2021 Spring Semester

Master's Thesis

EAP

Does Artificial Intelligence Dream of Bitcoin Trading?: Forecasting Bitcoin Returns with Machine Learning Techniques

Student ID: 3219B0140
Name: Hayashi Yuji

September 2021

Graduate School of Economics
Waseda University

Abstract

The paper investigated whether it is possible to make high profits in the Bitcoin market by forecasting future Bitcoin returns with the variables of the Bitcoin blockchain and the other cryptocurrencies (called Altcoins) market. First, we collected both the blockchain data and the historical prices and trading volumes of 15 major cryptocurrencies, checking their time-series properties. Then, we constructed the experiment to forecast the Bitcoin returns and evaluate the performances of the forecasting models. Finally, we simulated the trading strategies based on the forecasting models, calculated the cumulative returns, and conducted the Wilcoxon Signed-Rank test to confirm that the excess returns are statistically significant. From the experiments, we found two things. At first, the Bitcoin market was efficient in a weak form between 2019 and 2021. The Ljung & Box test clarified that Bitcoin returns did not have autocorrelation at the 0.05 significance level. However, the Bitcoin market was still inefficient enough to make higher profits than the benchmarks by forecasting future returns with the publicly available data from the Bitcoin blockchain and the Altcoins market. The trading strategies based on the forecasting models yielded higher cumulative returns than the benchmarks in trading simulations. Besides, the Wilcoxon Signed-Rank test clarified that our trading strategies achieved higher daily returns than the benchmarks at the 0.01 significance level. Second, the variables of the Bitcoin blockchain and the Altcoins market were essential to forecast Bitcoin returns in End of Day, two, and three days ahead. The variables selection algorithm chose the variables mainly from the categories of the Bitcoin blockchain and the Altcoins market rather than macroeconomic variables in a statistically significant way. Furthermore, the models using the variables of the Bitcoin blockchain and the Altcoins market outperformed those not using them. Our economic implication is that investors need to consider the internal factors of the Bitcoin blockchain and the external factors of the cryptocurrency market, rather than relying solely on the prices to forecast Bitcoin returns.

Table of Contents

1. Introduction

In 2008, after the collapse of Lehman Brothers, Bitcoin, the oldest and most popular cryptocurrency today, was invented by somebody known as the pseudonym of Satoshi Nakamoto (Nakamoto, 2008). Since Bitcoin trading started in 2009, it has captured much attention from investors worldwide. As a result, the market has been bearish and bullish, reaching a market capitalization of 1 trillion USD as of February 2021 (Bloomberg, 2021). In this situation, investors worldwide have wanted to make high profits in the Bitcoin market by forecasting future returns. Like other financial assets, the demand and supply determine Bitcoin prices. Therefore, capturing the hidden patterns of its demand and supply is essential to forecast future Bitcoin returns.

However, Bitcoin may have the unique determinants of its demand and supply: the mechanics of the Bitcoin blockchain and the relationship between Bitcoin and the other cryptocurrencies (Bloomenthal, 2021). Bitcoin works on the blockchain, a distributed ledger that enables the transfer of value in the digital world without a centralized trusted authority. The blockchain holds publicly available information regarding the demand and supply of Bitcoin, such as transaction costs or its network value. Besides, since the Bitcoin system appeared, alternative coins (called "Altcoins") that are all the cryptocurrencies other than Bitcoin have been generated based on the Bitcoin blockchain whose source codes are open sources. As of April 2021, the number of Altcoins was more than 9000, and the total market capitalization of cryptocurrencies excluding Bitcoin reached 1 trillion USD (CoinMarketCap, 2021). To trade Altcoins, investors also need to trade Bitcoin in cryptocurrency exchanges. Thus, the mechanics of the Bitcoin blockchain and the relationship between Bitcoin and Altcoins could be potential factors to determine Bitcoin prices. The purpose of this paper is to investigate whether it is possible to make high profits in the Bitcoin market by forecasting future Bitcoin returns with the variables of the Bitcoin blockchain and the Altcoins market.

From the experiments, we found two things. The first finding is that the Bitcoin market was still inefficient enough to make higher profits than the benchmarks by forecasting future returns with the publicly available data from the Bitcoin blockchain and the Altcoins market. The second finding is that the variables of the Bitcoin blockchain and the Altcoins market played an essential role in forecasting Bitcoin returns in End of Day, two, and three days ahead.

The remainder of the paper is organized as follows. Section 2 introduces the literature review. Section 3 describes the knowledge of the determinants of Bitcoin prices, the data contents and sources, the problems the data have, and how to treat them. Section 4 explains the methods of the experiments, from the return forecasting to the trading simulation. Section 5 illustrates the experimental results. Finally, Section 6 concludes the findings, the limitations, and the economic implications, and the Appendix shows the tables.

2. Literature Review

This section reviews the works of literature investigating the predictability of the Bitcoin market. Two main research perspectives are the efficient market hypothesis (the EMH) and the empirical forecasting approaches. The former provides theoretical viewpoints on financial markets, while the latter addresses data-driven experiments.

2.1 Efficient Market Hypothesis in the Bitcoin Market

The efficient market hypothesis states three forms of market efficiency: the weak-form, the semi-strong form, and the strong form (Fama, 1970). The weak form says that it is impossible to obtain higher returns than the market average by forecasting the future prices, but it is possible with other publicly available information. The semi-strong form states that it is impossible to achieve higher returns by forecasting the future with even other publicly available data, but it is possible with private information such as insider information. Finally, the strong form says no one can make higher returns even when they use private information; this is ultimately efficient because the information spreads quickly in the market.

A paper conducted a meta-analysis systematically on the efficiency and profitability of the Bitcoin market (Kyriazis, 2019). The previous research covered by Kyriazis (2019) employed various methods, including the Augmented Dickey-Fuller test (the ADF test), the Ljung & Box test, and Detrended Fluctuation Analysis, to test the efficiency. Kyriazis (2019) found that most previous research showed that the EMH did not hold before 2018. This result implied that the investors could make higher returns by forecasting future returns with the Bitcoin prices. However, it also mentioned that market efficiency might have become weaker as time has passed.

2.2 Empirical Forecasting Approaches

In data science, some research set up the forecasting task as a binary classification problem for return in subsequent days and evaluate their accuracy scores. For example, Mcnally (2016) focused on the data between 2013 and 2016 and employed Long-Short-Term-Memory (LSTM), Recurrent Neural Network (RNN), and Autoregressive Integrated Moving Average (ARIMA) models. In addition, it standardized the variables and used automated tools in variables selection and hyperparameter selection, achieving the highest forecasting accuracy, 52%, with LSTM.

Other research has expanded the forecasting models and the independent variables following Mcnally (2016). Mallqui (2019) analyzed the data between 2013 and 2017 and used broader macroeconomic variables, such as S&P500 futures prices. The achieved accuracy score here was 62%, showing ensemble models achieved high performances. Mudassir (2020) investigated the data between 2013 and 2019 and constructed the forecasting models: Feedforward Neural Network (FNN), Support Vector Machine (SVM), Stacked Feedforward Neural Network (SANN), LSTM. Every model achieved about 65% accuracy scores in End of

Day forecasting, even though the independent variables came from only the Bitcoin blockchain. Chen (2020) also achieved a 65% accuracy score with the Logistic Regression model, using sentiment variables from Google Trends and Baidu and macroeconomic variables like the gold prices in the data between 2017 and 2019.

Besides, some research simulated trading strategies based on forecasting models. Concretely, Shintate (2019) simulated trading strategies, relying on the forecasting models that performed around 55-62% accuracy scores minute by minute. The paper concluded that it could not outperform the Buy & Hold strategy in the data between 2013 and 2017. On the other hand, Amjad (2016), dealing with the data between 2014 and 2016, could achieve higher cumulative returns by forecasting Bitcoin returns every 5-second. Their cumulative returns were 6-7x, 4-6x, 3-6x in 2014, 2015, and 2016, respectively. The paper showed that the more profitable models were Random Forest and Logistic Regression. One interesting point in the paper is that the Partial Autocorrelation Function test implied that the Bitcoin market between 2014 and 2016 was more inefficient than a weak-form efficiency.

2.3 The Differences Between the Previous Literature and Our Paper

The existing research above still had some gaps to be filled. First, the data did not include the latest bull market situation starting January 2021. Therefore, they could not assess the efficiency of the Bitcoin market these days. Second, to the best of our knowledge, few existing studies have used the variables of both the Bitcoin blockchain and the Altcoins market as independent variables, even though the mechanics of the Bitcoin blockchain and the effects of the Altcoins market may be essential factors for the demand and supply of Bitcoin. Third, almost no research simulated the daily trading strategies based on binary returns forecasting. For filling those gaps, the paper conducted: collecting the latest data that include both bear and bull market situations between 2019 and 2021, using the variables of the Bitcoin blockchain and the Altcoins market, and simulating trading strategies based on the daily Bitcoin binary returns forecasting.

3. Data

This section explains the determinants of Bitcoin prices as the data generation process, the data sources, data preprocessing, and summary statistics of the variables. Besides, it tests the time-series properties of the variables.

3.1 Determinants of Bitcoin Prices

3.1.1 The Bitcoin Blockchain

The Bitcoin blockchain holds publicly available information regarding the demand and supply of Bitcoin, such as transaction costs or its network values. That information may give us knowledge of Bitcoin prices. Here introduces how the Bitcoin blockchain works.

The Bitcoin blockchain consists of the following key concepts: a user with a wallet containing his or her private key, transactions that propagate through the Bitcoin network, and miners who maintain the blockchain, the distributed ledger that holds all the transactions (Antonopoulos, 2017).

3.1.1.1 Transaction

A transaction is data that propagates the transfer of value between wallets into the Bitcoin network. Like a bookkeeping system, each transaction contains "inputs" and "outputs." The value transferred beneficiary can use each output as an input in the next transaction. In other words, the output of a transaction expresses the value of Bitcoin. The new value that a user can use in the next transaction is generated by mining as the unused-transaction-output (UTXO).

3.1.1.2 Wallet

When a user transfers an output received in a previous transaction as an input in the next transaction, she needs to create a digital signature to confirm her authenticity.

Wallets allow a beneficiary to hold the private key to generate this digital signature. Most software wallets have several other functions, too. For instance, it has a database containing UTXO that can be used for the next transaction. The wallet creates inputs by combining the appropriate UTXO to generate a transaction. Then, the user can create a digital signature with the private key to transfer the value.

3.1.1.3 Peer to Peer Network (P2P)

The Bitcoin network is a peer-to-peer network that connects an individual wallet with several other wallets on the Bitcoin network without a central server, enabling rapid data exchanges regarding transactions. Originators' wallets can send new transactions to any Bitcoin node with an Internet connection. Once a node in the Bitcoin network receives valid transactions that it has never seen before, the Bitcoin network constantly shares newly generated transactions to broadcast the fact of value transfer across the globe. Then, digital signatures and

cryptographic proof confirm the integrity of the transactions and blocks, as explained later. Therefore, no one can practically tamper with the finalized transactions on the network because it contradicts the previous integrity if someone changes them.

3.1.1.4 Mining

Mining is a mechanism to finalize and perpetuate transactions. It bundles multiple transactions into a data structure called a block. Its purpose is to provide a unique record of the transfer of value and improve processing efficiency. The data structure of a block requires a certain amount of computation to generate, but once it is generated, any user can easily verify its integrity. The process of creating a block is called mining, analogous to mining gold, and in Bitcoin, it requires particular computations described below. The Bitcoin network participants who perform mining are called miners.

Because mining is essential for maintaining the blockchain, successful miners are paid a certain fee and a commission for processing transactions as an incentive to perform the computations. Miners invest in computing power to earn these rewards. Each miner is free to choose which transactions to include in a block and perform the computations for mining, and a transaction creator can pay a miner any amount of fees. This mechanism allows market forces to work and includes transactions with higher fees in a block faster.

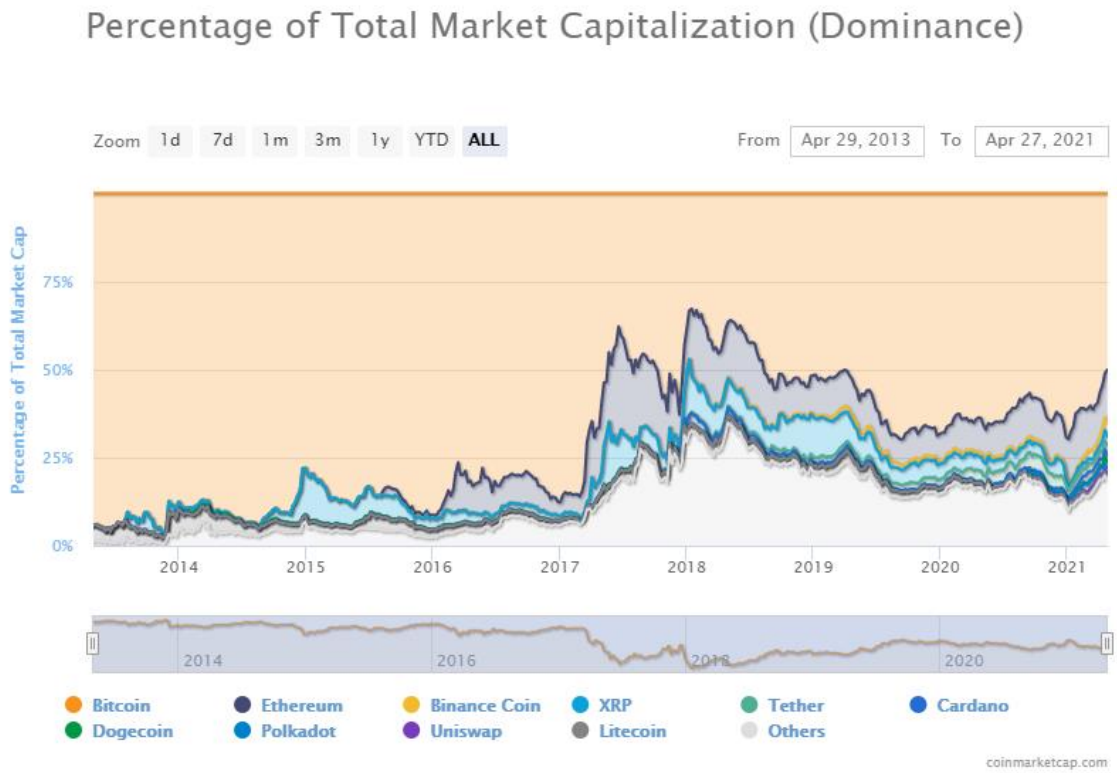
Bitcoin mining requires cryptographically intensive computations called Proof of Work (PoW). Depending on the sum of all mining computers' processing power, the mechanism requires 10 minutes of computation on average. Miners worldwide perform these computational processes. The first successful miner (i.e., the one who got the correct computation results) broadcasts the block with the rest of the world via P2P, which updates the ledger information recognized by all Bitcoin nodes worldwide. Miners then start mining a new subsequent block.

3.1.2 The Relationship Between Bitcoin and Altcoins

It is common for most investors to use cryptocurrency exchanges such as CoinBase or Binance to trade between fiat currencies, Bitcoin, and Altcoins, although cryptocurrency blockchains are maintained in a decentralized manner. These cryptocurrency exchanges deal with massive trading every day, facilitating cryptocurrency trading and liquidating the market.

Figure 1 shows that the market share of Bitcoin in the cryptocurrency market has changed over time. In the beginning, Bitcoin was overwhelmingly dominant, but as the number of Altcoins increases and gains popularity, the market capitalization of Altcoins has increased.

Figure 1: Bitcoin Dominance Rate in Total Market Cap



Global cryptocurrency market charts. Retrieved 04 27, 2021, from CoinMarketCap: https://coinmarketcap.com/charts/

In cryptocurrency trading, the relationship between the demand for Bitcoin and that for Altcoins seems complicated. Bitcoin is a crucial currency in the cryptocurrency market like USD in the traditional financial market. Many Altcoins are denominated in Bitcoin. If the demand for Altcoins increases due to their higher profitability, it may increase the demand for Bitcoin because investors need to buy Bitcoin as an intermediary to exchange fiat currencies like USD or JPY for Altcoins in cryptocurrency exchanges. However, in terms of competition, if the demand for Altcoins increases due to their higher profitability, it may decrease the demand for Bitcoin because investors exchange Bitcoin for Altcoins.

In either case, we can assume that the Altcoins market affects the demand and supply of Bitcoin. Furthermore, the effects on the demand and supply of Bitcoin may determine its prices. According to the articles published by COINTELEGRAPH (Megas, 2020; Vidal, 2020), practitioners have found that Bitcoin prices correlate with those of Altcoins.

3.2 Raw Data

3.2.1 Data sources and contents

We scraped the Bitcoin blockchain data and the adjusted close prices and trading volumes of 15 major Altcoins. The paper also addressed macroeconomic variables, such as

S&P500, following the variable choices by Jang (2017). The blockchain data came from Blockchain.com (Blockchain.com, 2021). The adjusted close prices and trading volumes of Altcoins and the macroeconomic variables came from Yahoo! Finance (Yahoo! Finance, 2021). Due to the data availability, we collected the prices and trading volumes of 15 major Altcoins that investors had traded for more than three years from the top 30 Altcoins chosen by the watchlist of Yahoo! Finance, "Top Cryptos by Market Cap," as of February 2021 (Yahoo! Finance, 2021).

The observation units of the data were daily, and their number was 1266 days between September 05, 2017, and February 21, 2021. Table A1 in Appendix shows all the definitions and the categories of the variables. The total number of independent variables was 79, with 27 variables from the Bitcoin blockchain, 30 variables from the Altcoins market, and 22 variables from the macroeconomics.

3.2.2 Summary statistics

Figure 2 visualizes the bar plot, histogram, and time-series plot of Bitcoin prices, and Table 1 shows the summary statistics of the raw data variables chosen by the variables selection algorithm Boruta explained later.

Figure 2: Visualization of Bitcoin prices

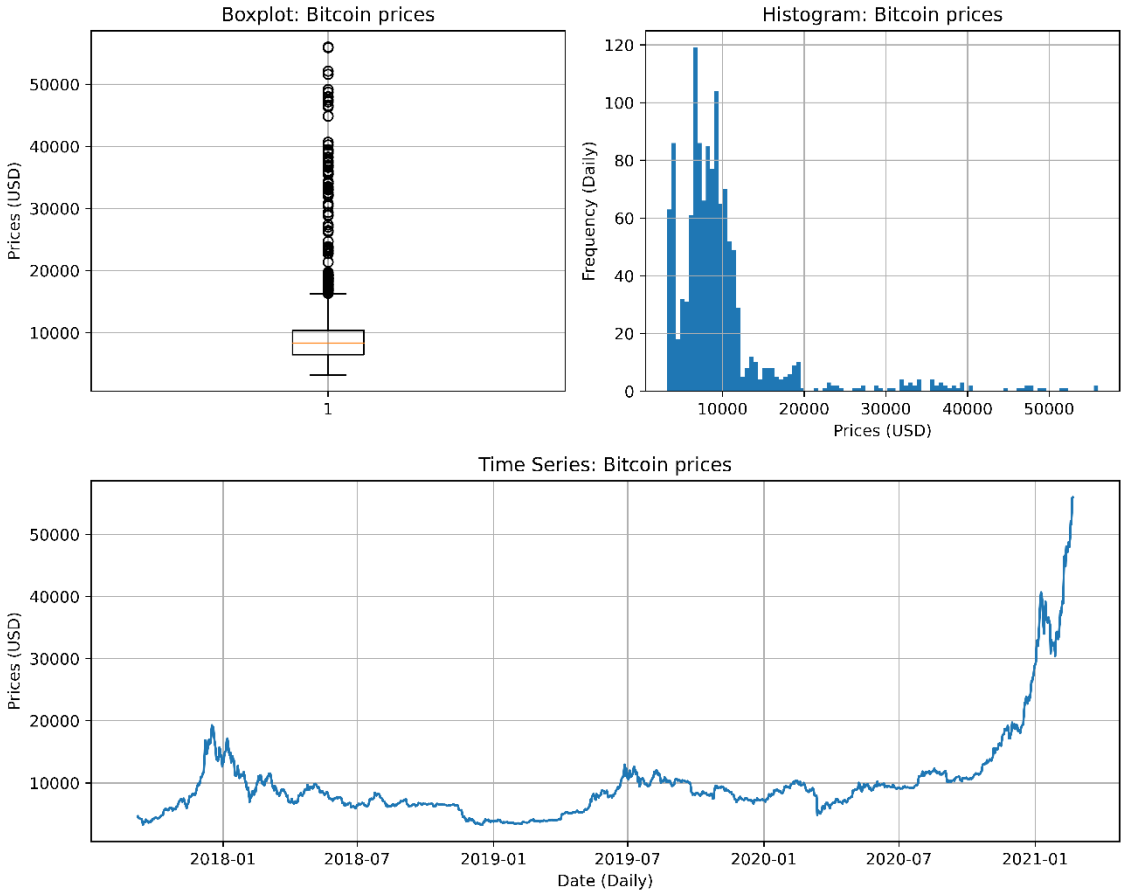


Table 1: Summary statistics of the raw data variables

Variables	count	mean	std	min	50%	max	skewness
BCH_Price	1259	522.788	505.552	77.366	318.306	3923.070	2.613
BCH_Volume	1259	1.86E+09	2.12E+09	5.93E+07	1.32E+09	2.96E+10	4.172
BNB_Price	1259	18.59	20.19	0.68	15.58	332.62	8.263
BTC_Price	1266	9.84E+03	7.12E+03	3.22E+03	8.32E+03	5.60E+04	3.295
EOS_Price	1259	4.669	3.236	0.493	3.540	21.543	1.833
ETC_Price	1259	10.386	7.754	3.472	7.022	44.048	1.939
ETH_Price	1259	377.170	316.769	84.308	251.864	1960.165	2.268
IOTA_Price	1259	0.67	0.81	0.11	0.32	5.37	2.821
IOTA_Volume	1259	4.78E+07	1.21E+08	2.90E+06	1.74E+07	2.13E+09	8.733
LTC_Price	1259	83.453	53.773	23.464	61.142	358.336	1.885
NEO_Price	1259	25.778	28.808	5.377	15.124	187.405	2.543
USDT_Price	1259	1.00	0.01	0.97	1.00	1.08	1.628
USDT_Volume	1259	2.36E+10	2.86E+10	8.54E+07	1.62E+10	1.84E+11	2.120
WAVES_Price	1259	3.305	2.764	0.533	2.509	16.027	1.850
XEM_Price	1259	0.169	0.228	0.031	0.090	1.843	3.708
XLM_Price	1259	0.152	0.125	0.011	0.102	0.896	1.737
XMR_Price	1259	114.337	77.307	33.010	90.047	469.198	1.958
XRP_Price	1259	0.41	0.33	0.14	0.30	3.38	4.270
XRP_Volume	1259	1.99E+09	3.23E+09	2.06E+07	1.15E+09	3.50E+10	4.639
ZEC_Price	1259	131.527	128.361	24.504	70.971	880.761	2.110
cost-per-transaction	1266	56.26	29.10	18.00	50.19	209.43	1.367
gbp_usd_Price	881	1.306	0.048	1.149	1.305	1.434	0.140
gold_Price	861	1.47E+03	2.41E+02	1.18E+03	1.35E+03	2.05E+03	0.728
market-cap	1265	1.76E+11	1.34E+11	5.57E+10	1.47E+11	1.07E+12	3.356
mempool-count	1266	1.62E+04	2.34E+04	0.00E+00	6.72E+03	1.72E+05	2.775
mempool-size	1266	1.95E+07	2.96E+07	0.00E+00	5.38E+06	1.37E+08	1.977
mvrv	1264	1.76	0.71	0.69	1.62	4.47	1.181
nvts	1264	14.129	4.458	3.844	13.604	24.345	0.248
utxo-count	1265	5.91E+07	7.42E+06	4.89E+07	5.94E+07	7.32E+07	0.065

The variables, including 3 or 4 capital letters such as BCH, BNB, and ETH, are Altcoins variables. The units of cryptocurrencies are often abbreviated, as Bitcoin is BTC. The adjusted close prices are denominated in USD here. The variables in this table are the variables that were chosen by the variables selection algorithm Boruta at least once as the independent variables in the experiments.

As Table 1 and Figure 2 indicate, the raw data had some problems. First, they had many missing values. For example, macroeconomic variables, such as the adjusted close prices of gold, had missing values every Saturday and Sunday because traditional financial markets close at the end of every week.

Second, some variables had high skewness. As the histogram and the time-series plot in Figure 2 demonstrate, the skewness also implied that the distributions of the variables changed as time passes. Different distributions over train and test data cause an extrapolation problem in

the forecasting task because machine learning models assume that the distributions of variables in the training periods are similar to those of the test period variables.

Third, the scales among the variables were too different. Some variables took small values, while others took huge ones. The Neural Network model does not handle these differences in scale between variables.

Fourth, it was difficult to use Bitcoin prices directly as the dependent variable. The extrapolation problem explained above could occur, too. Besides, the forecasting results of the price variable were hard to interpret when implementing trading strategies based on forecasting.

3.3 Data Preprocessing

For the independent variables, to fill in missing values, we used a forward fill imputation method. The method imputes the previous values of missing values into them. This method was employed because investors cannot access information about the future in the real world. Other methods like backward fill imputation or linear interpolation need future data, causing a data leakage problem. Next, we took the log differences of all the independent variables by a day to alleviate an extrapolation problem. Then, as explained later, we standardized the scale difference in the independent variables in each cross-validation trial. If we standardize the scales of each variable for the whole period, it may cause a data leakage problem.

Table A2 in the Appendix shows the summary statistics of the preprocessed independent variables. The preprocessed variables had no missing values, and their distributions were less skewed than those of raw data. Investors also can understand the meaning of log return variables as the change of recent patterns.

For the dependent variable, we took the log difference of Bitcoin prices by one, two, and three days to convert prices into returns in End of Day, two, and three days ahead. Then, 0 was assigned to negative Bitcoin returns while 1 to positive ones to set up the forecasting task as a binary classification problem. These transformations facilitated dealing with the skewness and interpreting the meaning of the variable.

Table 2 shows the summary statistics of the preprocessed dependent variable, the binary log return of Bitcoin prices. The mean of the binary log return of Bitcoin prices was 0.538, implying the number of days when Bitcoin return is positive is more than that of days when Bitcoin return is negative. Besides, the skewness became lower than that of the original time series. Furthermore, investors can easily interpret the binary log return of Bitcoin prices as price movement directions, utilizing forecasting results quickly to construct trading strategies, deciding to buy or not to buy.

Table 2: Summary statistics of the preprocessed dependent variable

Variables	count	mean	std	min	50%	max	skewness
BTC_Price_LogReturn_binary	1263	0.538	0.499	0.000	1.000	1.000	-0.151

The data range is between September 08, 2017 and February 21, 2021.

3.4 Time Series Analysis

3.4.1 Stationarity

Stationarity is a critical time-series property because stationarity may ensure that the forecasting is interpolative. We conducted the augmented Dickey-Fuller test (the ADF test) on the preprocessed variables to test their stationarity. The ADF test has the null hypothesis that the univariate time series in question is non-stationary (MacKinnon, 2010).

Table A3 in the Appendix shows the results of the ADF test on the raw data variables. The results indicated that most variables were non-stationary even at the 0.1 significance level. Therefore, we took the log differences of all the variables in the preprocessing as explained in Subsection 3.3 Data Preprocessing.

Table A4 in the Appendix shows the results of the ADF test on the preprocessed variables. The result was that all the variables used for forecasting, including the dependent variables, were tested as stationary at the 0.05 significance level. Their stationarity assures that the distribution of train data is almost the same as that of test data.

3.4.2 Autocorrelation

We also conducted the Ljung & Box test on the dependent variable, the log return of Bitcoin prices (Tsay, 2010). The null hypothesis is that all the autocorrelation coefficients are zero between a specific time point and its lagged points within the predefined lags. The economic interpretation is that the test measures the efficiency of the Bitcoin market in a weak form because if the market has autocorrelation, it means investors can forecast the returns by using only the past prices, contradicting a weak-form efficiency.

Table 3 illustrates the Ljung & Box test results on the log return of Bitcoin prices.

Table 3: The results of the Ljung & Box test on the log return of Bitcoin prices

lags	test statistic	p value
20	30.917	0.056*
40	46.42	0.225
60	66.231	0.271

The number of data points is 1263. * indicates the 0.1 significance level. The column "lags" means the predefined rags tested.

Table 3 shows that we could not reject the null hypothesis at the 0.05 significance level. Thus, the log return of Bitcoin prices did not have autocorrelation at the 0.05 significance level. Compared with the periods before 2018 explained in Section 2 Literature Review, the market has been more efficient between 2019 and 2021. Thus, using only its past prices could not improve the forecasting ability of the models, meaning any technical analyses could not make sense. This fact justifies our hypothesis that other data like Altcoins information are vital for accurate forecasting in the latest periods.

4. Methods

The previous section collected and preprocessed the data. Then, we checked their time-series properties. The above analysis showed that the Bitcoin market between 2019 and 2021 was efficient, at least in a weak form. Therefore, this section constructed the experiment to test whether the market was efficient in a semi-strong form or not.

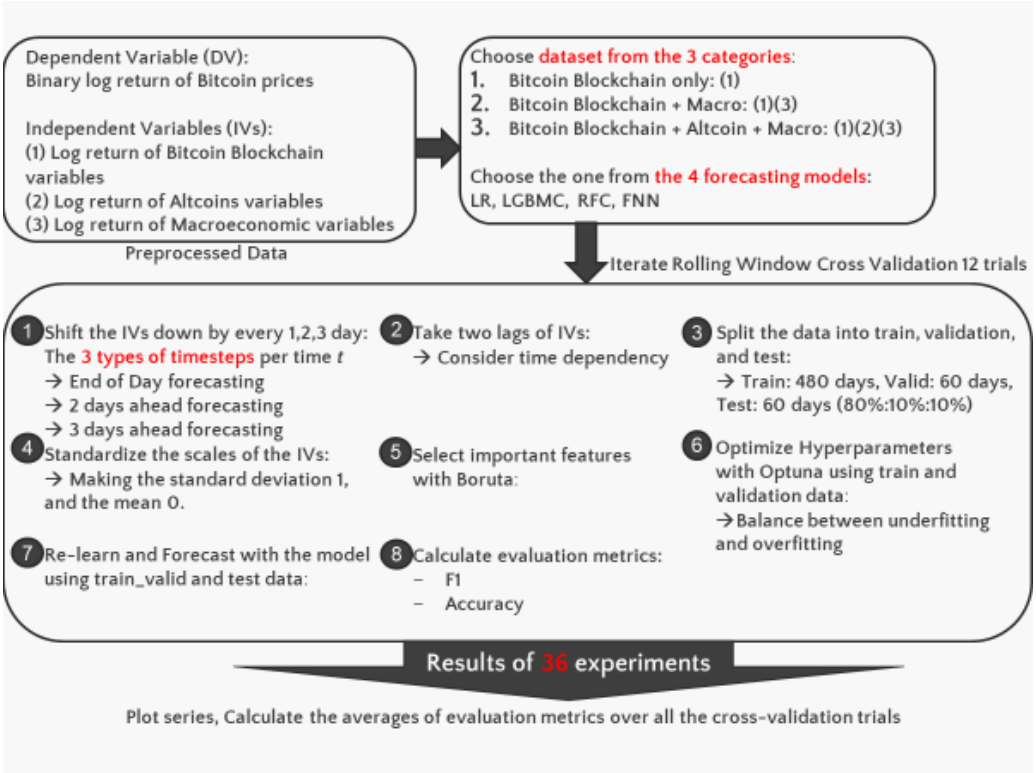
The whole experiment was composed of two steps. The first step was to forecast Bitcoin binary log returns in End of Day, two and three days ahead. Then, we calculated their evaluation metrics. The second step was to simulate trading strategies based on the forecasting models and assess their profitability.

4.1 Bitcoin Return Forecasting

4.1.1 The overview of the experiment

Figure 3 shows the overview of the experiment. Once we finished preprocessing the dataset, the master table contained the dependent variable, binary log return of Bitcoin prices, and the independent variables from the three dataset categories: the Bitcoin blockchain, the Altcoins market, and the macroeconomics. Then, we chose the combination of the forecasting models and the dataset categories to use in each experiment.

Figure 3: The overview of the experiment



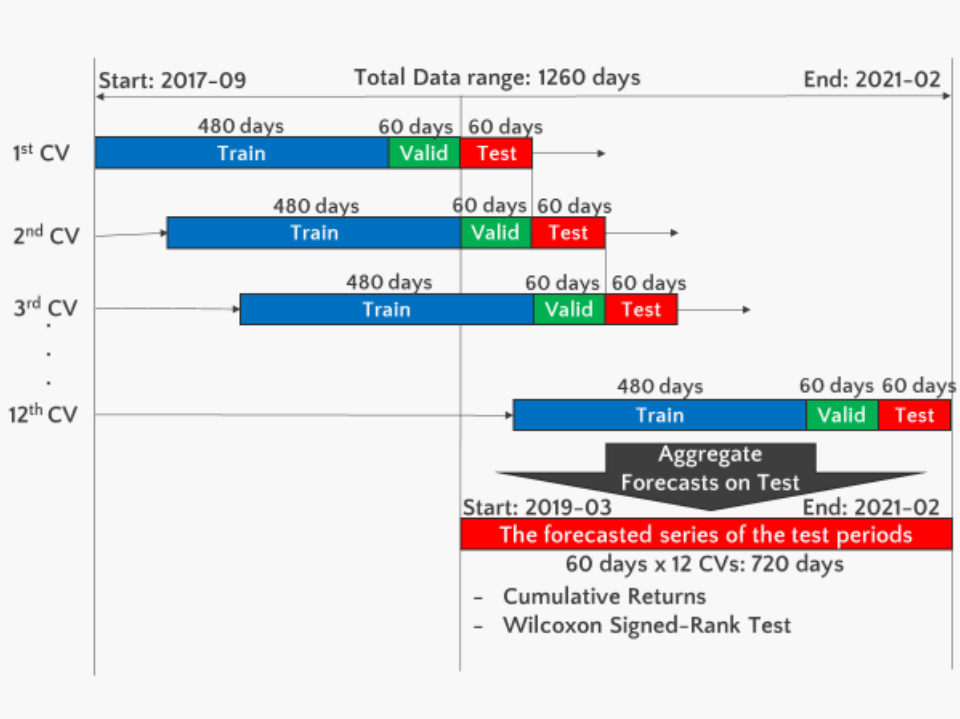
Each rolling window cross-validation trial was iterated as the below eight steps. First, we shifted the independent variables down by one, two, and three days to set up a binary classification problem for End of Day, two and three days ahead return forecasting, respectively. As a result, we conducted 36 combinations of the experiment (three dataset combinations, four forecasting models, and three different timesteps). Second, adding one and two lags to the independent variables enabled the models to consider time dependencies on Bitcoin returns. Third, the data were split into train, validation, and test. Fourth, we standardized the scales of the independent variables. Fifth, the Boruta algorithm chose statistically significantly important independent variables. Sixth, the Optuna algorithm optimized the hyperparameters of the forecasting models. Seventh, the forecasting models learned the hidden patterns behind the train and validation data with the optimized hyperparameters. Eighth, we compared the evaluation metrics based on the forecasting model, dataset category, and the number of days ahead of the forecasting we chose in the experiment. Besides, we employed two benchmark forecasting strategies: Random guess and Naive guess. As we followed Garcia (2015), the Random guess "samples a random number with 0 mean at every time *t* and formulate a prediction based on the sign of the random number". On the other hand, the Naive guess "predicts that price changes at time $t + 1$ will be the same as at time t" (Garcia & Schweitzer, 2015).

4.1.2 Rolling window cross-validation

In evaluating the forecasting ability of the models, it is necessary to consider the forecasting accuracy for unseen data, generalization performance, in addition to the goodness of fit in train data. In most cases, previous studies used a k-fold cross-validation method or a hold-out method to address this task (Mcnally, 2016; Amjad & Shah, 2016; Mallqui & Fernandes, 2019; Chen et al., 2020; Mudassir, 2020). However, time-series data have some difficulty in applying both solutions. First, the k-fold cross-validation method assumes that the sample data are independently identically distributed, but this assumption is not necessarily satisfied in time-series data. Second, the market situations vary from period to period frequently. Therefore, high forecasting accuracy in the most recent data does not necessarily guarantee the robustness of forecasting unseen data in the hold-out method. Finally, we needed to use older data as train data and newer data as test data for time-series data because the order of the data is critical, not allowing us to split the data into train and test data randomly.

Therefore, we adopted a rolling window cross-validation method (also called a walk-forward optimization method (Dixon et al., 2017; Shintate & Pichl, 2019)). Figure 4 illustrates how rolling window cross-validation works. The total data range was 1260 days from September 2017 to February 2021. In the cross-validation trials of this range, the first day of the train data was shifted by 60 days 12 times. We generated a forecasted series and evaluated its accuracy in the 12 cross-validations, relying on the procedures described in Subsections from 4.1.3 to 4.1.7. Finally, in the range of March 2019 between February 2021, the forecasted series of the test period were linked together and used for the trading simulation of Subsection 4.2.

Figure 4: The mechanism of rolling window cross-validation



4.1.3 Preparation of a master table for modeling

We still had some work to use the independent variables in the forecasting models. First, we shifted the independent variables down by one, two, and three days to set up a binary classification problem for End of Day, two and three days ahead return forecasting, respectively. Second, we added one and two lags to the independent variables, enabling the models to consider the effects of time dependencies on Bitcoin returns. Third, we needed to split the data into train, validation, and test periods to optimize the hyperparameters of the forecasting models. Forecasting models have both model parameters and hyperparameters. It is possible to estimate model parameters by making models learn from train data. However, we needed to optimize hyperparameters externally because the models cannot learn them by themselves. The procedures were as follows. Initially, the forecasting models estimated the model parameters with the train data. Then, by fixing those inside parameters, the Optuna algorithm optimized the hyperparameters with the validation data to maximize its f1 score. Finally, the forecasting models estimated the model parameters again with both the train and validation data to forecast the test periods with the optimized hyperparameters. Fourth, we standardized the scales of the independent variables, making the means zero and the standard deviations one.

4.1.4 Variables selection with Boruta

So far, we preprocessed 79 independent variables and took up to two lags of them. As a result, we had 237 independent variables for 480 days of train data. However, using many independent variables in models often complicates models and causes an overfitting problem. Moreover, it increases the computation time. To reduce the number of independent variables, we needed to select only the relevant ones among all the independent variables. The most commonly used methods are filter methods, which select variables with high correlation or high regression coefficients. However, in machine learning prediction tasks, filter methods may lose important information and reduce prediction accuracy. "Very high variable correlation (or anti-correlation) does not mean an absence of variable complementarity" and "a variable that is utterly useless by itself can provide a significant performance improvement when taken with others" (Guyon & Elisseeff, 2003).

Therefore, as Chen (2020) and Mcnally (2016) employed, we adopted the Boruta algorithm, a wrapper built around the Random Forest classification algorithm (Kursa & Rudnicki, 2010). According to Kursa (2010), Random Forest is an ensemble method that performs classification by voting on multiple unbiased weak classifiers, decision trees. The algorithm creates these decision trees independently using different bagging samples from the train data, obtaining the importance of a variable as the loss of classification accuracy caused by random permutations of variable values between objects. Furthermore, the *Z*-score, the average loss of accuracy divided by its standard deviation, can also measure importance because it is computed separately for all trees (Kursa & Rudnicki, 2010).

According to Kursa (2010), the Boruta algorithm follows the below steps. First, the algorithm extends the information system by creating "shadow variables" with copies of all variables. Second, it shuffles the shadow variables to remove any correlation with the dependent variable. Third, it runs a Random Forest classifier on the expanded information system and collects the calculated *Z*-scores. Fourth, it finds the largest *Z*-score among the shadow variables (*MZSA*) and assigns a hit to all variables with a *Z*-score better than *MZSA*. Fifth, it performs a two-sided test with *MZSA* for each variable whose importance has not been determined. The null hypothesis here is that the importance of this variable is the same as the importance of a variable that does not contribute to the classification (the shadow variable with *MZSA*). The alternative hypothesis 1 is that the importance of this variable is greater than the importance of the shadow variable with *MZSA*. The alternative hypothesis 2 is that the importance of this variable is less than the importance of the shadow variable with *MZSA*. The test statistic *T* is the number of times a hit is assigned. Sixth, it considers variables that are significantly less important than the shadow variable with *MZSA* as "unimportant" and permanently removes them from the information system. In contrast, it considers variables that are significantly more important than the shadow variable with *MZSA* as "important." Seventh, it removes all shadow variables. Finally, it repeats this procedure until all the independent variables are assigned importance or the predefined Random Forest iteration ends.

4.1.5 Hyperparameter optimization with Optuna

The machine learning model estimates the parameters of the independent variables (model parameters) to minimize the prediction error between the predicted values and the actual values of the dependent variable. However, machine learning models have hyperparameters that need to be optimized externally apart from the model parameters. These hyperparameters often have to do with balancing overfitting and underfitting, affecting prediction performance. The methods that have been commonly used to optimize the hyperparameters are a grid search and a random search. The grid search specifies the candidate values for each hyperparameter in advance and tries all the combinations without limiting the number of searches. In contrast, the random search specifies the candidate values for each hyperparameter in advance and randomly combines the values within the range with the limit of the number of searches. However, both methods have flaws. The grid search requires too much computation time for search, and the random search is more challenging to find globally optimized hyperparameter combinations than grid search.

Therefore, the paper employed the Optuna algorithm, which works with Bayesian optimization called Tree-structured Parzen Estimator. We set up the optimization problem that Optuna faces with the three key elements. First, the maximized objective function was the f1 score of validation data. Second, decision variables were the hyperparameters of each forecasting model explained below. Third, the constraints were the predefined candidate regions for each hyperparameter.

Optuna finds an optimal solution using the train data. According to Akiba (2019), Optuna uses the history of completed trials to determine the values of hyperparameters to try in the subsequent trial. Based on the history of completed trials up to that point, Optuna estimates a promising region and repeats trying the values in that region. Then, based on the newly obtained results, it estimates more promising regions. Optuna also can prune branches (Akiba et al., 2019). When we use iterative algorithms for training, such as deep learning or gradient boosting, Optuna can roughly predict how well the final result of training will be using its learning curve. With this prediction, Optuna can terminate trials early before finishing if it predicts they will not produce good results.

4.1.6 Forecasting models

We formulated a binary classification problem for End of Day, two, and three days ahead return forecasting. To solve this problem, we used four classification algorithms: Logistic Regression (Logistic), Random Forest Classifier (RFC), Light Gradient Boosting Machine Classifier (LGBMC), and Feedforward Neural Network (FNN).

Here, we introduce the mechanics of these forecasting models. All the models have essential components: Estimation equation that assumes the relationship between independent variables and dependent variables, an objective function that the algorithm optimizes in the model, model parameters that play the role of decision variables in the optimization problem, hyperparameters that Optuna optimizes as explained later.

In the below descriptions, we designate the matrix of the independent variables $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T$ where each independent variables vector $\mathbf{x_n} = (x_{n0}, \dots, x_{nD})$ for $n \in \{1, \dots, N\}$. Here, we assume that every $x_{n0} = 1$ for $n \in \{1, \dots, N\}$ as they are constants. Besides, $\boldsymbol{y} = (\mathrm{y}_1, \dots, \mathrm{y}_N)^T$. The number of data points is N, and the number of dimensions of the independent variables is D.

4.1.6.1 Logistic Regression (Logistic)

Logistic regression is a linear model for a binary classification problem whose dependent variable should take the binary values, 0 or 1. In Statistics, the model is also known as logit regression. We denote the coefficients of the independent variables vector $\mathbf{w} = (w_0, \dots, w_D)$, w_0 is the intercept. The estimation equation is:

$$\widehat{\boldsymbol{y}}(\mathbf{w}, \mathbf{X}) = \sigma\,(\mathbf{X}\mathbf{w}^T)$$

where σ is the sigmoid function:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Thus, we can forecast the class label (binary Bitcoin returns) as follows:

$$\widehat{y_n}\,(\mathbf{w}, \mathbf{x_n}) = 1 \text{ if } P(y_n = 1) = \sigma(\mathbf{x_n}\mathbf{w}^T) \geq 0.5$$

$$\widehat{y_n}(\mathbf{w}, \mathbf{x_n}) = 0 \text{ if } P(y_n = 1) = \sigma(\mathbf{x_n}\mathbf{w}^T) < 0.5$$

According to Pedregosa (2011), given the ℓ_2 regularization, the minimized objective function is:

$$\min_{\mathbf{w}} \left[\frac{1}{2}\,\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N} \log\left(1 + \exp\left(-y_n(\mathbf{x_n}\mathbf{w}^T)\right)\right)\right]$$

where C is the inverse of regularization strength, and y_n is the actual value of the dependent variable in the n-th data point. The model parameters are the coefficients of the independent variables vector $\mathbf{w}$.

Table 4 shows the hyperparameters of Logistic regression. The solver decides which algorithm to use in the optimization problem. The default solver is lbfgs. However, we made liblinear another candidate of the solver because it is good at addressing small datasets, according to Pedregosa (2011). The C is the inverse of regularization strength. The smaller the C is, the stronger the regularization, preventing overfitting.

Table 4: Hyperparameters of Logistic Regression

Hyperparameters	Candidate regions
solver	{liblinear, lbfgs}
C	[1e-3, 1e0]

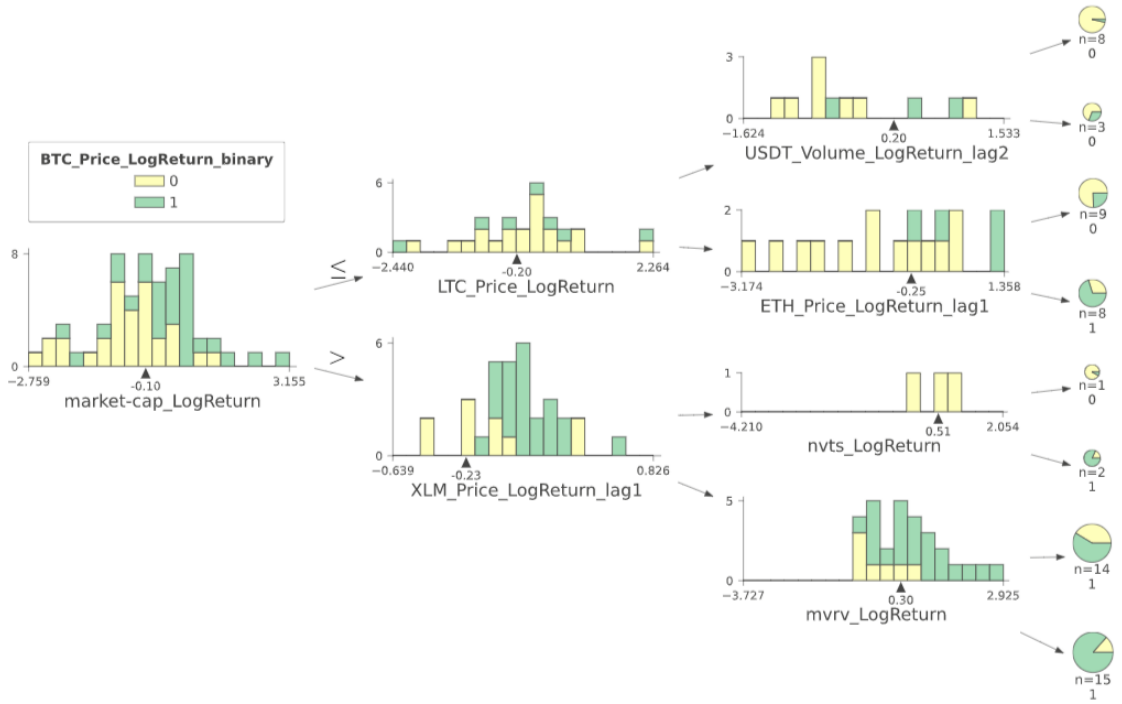
The solver is the algorithm to optimize, and C is the inverse of regularization strength.

4.1.6.2 Random Forest Classifier (RFC)

According to Pedregosa (2011), Random Forest is an ensemble method employing decision trees. Each decision tree is constructed using random samples from the train data in the ensemble process. Besides, the optimal split is found randomly among all the independent variables when splitting each node in a decision tree. These two randomness aims to reduce the variance of the forest estimator. Individual decision trees often show high variances and tend to be overfitted. The randomness in the forest generates a decision tree with separated prediction errors. Taking the average of these predictions offsets some of the errors. Random forest reduces variance by combining various decision trees, preventing overfitting problems. Pedregosa (2011) stated, "in practice, the variance reduction is often significant hence yielding an overall better model."

Figure 5 illustrates how each decision tree captured the relationship between the dependent variable and the independent variables.

Figure 5: An example of a decision tree generated in Random Forest



According to Pedregosa (2011), a decision tree repeatedly divides the independent variable space to group together samples where the dependent variable takes the same value. As the objective function, we adopted the Gini impurity to measure the success of a split.

Finally, Table 5 introduces the hyperparameters of RFC. According to Pedregosa (2011), max_depth determines the maximum depths of the tree, and n_estimators decide the number of trees in the forest. As more the maximum depths of the tree, the model becomes

more complex and easier to suffer from overfitting problems. On the other hand, as the number of trees increase, it can alleviate overfitting problems.

Table 5: Hyperparameters of RFC

Hyperparameters	Candidate regions
max_depth	[3, 10]
n_estimator	[1, 1000]

max_depth means the maximum depths of the tree, and n_estimator means the number of trees in the forest.

4.1.6.3 Light Gradient Boosting Machine Classifier (LGBMC)

Light Gradient Boosting Machine Classifier (LGBMC) is an applied gradient boosting method employing decision trees like ensemble methods. The difference between ensemble methods and gradient boosting methods is in how to shape their decision trees. Ensemble methods like Random Forest create decision trees independently and take the average of each prediction. However, gradient boosting methods utilize prediction errors in a previous tree to continuously improve subsequent decision tree predictions. The advantages of LGBMC compared with conventional gradient boosting methods are reduced training time and memory usage achieved with its features: Histogram-Based Algorithms and Leaf-wise Tree Growth (Ke, et al., 2017). Besides, according to LightGBM (n.d.), the minimized objective function is Log Loss, also known as binary cross-entropy, explained in our Subsection 4.1.6.4 Feedforward Neural Network.

Finally, Table 6 shows the hyperparameters of LGBMC. The effect of the maximum depths of the tree is the same as Random Forest, but that of the number of trees is different. As more trees, the overfitting problem may happen because the algorithm does not average decision node results. Besides, the learning rate adjusts the speed to optimize the model parameters to reach the minimum loss function (Xiaolei et al., 2020). As larger the learning rate is, the model suffers from overfitting easier.

Table 6: Hyperparameters of LGBMC

Hyperparameters	Candidate regions
max_depth	[5, 10]
n_estimator	[1, 1000]
learning_rate	[1e-4, 1e-1]

max_depth means the maximum depths of the tree, and n_estimator means the number of trees in Gradient Boosting. learning_rate is the speed to optimize the model parameters to reach the minimum loss function.

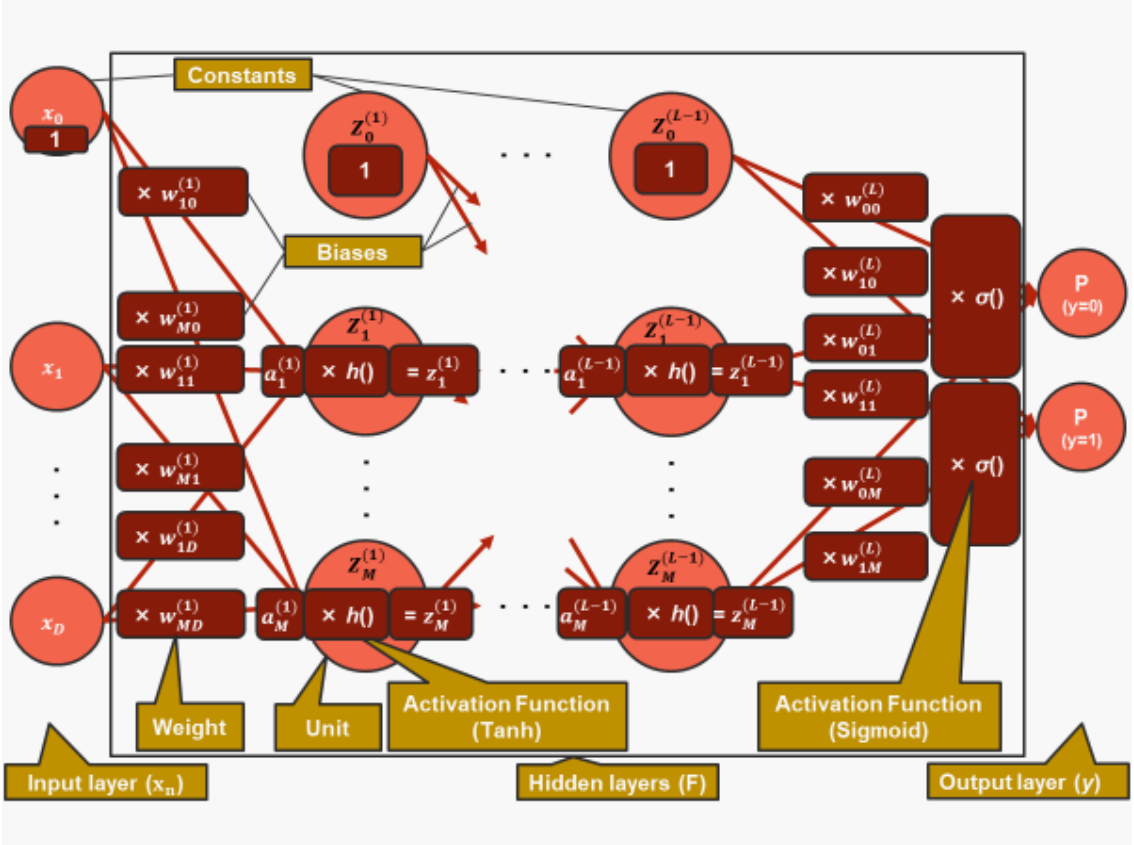
4.1.6.4 Feedforward Neural Network (FNN)

FNN is a non-linear model that minimizes the error between the predicted and actual values by optimizing the model parameters. At first, FNN has five essential components: an input layer, hidden layers, an output layer, units, and model parameters. The input layer consists of the independent variables, and the output layer consists of the class labels of the dependent

variables. The hidden layers between the input and output layers capture the relationship between the independent variables and the dependent variable. Each unit performs a specific calculation on the data flowing from the previous layer, transforms it, and conveys it to the next layer. Finally, the model parameters consist of biases (the coefficients of the constants) and weights for each unit, connecting each unit and sending the transformed data from the previous unit to the subsequent unit.

Second, Figure 6 illustrates how FNN works.

Figure 6: The components of FNN



The forecasting procedure consists of the following four steps: forward propagation, the objective function, backward propagation, and optimization. We borrowed the estimation equations from Bishop (2006) and generalized it for FNN having *L* hidden layers.

At first, to expand the logistic regression model, we construct *M* linear combinations of the independent variables $\{x_1, \dots, x_D\}$:

$$a_m^{(1)} = \sum_{d=0}^{D} w_{md}^{(1)}\, x_d$$

where $m \in \{1, \dots, M\}$ and $x_0 = 1$ (defined as the constant). The superscript (l) of $w_{md}^{(l)}$ for $l \in \{1, \dots, L\}$ means the order of layer where the weight exists. We denote the number of layers as L, the number of hidden units as M, the weights as the model parameters $w_{md}^{(l)}$ and biases as the model parameters $w_{m0}^{(l)}$. Here, we assume that the number of units in each hidden layer is M for simplicity.

We can transform each activation $a_m^{(1,\dots,L-1)}$ by a differentiable, non-linear activation function. The activation function we used for the hidden layers is the tanh function:

$$z_m = tanh(a_m) = \frac{\exp(a_m) - \exp(-a_m)}{\exp(a_m) + \exp(-a_m)}$$

where all the $z_0^{(1,\dots,L-1)} = 1$. Here, all the $z_0^{(1,\dots,L-1)}$ are defined as the constants. The superscript (l) of z_m for $l \in \{1, \dots, L-1\}$ means the order of the connected layer.
Thus:

$$z_m^{(1)} = tanh\left(\sum_{d=0}^{D} w_{md}^{(1)}\ x_d\right)$$

$$z_m^{(2)} = tanh\left(\sum_{m=0}^{M} w_{km}^{(2)}\ z_m^{(1)}\right)$$

…

$$z_m^{(L-1)} = tanh\left(\sum_{m=0}^{M} w_{km}^{(L-1)}\ z_m^{(L-2)}\right)$$

Regarding the last layer L:

$$a_k^{(L)} = \sum_{m=0}^{M} \left(w_{km}^{(L)} z_m^{(L-1)}\right)$$

where the class label is $k \in \{0,1\}$. The activation function we used for the output layer is the sigmoid function σ that was explained in Subsection 4.1.6.1 Logistic Regression.

Finally, from the above, we get:

$$P(y = k) = \sigma\left(\sum_{m=0}^{M} \left(w_{km}^{(L)} z_m^{(L-1)}\right)\right) = \sigma\left(\sum_{m=0}^{M} \left(w_{km}^{(L)}\ tanh\left(\sum_{m=0}^{M} w_{km}^{(L-1)}\ z_m^{(L-2)}\right)\right)\right)$$

where $P(y = k)$ means the probability that the class label is $k \in \{0,1\}$, given the model parameters: the independent variables vectors $\mathbf{x_n}$ and the weights matrix $\mathbf{W}$. Therefore, we can forecast the class label by rounding the probability:

$$\widehat{y_n}\ (\mathbf{W}, \mathbf{x_n})\ = 1\ if\ P(y_n = 1)\ \geq\ 0.5$$

$$\widehat{y_n}\ (\mathbf{W}, \mathbf{x_n}) =\ 0\ if\ P(y_n = 1) <\ 0.5$$

The minimized objective function of the FNN model here is Binary cross-entropy (known as Log Loss) expressed as:

$$e(\mathbf{W})\ =\ -\sum_{n=1}^{N}\sum_{k=0}^{1}\{t_{nk}\log(\ y_{nk}) + (1 - t_{nk})\log\ (1 - y_{nk})\}$$

In the objective function, t_{nk} means the actual value at the *n*-th data point, while y_{nk} means the predicted value $\widehat{y_n}\ (\mathbf{W}, \mathbf{x_n}) = k$ for $k\ \in\ \{0,1\}$ at the *n*-th data point. The objective function is optimized by the algorithm called backward propagation. In the backward propagation, we chose the Adam optimization algorithm as Mudassir (2020) used.

Table 7 shows the hyperparameters of FNN. As the number of layers and units increases, the model becomes more complicated, causing overfitting easier. On the other hand, the dropout rate alleviates overfitting. Too large Batch size and many epochs lead to overfitting. We set up the predefined ranges of each hyperparameter as below because they performed well after iterating try and errors in the experiments.

Table 7: Hyperparameters of FNN

Hyperparameters	Candidate regions
num_hidden_layers	[3, 5]
num_hidden_units	[8,128]
dropout_rate	[0.2, 0.4]
batch_size	{8, 16, 32, 64}
epochs	[10, 30]

num_hidden_layers are the number of hidden layers, num_hidden_units are the number of hidden units, dropout_rate is the rate of skipping the units randomly in learning, batch_size is the number of sample data used in each learning iteration, and epochs is the number of times the whole learning process is iterated.

4.1.7 Evaluation metrics

After creating forecasts, we could classify the results into four patterns. First, if an actual return was positive and the predicted return was positive, it belongs to True Positive (TP). Second, if an actual return was negative and the predicted return was negative, it belongs to True Negative (TN). Third, if an actual return was negative, but the predicted return was positive, it belongs to false Positive (FP). Finally, if an actual return was positive, but the predicted return was negative, it belongs to False Negative (FN). Thus, the prediction was correct in the TP and TN cases, while the prediction was wrong in the FP and FN cases. We calculated four evaluation metrics from these four patterns: accuracy score, precision score,

recall score, and f1 score. The accuracy score seems the most intuitive metric to evaluate the ability of models to classify. The accuracy score was calculated as:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}$$

However, when the classes of the dependent variable are imbalanced, for example, in a cross-validation trial, when the number of days when Bitcoin return is positive is much more frequent than that of days Bitcoin return is negative, the accuracy score cannot consider the difference between FN and FP. If every error comes from only FN or FP in a forecasting model, we cannot admit that the model has enough forecasting ability even when its accuracy score is high.

Therefore, we used both the f1 and the accuracy scores to compare the forecasting models in Subsection 5 Results. By taking an average between the precision score and the recall score, we can calculate the f1 score as:

$$\text{F1} = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

where:

$$\text{Precision} = \frac{(\text{TP})}{(\text{TP} + \text{FP})}$$

$$\text{Recall} = \frac{(\text{TP})}{(\text{TP} + \text{FN})}$$

The precision score and the recall score evaluate FP and FN, respectively. However, the f1 score is better to evaluate FP and FN simultaneously because the relationship between FP and FN is a trade-off.

4.2 Trading Simulation

4.2.1 Simple trading strategy based on return forecasting.

When applying the forecasting to trading, we took the straightforward approach; We decide to buy or not by the forecasting for End of Day, two, and three days ahead. We do not rely on forecasting by the models when selling Bitcoin. At the time *t*, we sell all the Bitcoin if we bought it at the time $t - 1$, regardless of the actual return at the time *t*. Next, we decide to buy if the forecasted return at the time $t + 1$ is positive. If the forecasted return at the time $t + 1$ is negative, we do not buy. That is, do nothing. The trading interval between time *t* and $t + 1$ is every day when trading based on the End of Day return forecasting, every second day when trading based on the two days ahead return forecasting, and every three days when trading based on the three days ahead return forecasting.

We also set up three benchmark trading approaches: Random guess, Naive guess, and Buy & Hold strategy. Regarding Random guess and Naive guess, the procedures were explained

in Subsection 4.1.1, The overview of the experiment. As above, we decide to buy in both strategies if the forecasted return at the time $t + 1$ is positive. On the other hand, if the forecasted return at the time $t + 1$ is negative, we do not buy. That is, do nothing. Finally, when we employed Buy & Hold strategy, as Garcia (2015) explains, it "simply buys Bitcoin with the initial capital at time $t = 1$, selling it only once at the time when profits are evaluated".

4.2.2 Evaluation of trading performances

At first, we calculated the cumulative returns of the trading strategies. The simulated cumulative returns demonstrated the ability of the forecasting models to make profits in trading simulations. Then, we conducted the Wilcoxon Signed-Rank test on the simulated Bitcoin returns generated from the forecasting models and the benchmarks (Wilcoxon, 1945). Wilcoxon Signed-Rank test has its null hypothesis that the paired samples from the two series follow the same distribution, meaning the difference of two samples $d = s1 - s2$ are distributed around zero symmetrically (Scipy.org, 2021). We also set up the alternative hypothesis that $d > 0$. Therefore, we could conclude whether the returns series generated from the forecasting models are greater than those from the benchmarks or not.

5. Results

In the previous section, we discussed the experimental methods. Our whole experiment was composed of two steps. The first step was to forecast Bitcoin binary log returns in End of Day, two and three days ahead. Then, we calculated their evaluation metrics. The second step was to simulate trading strategies based on the forecasting models and assess their profitability. This section shows their results.

5.1 Bitcoin Return Forecasting

5.1.1 The independent variables chosen by Boruta

Table A5 in the Appendix shows the list of independent variables chosen by Boruta at least once in the experiments when we use all the three dataset categories: the Bitcoin blockchain, the Altcoins market, and the macroeconomics. As Table A5 shows, the variables that were chosen by Boruta at least once mainly came from the categories of the Bitcoin blockchain and the Altcoins market. As to the macroeconomic variables, only the log return of the gold prices and that of the GBP-USD prices were statistically significantly important to forecast Bitcoin returns. Furthermore, Boruta chose more one-lagged variables from the Altcoins variables than from the other categories and used them for forecasting. Thus, the results supported our hypothesis that the Bitcoin blockchain and the Altcoins market are potential factors affecting future Bitcoin returns.

5.1.2 The f1 scores of each cross-validation trial

Figures 7-9 illustrate the f1 scores of the test periods over the cross-validation trials. The x-axis expresses the cross-validation trial number. Each cross-validation trial corresponded to 60 days of the test periods; for example, the first trial contained the data range between March 03, 2019, and May 1, 2019, in End of Day forecasting. The y-axis expresses the f1 scores in each cross-validation trial. The visualizations show the case that the forecasting models used all the three dataset categories as the independent variables.

In all the timesteps: End of Day (shown as Figure 7), two days ahead (shown as Figure 8), and three days ahead (shown as Figure 9), our four forecasting models (Logistic, LGBMC, RFC, FNN) outperformed Random guess and Naive guess in the f1 scores, even when forecasting did not work well (e.g., in the fifth cross-validation trial).

Besides, when comparing End of Day, two, and three days ahead forecasting, End of Day forecasting was the most successful of the three. This result implied that it is easier to forecast a short-term future.

Tables A6-A8 in the Appendix show all the f1 scores of the test period in every cross-validation trial. The columns "CV," "Test start," and "Test end" in Tables A6-A8 represent the same cross-validation trial number as the x-axis of Figures 7-9 and the corresponding dates. The tables show the case that the forecasting models used all the three dataset categories as the independent variables.

Figure 7: The f1 scores of each cross-validation trial for End of Day return forecasting

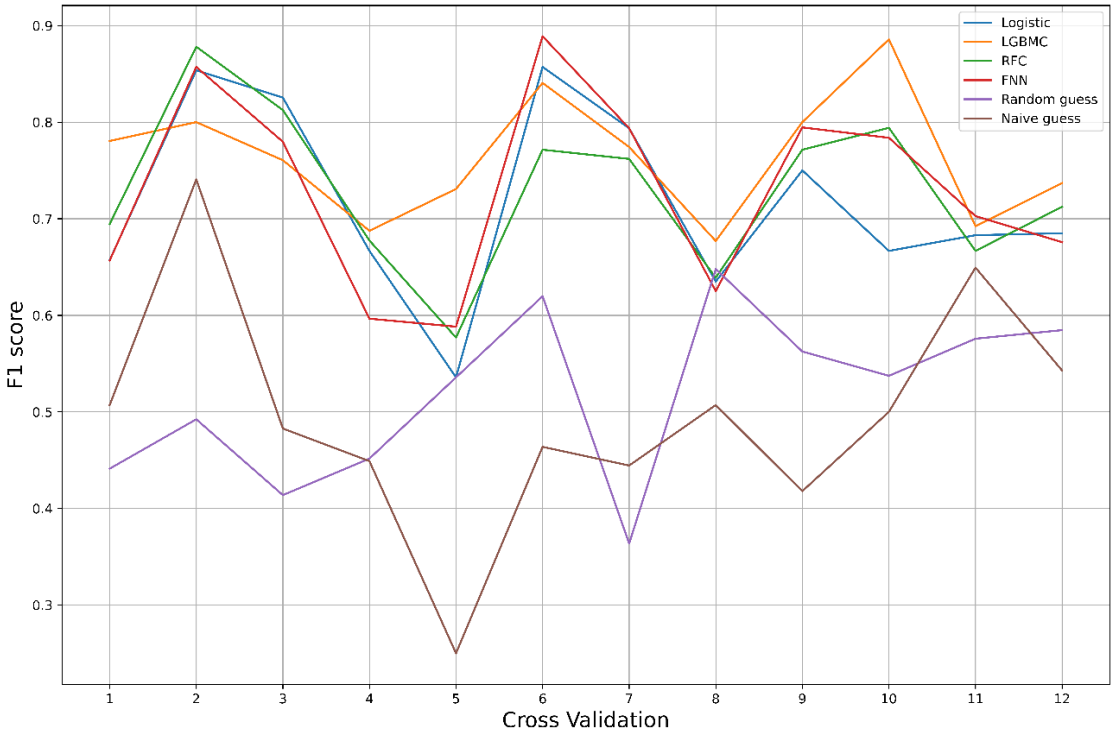


Figure 8: The f1 scores of each cross-validation trial for two days ahead return forecasting

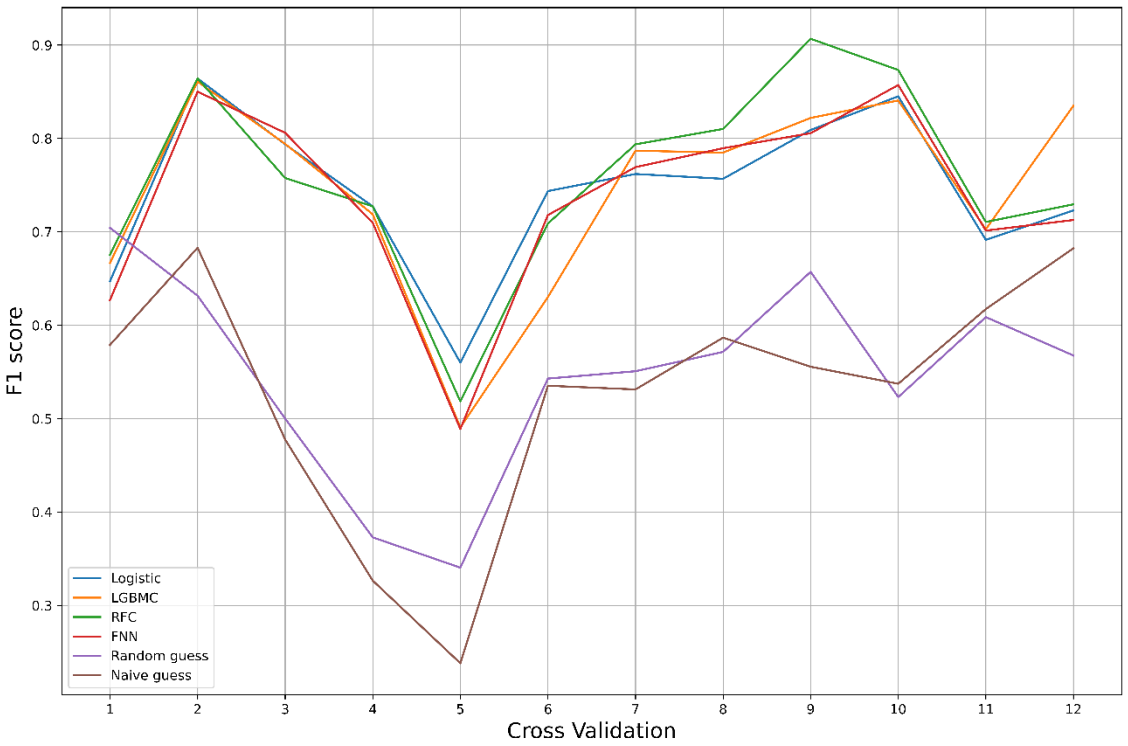
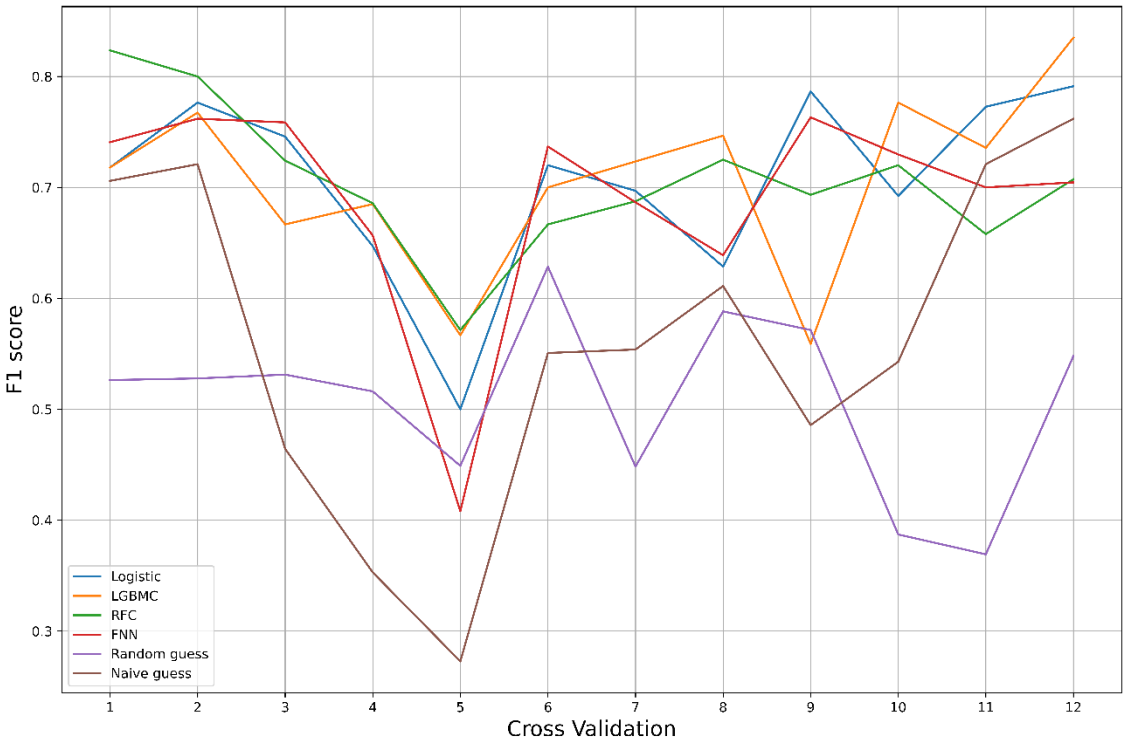


Figure 9: The f1 scores of each cross-validation trial for three days ahead return forecasting



5.1.3 The average f1 and accuracy scores over the cross-validation trials

Tables 8-10 show the average f1 and accuracy scores over the cross-validation trials for End of Day, two, and three days ahead return forecasting.

Table 8: The average f1 and accuracy scores over all the cross-validation trials for End of Day forecasting

model	dataset	test f1	test accuracy	train-valid f1	train-valid accuracy
LGBMC	**BlockchainAltcoinMacro**	**0.764**	**0.715**	0.901	0.865
FNN	BlockchainMacro	0.763	0.744	0.750	0.740
FNN	Blockchain	0.757	0.739	0.752	0.740
Logistic	BlockchainMacro	0.753	0.707	0.732	0.686
Logistic	Blockchain	0.741	0.701	0.731	0.689
RFC	**BlockchainAltcoinMacro**	**0.730**	**0.694**	0.959	0.956
FNN	**BlockchainAltcoinMacro**	**0.729**	**0.704**	0.819	0.812
Logistic	**BlockchainAltcoinMacro**	**0.717**	**0.676**	0.768	0.742
RFC	Blockchain	0.715	0.690	0.919	0.915
LGBMC	BlockchainMacro	0.715	0.647	0.830	0.778
LGBMC	Blockchain	0.714	0.642	0.801	0.742
RFC	BlockchainMacro	0.703	0.674	0.895	0.889

The column "model" expresses which forecasting model is used, and the column "dataset" tells the category from which the independent variables come. The column "test f1" and "test accuracy" have the average f1 and accuracy scores in test data over cross-validations, while the column "train-valid f1" and "train-valid accuracy" have average f1 and accuracy scores in train and validation data over cross-validations. The bold font shows the dataset that includes the variables of the Altcoins market.

Table 9: The average f1 and accuracy scores over all the cross-validation trials for two days ahead forecasting

model	dataset	test f1	test accuracy	train-valid f1	train-valid accuracy
RFC	**BlockchainAltcoinMacro**	**0.756**	**0.711**	0.920	0.914
LGBMC	**BlockchainAltcoinMacro**	**0.744**	**0.704**	0.931	0.915
Logistic	**BlockchainAltcoinMacro**	**0.744**	**0.699**	0.763	0.738
FNN	**BlockchainAltcoinMacro**	**0.736**	**0.700**	0.787	0.773
Logistic	BlockchainMacro	0.669	0.604	0.657	0.597
Logistic	Blockchain	0.654	0.599	0.654	0.602
FNN	Blockchain	0.652	0.618	0.642	0.618
LGBMC	BlockchainMacro	0.652	0.579	0.743	0.673
LGBMC	Blockchain	0.643	0.572	0.754	0.692
FNN	BlockchainMacro	0.643	0.607	0.648	0.625
RFC	Blockchain	0.609	0.579	0.861	0.864
RFC	BlockchainMacro	0.607	0.585	0.862	0.861

The column "model" expresses which forecasting model is used, and the column "dataset" tells the category from which the independent variables come. The column "test f1" and "test accuracy" have the average f1 and accuracy scores in test data over cross-validations, while the column "train-valid f1" and "train-valid accuracy" have average f1 and accuracy scores in train and validation data over cross-validations. The bold font shows the dataset that includes the variables of the Altcoins market.

Table 10: The average f1 and accuracy scores over all the cross-validation trials for three days ahead forecasting

model	dataset	test f1	test accuracy	train-valid f1	train-valid accuracy
LGBMC	**BlockchainAltcoinMacro**	**0.707**	**0.624**	0.850	0.804
Logistic	**BlockchainAltcoinMacro**	**0.706**	**0.643**	0.727	0.692
RFC	**BlockchainAltcoinMacro**	**0.705**	**0.640**	0.887	0.875
Logistic	Blockchain	0.701	0.614	0.671	0.584
Logistic	BlockchainMacro	0.693	0.607	0.666	0.583
FNN	**BlockchainAltcoinMacro**	**0.690**	**0.636**	0.736	0.719
LGBMC	BlockchainMacro	0.679	0.547	0.719	0.606
LGBMC	Blockchain	0.675	0.550	0.722	0.615
FNN	Blockchain	0.641	0.574	0.650	0.602
FNN	BlockchainMacro	0.628	0.565	0.640	0.602
RFC	Blockchain	0.627	0.571	0.743	0.719
RFC	BlockchainMacro	0.603	0.546	0.747	0.725

The column "model" expresses which forecasting model is used, and the column "dataset" tells the category from which the independent variables come. The column "test f1" and "test accuracy" have the average f1 and accuracy scores in test data over cross-validations, while the column "train-valid f1" and "train-valid accuracy" have average f1 and accuracy scores in train and validation data over cross-validations. The bold font shows the dataset that includes the variables of the Altcoins market.

For End of Day return forecasting, all the f1 scores were better than 70%. The best f1 score reached better than 76%, and the best accuracy score was better than 74%. FNN and Logistic Regression achieved relatively higher performances in both metrics. Regarding the dataset, we could not point out which dataset combinations achieved better performances than the others.

For two days ahead of return forecasting, the difference of f1 scores between the best and worst models was more prominent than in End of Day forecasting. The best f1 score was better than 75%, and the best accuracy score was better than 71%. We could not identify which forecasting models achieved better performances than the others regarding the forecasting models. However, the models using Altcoins variables achieved more excellent performances than those not using them.

For three days ahead of return forecasting, the best f1 score reached 70%, and the best accuracy score was better than 64%. Unfortunately, we could not clarify which forecasting models achieved better performances than the others regarding the forecasting models. However, the models using Altcoins variables achieved better performances than those not using them again.

To sum up, the more we forecast the future, the more variables we may need. However, the results also implied that we need to build multiple models and compare their performances to determine the best model in each situation. One size does not fit all.

5.2 Trading Simulation

5.2.1 Cumulative returns

Figures 10-12 show the simulated cumulative returns over the whole test period based on the forecasting for End of Day, two, and three days ahead, respectively. Again, the visualizations show the case that the forecasting models used all the three dataset categories as the independent variables.

In all the cases: everyday trading (shown as Figure 10), trading every second day (shown as Figure 11), and trading every three days (shown as Figure 12), the trading strategies based on our four forecasting models (Logistic, LGBMC, RFC, FNN) outperformed the benchmarks in cumulative returns. Although the benchmark strategies achieved less than 2.5x cumulative returns, our trading strategies reached 6.5–8x cumulative returns in everyday trading, 4.5-5.5x in trading every second day, and 3.5-4.5x in trading every three days.

Figure 10: Cumulative returns over the whole test period in everyday trading

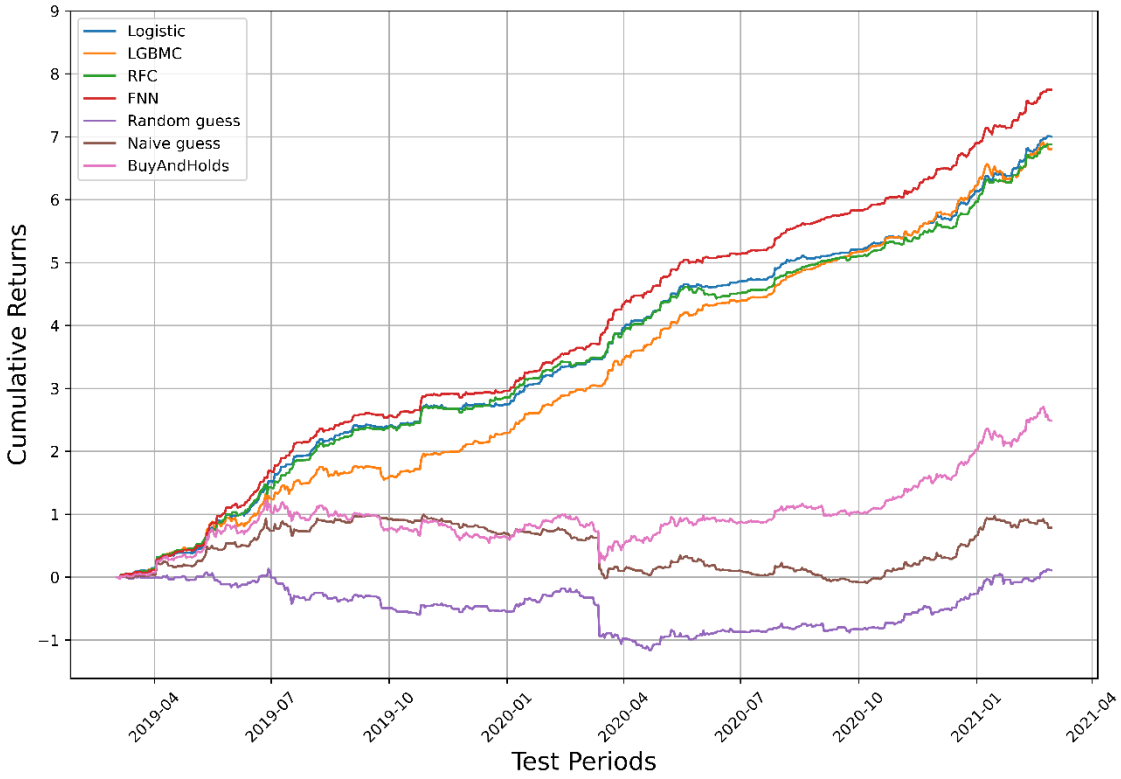


Figure 11: Cumulative returns over the whole test period in trading every second day

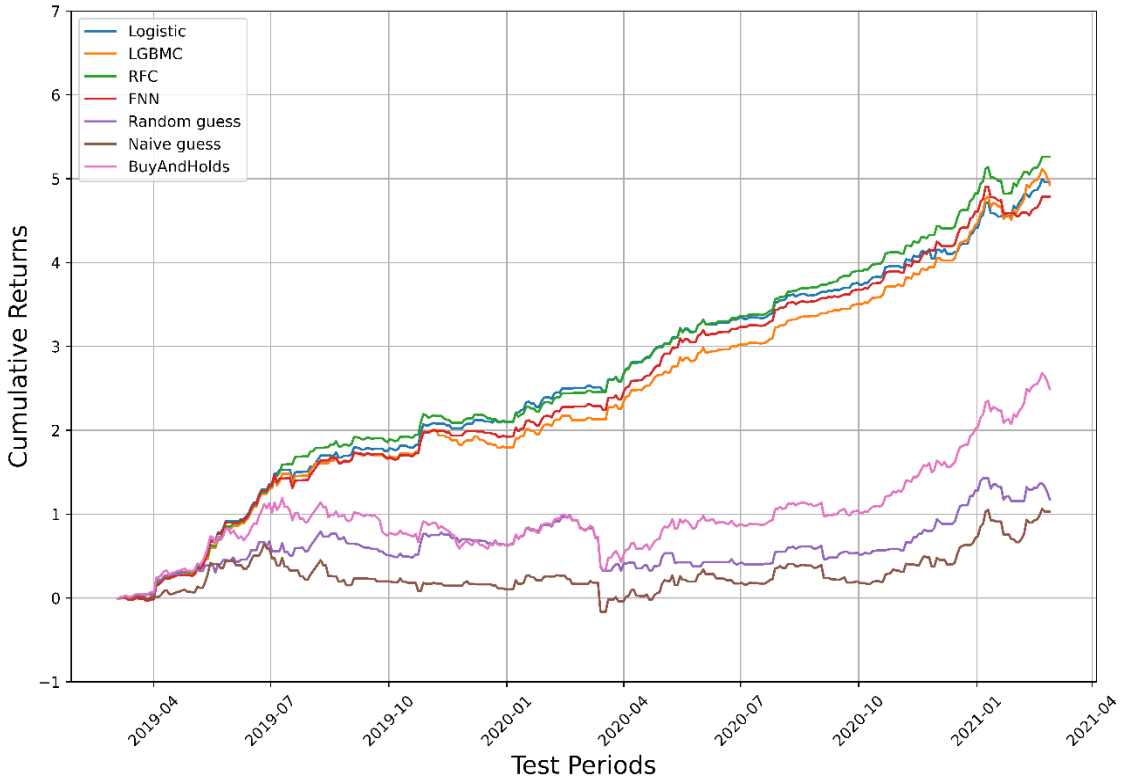
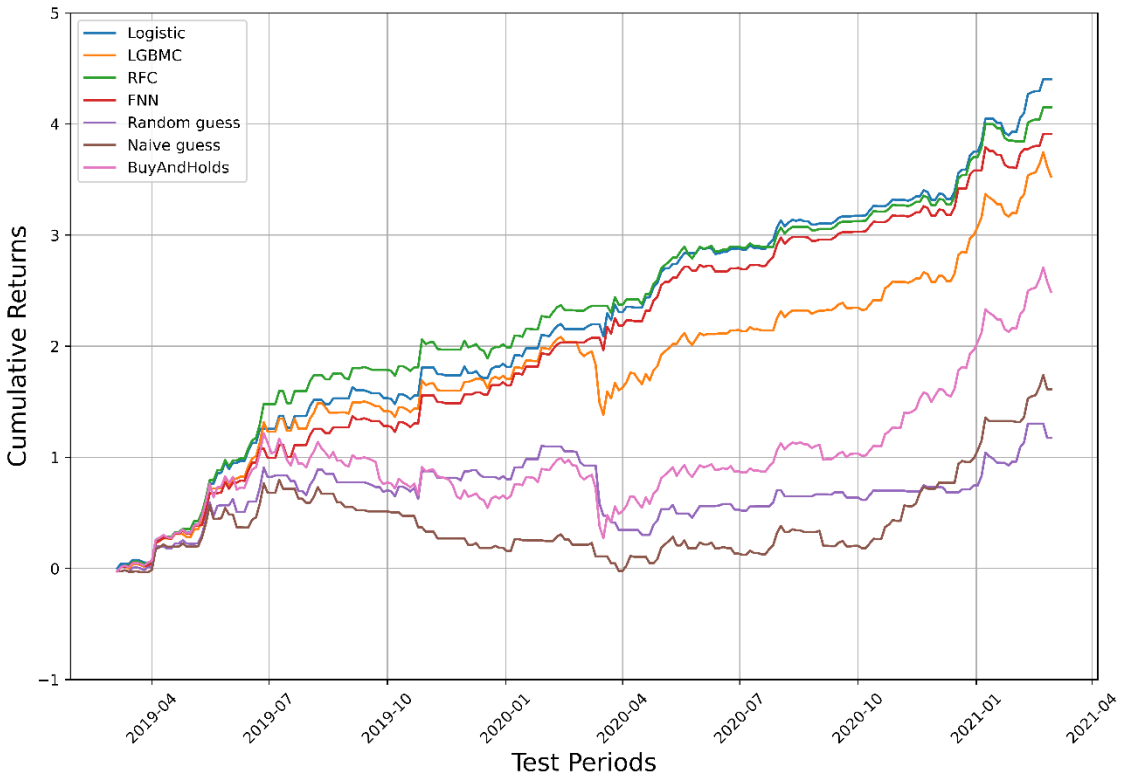


Figure 12: Cumulative returns over the whole test period in trading every three days



5.2.2 Wilcoxon Signed-Rank Test

Tables 11-13 show the results of the Wilcoxon Signed-Rank test on the simulated Bitcoin returns based on End of Day, two, and three days ahead forecasting. Here, we can interpret the economic meaning of the null hypothesis as that a row model does not make higher profits than a column model on average. The values express p value, meaning that when the values are lower than 0.05, the row model outperformed the column model statistically significantly at the critical p value of 0.05. Here, we show the case that the forecasting models used all the three dataset categories as the independent variables.

Table 11: Wilcoxon Signed-Rank Test for everyday trading

model	Logistic	LGBMC	RFC	FNN	**BuyAndHolds**	**Random guess**	**Naive guess**
Logistic	-	0.871	0.849	0.996	**0.000*** **	**0.000*** **	**0.000*** **
LGBMC	0.129	-	0.351	0.749	**0.000*** **	**0.000*** **	**0.000*** **
RFC	0.151	0.649	-	0.903	**0.000*** **	**0.000*** **	**0.000*** **
FNN	0.004***	0.251	0.097*	-	**0.000*** **	**0.000*** **	**0.000*** **
BuyAndHolds	1.000	1.000	1.000	1.000	-	0.054*	0.000***
Random guess	1.000	1.000	1.000	1.000	1.000	-	0.049**
Naive guess	1.000	1.000	1.000	1.000	1.000	0.951	-

The values are the p values. Thus, *** indicates the 0.01 significance level, ** indicates the 0.05 significance level, and * indicates the 0.1 significance level. The bold font shows the comparisons between our trading strategies and the benchmarks.

Table 12: Wilcoxon Signed-Rank Test for trading every second day

model	Logistic	LGBMC	RFC	FNN	**BuyAndHolds**	**Random guess**	**Naive guess**
Logistic	-	0.307	0.750	0.427	**0.000*** **	**0.000*** **	**0.000*** **
LGBMC	0.693	-	0.880	0.650	**0.000*** **	**0.000*** **	**0.000*** **
RFC	0.250	0.120	-	0.223	**0.000*** **	**0.000*** **	**0.000*** **
FNN	0.573	0.350	0.777	-	**0.000*** **	**0.000*** **	**0.000*** **
BuyAndHolds	1.000	1.000	1.000	1.000	-	0.003***	0.000***
Random guess	1.000	1.000	1.000	1.000	1.000	-	0.372
Naive guess	1.000	1.000	1.000	1.000	1.000	0.628	-

The values are the p value. *** indicates the 0.01 significance level. The bold font shows the comparisons between our trading strategies and the benchmarks.

Table 13: Wilcoxon Signed-Rank Test for trading every three days

model	Logistic	LGBMC	RFC	FNN	**BuyAndHolds**	**Random guess**	**Naive guess**
Logistic	-	0.060	0.252	0.118	**0.000*** **	**0.000*** **	**0.000*** **
LGBMC	0.940	-	0.865	0.779	**0.000*** **	**0.000*** **	**0.000*** **
RFC	0.748	0.135	-	0.381	**0.000*** **	**0.000*** **	**0.000*** **
FNN	0.882	0.221	0.619	-	**0.000*** **	**0.000*** **	**0.000*** **
BuyAndHolds	1.000	1.000	1.000	1.000	-	0.025**	0.005***
Random guess	1.000	1.000	1.000	1.000	1.000	-	0.363
Naive guess	1.000	1.000	1.000	1.000	1.000	0.637	-

The values are the p value. *** indicates the 0.01 significance level, ** indicates the 0.05 significance level. The bold font shows the comparisons between our trading strategies and the benchmarks.

The above tables illustrate that we could reject the null hypothesis regarding our trading strategies and the benchmarks. Thus, all the trading strategies based on our forecasting models made higher profits than a Random guess, Naive guess, and Buy & Hold strategy statistically significantly at the critical p value of 0.01 for all the cases: Everyday trading, trading every second day, and trading every three days. In addition, FNN outperformed Logistic Regression and RFC statistically significantly in everyday trading at the critical p value of 0.01 and 0.1, respectively. However, we could not identify which models are better than others in trading every second day and every three days. Therefore, it is better to build multiple models and compare them when trading in different timesteps.

5.3 Comparison with Previous Studies

In forecasting accuracy, Mudassir (2020) built FNN, SVM, SANN, and LSTM models with the Bitcoin blockchain data between 2013 and 2019 to forecast End of Day return and achieved 65% forecasting accuracy, while our forecasting models achieved more than 70% accuracy in End of Day return forecasting between 2019 and 2021. On the other hand, in terms of profitability, Shintate (2019) concluded that their cumulative returns generated from their trading strategies based on forecasting did not outperform Buy & Holds strategy in a statistically significant way between 2013 and 2017. Besides, Amjad (2016) found that their cumulative returns were 6-7x, 4-6x, 3-6x in 2014, 2015, and 2016, respectively, achieved by forecasting Bitcoin returns every 5-second. In contrast, our trading strategies achieved cumulative returns of 6.5-8x in everyday trading, 4.5-5.5x in trading every second day, and 3.5-4.5x in trading every three days between 2019 and 2021. Even though the efficiency of the Bitcoin market has increased since 2019, our forecasting accuracy and profitability were higher than in previous studies.

The possible reasons for this difference are as below. First, we used various variables of the Bitcoin blockchain and the Altcoins market in the forecasting models. Then, we employed the variables selection algorithm Boruta that contributed to finding relevant variables. The previous studies focused on only Bitcoin variables. Furthermore, they dropped some blockchain variables and did not use them. Second, we optimized the hyperparameters with Optuna that could find more optimal values. Specifically, it is known that the hyperparameters affect the performance of Neural Network models. Therefore, finding the optimal hyperparameters should be critical to achieving higher forecasting accuracy.

6. Conclusion

From the experiment, we concluded two points. First, the Bitcoin market was efficient in a weak form between 2019 and 2021. The Ljung & Box test clarified that Bitcoin returns did not have autocorrelation at the 0.05 significance level, indicating that using only its past prices could not improve the forecasting ability of the models. This result means the market became more efficient between 2019 and 2021 than before. On the other hand, the Bitcoin market was still inefficient enough to make higher profits than the benchmarks by forecasting future returns with the publicly available data from the Bitcoin blockchain and the Altcoins market. The trading strategies based on the forecasting models made higher cumulative returns than the benchmarks. Besides, the Wilcoxon Signed-Rank test clarified that our trading strategies yielded higher daily returns than the benchmarks at the 0.01 significance level. Second, the variables of the Bitcoin blockchain and the Altcoins market were essential to forecast Bitcoin returns in End of Day, two, and three days ahead. The variables selection algorithm, Boruta, chose the variables mainly from the categories of the Bitcoin blockchain and the Altcoins market rather than macroeconomic variables in a statistically significant way. In addition, the models using the variables of the Bitcoin blockchain and the Altcoins market outperformed those not using them in the f1 and accuracy scores.

This paper still has its limitations. First, we did not choose a specific cryptocurrency exchange to collect the Bitcoin price data. In order to implement trading strategies based on return forecasting, investors need to choose a specific cryptocurrency exchange. Therefore, they need to input different data into the forecasting models for constructing trading strategies. Second, we could not identify how each independent variable affects the dependent variable. Because the priority of this paper was to improve the forecasting accuracy, we did not remove variables to eliminate multicollinearity. When multicollinearity exists, interpreting the coefficients becomes unreliable, making it challenging to understand the effects of each independent variable. Using other more explainable approaches may clarify the relationship between Altcoins returns and Bitcoin returns. Third, we did not consider transaction costs in our trading simulation. Although we assumed this is not an essential issue because many cryptocurrency exchanges worldwide impose almost zero transaction costs, it is better to include transaction costs for a more rigorous profitability analysis.

Finally, our economic implication is that investors need to consider the internal factors of the Bitcoin blockchain and the external factors of the cryptocurrency market, rather than relying solely on the prices to forecast Bitcoin returns. As the miners mine the Bitcoin for miner revenues, investors should mine the data for investing profits.

Bibliography

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Amjad, M. J., & Shah, D. (2016). Trading Bitcoin and Online Time Series Prediction. *Proceedings of the Time Series Workshop at NIPS 2016 in PMLR*, 1-15.

Antonopoulos, A. M. (2017). *Mastering Bitcoin, 2nd Edition.* O'Reilly Media.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer-VerlagBerlin, Heidelberg.

Blockchain.com. (2021). *Blockchain.com Charts Summary*. Retrieved 03 10, 2021, from Blockchain.com: https://www.blockchain.com/charts

Bloomberg. (2021, 02 19). *Bitcoin Hits $1 Trillion Value as Crypto Leads Other Assets.* (E. Lam, Editor) Retrieved 04 23, 2021, from Bloomberg.com: https://www.bloomberg.com/news/articles/2021-02-19/bitcoin-nears-1-trillion-value-as-crypto-jump-tops-other-assets

Bloomenthal, A. (2021, 03 16). *What Determines the Price of 1 Bitcoin?* Retrieved 04 26, 2021, from Investopedia: https://www.investopedia.com/tech/what-determines-value-1-bitcoin/

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics, 365*(112395). doi:10.1016/j.cam.2019.112395

CoinMarketCap. (2021). *Global Cryptocurrency Charts*. Retrieved 04 27, 2021, from CoinMarketCap: https://coinmarketcap.com/charts/

Dixon, M., Klabjan, D., & Bang, J. H. (2017). Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance, 6*(3-4), 67-77. doi:10.2139/ssrn.2756331

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance, 25*(2), 383-417. doi:10.2307/2325486

Garcia, D., & Schweitzer, F. (2015). Social signals and algorithmic trading of Bitcoin. *ROYAL SOCIETY OPEN SCIENCE, 2*(9). doi:10.1098/rsos.150288

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157-1182.

Jang, H., & Lee, J. (2017). An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *IEEE Access, 6*, 5427–5437. doi:10.1109/ACCESS.2017.2779181

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., . . . Liu, T. Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems, 30*, 3149-3157.

Kursa, M. B., & Rudnicki, W. R. (2010, 9). Feature selection with the boruta package. *Journal of Statistical Software, 36*(11). doi:10.18637/jss.v036.i11

Kyriazis, N. (2019). A Survey on Efficiency and Profitable Trading Opportunities in Cryptocurrency Markets. *Journal of Risk and Financial Management, 12*(67). doi:10.3390/jrfm12020067

LightGBM. (n.d.). *LightGBM*. Retrieved 05 21, 2021, from Features - LightGBM 3.2.1.99 documentation: https://lightgbm.readthedocs.io/en/latest/Features.html

MacKinnon, J. G. (2010). Critical values for cointegration tests. *Queen's Economics Department Working Paper, No. 1227*(1227).

Mallqui, D. C., & Fernandes, R. A. (2019). Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Applied Soft Computing Journal, 75*, 596–606.

McNally, S. (2016). *Predicting the price of Bitcoin using Machine Learning.* National College of Ireland.

Megas, J. (Ed.). (2020, 02 08). *Truth About Crypto Price Correlation: How Closely Does ETH Follow BTC?* Retrieved 04 18, 2021, from Cointelegraph: https://cointelegraph.com/news/truth-about-crypto-price-correlation-how-closely-does-eth-follow-btc

Mudassir, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications, 4*, 1-15. doi:10.1007/s00521-020-05129-6

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* www.bitcoin.org.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Scipy.org. (2021, 4 26). *scipy.stats.wilcoxon*. Retrieved 05 25, 2021, from scipy.stats.wilcoxon - SciPy v1.6.3 Reference Guide: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html

Shintate, T., & Pichl, L. (2019). Trend Prediction Classification for High Frequency. *Journal of Risk and Financial Management, 12*(17). doi:10.3390/jrfm12010017

Tsay, R. S. (2010). *Analysis of Financial Time Series.* John Wiley & Sons, Inc.

Vidal, T. (Ed.). (2020, 02 17). *Does correlation between bitcoin price and Altcoins mean buy the Dips?* Retrieved from Cointelegraph: https://cointelegraph.com/news/does-correlation-between-bitcoin-price-and-altcoins-mean-buy-the-dips

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin, 1*(6), 80-83.

Xiaolei, S., Mingxi, L., & Zeqian, S. (2020). A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters, 32*.

Yahoo! Finance. (2021). *Cryptocurrencies with Highest Market Cap*. Retrieved 03 10, 2021, from Yahoo! Finance: https://finance.yahoo.com/u/yahoo-finance/watchlists/crypto-top-market-cap?.tsrc=fin-srch

Yahoo! Finance. (2021). *Stock Market Live, Quotes, Business & Finance News*. Retrieved 03 10, 2021, from Yahoo! Finance: https://finance.yahoo.com/

Appendix

Table A1: Definition of variables and Abbreviations

Variables	Categories	Definition of Variable
BTC_Price	BTC prices	The average USD market price across major bitcoin exchanges. (USD)
total-bitcoins	BTC blockchain	The total number of mined bitcoins that are currently circulating on the network. (BTC)
market-cap	BTC blockchain	The total USD value of bitcoin in circulation. (USD)
trade-volume	BTC blockchain	The total USD value of trading volume on major bitcoin exchanges. (USD)
blocks-size	BTC blockchain	The total size of the blockchain minus database indexes in megabytes. (MB)
avg-block-size	BTC blockchain	The average block size over the past 24 hours in megabytes. (MB)
n-transactions-per-block	BTC blockchain	The average number of transactions per block over the past 24 hours. (transaction)
n-transactions-total	BTC blockchain	The total number of transactions on the blockchain. (transaction)
median-confirmation-time	BTC blockchain	The median time for a transaction with miner fees to be included in a mined block and added to the public ledger. (min)
avg-confirmation-time	BTC blockchain	The average time for a transaction with miner fees to be included in a mined block and added to the public ledger. (min)
hash-rate	BTC blockchain	The estimated number of terahashes per second the Bitcoin network is performing in the last 24 hours. (TH/s)
difficulty	BTC blockchain	A relative measure of how difficult it is to mine a new block for the blockchain. (relative)
miners-revenue	BTC blockchain	The total value of coinbase block rewards and transaction fees paid to miners. (USD)
transaction-fees	BTC blockchain	The total BTC value of all transaction fees paid to miners. (BTC)
cost-per-transaction	BTC blockchain	Miners' revenue divided by the number of transactions. (USD)
n-unique-addresses	BTC blockchain	The total number of unique addresses used on the blockchain. (address)
n-transactions	BTC blockchain	The total number of confirmed transactions per day. (transaction)
transactions-per-second	BTC blockchain	The number of transactions added to the mempool per second. (transaction/s)
output-volume	BTC blockchain	The total value of all transaction outputs per day. (BTC)
mempool-count	BTC blockchain	The total number of unconfirmed transactions in the mempool. (transaction)
mempool-growth	BTC blockchain	The rate at which the mempool is growing in bytes per second. (byte/s)
mempool-size	BTC blockchain	The aggregate size in bytes of transactions waiting to be confirmed. (byte)
utxo-count	BTC blockchain	The total number of valid unspent transaction outputs. (transaction)
n-transactions-excluding-popular	BTC blockchain	The total number of transactions excluding those involving the network's 100 most popular addresses. (transaction)
estimated-transaction-volume	BTC blockchain	The total estimated value in BTC of transactions on the blockchain. (BTC)
mvrv	BTC blockchain	Market Value divided by Realised Value. (ratio)
nvts	BTC blockchain	The Moving Average over the last 90 days of the denominator of Network Value (= Market Value) by the total transactions volume in USD over the past 24hour. (ratio)
Prices and Volumes	Altcoins	Adjusted close Prices and Volumes of ETH-USD, XRP-USD, USDT-USD, LTC-USD, BCH-USD, BNB-USD, XLM-USD, EOS-USD, XMR-USD, XEM-USD, NEO-USD, IOTA-USD, ZEC-USD, ETC-USD, WAVES-USD (USD)
Prices and Volumes	Macroeconomics	Adjusted close Prices and Volumes of sp500, eurostoxx, dow30, Nasdaq, crude oil, SSE, gold, vix, nikkei225, GBP-USD, JPY-USD, CHF-USD, CNY-USD, EUR-USD (USD)

All the definitions follow Blockchain.com (2021). Nakamoto (2008) explains the details of the blockchain mechanics as the original Bitcoin white paper.

Table A2: Summary statistics of the preprocessed independent variables

Variables	mean	std	min	50%	max	skewness
BCH_Price_LogReturn	0.000	0.069	-0.559	-0.002	0.431	0.010
BCH_Volume_LogReturn	0.001	0.323	-1.239	-0.022	2.147	0.802
BNB_Price_LogReturn	0.004	0.058	-0.503	0.001	0.527	0.771
BTC_Price_LogReturn	0.002	0.044	-0.497	0.002	0.197	-1.115
EOS_Price_LogReturn	0.001	0.053	-0.355	0.000	0.304	0.219
ETC_Price_LogReturn	0.000	0.055	-0.423	0.000	0.300	-0.824
ETH_Price_LogReturn	0.001	0.052	-0.547	0.001	0.234	-1.107
IOTA_Price_LogReturn	0.000	0.031	-0.304	0.000	0.267	0.475
IOTA_Volume_LogReturn	0.001	0.388	-1.731	-0.023	1.825	0.449
LTC_Price_LogReturn	0.001	0.056	-0.438	0.000	0.388	0.096
NEO_Price_LogReturn	0.001	0.060	-0.410	0.000	0.334	0.042
USDT_Price_LogReturn	0.000	0.003	-0.027	0.000	0.029	0.603
USDT_Volume_LogReturn	0.005	0.230	-1.294	-0.006	1.636	0.393
WAVES_Price_LogReturn	0.001	0.048	-0.249	0.000	0.327	0.210
XEM_Price_LogReturn	0.000	0.018	-0.161	0.000	0.282	4.382
XLM_Price_LogReturn	0.000	0.013	-0.104	0.000	0.192	3.018
XMR_Price_LogReturn	0.001	0.055	-0.483	0.000	0.247	-0.669
XRP_Price_LogReturn	0.000	0.024	-0.236	0.000	0.279	1.628
XRP_Volume_LogReturn	0.003	0.375	-1.963	-0.018	2.322	0.647
ZEC_Price_LogReturn	0.000	0.058	-0.401	-0.002	0.260	-0.187
cost-per-transaction_LogReturn	0.001	0.122	-0.744	0.007	0.413	-0.474
gbp_usd_Price_LogReturn	0.000	0.003	-0.023	0.000	0.016	-0.188
gold_Price_LogReturn	0.000	0.008	-0.051	0.000	0.058	-0.129
market-cap_LogReturn	0.002	0.040	-0.258	0.002	0.201	-0.232
mempool-count_LogReturn	0.001	1.155	-8.620	-0.009	7.189	0.007
mempool-size_LogReturn	0.002	1.485	-17.160	0.005	15.920	-0.252
mvrv_LogReturn	0.000	0.026	-0.180	0.000	0.130	-0.256
nvts_LogReturn	0.000	0.039	-0.283	0.001	0.181	-0.456
utxo-count_LogReturn	0.000	0.002	-0.009	0.000	0.007	-1.154

In every variable, the number of data points is 1263. The variables in this table are the variables that were chosen by the variables selection algorithm Boruta at least once as the independent variables in the experiments.

Table A3: The results of the ADF test on the raw data variables

variables	test statistic	p value
BCH_Price	-2.216	0.201
BCH_Volume	-2.787	0.060*
BNB_Price	2.093	0.999
BTC_Price	4.327	1.000
EOS_Price	-2.782	0.061*
ETC_Price	-1.686	0.438
ETH_Price	0.757	0.991
IOTA_Price	-2.919	0.043**
IOTA_Volume	-3.594	0.006***
LTC_Price	-1.822	0.370
NEO_Price	-1.778	0.392
USDT_Price	-5.949	0.000***
USDT_Volume	1.880	0.998
WAVES_Price	-1.593	0.487
XEM_Price	-2.740	0.067*
XLM_Price	-2.251	0.188
XMR_Price	-1.713	0.424
XRP_Price	-2.942	0.041**
XRP_Volume	-1.885	0.339
ZEC_Price	-1.866	0.348
cost-per-transaction	0.096	0.966
gbp_usd_Price	-2.212	0.202
gold_Price	-0.559	0.880
market-cap	5.098	1.000
mempool-count	-2.416	0.137
mempool-size	-3.035	0.032**
mvrv	-1.125	0.705
nvts	-3.182	0.021**
utxo-count	-0.616	0.867

*** indicates the 0.01 significance level, ** indicates the 0.05 significance level, and * indicates the 0.1 significance level. The variables in this table are the variables that were chosen by the variables selection algorithm Boruta at least once as the independent variables in the experiments.

Table A4: The results of the ADF test on the preprocessed variables

variables	test statistic	p value
BCH_Price_LogReturn	-34.874	0.000***
BCH_Volume_LogReturn	-9.799	0.000***
BNB_Price_LogReturn	-7.048	0.000***
BTC_Price_LogReturn	-24.518	0.000***
EOS_Price_LogReturn	-13.713	0.000***
ETC_Price_LogReturn	-13.740	0.000***
ETH_Price_LogReturn	-24.094	0.000***
IOTA_Price_LogReturn	-6.171	0.000***
IOTA_Volume_LogReturn	-14.196	0.000***
LTC_Price_LogReturn	-36.156	0.000***
NEO_Price_LogReturn	-7.351	0.000***
USDT_Price_LogReturn	-12.419	0.000***
USDT_Volume_LogReturn	-8.724	0.000***
WAVES_Price_LogReturn	-12.370	0.000***
XEM_Price_LogReturn	-4.998	0.000***
XLM_Price_LogReturn	-7.693	0.000***
XMR_Price_LogReturn	-13.217	0.000***
XRP_Price_LogReturn	-7.611	0.000***
XRP_Volume_LogReturn	-11.092	0.000***
ZEC_Price_LogReturn	-12.974	0.000***
cost-per-transaction_LogReturn	-7.127	0.000***
gbp_usd_Price_LogReturn	-12.770	0.000***
gold_Price_LogReturn	-7.719	0.000***
market-cap_LogReturn	-9.235	0.000***
mempool-count_LogReturn	-10.554	0.000***
mempool-size_LogReturn	-10.918	0.000***
mvrv_LogReturn	-36.569	0.000***
nvts_LogReturn	-34.029	0.000***
utxo-count_LogReturn	-3.331	0.014**

*** indicates the 0.01 significance level, and ** indicates the 0.05 significance level. The variables in this table are the variables that were chosen by the variables selection algorithm Boruta at least once as the independent variables in the experiments.

TableA5: The list of the independent variables chosen by Boruta

Variables	Categories	End of Day	Two days ahead	Three days ahead
BCH_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
BCH_Volume_LogReturn	Altcoins	-	-	**2**
BNB_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
EOS_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
ETC_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
ETH_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
IOTA_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
IOTA_Volume_LogReturn	Altcoins	-	**0**	-
LTC_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
NEO_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
USDT_Price_LogReturn	Altcoins	-	**1**	-
USDT_Volume_LogReturn	Altcoins	**1, 2**	**0**	**0**
WAVES_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0**
XEM_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
XLM_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
XMR_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
XRP_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
XRP_Volume_LogReturn	Altcoins	**2**	-	-
ZEC_Price_LogReturn	Altcoins	**0, 1**	**0, 1**	**0, 1**
BTC_Price_LogReturn	BTC prices	**0**	-	-
cost-per-transaction_LogReturn	BTC blockchain	-	**0**	-
gbp_usd_Price_LogReturn	Macroeconomics	-	-	**1**
gold_Price_LogReturn	Macroeconomics	**0**	-	-
market-cap_LogReturn	BTC blockchain	**0**	**0**	**0**
mempool-count_LogReturn	BTC blockchain	-	**1**	-
mempool-size_LogReturn	BTC blockchain	**2**	**1**	-
mvrv_LogReturn	BTC blockchain	**0**	**0**	**0**
nvts_LogReturn	BTC blockchain	**0**	**0**	**0**
utxo-count_LogReturn	BTC blockchain	**0, 1**	-	-

In each element, the value 0 means no-lagged variable was used, the value 1 means one-lagged variable was used, and the value 2 means two lagged variable was used. - means that the variable was not used in the nth days ahead forecasting. The variables in this table are the variables that were chosen by the variables selection algorithm Boruta at least once as the independent variables in the experiments.

Table A6: The f1 scores of the test period in End of Day return forecasting

CV	Test start	Test end	Logistic	LGBMC	RFC	FNN
1	2019-03-03	2019-05-01	0.657	0.780	0.694	0.657
2	2019-05-02	2019-06-30	0.854	0.800	0.878	0.857
3	2019-07-01	2019-08-29	0.825	0.761	0.813	0.780
4	2019-08-30	2019-10-28	0.667	0.688	0.677	0.596
5	2019-10-29	2019-12-27	0.536	0.731	0.577	0.588
6	2019-12-28	2020-02-25	0.857	0.841	0.771	0.889
7	2020-02-26	2020-04-25	0.794	0.774	0.762	0.794
8	2020-04-26	2020-06-24	0.635	0.677	0.639	0.625
9	2020-06-25	2020-08-23	0.750	0.800	0.771	0.795
10	2020-08-24	2020-10-22	0.667	0.886	0.794	0.784
11	2020-10-23	2020-12-21	0.683	0.692	0.667	0.703
12	2020-12-22	2021-02-19	0.685	0.737	0.712	0.676

The first day of the test period is different from that in two days ahead or three days ahead forecasting due to the preprocessing.

Table A7: The f1 scores of the test period in two days ahead return forecasting

CV	Test start	Test end	Logistic	LGBMC	RFC	FNN
1	2019-03-04	2019-05-02	0.647	0.667	0.675	0.627
2	2019-05-03	2019-07-01	0.864	0.861	0.864	0.850
3	2019-07-02	2019-08-30	0.794	0.794	0.758	0.806
4	2019-08-31	2019-10-29	0.727	0.719	0.727	0.710
5	2019-10-30	2019-12-28	0.560	0.491	0.519	0.489
6	2019-12-29	2020-02-26	0.744	0.630	0.709	0.718
7	2020-02-27	2020-04-26	0.762	0.787	0.794	0.769
8	2020-04-27	2020-06-25	0.757	0.785	0.810	0.789
9	2020-06-26	2020-08-24	0.809	0.822	0.907	0.806
10	2020-08-25	2020-10-23	0.845	0.841	0.873	0.857
11	2020-10-24	2020-12-22	0.691	0.703	0.711	0.701
12	2020-12-23	2021-02-20	0.723	0.835	0.729	0.713

The first day of the test period is different from that in End of Day or three days ahead forecasting due to the preprocessing.

Table A8: The f1 scores of the test period in three days ahead return forecasting

CV	Test start	Test end	Logistic	LGBMC	RFC	FNN
1	2019-03-05	2019-05-03	0.718	0.718	0.824	0.741
2	2019-05-04	2019-07-02	0.776	0.767	0.800	0.762
3	2019-07-03	2019-08-31	0.746	0.667	0.724	0.759
4	2019-09-01	2019-10-30	0.647	0.685	0.686	0.657
5	2019-10-31	2019-12-29	0.500	0.567	0.571	0.408
6	2019-12-30	2020-02-27	0.720	0.700	0.667	0.737
7	2020-02-28	2020-04-27	0.697	0.723	0.688	0.687
8	2020-04-28	2020-06-26	0.629	0.747	0.725	0.639
9	2020-06-27	2020-08-25	0.787	0.559	0.693	0.763
10	2020-08-26	2020-10-24	0.692	0.776	0.720	0.730
11	2020-10-25	2020-12-23	0.773	0.736	0.658	0.700
12	2020-12-24	2021-02-21	0.791	0.835	0.707	0.705

The first day of the test period is different from that in two days ahead or three days ahead forecasting due to the preprocessing.