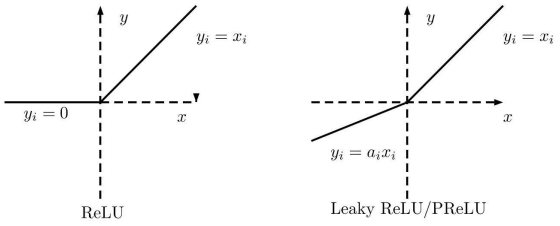



*Figure 5. ReLU (left) and Leaky ReLU (right)*

$$ReLU: f(x_i) = max(0, x_i)$$

$$Leaky\ ReLU: f(x_i) = max(\alpha x, x_i)$$

A number of researchers have reported the circumstance of dying nodes, which are no longer in use for the learning process [7]. In fact, few kinds of research using the leaky

ReLU activation function exhibited better prevention of vanishing gradients.

### 6.2.2 Batch Normalization

Batch normalization reduces internal covariate shift, which is adding a normalization layer by fixing the means and variances of the layer inputs [2].

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2$$

Thus, the batch normalization layer includes the above means and variances for minibatch *B*. During the update procedure, it results in following x and y.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i = \gamma \hat{x}_i + \beta = BN_{\gamma,\beta}(x_i)$$

Batch normalization is effective in the prevention of the gradient vanishing problem since it reduces the dependence of gradients on the scale of the parameters and of the initial values [2]. Therefore, it significantly reduces the risk of divergence in the learning process.

### 6.2.3 Dropout

The implementation of the dropout layer is crucial in preventing the overfitting model [24]. A dropout layer drops a certain portion of a batch in order to avoid voting and co-adaptation effects. This project uses a dropout rate = 0.1, which means that it drops 1/10 of rows from a batch.

The batch normalization layer and the dropout layer are not included in the last linear layer in Figure 4, since it directly outputs the prediction, there are no nodes to be normalized nor dropped. However, the input layer $X_0$ passes through both batch normalization and dropout layer before producing $X_1$.

## 6.3. Back Propagation Phase

The model is actually learning through the backpropagation procedure. Backpropagation compares the predicted and actual values of the target variable to see the discrepancies. Then, it propagates backward of the neural network to update the weights of the model. An optimization method determines how the model converges to optimal weight, meaning minimizing a loss function.

### 6.3.1 Optimization

Optimization methods in the backpropagation attempt to search for an optimal solution (weights), given input variables. The application of classical optimization techniques confronted the problem of gradient vanishing/explosion. Many optimization techniques were introduced by researchers, and this project uses the Adadelta optimization method.

The classic gradient descent method is rarely used in optimizing neural networks. It is due to its massive computational complexity [8]. In order to supplement such a disadvantage, the stochastic gradient descent (SGD) method was introduced, which can update parameters using an approximated gradient of the loss function by a gradient obtained from a minibatch of observations. SGD reduces the computational complexity by a noticeable amount. Nevertheless, its

drawbacks are high oscillations during the convergence and lower speed to convergence, which makes the learning process unstable and inconsistent. For an improvement, momentum is calculated, and it adopts a certain formula to update the parameters. The model remembers the gradient from a previous iteration and is linearly combined with the current gradient to determine the next update. The idea of momentum can push the weights towards the convergence point much faster, along with reduced risks of getting stuck in a local minima/maxima [8].

The Adadelta algorithm, suggested by Zeiler [28], adds a more technical adaptive learning rate to the SGD and momentum, and it has been developed in order to supplement the drawbacks of the AdaGrad algorithm [8], which suggested the idea of adaptive learning rate. The algorithm corrects units with a Hessian approximation [28]. It was eventually able to resolve the gradient vanishing problem and the continual decay of learning rates throughout the training procedure [28].

### 6.3.2 Loss Function

Neural networks optimize the gradients in order to minimize the loss function, which implies the error between the predicted and actual data. Since this project belongs to the class of regression problem, the mean squared error (MSE) was used as a loss function.

## 7. Extrapolation Model

As mentioned earlier, one of the auxiliary objectives is to examine the competitiveness of an interpolation model over an extrapolation model. The inference method of the main model is the interpolation method, which deeply learns a lot of cases in order to interpolate the value of the target variable when it is given with the inputs. On the other hand, this project does not implement the extrapolation model which uses deep learning architecture. Since the main objective of the project is to implement trading algorithms and to examine deep neural networks' competitiveness in market prediction tasks, I have decided to implement a classical time-series forecasting model with a little bit of variation.

The ARIMA-GARCH model is implemented in order to make reference to how well the main model is performing compared to the extrapolation model. As mentioned in part 5, the ARIMA model predicts the future values based on autoregressive and moving averages. However, this model incorporates a volatility model called generalized autoregressive conditional heteroskedasticity (GARCH). The idea is to model the residuals of the ARIMA model using the GARCH model

$$\hat{y}_t = c + \sum_{i=1}^{p} a_i y_{t-1} + \sum_{i=1}^{q} b_i \varepsilon_{t-1} + \varepsilon_t$$

$$c = \mu(1 - \sum_{i=1}^{p} a_i), \varepsilon_t = \sqrt{\sigma_t} z_t$$

The above expression represents the inference of the ARIMA model. $\varepsilon_t$, the residuals of the ARIMA model contain the standard deviation at *t*. Here, the GARCH model comes into model the volatility.

$$\sigma^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{q} \boldsymbol{\beta}_i \sigma_{t-i}^2$$

The original residuals of the ARIMA model assume that the residuals will be normally distributed. However, this statement is untrue in most financial data. Since the external risk has an impulse response on the price of a financial instrument for a certain period, thus the volatility of price will not be normally distributed. Eventually, this induces heteroskedasticity of the variance. Therefore, in order to reflect such a characteristic of financial data, Engle developed the ARCH model. However, the ARCH model had drawbacks of preserving nonnegativity and the complex representation of the model with the large time difference, $q$. Therefore, Bollerslev introduced GARCH, which is written in the above equation. That is why the extrapolation model in this project modeled the volatility with the GARCH model.

This project uses the pdarima library from python, which automatically tunes the parameters of the ARIMA model, and arch library to implement the GARCH model. The parameters of the GARCH model are fixed to (1,1). Unlike the training dataset of the main model, the length of the train data set of the ARIMA-GARCH model is the previous 100 data of each data point of the test set. The test set is equivalent to the test data for the main model. Thus, the model only predicts the one time step away from each data point, which is exactly predicting the same thing, returning in the next tick, as what the main model is predicting.

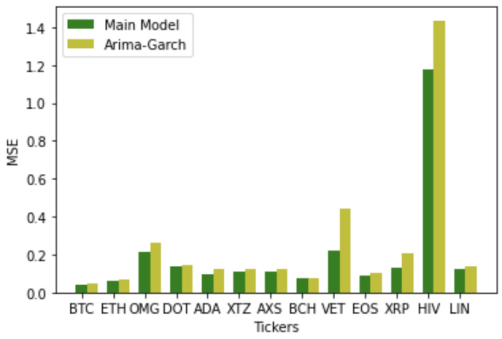
# 8. Experiments

## 8.1 Experiment Setups

The train set and the test set are split from the preprocessed data (Appendix A.2), which has calculated the 15 features from the raw data. 13 discrete main models were trained for each ticker. Instead of building a united model which can be applied to all tickers, this project has trained the discrete models by tickers, since there exist possibilities of differences in feature importances toward the target variable. Moreover, it will leave a chance for the model to make a higher profit using the discrete models. Then, it obtains the inferences on the test dataset. During the inference process, the network deactivates the batch normalization and dropout layer, which will have influences on the inference result. Also, the extrapolation model, ARIMA-GARCH calculates the prediction of each data point of the test data based on the 100 previous sequences of each data point. The inference results are then compared to each other to see the competitive performances, and the metrics were obtained to see the two model's accuracy and performance. Finally, the project underwent mock trading on the test dataset.

The simple trading strategy was implemented during the mock trading. If the predicted return in the next tick is greater than 0.05%, then it buys at the close price at the current closing price and sells it at the next tick's closing price. Implementing the simplest strategy lets us see the effectiveness of the model more directly. The number 0.05% is set due to the transaction cost

required from the platform being 0.05%. However, the turnover was not included in the results in order to analyze the result without external influences. It leaves an availability of adopting different trading strategies, and different results will be



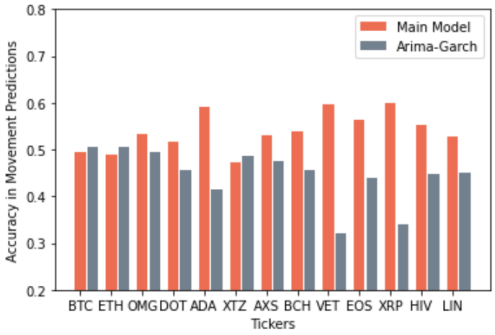




*Figure 6. Mean Square Error of the main and the extrapolation model by tickers (left)*

*Figure 7. Accuracy in making predictions of movements whether positive or negative (right)*

presented depending on which strategy to use.

### 8.2 Experiment Results

The experiment results are divided into two parts: Model Accuracies and Mock Trading Results. The first part examines the model performances in terms of the error in making predictions. The second part examines the effectiveness of the model when it comes to trade cryptocurrencies.

#### 8.2.1 Model Accuracies

Figure 6 represents the mean square error of the 13 main models and the ARIMA-GARCH model. The average mean squared error of the main model is 0.2, while the average mean squared error of the ARIMA-GARCH model is 0.25. Considering the average standard deviation of lagged returns in the training set was 0.3, the mean squared errors of the models are quite high. This indicates that the model is not making accurate predictions, which is reasonable since the 15 features selected do not represent the whole market mechanism, which is supposed to be extensively complex.

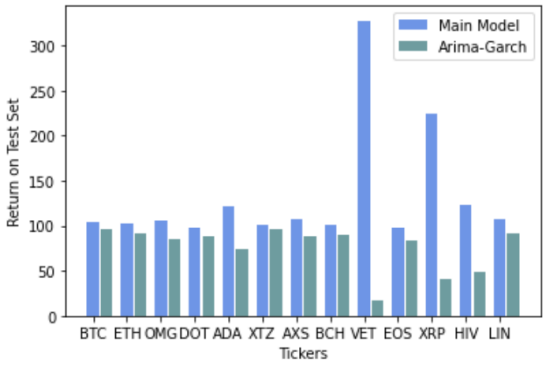
The model using a neural network has overperformed the ARIMA-GARCH model. Considering the fact that the extrapolation model is a univariate model, the prediction made by the model was decent. This implies that the influence of time is significant to a large extent.

Next, I have attempted to make a prediction resembling the binary classification task; if the model is predicted to return as positive, label 1, else 0. Then, I compared it with the actual data, which was samely labeled as either 1 or 0. Figure 7 represents the accuracy of the model whether the model well predicted the price movements in the next tick. The average accuracy of the 13 main models was 0.54 and the ARIMA-GARCH model had the value of 0.45.

The result of the accuracy in making predictions of price movements in the next tick is quite low. The ARIMA-GARCH

model even failed to correctly predict half of the actual movements.

One of the reasons for such a low accuracy is that there exists a lot of rows in the training dataset containing zero in the values of the target variable. This implies that



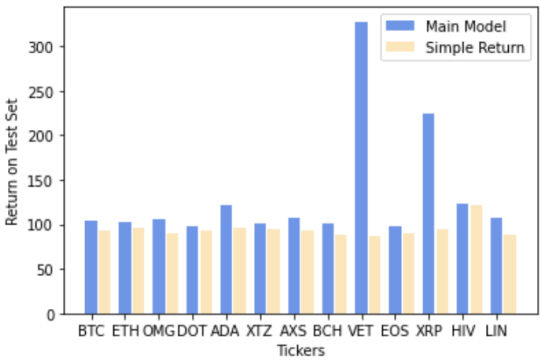




*Fig 8. Profit made by the main and the extrapolation model on the test set (5 days) by tickers (left)*
*Fig 9. Profit made by the main model and change in market price on the test set (5 days)by tickers (right)*

there is no price change between a current tick and a next tick. Although prices have fluctuated in the large range, if the closing price is at the same level as the closing price of the previous tick, the features do not reflect its fluctuations. Thus, this acts as a significant defect of the model. One of the aids to supplement such a disadvantage, the model could have a target variable to predict the range of returns in the next tick.

Another reason is the characteristics of the features. As discussed earlier in part 5, the features are not having meaningful information unless they are at the extreme values. Since the values are not at the extreme values most of the time, there will not be much for the model to learn when the values of features are not extreme. Also, the mean squared error of the main model is 0.2, which has no big difference from the standard deviation of the training dataset. Thus, any predicted values near 0 will have a higher likelihood of actual value with an opposite sign.

The reason for the ARIMA-GARCH model's low accuracy is that it does not count the effects of other variables apart from its returns.

The last reason that results in low accuracy is the range of predictions. The average standard deviation of the lagged returns of the test data (actual data) is 0.3. While the average standard deviations of predictions made by the main model and the extrapolation model are 0.08 and 0.04 accordingly. The models are passively making predictions due to the reason stated in the previous paragraphs, and eventually, increase the chance of mispredicting the price movements if positive or negative.

### 8.2.2 Mock Trading Results

Figure 8 represents the profit earned from the mock trading by using the main model and the ARIMA-GARCH model, and figure 9 represents the profit made by the main model compared to the simple price

change from the first tick to the last tick of the test dataset. Figure 10 is the data table of return from the main model, return from the ARIMA-GARCH model and the simple rate of change in the price of the tickers during the tested period.

During the tested period (5 days), the overall market was shrinking except for the ticker HIV. The average return from the main

| | BTC | ETH | OMG | DOT | ADA | XTZ | AXS | BCH | VET | EOS | XRP | HIV | LIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RET_MM | 103.82165 | 102.48442 | 106.11892 | 98.20000 | 121.09617 | 100.79667 | 107.46369 | 100.44949 | 327.77412 | 98.30014 | 224.24010 | 123.72185 | 107.05074 |
| RET_AG | 95.50546 | 90.83265 | 84.65833 | 88.74908 | 74.22856 | 95.61813 | 88.73584 | 90.41076 | 16.97787 | 84.30209 | 41.08497 | 48.92827 | 91.25211 |
| RET_TEST | 93.19645 | 95.72864 | 89.71014 | 93.79423 | 96.16162 | 95.00713 | 92.36553 | 88.54550 | 86.06965 | 90.27895 | 94.42509 | 120.93023 | 88.44221 |

*Fig 10. Data table of the profit made by the main model, the extrapolation model, and the change in market price on the test set (5 days) by tickers*

model was 132.42. This means that if the asset is equally allocated among the 13 tickers, after 5 days, the model is earning 32.4% of an initial investment. The result is impressive considering the fact that the market has shrunk 5.8% in the same period. While the average return from the ARIMA-GARCH model was 76.25, which means that there occur losses of 23.75% of an initial investment.

The positive results in the mock trading using the main model implies that the model had well mimicked the quantitative trading strategies. Since those strategies often make more profits in more volatile markets, the main model also shows strength in the corresponding circumstances.

One of the points to notice from the result is that there are significant differences in the performances between the main model and the ARIMA-GARCH model. Although the extrapolation model has modeled the volatility, there is a limit in predicting the volatility just using the volatility of the returns. Therefore, the multivariate volatility model, such as the vector ARMA model could be suggested, but this project could not implement it due to its computational overload. Thus, the discrepancies between the performances of those two models is due to the lack of technical indicators from the extrapolation model.

Meanwhile, there are some noticeable results from the tickers VET and XRP, which were able to earn profits of 327.77 and 224.24. While using the ARIMA-GARCH model led to significant losses, where the discrepancies between the profits were more dramatic than others. This will be further discussed in the next section.

## 9. Discussion

From the results of the experiment, there are a few points to discuss. The discussion is divided into three parts: a case study of VET, a case study of HIVE, and a discussion of the influence of volatility on this project.

### 9.1 Case Study: VET

From the results of the mock trading, it is noticed that the main model made a profit of 327.77 in five days, which is surprising. Hence, I have conducted a case analysis to find potential reasons behind this enormous profit.

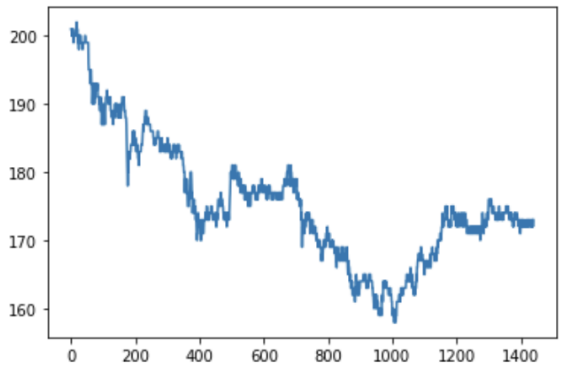



Fig 11. Movement of closing prices of VET on the test set (16-20 Nov)

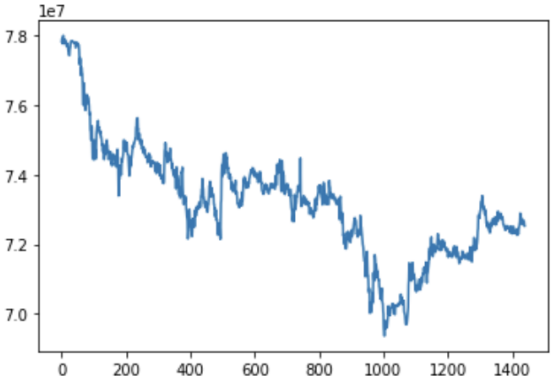



Fig 12. Movement of closing prices of BTC on the test set (16-20 Nov)

Figure 11 represents the time series data of VET's prices of the test dataset, while figure 12 represents the BTC's price time-series data of the test set. As noticed from the figures, the trends of VET and BTC resemble each other. However, there has been a huge difference in profit made through mock trading. The reason is the difference in the range of returns in the datasets.

- S.d. of VET's returns: 0.512 (Training), 0.516 (Test)
- S.d. of BTC's returns: 0.169 (Training), 0.207 (Test)

The more deviated data have enabled the main model fitted to the VET to learn a more variety of circumstances of the market. Thus, this implies that the model could have made an inference further away from the mean with more confidence. Another ticker in which the main model was able to make a high profit was XRP, and its standard deviation of the returns is also 0.39, which is relatively higher than the standard deviation of the BTC's returns. Higher standard deviation implies higher volatility, which led the model to exhibit the characteristics of quantitative trading strategies, where it performs better in more volatile markets. Therefore, the accuracy of the model and the performance in the mock trading can be viewed separately to some extent.

Furthermore, the discrepancy in the models' performances of the mock trading was more dramatic in VET than other tickers, which are 327.77 and 16.98. During the market being highly volatile, $\epsilon_t$ of The ARIMA-GARCH model will have an extensively wide range, since the GARCH model reflects the market volatility. Therefore, the model accuracy will significantly drop during this period, and as mentioned previously, the use of multiple indicators of volatility will be helpful in improving the model's performance.

### 9.2 Case Study: HIVE

The ticker HIVE ('HIV') was the only ticker that the price increased during the

tested period. There was an interesting point to discuss when I have attempted to conduct individual case studies of each ticker.

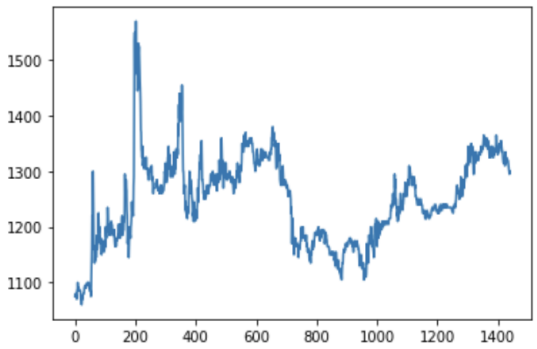



Fig 13. Movement of closing prices of HIVE on the test set (16-20 Nov)

Figure 13 represents the price movements of HIVE during the tested period. Obviously, the market trend was different from the BTC and VET movements. Looking at the first 400 ticks of the price movements of HIVE, there were significant increases in prices a few times. Nevertheless, the model could not predict its increases, as shown from the result of mock trading. While the market price was increased by 20.9 percent during the tested period, the model could only make 23 percent of profits.

The standard deviation of the training set of HIVE was 0.525, and 1.08 for the test set. The difference in the standard deviation implies that there potentially existed external risk, which may have influenced the market's volatility for the tested period. In fact, on November 16, which is the same date as the test data started, HIVE announced the record-breaking revenue during the 2nd quarter of 2021 [31]. The positive issue was promptly reflected in the market price, which is shown in tick 195 of the test data.

As shown in this case, external risks are unpredictable by the model. Nevertheless, considering the characteristics of financial instruments, the market is highly influenced by external issues. Hence, numerous researchers have attempted to use natural language processing techniques to conduct sentiment analysis to forecast market movements [27]. However, the model proposed in this project does not count the external factors. The only indicator that assumes the presence of external factors is volatility. As there exists a significant difference in the HIVE's standard deviations of returns in the training and test dataset. Therefore, as countermeasures, the volatility models can be implemented in order to evaluate the risks. Or, one can implement different trading strategies.

### 9.3 Influence of Volatility

Number of researchers have identified the importance of volatility in financial data analysis [4, 10]. As shown from the cases of VET and HIVE, volatility is an important factor in financial market prediction . As shown from the case of VET, more volatile market data lets models learn a more variety of circumstances, which eventually enable models to make inferences about the next return much less passively. On the other hand, using more volatile market data will increase the randomness as discussed in part 5. Also, in figure 6, a model trained with more volatile data had low accuracy. Therefore, the trade-off between profitability and model accuracy in the context of volatility is still presented even using the neural network. This result requires the auxiliary models to support better

decisions of the neural network. In addition, an approach to tackle the market prediction task has to focus on both maximization of profit and minimization of volatility and should find the equilibrium.

In addition, as shown from the case of HIVE, the presence of external risks must be considered in order to implement the model for practical uses. However, predicting external risks is almost impossible since occurrences of events are not continuous and are random. Therefore, in-depth analysis and understanding of an asset in the trading universe are crucial. Also, as mentioned above, the implementation of auxiliary models that counteract external risks will be helpful to resolve the corresponding problem.

## 10. Conclusion

To conclude, the features used in this project are not having useful information unless they are not at an extreme value. As result, the model accuracy was relatively low, but it performed well as first assumed. Thus, it can be concluded that the model has well mimicked quantitative trading strategies, and it has a competitiveness of using quantitative indexes as variables in financial market tasks. In future research, the uses of other features may influence the consequent returns.

As mentioned in the previous section, volatility is an important factor in financial market prediction. Nevertheless, the model proposed in this project could not focus on volatility, rather it focuses on maximizing profits by mimicking quantitative traders. Hence, the implementation of the aforementioned countermeasures is necessary in order to make it practical.

Despite the presence of the discipline of XAI (Explainable AI) [25], the explainability of the neural network is still lower than existing statistical models [25]. Hence, the consistency of inferences made by the model is not guaranteed.

Another disadvantage of the proposed main model is that the model only predicts the return in the next tick, which means that only the inference on the one time step away from the present is available. This drawback makes the model practical, since it has a high turnover value as aforementioned, and this requires the model to predict several time steps more to increase the profit. Although we can amend the target variables to predict several time steps away from the present, it is unlikely that the quantitative indexes have influences on those values. Hence, the need of using the extrapolation model comes into place. Predicting several consequent sequences of returns enables one to implement more various trading strategies and expect higher returns. In a few current pieces of research, it has attempted to implement deep time-series forecasting models to tackle the market prediction tasks [16, 17, 23, 29]. In future research, the extrapolation model could be incorporated into an interpolation model.

Finally, the model proposed in this project is definitely unready for practical uses. In practice, the implementation of several risk models, appropriate trading strategies, and AI operating systems must be considered. Nevertheless, with more research, the approach using quantitative

indexes as variables has competitiveness as a practical use for financial market prediction.

## References

### Journals


[1] Anghel, G. D. (2015). Stock market efficiency and the MACD. evidence from countries around the world. *Procedia Economics and Finance*, *32*, 1414–1431. https://doi.org/10.1016/s2212-5671(15)01518-x

[2] Awais, M., Bin Iqbal, M. T., & Bae, S.-H. (2021). Revisiting internal covariate shift for batch normalization. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(11), 5082–5092. https://doi.org/10.1109/tnnls.2020.3026784

[3] Chan, L. K., Jegadeesh, N., & Lakonishok, J. (1996). Momentum strategies. *The Journal of Finance*, *51*(5), 1681–1713. https://doi.org/10.1111/j.1540-6261.1996.tb05222.x

[4] Charandabi, S. E., & Kamyar, K. (2021). Prediction of cryptocurrency price index using Artificial Neural Networks: A survey of the literature. *European Journal of Business and Management Research*, *6*(6), 17–20. https://doi.org/10.24018/ejbmr.2021.6.6.1138

[5] Chong, T., Ng, W.-K., & Liew, V. (2014). Revisiting the performance of MACD and RSI Oscillators. *Journal of Risk and Financial Management*, *7*(1), 1–12. https://doi.org/10.3390/jrfm7010001

[6] Deng, S., & Sakurai, A. (2014). Short-term foreign exchange rate trading based on the support/resistance level of Ichimoku Kinkohyo. *2014 International Conference on Information Science, Electronics and Electrical Engineering*. https://doi.org/10.1109/infoseee.2014.6948127

[7] Dubey, A. K., & Jain, V. (2019). Comparative study of convolution neural network's relu and leaky-relu activation functions. *Lecture Notes in Electrical Engineering*, 873–880. https://doi.org/10.1007/978-981-13-6772-4_76

[8] Duchi, J. C., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.

[9] Fang, J., Jacobsen, B., & Qin, Y. (2017). Popularity versus profitability: Evidence from Bollinger Bands. *The Journal of Portfolio Management*, *43*(4), 152–159. https://doi.org/10.3905/jpm.2017.43.4.152

[10] Ghashami, F., Kamyar, K., & Riazi, S. A. (2021). Prediction of stock market index using a hybrid technique of artificial neural networks and particle swarm optimization. *Applied Economics and Finance*, *8*(3), 1. https://doi.org/10.11114/aef.v8i3.5195

[11] Hamori, S., Kawai, M., Kume, T., Murakami, Y., & Watanabe, C. (2018). Ensemble learning or Deep learning? application to default risk analysis. *Journal of Risk and Financial Management*, *11*(1), 12. https://doi.org/10.3390/jrfm11010012

[12] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *06*(02), 107–116. https://doi.org/10.1142/s0218488598000094

[13] Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T.-Y. (2018). Listening to chaotic whispers. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. https://doi.org/10.1145/3159652.3159690

[14] Karpoff, J. M. (1987). The relation between price changes and Trading Volume: A survey. *The Journal of Financial and Quantitative Analysis*, *22*(1), 109. https://doi.org/10.2307/2330874

[15] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional Neural Networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

[16] Liu, S., Zhang, C., & Ma, J. (2017). CNN-LSTM neural network model for quantitative strategy analysis in stock markets. *Neural Information Processing*, 198–206. https://doi.org/10.1007/978-3-319-70096-0_21

[17] Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., & Pintelas, P. (2021). An advanced CNN-LSTM model for cryptocurrency forecasting. *Electronics*, *10*(3), 287. https://doi.org/10.3390/electronics10030287

[18] Ma, A. (J., & Orland, W. (2017). Using pivot table to test market anomaly. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2992371

[19] Malkiel, B. G., & Fama, E. F. (1970). Efficient Capital Markets: A review of theory and empirical work*. *The Journal of Finance*, *25*(2), 383–417. https://doi.org/10.1111/j.1540-6261.1970.tb00518.x

[20] Markus, L., & Weerasinghe, A. (1988). Stochastic oscillators. *Journal of Differential Equations*, *71*(2), 288–314. https://doi.org/10.1016/0022-0396(88)90029-0

[21] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

[22] Renault, E., & Werker, B. J. M. (2011). Causality effects in return volatility measures with Random Times. *Journal of Econometrics*, *160*(1), 272–279. https://doi.org/10.1016/j.jeconom.2010.03.036

[23] Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00333-6

[24] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

[25] Tjoa, E., & Guan, C. (2021). A survey on Explainable Artificial Intelligence (XAI): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(11), 4793–4813. https://doi.org/10.1109/tnnls.2020.3027314

[26] Xiaohua Wang, Paul Kang, Hoh Phua, & Weidong Lin. (n.d.). Stock market prediction using neural networks: Does trading volume help in short-term prediction? *Proceedings of the International Joint Conference on Neural Networks, 2003.* https://doi.org/10.1109/ijcnn.2003.1223946

[27] Xu, Y., & Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://doi.org/10.18653/v1/p18-1183

[28] Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method.

[29] Zhengyang, W., Xingzhou, L., Jinjin, R., & Jiaqing, K. (2019). Prediction of cryptocurrency price dynamics with multiple machine learning techniques. *Proceedings of the 2019 4th International Conference on Machine Learning Technologies*. https://doi.org/10.1145/3340997.3341008

## Websites

[30] Best, R. de. (2021, November 3). *Number of crypto coins 2013-2021*. Statista. Retrieved December 8, 2021, from https://www.statista.com/statistics/863917/number-crypto-coins-tokens/.

[31] Newsfile Corp. (2021, November 16). *Retransmission: Hive announces record-breaking revenue of $52.6 million and earnings for our 2nd quarter ended September 30, 2021*. Yahoo! Finance. Retrieved December 8, 2021, from https://finance.yahoo.com/news/retransmission-hive-announces-record-breaking-110000376.html.

[32] Ossinger, J. (2021, November 9). *Cryptocurrency now worth more than $3 trillion*. Time. Retrieved December 8, 2021, from https://time.com/6115300/cryptocurrency-value-3-trillion/.

# Appendices

## Appendix A.1. Raw Data of BTC (First 10 rows)

KRW-BTC_5

| | open | high | low | close | volume | value |
|---|---|---|---|---|---|---|
| 2021-10-12 11:55:00 | 69743000 | 69891000 | 69620000 | 69869000 | 55.87300728 | 3897258900.58806 |
| 2021-10-12 12:00:00 | 69888000 | 69900000 | 69603000 | 69617000 | 44.92208238 | 3135569206.2689 |
| 2021-10-12 12:05:00 | 69617000 | 69664000 | 69500000 | 69502000 | 91.09691026 | 6337187330.81641 |
| 2021-10-12 12:10:00 | 69501000 | 69502000 | 69300000 | 69390000 | 93.69227287 | 6502758634.48731 |
| 2021-10-12 12:15:00 | 69390000 | 69710000 | 69377000 | 69701000 | 62.43465872 | 4339778956.08619 |
| 2021-10-12 12:20:00 | 69696000 | 69722000 | 69580000 | 69696000 | 68.90097344 | 4799224586.61311 |
| 2021-10-12 12:25:00 | 69700000 | 69860000 | 69696000 | 69703000 | 38.45926997 | 2684693232.96451 |
| 2021-10-12 12:30:00 | 69704000 | 69788000 | 69675000 | 69732000 | 21.42191475 | 1493677258.59027 |
| 2021-10-12 12:35:00 | 69764000 | 69862000 | 69732000 | 69862000 | 23.94450638 | 1672159443.19024 |
| 2021-10-12 12:40:00 | 69862000 | 69863000 | 69852000 | 69852000 | 20.87994699 | 1458693776.54053 |

## Appendix A.2. Preprocessed Data of BTC (First 50 rows)

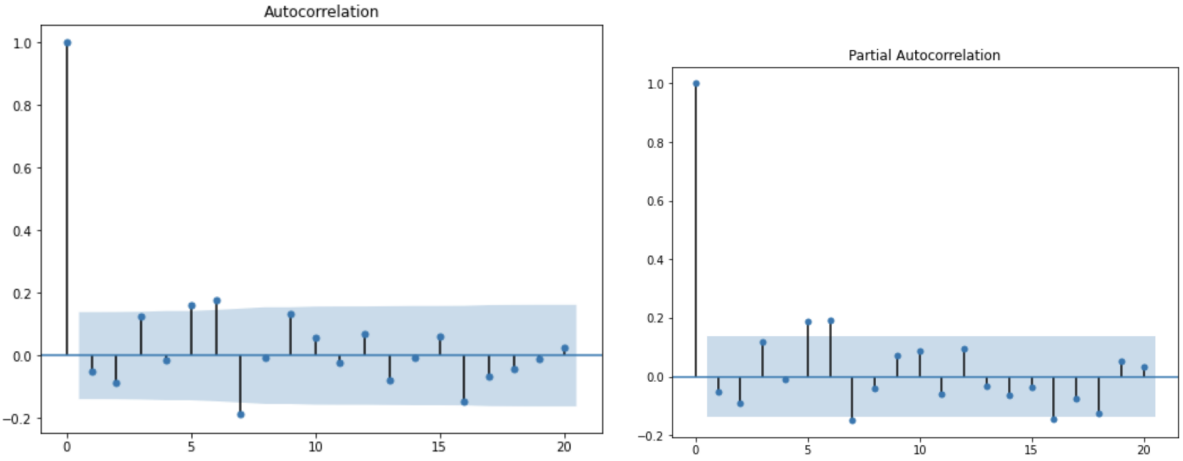
BTC_train

| | index | bb_width | d_to_ubb | d_to_lbb | MACD | MACD_Oscillator | RSI | d_to_sup2 | d_to_res2 | fast_k | slow_k | Momentum | ten_to_sen | cloud_thickness | d_to_cloud | volume | return_next_tick | close |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 77 | 1124675.637190737 | 1048137.818595369 | 76537.81859503686 | -128927.92605170608 | -97062.06435538275 | 28.327747814438666 | 169333.33333332837 | 278666.66666668653 | 4.273504273504273 | 15.763430985362476 | -0.008548630983743822 | -112000 | 364000 | 175000 | 80.91863416 | 0.30365671109844340 | 70145000 |
| 1 | 78 | 1120267.904665098 | 782333.9522332549 | 337933.9522332549 | -138766.93644848466 | -85520.85932953443 | 38.09162989995328 | 372000 | 328000 | 37.042682926829265 | 14.365149388910146 | -0.004736222653496935 | -112000 | 344000 | 368000 | 30.81047753 | 0.29705221863171 | 70358000 |
| 2 | 79 | 1067171.880355835 | 521185.9401779175 | 545985.9401779175 | -128253.92680075765 | -60006.27948039124 | 45.882101741632056 | 360666.66666667163 | 193333.33333331347 | 97.20430107526882 | 33.761761816353329 | 0.0008916607371555342 | -116500 | 360750 | 593750 | 43.25233002 | -0.05668371901880187 | 70567000 |
| 3 | 80 | 969511.1253730953 | 477855.56268654764 | 491655.56268654764 | -121740.48873969913 | -42794.27298424422 | 44.722161628811406 | 50666.666666671634 | 69333.33333331347 | 88.60215053763444 | 46.325654117065733 | 0.0018824501661319324 | -116500 | 360750 | 553750 | 25.86326437 | -0.21552029719115184 | 70527000 |
| 4 | 81 | 818697.6107072532 | 501848.8053536266 | 316848.8053536266 | -127354.63801816106 | -38726.73770068618 | 40.52913274174608 | 169333.33333332837 | 250666.66666668653 | 55.91397849462365 | 56.607323461572706 | -0.0018460445402222913 | -179500 | 351750 | 354250 | 40.5432816 | -0.015630550621670297 | 70375000 |
| 5 | 82 | 739031.9193226993 | 436715.95966134965 | 302315.95966134965 | -131177.5170979418 | -34039.693343272476 | 40.23513478724604 | 96666.66666667163 | 113333.33333331347 | 49.76744186046512 | 65.706110978964623 | -0.0020853708761831347 | -202500 | 273500 | 163500 | 21.70553732 | 0.09379796486839176 | 70364000 |
| 6 | 83 | 583951.4439479411 | 260675.72197397053 | 323275.72197397053 | -127420.00291398168 | -24225.743287737525 | 42.91100667404106 | 205333.33333332837 | 172666.66666668653 | 45.84837545126354 | 67.467249483851087 | -0.0007973893279027 | -298000 | 226000 | 134500 | 35.16228098 | -0.11926735766009111 | 70430000 |
| 7 | 84 | 568384.2205957472 | 321392.1102978736 | 246992.11029787362 | -129715.61602194607 | -21217.08508585191 | 40.4299181045728 | 318666.66666667163 | 245333.33333331347 | 46.55581947743468 | 57.337553164284826 | -0.0013345880121274276 | -298000 | 258500 | 83000 | 96.43870518 | -0.11940977454297785 | 70346000 |
| 8 | 85 | 540117.518076688 | 363458.759038344 | 176658.75903834403 | -136728.14905504882 | -22583.694469013615 | 38.060032396517047 | 272666.66666667163 | 247333.33333331347 | 34.81308411214953 | 46.579739878192728 | -0.0034149836202858703 | -298000 | 310500 | 26000 | 46.47360471 | -0.008539466567979165 | 70262000 |
| 9 | 86 | 503558.8898585439 | 358779.4449292719 | 144779.44492927194 | -141141.85316364467 | -21597.91884208063 | 37.889189665558894 | 90666.66666667163 | 99333.33333331347 | 36.9509439276486 | 42.787125058815552 | -0.0030830446745874094 | -298000 | 368750 | 78250 | 24.50447249 | 0.06262810293782284 | 70256000 |
| 10 | 87 | 414571.26448094845 | 285785.6322404742 | 128785.63224047422 | -139484.8646350506 | -15952.744238931147 | 40.015622250222105 | 213333.33333332837 | 192666.66666668653 | 48.320413436692505 | 42.49771937401 | -0.0013271361276347093 | -298000 | 391500 | 145000 | 35.41562970000004 | -0.06827880512091378 | 70300000 |
| 11 | 88 | 444492.01967588067 | 338146.00983794034 | 106346.00983794034 | -140422.13560251892 | -13512.012157144927 | 38.46836887325105 | 204666.66666667163 | 185333.33333331347 | 43.573667711598745 | 42.042777826212804 | -0.0003434924705903231 | -302500 | 382750 | 88250 | 39.81840249 | 0.4071058475203504 | 70252000 |
| 12 | 89 | 422361.61441746354 | 38180.80720873177 | 384180.80720873177 | -116761.95140604675 | 8118.537627261128 | 50.700087736600864 | 475333.33333332837 | 262666.66666668653 | 94.44444444444444 | 51.620502819529999 | 0.0035875132951551873 | -306500 | 384000 | 375500 | 27.86475369 | -0.04678329411097470 5 | 70538000 |
| 13 | 90 | 408033.9855126405 | 61816.992756322026 | 346216.992756322 | -99524.428397879 | 20284.84850092657 | 49.47791376006098 | 96000 | 98000 | 86.28841607565012 | 61.915569212230 1 | 0.0025397684795333 2 | -306500 | 387500 | 346000 | 24.93307472 | 0.37302189844690 1 | 70505000 |
| 14 | 91 | 656245.0761719942 | -37777.461914002895 | 694022.5380859971 | -63921.39955089986 | 44710.30186475311 | 58.138802524775 54 | 409333.33333332837 | 292666.66666668653 | 87.70949720670392 | 72.067287775017 92 | 0.0064026497276390 42 | -160000 | 338250 | 501750 | 37.4567264 | 0.19641645941668173 | 70768000 |
| 15 | 92 | 910416.1685734689 | 4608.084286734462 | 905808.0842867345 | -24218.723041325808 | 67530.3826830631 | 61.860178708149256 | 292666.66666667163 | 181333.33333331347 | 95.6359102244389 | 81.530387132567 2 | 0.0064730466802871 2 | -93500 | 337750 | 637250 | 57.14839018 | -0.24398155330220828 | 70907000 |
| 16 | 93 | 973749.3175008297 | 239674.6587504148 5 | 734074.6587504148 | -6628.205717161298 | 68096.71999255315 | 55.27406373946684 | 125333.33333332837 | 242666.66666668653 | 72.19251336898395 | 87.254156264044 24 | 0.0044963095547485 78 | -11500 | 309000 | 435500 | 34.14677213 | -0.13289224418242 54 | 70734000 |
| 17 | 94 | 969176.1449808776 | 360788.0724904388 | 608388.0724904388 | -264.95371167361736 | 59567.97758917 5 | 52.03246698728908 | 121333.33333332837 | 164666.66666668653 | 37.60330578512397 | 75.885928532180 15 | 0.0015370208461271352 | 8500 | 307000 | 339500 | 21.93119079 | 0.00141562853908094 87 | 70640000 |
| 18 | 95 | 909164.109376192 1 | 367682.0546809605 | 541482.054668096 | 4803.208680123091 | 51708.91197834836 | 52.06467397562718 | 139333.33333332837 | 96666.66666668653 | 31.12128146453089 | 64.852501609956 3 | -0.0008384580933949448 | 8500 | 300000 | 333500 | 26.93577768 | 0.06370238246911075 | 70641000 |
| 19 | 96 | 816491.0287320912 | 319345.5143660456 | 497145.5143660456 | 12307.055712029338 | 47370.20720349203 | 53.57527327441147 | 194666.66666667163 | 109333.33333331347 | 35.353535353535536 | 54.381309239322 58 | 0.0013975093537877737 | 13000 | 299000 | 377500 | 30.00175384 | 0.07073536485300291 | 70686000 |
| 20 | 97 | 714310.9811404049 | 261855.49057020247 | 452455.49057020247 | 22032.477452114224 | 45676.50315122698 | 55.2622530493035 | 110666.66666667163 | 119333.33333331347 | 51.630434782608695 | 45.580214150956536 | 0.0018930828833740422 | 40000 | 299000 | 427500 | 28.11578774 | -0.10885546256502732 | 70736000 |
| 21 | 98 | 461287.8590304535 | 253043.92951522768 | 208243.92951522768 | 23261.31427963078 | 37524.27198060613 | 52.12116549099154 | 76000 | 118000 | 44.488189763779 5 | 40.039349272435533 | 0.0002253760324820164 3 | 172500 | 276000 | 327500 | 24.41291609 | -0.16275350626625 | 70659000 |
| 22 | 99 | 458022.8038961589 3 | 367011.4019480794 7 | 91011.40194807947 | 14789.052539945291 | 23241.608191159008 | 47.755350839892 87 | 92666.66666667163 | 179333.33333331347 | 1.1583011583011582 | 32.750348347070777 | -0.0011161102776159892 | 196000 | 280500 | 217000 | 20.65861618 | 0.16301882513041 21 | 70544000 |
| 23 | 100 | 388346.6950491071 | 232573.34752455354 | 155773.34752455354 | 17152.930448129773 | 20484.388879034148 | 52.07818995202205 | 161333.33333332837 | 88666.66666668653 | 45.559845559845556 | 35.638061166133 7 | 0.0009509938761737178 | 214000 | 230500 | 282000 | 15.22500134 | -0.07925388131730937 | 70659000 |
| 24 | 101 | 391093.9642705917 4 | 273446.98213529587 | 117646.98213529587 | 14343.856818303466 | 14140.252198905406 | 49.91242751209268 4 | 222000 | 178000 | 40.483836858080604 | 36.664030832586 75 | -0.0003733509502226955 | 164000 | 168750 | 164250 | 24.20071888 | 0.19262637564976 26 | 70603000 |
| 25 | 102 | 251109.1838667392 7 | 50654.59193336964 | 200454.59193336964 | 22824.920400381 09 | 18097.052624314 56 | 54.82581307712819 | 314000 | 208000 | 84.112149532710 28 | 43.160373782607 07 | 0.0022708329678436 7 | 107000 | -250 | 131250 | 25.12574477 | 0.15691485602000732 | 70739000 |
| 26 | 103 | 336771.8647525310 5 | -5914.067623734474 | 342685.9323726555 | 38061.428280159 83 | 26666.848402718395 | 58.411655859827 53 | 152000 | 76000 | 100 | 54.262735987331475 | 0.0020667927854524093 | 132000 | -18000 | 224500 | 35.63389715 | -0.02964008468595569 2 | 70850000 |
| 27 | 104 | 382855.7024374604 | 57027.85121873021 | 325827.8512187302 | 47890.317383274436 | 29196.590004179183 | 57.482017782220545 | 160000 | 276000 | 64.74820143884892 | 66.980716043441 02 | 0.0021022684945153159 | 178000 | -18000 | 203500 | 69.77394457 | 0.16236287396405036 | 70829000 |
| 28 | 105 | 485518.0966989994 | 23659.048349499702 | 461859.0483494997 | 64216.553522422 91 | 36418.260914175946 | 61.130302899461395 | 262666.66666667163 | 137333.33333331347 | 85.431654676259 | 74.955077866723 71 | 0.0037438283936661 15 | 178000 | -20250 | 316250 | 31.34411685 | -0.16914749661 70503 | 70944000 |
| 29 | 106 | 497102.8956754207 6 | 163251.44783771038 | 333851.4473771038 | 66705.589662671 09 | 31125.837643206876 | 55.75426513709825 | 102666.66666667163 | 171333.33333331347 | 59.475806451612 9 | 78.753562198861 5 | 0.0010380127647759418 | 178000 | -20250 | 196250 | 31.51142809 | 0.08754094657177358 | 70824000 |
| 30 | 107 | 530727.1322335899 | 133063.56611679494 | 397663.56611679494 | 72840.218358693393 | 29808.373073132109 | 57.818335974569 45 | 138000 | 104000 | 51.903114186851205 | 72.311755307143 4 | 0.0022985241992807158 | 178000 | -51750 | 226750 | 21.27985226 | -0.19891092740456084 | 70886000 |
| 31 | 108 | 514342.0390436683 | 274471.01962183416 | 239871.01962183416 | 65570.933835700 15 | 18031.270838346 8 | 51.889802093375 65 | 102666.66666667163 | 205333.33333331347 | 0 | 52.311755307143 36 | -0.0012101335460690588 | 178000 | -63250 | 74250 | 19.43551804 | -0.12439041628383718 | 70745000 |
| 32 | 109 | 444150.9252965506 | 338675.46264977753 | 105475.46264977753 | 52109.798772769268 | 3656.1086202513106 | 48.544250290457256 | 212000 | 260000 | 21.866666666666667 | 43.735443899627789 | -0.0022195641093409103 | 178000 | -111000 | -61500 | 22.31803481 | 0.09623957994253641 | 70657000 |
| 33 | 110 | 421125.6872294247 | 265762.84361471236 | 155362.84361471236 | 46393.007283532045 | -1648.545533689517 5 | 51.164440595721 68 | 303333.33333332837 | 156666.66666668653 | 50.675675675675676 | 36.784252596161224 | -0.0013893110328143 | 193500 | -124500 | -20500 | 46.85500261 | -0.0749381406857541 | 70725000 |
| 34 | 111 | 376153.3960504023 | 303176.6983027011 | 72976.69830270112 | 37158.129967689514 | -8706.739041025889 | 49.067229705554 71 | 82666.66666667163 | 93333.33333331347 | 40.85510688836104 | 33.060112683516 85 | -0.0013032464187991981 | 180000 | -245000 | -313500 | 13.02051292 | 0.10329409101199616 | 70672000 |
| 35 | 112 | 374863.7678470055 | 230131.88392350078 | 144731.88392350078 | 35321.833826534494 | -8434.428144272759 | 51.986458516872 93 | 137333.33333332837 | 74666.66666668653 | 61.403508771929383 | 34.960191600526 58 | 0.0005315170011960691 | 180000 | -245000 | -240500 | 16.61547841 | -0.15407449289702724 | 70745000 |
| 36 | 113 | 407913.0626330375 7 | 334256.5313165188 | 73656.53131651878 | 24786.698570773005 | -15175.650721441765 | 47.599514931321 74 | 84666.66666667163 | 157333.33333331347 | 43.729903536977496 | 43.706172307922074 | -0.0015937539972139092 | 142500 | -245000 | -349500 | 16.10919614 | -0.19678350982501 59 | 70636000 |
| 37 | 114 | 518399.0740732555 06 | 495299.53703662753 | 23099.537036627 53 | 5163.3704857233019 | -27839.183045143523 | 42.656104642290664 | 140666.66666667163 | 209333.33333331347 | 12.413793103448276 | 41.815597595784 | -0.0029739838513187 4 | 93000 | -247250 | -490750 | 35.62588141 | 0.35604351958238 3 | 70497000 |
| 38 | 115 | 428052.956225425 | 179526.4781127125 | 248526.4781127125 | 9750.474596272013 | -18601.66314524896 | 52.291388468560065 | 358000 | 182000 | 98.9655172413793 | 51.473565908419 13 | 0.0011964426834207 84 | -50500 | -249250 | -241750 | 16.42398307 | -0.13003901170350884 | 70748000 |
| 39 | 116 | 402966.9961671829 | 242183.49808359146 | 160783.49808359146 | 5895.041659936309 | -17965.676867647 13 | 49.038892523854 8 | 94666.66666667163 | 139333.33333331347 | 67.241379310344 83 | 56.750820392815 89 | -0.0006396125294216848 | -63000 | -249250 | -333750 | 18.34808284 | 0.13020833333332593 | 70656000 |
| 40 | 117 | 316179.69574278593 | 92989.84787139297 | 223189.84787139297 | 10145.482384681702 | -10972.188914311326 | 52.238172233746 64 | 264000 | 132000 | 98.9655172413793 | 64.263222086570578 | 0.0015502862699318552 | -107000 | -157000 | -130500 | 30.98553961 | -0.06784644088878578 | 70748000 |
| 41 | 118 | 305407.2690868815 | 131103.63453434408 | 174303.63453434408 | 9531.290354773402 | -9269.104755368895 | 50.458318866541 85 | 140000 | 120000 | 82.413793103448827 | 71.999999999999 4 | -0.0008802221975301228 | -137000 | -132250 | -138750 | 24.35278079 | -0.05940590459405946 | 70700000 |
| 42 | 119 | 305285.2946198284 6 | 173142.64730991423 | 132142.64730991423 | 5591.342145621777 | -10567.242371610211 | 48.888688555863706 | 190666.66666667163 | 137333.33333331347 | 65.555555555555556 | 82.628352490421 4 | -0.0005356965493616529 | -137000 | -111250 | -97750 | 28.89582397 | 0.10189928953550087 | 70658000 |
| 43 | 120 | 306701.4473030269 | 102350.72365151346 | 204350.72365151346 | 8183.879719227552 | -6379.763838400551 | 51.664552841270584 | 185333.33333332837 | 100666.66666668653 | 90.654205607476644 | 80.966090163640 87 | 0.0008219136541346094 | -137000 | -107750 | -22250 | 20.4057028 | 0.23610914746217126 | 70730000 |
| 44 | 121 | 411662.2132012844 | 10331.106600642204 | 401331.1066006422 | 23442.706182837486 | 7103.250100164 84 | 57.438429558462 89 | 249333.33333332837 | 164666.66666668653 | 90.024937655586036 | 85.522801832743 97 | 0.0027043689417822 98 | -44000 | -107750 | 144750 | 34.38439225 | 0.07475633665574111 | 70897000 |
| 45 | 122 | 518085.3640515804 | 31042.682025790215 | 487042.6820257902 | 39358.075763627 89 | 18414.895744758658 | 59.107924109532 32 | 256000 | 128000 | 100 | 85.729698384468 11 | 0.0039508600115867 76 | -37500 | -110000 | 195500 | 35.35980244 | -0.03523608174771354 | 70950000 |
| 46 | 123 | 560068.0910984576 | 105934.04554922879 | 454134.04554922879 | 49384.670249084738 | 22753.19391277705 | 57.953159926065 | 147333.33333332837 | 108666.66666668653 | 91.745283018867924 | 87.595996367552 03 | 0.0028450165012303243 | -22500 | -96500 | 184000 | 31.76466998 | -0.14099400775466897 | 70925000 |
| 47 | 124 | 435553.4920580649 6 | 176476.74602930248 | 259076.74602930248 | 48700.755151002264 | 17655.418443142 15 | 53.454563246571475 | 156000 | 202000 | 63.013698630136986 | 87.087624982468 53 | 0.0002487956866157548 | 5000 | -30750 | 149750 | 20.1686458 | 0.32897987985880 1 | 70825000 |
| 48 | 125 | 551468.5041877925 | 32434.252093889627 | 519034.2520938962 7 | 66195.750794976 95 | 28120.334149689144 | 61.042615800145335 | 369333.33333332837 | 254666.66666668653 | 82.412060301507 54 | 85.439159521274 5 | 0.0036542790039988 85 | 37500 | -19000 | 394500 | 48.60427145 | -0.12384249486334653 | 71058000 |
| 49 | 126 | 533564.7831124067 | 142882.39155620337 | 390682.39155620337 | 72128.703514054 42 | 27242.629495009 94 | 57.246711959148556 | 196666.66666667163 | 185333.33333331347 | 57.2972972972973 | 78.893667849561 9 | 0.0026770931970774513 | 37500 | -10000 | 315500 | 46.05075287 | -0.04931661226532385 | 70970000 |
| 50 | 127 | 524801.084009707 | 192200.5420048535 | 332600.5420048535 | 73163.153069183 23 | 22621.663240108777 | 55.761506221300344 | 244000 | 186000 | 46.089385474860336 | 68.111544944533996 | 0.0014659954904904 | 37500 | -35000 | 255500 | 48.81719199 | -0.33128920843025 21 | 70935000 |

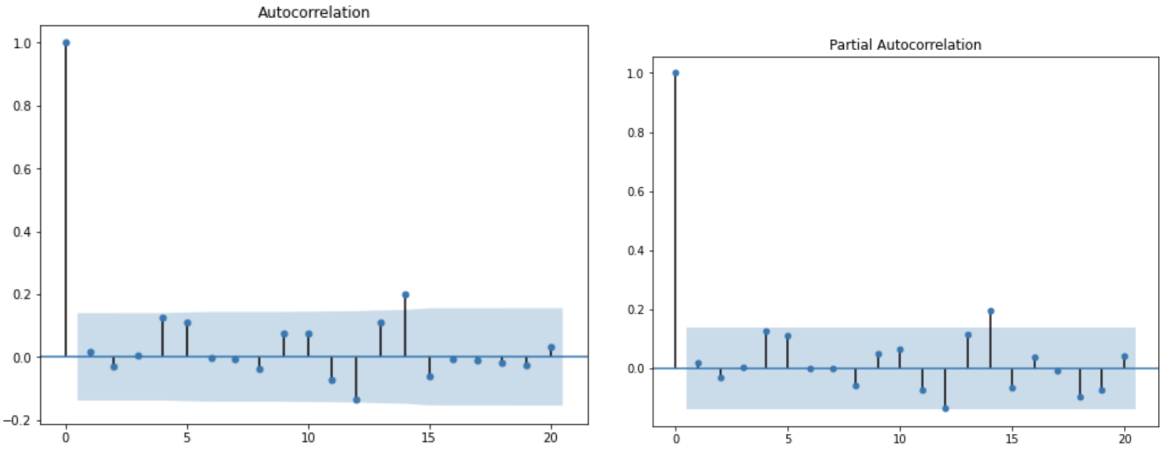
Appendix B. 13 Selected Tickers

- BTC: Bitcoin
- ETH: Etherium
- XRP: Ripple
- ADA: Cardano
- XTZ: Tezos
- BCH: Bitcoin Cash
- HIV: Hive
- VET: Vetchain
- DOT: Polkadot
- EOS: Eos
- LINK: Chain Link
- OMG: Omisego
- AXS: Axie Infinity

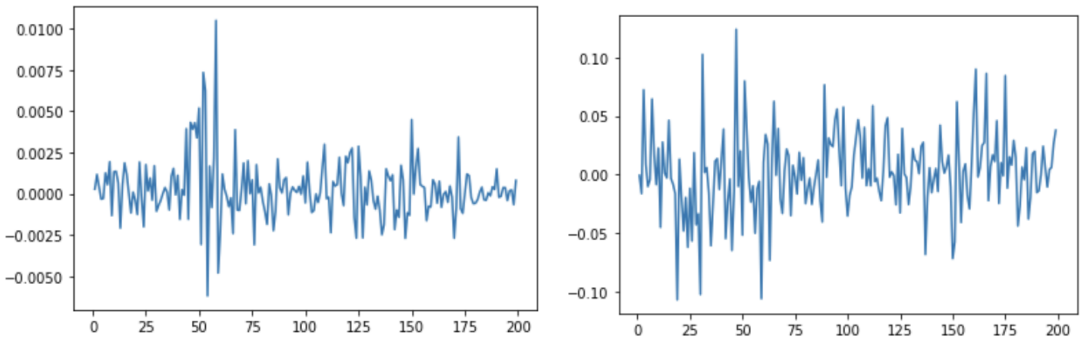
Appendix C.1. Autocorrelation and Partial Autocorrelation Plot of 5-minute candlestick of BTC




Appendix C.2. Autocorrelation and Partial Autocorrelation Plot of daily candlestick of BTC




Appendix C.3. Plots of lagged returns of 5-minute (left) and daily (right) candlestick



Appendix C.4. ARIMA(5,1) model results (left) fitted to 5-min candlestick of BTC and plot of the plot of the ARIMA model (right)

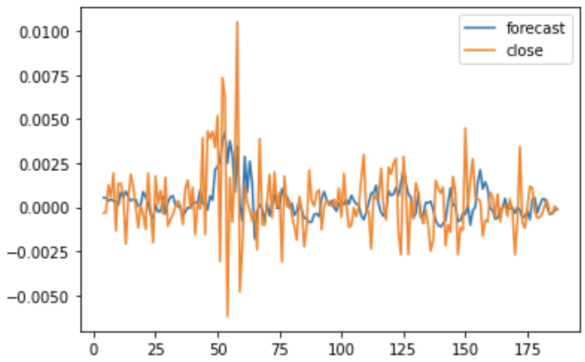
ARIMA Model Results

| | | | |
|---|---|---|---|
| Dep. Variable: | D.close | No. Observations: | 186 |
| Model: | ARIMA(5, 1, 0) | Log Likelihood | 900.776 |
| Method: | css-mle | S.D. of innovations | 0.002 |
| Date: | Sat, 04 Dec 2021 | AIC | -1789.552 |
| Time: | 15:17:04 | BIC | -1770.197 |
| Sample: | 1 | HQIC | -1781.709 |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| ar.L1.D.close | -0.9777 | 0.070 | -13.909 | 0.000 | -1.115 | -0.840 |
| ar.L2.D.close | -0.9639 | 0.091 | -10.644 | 0.000 | -1.141 | -0.786 |
| ar.L3.D.close | -0.7285 | 0.101 | -7.189 | 0.000 | -0.927 | -0.530 |
| ar.L4.D.close | -0.5821 | 0.090 | -6.484 | 0.000 | -0.758 | -0.406 |
| ar.L5.D.close | -0.2751 | 0.069 | -3.963 | 0.000 | -0.411 | -0.139 |

Roots

| | Real | Imaginary | Modulus | Frequency |
|---|---|---|---|---|
| AR.1 | 0.4457 | -1.1374j | 1.2216 | -0.1906 |
| AR.2 | 0.4457 | +1.1374j | 1.2216 | 0.1906 |
| AR.3 | -0.7314 | -1.0209j | 1.2559 | -0.3489 |
| AR.4 | -0.7314 | +1.0209j | 1.2559 | 0.3489 |
| AR.5 | -1.5444 | -0.0000j | 1.5444 | -0.5000 |




Appendix C.5. ARIMA(4,2) model results (left) fitted to daily candlestick of BTC and plot of the plot of the ARIMA model (right)

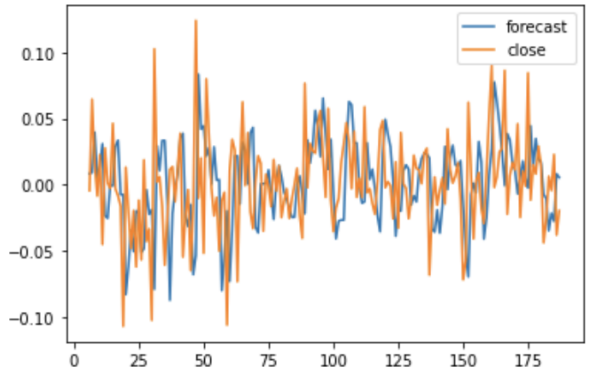
ARIMA Model Results

| | | | |
|---|---|---|---|
| Dep. Variable: | D2.close | No. Observations: | 185 |
| Model: | ARIMA(4, 2, 0) | Log Likelihood | 299.605 |
| Method: | css-mle | S.D. of innovations | 0.048 |
| Date: | Sat, 04 Dec 2021 | AIC | -589.209 |
| Time: | 15:33:36 | BIC | -573.108 |
| Sample: | 2 | HQIC | -582.684 |

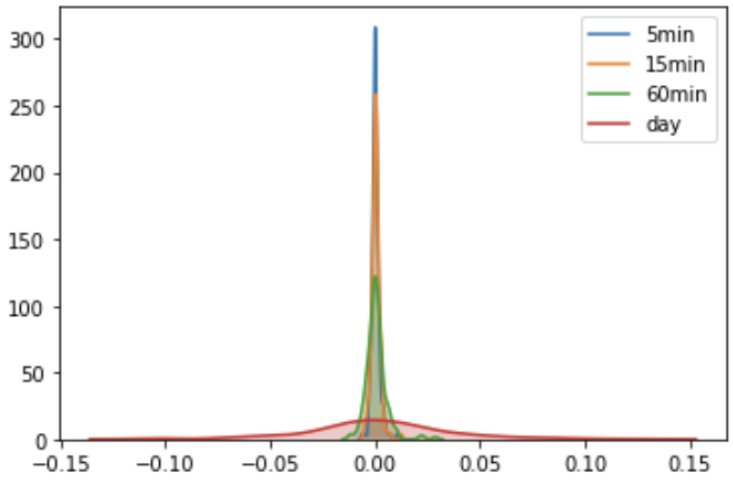
| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| ar.L1.D2.close | -1.3430 | 0.067 | -20.133 | 0.000 | -1.474 | -1.212 |
| ar.L2.D2.close | -1.2549 | 0.099 | -12.730 | 0.000 | -1.448 | -1.062 |
| ar.L3.D2.close | -0.9204 | 0.098 | -9.370 | 0.000 | -1.113 | -0.728 |
| ar.L4.D2.close | -0.4143 | 0.067 | -6.196 | 0.000 | -0.545 | -0.283 |

Roots

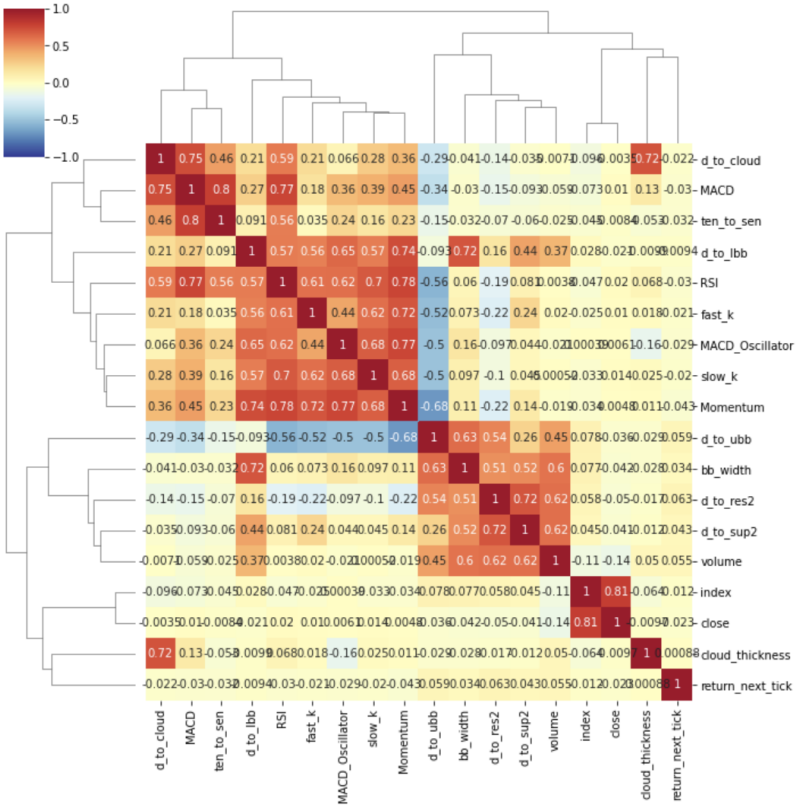
| | Real | Imaginary | Modulus | Frequency |
|---|---|---|---|---|
| AR.1 | 0.0185 | -1.2091j | 1.2092 | -0.2476 |
| AR.2 | 0.0185 | +1.2091j | 1.2092 | 0.2476 |
| AR.3 | -1.1294 | -0.6125j | 1.2848 | -0.4209 |
| AR.4 | -1.1294 | +0.6125j | 1.2848 | 0.4209 |



Appendix D. Kernel Distribution Estimation Plot of 5-min, 15-min, 60-min, and daily candlesticks of BTC




Appendix E. Correlogram of ETH training data



Appendix F.1. Augmented Dickey Fuller Test conducted on every feature.

bb_width
ADF test statistic: -11.702151781592292
p-value: 1.5594143116151221e-21

d_to_ubb
ADF test statistic: -13.816213837086067
p-value: 8.06002342306174e-26

d_to_lbb
ADF test statistic: -15.131619992654539
p-value: 7.218119895933034e-28

MACD
ADF test statistic: -14.527195952330016
p-value: 5.404273092620692e-27

MACD_Oscillator
ADF test statistic: -21.399891996671936
p-value: 0.0

RSI
ADF test statistic: -22.326770311490044
p-value: 0.0

d_to_sup2
ADF test statistic: -13.696770184999181
p-value: 1.312438713837769e-25

d_to_res2
ADF test statistic: -12.084466823625526
p-value: 2.193794918581059e-22

fast_k
ADF test statistic: -30.31166258712774
p-value: 0.0

slow_k
ADF test statistic: -15.72659293366242
p-value: 1.3045472324015956e-28

Momentum
ADF test statistic: -17.043330850451564
p-value: 8.166829146866593e-30

ten_to_sen
ADF test statistic: -16.873502902493346
p-value: 1.0765234401493718e-29

cloud_thickness
ADF test statistic: -15.136478138931684
p-value: 7.110115215828448e-28

d_to_cloud
ADF test statistic: -12.743165198529095
p-value: 8.854924107310275e-24

volume
ADF test statistic: -14.54214581683456
p-value: 5.125183044100752e-27

return_next_tick
ADF test statistic: -26.040259363363948
p-value: 0.0

Appendix F.2. Summary of VAR model fitted to the variable (d_to_ubb) and the target variable (return_next_tick)

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 08, Dec, 2021
Time:                     22:18:58
--------------------------------------------------------------------
No. of Equations:         2.00000    BIC:                    2.33031
Nobs:                     11911.0    HQIC:                   2.32784
Log likelihood:          -47652.0    FPE:                    10.2430
AIC:                      2.32659    Det(Omega_mle):         10.2378
--------------------------------------------------------------------
Results for equation d_to_ubb
================================================================================================
                         coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------------------------
const                      3.607594         0.133191           27.086           0.000
L1.d_to_ubb                0.915300         0.001996          458.619           0.000
L1.return_next_tick      -72.141246         0.339447         -212.526           0.000
================================================================================================

Results for equation return_next_tick
================================================================================================
                         coefficient       std. error           t-stat            prob
------------------------------------------------------------------------------------------------
const                     -0.020380         0.003595           -5.668           0.000
L1.d_to_ubb                0.000423         0.000054            7.856           0.000
L1.return_next_tick       -0.053206         0.009163           -5.807           0.000
================================================================================================

Correlation matrix of residuals
                  d_to_ubb  return_next_tick
d_to_ubb          1.000000          0.019224
return_next_tick  0.019224          1.000000
```