Lund University

Master's thesis

NEKN02 Finance

Out of the Books and Into the Woods

Predicting short-term crypto returns from limit order books using Random forest

Author
Elisabeth Molin

Supervisor
Anders Vilhelmsson

Spring 2023
Department of Economics

Lund University

Abstract

The objective of this thesis is to analyze the predictive power of the cryptocurrency limit order book for return predictions. The analysis is performed for the BTC/USD and ETH/USD directional 10-second returns, using limit order book data from three of the largest cryptocurrency spot exchanges. The analysis employs the Random forest algorithm as a classification problem. The results demonstrate that Coinbase achieves the highest average F1 scores (accuracy), followed by Bitfinex and Gemini. When utilizing the most recent period's features for the predictions, the F1 scores consistently exceed those of random chance, providing empirical evidence for the predictive potential of limit order book data. It is, however, observed that the accuracy varies greatly depending on the test set and the number of periods by which features are lagged. Furthermore, the study investigates the lead-lag relationships among the exchanges and the effects on predictions. Findings suggest Coinbase as the leading exchange for BTC, while indicating that Gemini was the leading exchange for ETH. However, interpreting the results for ETH is challenging due to highly imbalanced data and methodological choices. Overall, this study underscores the predictive power of limit order book data for cryptocurrency spot returns.

Keywords: Limit Order Book, Cryptocurrency, Random forest, Market Microstructure, Lead-lag

Contents

1 Introduction

This paper explores the predictive potential of cryptocurrency spot returns by building upon three interconnected areas of research. Firstly, it contributes to the body of literature on the general predictability of crypto spot returns and its implications for fundamental theories in financial markets. As a relatively new asset class with unique characteristics, cryptocurrencies have become subject to extensive research, and several studies have endeavored to predict crypto returns. This task has proven to be challenging due to the market's high volatility and the lack of fundamental factors (Gradojevic et al., 2023). Nevertheless, many researchers have found evidence of predictability in crypto returns (Akyildirim et al., 2021; Gradojevic et al., 2023; Jaquart et al., 2021). This has raised questions about the efficiency of the market, with some arguing that the efficiency (and thus predictability) varies over time (Chu et al., 2019; Gradojevic et al., 2023; Kyriazis, 2019; Tran & Leirvik, 2020).

Secondly, this thesis expands upon the relatively scarce area of research involving the use of Limit Order Book (LOB) data for predictions in the cryptocurrency market. The LOB is a log file where limit orders are recorded and stored until being canceled or executed against subsequent orders. In order-driven markets, the LOB plays an essential role since it can reveal the price dynamics on a microstructural level through the interaction of buy and sell orders at different prices and quantities (Abergel et al., 2016). While some researchers have used crypto LOB data, their studies have been limited to small samples and focused on predicting mid-prices (Fang et al., 2021; Jha et al., 2020). These studies have attained highly accurate results by employing advanced deep learning models and ultra high-frequency data. Nonetheless, it remains uncertain whether less advanced models, such as the Random forest, with lower-frequency data and larger sample sizes, can produce predictions of comparable accuracy for actual returns.

Lastly, this study addresses the lead-lag phenomenon. In this context, lead-lag refers to the phenomenon where one exchange leads the price movement over another exchange with a time delay. Research has found that large exchanges tend to lead smaller exchanges, meaning that prices at the small exchanges trail behind (Alexander & Heck, 2020; Blasco & Corredor, 2022; Brandvold et al., 2015; Schei Norheim, 2019). Crypto research that incorporates lead-lag relationships in predictions has focused on the crypto futures market (Albers et al., 2021), leaving the question of whether lead-lag relationships can improve return predictions in the crypto spot market unanswered.

The purpose of this paper is to investigate the applicability of using LOB data in predicting cryptocurrency returns at 10-second frequencies, and in identifying potential lead-lag relationships between spot exchanges. Additionally, many crypto papers have focused on Bitcoin. I aim to broaden the analysis by considering not only Bitcoin but also Ethereum, the second-largest crypto. Both cryptocurrencies are analyzed in dollar denomination and will be referred to as BTC and ETH throughout the paper.

To achieve the purpose, the following research questions are addressed:

- How accurately can the BTC and ETH directional spot returns at individual exchanges be predicted with the use of LOB data?
- Does the accuracy improve by the inclusion of cross-exchange information?

To answer these questions, 10-second frequency data from three of the largest crypto spot exchanges is used. Features are extracted from the LOBs and the machine learning model Random forest classifier is used to predict the sign of returns. The Random forest model has demonstrated high accuracy in predicting returns (Gradojevic et al., 2023; Qureshi, 2018), making it a suitable choice for the analysis. In the first step of the analysis, single-exchange predictions are performed for each exchange and crypto. Then, by the addition of another exchange's features, cross-exchange predictions are made, which can reveal potential lead-lag relationships. By comparing single- and cross-exchange predictions, the methodology provides a means to assess the effect of lead-lag on the accuracy of the predictions.

The remainder of this thesis is organized as follows. Section 2 presents previous literature on the predictability of crypto returns, the use of LOB in predictions and the lead-lag phenomenon. Section 3 describes the trade and LOB data, features and target variable used. Section 4 describes the Random forest, how it is implemented, how data is pre-processed and how class imbalances are dealt with. The results are presented and analyzed in section 5. In section 6, the results are discussed by comparing them to previous literature. Finally, section 7 draws a conclusion.

2 Literature

This paper builds upon three interconnected areas of research. The first is the general predictability of crypto spot returns and the conjunction to fundamental theories on financial markets. The second concerns the use of limit order book data for predictions. The third area pertains to the lead-lag phenomenon and cross-exchange predictability. It is worth noting that the crypto literature presented herein is with regards to the crypto *spot* market, unless explicitly stated otherwise.

2.1 The predictability of crypto spot returns

The predictability of financial asset returns has for long attained considerable interest in the research world, and cryptocurrency returns are no exception. Many studies have performed predictions of crypto returns, often linking the results to fundamental theories on financial markets. A widely considered financial theory when making predictions is the Efficient Market Hypothesis (EMH), introduced by Fama (1970), which suggests that financial markets are highly efficient; meaning that all available information is quickly reflected in asset prices. Fama (1970) proposed three degrees of market efficiency, and in the weak form, it should not be possible to predict an asset's price based on its past price information. The weak form, which is the most commonly employed form, appears to be disputed in crypto market research. The majority of papers using daily sampling frequencies provide evidence against weak-form efficiency in the crypto market (Kyriazis, 2019). When considering lower sampling frequencies, the findings are more split. For example, Gradojevic et al. (2023) focused on BTC and found pricing inefficiencies at daily frequencies but not hourly. Conversely, Akyildirim et al. (2021) found inefficiencies at both hourly and minute-level frequencies. Jaquart et al. (2021) also studied minute-level frequencies and argued that when accounting for trading costs, the predictions did not challenge weak-form efficiency. Furthermore, other studies conducted in the crypto market have provided evidence contradicting the semi-strong form of the EMH (Fischer et al., 2019; Kang et al., 2022), which expands upon the weak form by stating that prices reflect all publicly available information (Fama, 1970).

The apparent lack of consensus in terms of the EMH is not isolated to the crypto market, and other theoretical frameworks have been developed to depart from perceiving efficiency as an all-or-nothing setup. One such theory is the Adaptive Market Hypothesis (AMH) proposed by Lo (2004). In this framework, the degree of market efficiency (and thus predictability) evolves over time as environmental conditions and the behavior of market participants changes (Lo, 2004). Several researchers have found support for this theory in the crypto market (Chu et al., 2019; Gradojevic et al., 2023; Khuntia & Pattanayak, 2018; Tran & Leirvik, 2020), even when employing different methods, sample frequencies, cryptocurrencies and time periods. Gradojevic et al. (2023) provided empirical evidence supporting the idea that the degree of market efficiency varies over time, and specifically, they found that the predictive performances of their models (including the Random forest) substantially decreased during periods of high volatility.

2.2 Prediction with information from the LOB

As described above, many studies have performed predictions of cryptocurrency returns, but most of these use market-, asset-, blockchain-, and/or sentiment-based features without incorporating information from the LOB (Akyildirim et al., 2021; Guo et al., 2021; Huang et al., 2019; Jaquart et al., 2021; Kumar, 2021; Rathore et al., 2022; Valencia et al., 2019). However, when studying order-driven markets, it can be advantageous to utilize LOB data since it provides high-resolution price dynamics (Abergel et al., 2016). Yet, there are only a few crypto forecasting papers leveraging LOB data, and they typically use advanced deep learning methods with ultra-high frequency data. For example, Jha et al. (2020) predict BTC 100 millisecond mid-price movements and obtain 71% accuracy when applying a neural network (Temporal Convolutional Neural Network) which identifies patterns in the LOB. Fang et al. (2021) use data for both ETH and BTC, manually construct factors from the LOB, and apply a deep learning method (Long short-term memory) separately to each crypto and predict tick-level mid-price changes. The authors find prediction accuracies (F1 scores) ranging between 70.2-81.4% for BTC/USD and 55.5-73.6% for ETH/USD. The mentioned models are markedly advanced, which is sometimes necessary for detailed high-frequency data. Nevertheless, it raises the question of whether less advanced models with lower frequency LOB and trade data can still produce highly accurate predictions. I address this question by implementing the less advanced Random forest algorithm on 10-second frequency data.

It is worth noting that Jha et al. (2020) use only 9 days of data in June 2019, and Fang et al. (2021) use only 1 day of data in July 2018. This thesis' data spans over one month, which can facilitate in the detection of overall predictability of crypto returns and the generalizability of the model used. Additionally, Jha et al. (2020) and Fang et al. (2021) predict mid-prices, which is the average of the best ask and best bid price in the LOB. The mid-price is usually used as an approximation of the execution prices, but may not reflect the actual prices at which a trade is executed (Zaznov et al., 2022). I use actual prices to calculate the returns that are being predicted, and this study can therefore add to the prediction literature in a different way than the aforementioned.

Moreover, it should be noted that there are additional crypto papers that consider LOB data, but they are mainly designated to identifying informative features (microstructural signals) from the LOB and/or the dynamics of the LOB, rather than using it for pure prediction (see e.g., Nejat, 2021; Lim, 2022; Silantyev, 2019).

As explained above, this study differs from other crypto LOB prediction papers in that it uses a larger sample size, lower frequency data and employs the more interpretable Random forest model. The combination of Random forest and LOB data has been used in predictions of traditional assets, but primarily through regression rather than classification (see e.g., Petrova & Vilhelmsson, 2023), or with mid-prices as target variable (see e.g., Qureshi, 2018). Since I perform predictions of actual returns as a classification problem, and with other sampling frequencies, the results in such studies are not directly comparable. Yet, it is noteworthy that in their study on fiat currencies, Petrova & Vilhelmsson (2023) overall did not find predictability of returns when using LOB features.

2.3 Lead-lag and cross-exchange predictability

One dimension of crypto market microstructure literature that has grown in recent years is price discovery, the process by which new information is impounded in prices. An interesting topic within this area is the non-synchronous adoption of new information across exchanges, which can lead to contemporaneous price deviations for a given asset at different exchanges. This is particularly relevant in crypto markets because most cryptos are traded on multiple exchanges, and the major crypto pairs like BTC/USD and ETH/USD are currently, in May 2023, traded at 49 exchanges[1]. Several studies have found price discrepancies across crypto exchanges (Alexander & Heck, 2020; Brandvold et al., 2015; Blasco & Corredor, 2022; Schei Norheim, 2019) and have attributed them to several interconnected factors related to the non-synchronous adoption of new information.

One of the main reasons for these discrepancies is that exchanges with large trading volume provide the market with more information than smaller exchanges; for which reason prices at small exchanges trail behind (Blasco & Corredor, 2022; Brandvold et al., 2015). This is commonly referred to as the 'lead-lag' phenomenon, and in the current context it means that large exchanges lead the price discovery over smaller (lagging) exchanges. Alexander & Heck (2020) discovered that among spot exchanges, Coinbase has a leading role in the BTC spot price discovery, followed by Bitstamp and Bitfinex. The authors describe that this leadership can be explained by the exchanges' dominant trading volumes. However, the lead-lag relationship is relatively weak between the exchanges with the largest trading volumes (Schei Norheim, 2019). Furthermore, the lead-lag relationship between exchanges is not always clear, as some exchanges are neither laggers nor leaders (Brandvold et al., 2015). The relationship is also subject to change over time (Blasco & Corredor, 2022; Brandvold et al., 2015). Blasco & Corredor (2022) explain that as the number of exchanges increases, the herding behavior at small exchanges increases, meaning that market participants at small exchanges to a larger extent trade based on market consensus, which is more likely to be determined by large exchanges.

Up until now only spot exchanges have been discussed, but in the crypto market, the derivative exchanges play an important role in the lead-lag relationships. Although derivative exchanges are not within the scope of this thesis, it is worth mentioning their role in the crypto market and their possible impact on the spot exchanges analyzed. Alexander & Heck (2020) describe that the majority of the BTC price discovery happens in the derivative market and that many of the derivative exchanges are unregulated. To grasp the extent of regulatory disparities between many spot and derivative exchanges, consider the following quote:

> While well-established spot exchanges such as Coinbase or Gemini are headquartered in the US and apparently undertake considerable efforts to ensure market integrity, the major derivatives exchanges operate without any regulatory requirements at all; indeed some even fail to perform standard know-your-customer procedures. (Alexander & Heck, 2020, p.2)

[1] https://coinpaprika.com/exchanges/

Alexander & Heck (2020) explain that the lack of regulation enables for price manipulation and makes unregulated derivative exchanges the main driver of price movements across all exchanges, even regulated spot exchanges. In other words, even though the authors find lead-lag relationships between spot exchanges, the prices at these exchanges are still highly determined by unregulated derivative exchanges.

The previous discussion highlights the existence of lead-lag relationships across crypto exchanges. However, it does not explore if and how this knowledge can be used to improve predictions of prices/returns. Interestingly, this seems to be a relatively unexplored area of the crypto research and I could only find one crypto paper, by Albers et al. (2021), investigating this subject. In their study, Albers et al. (2021) analyzed the 500 millisecond predictability of BTC returns at one exchange by using information from another. Their analysis primarily focuses on the predictability of the BTC futures markets, and they only consider one spot exchange. Therefore, they did not measure the predictability between spot exchanges. Additionally, their study relies on a linear regression model, which may have limitations in capturing complex nonlinear relationships.

My analysis differs from that of Albers et al. (2021) in several ways. Firstly, this study expands the scope of the analysis by considering not only BTC but also ETH returns. Secondly, the analysis exclusively uses spot exchange data, collected at a lower frequency. Thirdly, the utilization of the Random forest model allows for the exploration of nonlinear relationships, thereby capturing more complex patterns in the data. Finally, a comparison is made between single- and cross-exchange predictions, offering insights into the potential impact of lead-lag relationships on crypto return predictions. The analysis thus has the potential to contribute to a more comprehensive understanding of lead-lag relationships between crypto spot exchanges and their implications for predictions.

Furthermore, it is worth noting that potential lead-lag relationships between exchanges do not necessarily translate into arbitrage profits if traded upon. There are multiple barriers, including transaction time, fees and geographical restrictions, which can prevent the exploitation of arbitrage opportunities arising from price discrepancies across exchanges (Crépellière et al., 2023; Makarov & Schoar, 2020).

3 Data and Features

I use trade data and limit order book data from three exchanges for the BTC/USD and ETH/USD pairs during September 2019. During the period studied, BTC and ETH constituted approximately 68.9% and 7.9% of the total cryptocurrency market capitalization across all denominations.[2] Considering their dominance, they are of particular importance in the understanding of the overall crypto market dynamics. As crypto exchanges are central in this thesis, I will continue this section by providing a motivation of the choice of exchanges. This section will also provide a description of the data and how it is processed, how features are extracted and how the target variable is defined.

3.1 Exchanges

The data is from three of the largest centralized cryptocurrency exchanges: Coinbase, Bitfinex and Gemini. The reason for choosing these three exchanges is threefold. Firstly, they are considered to better report reliable trading volumes compared to many other exchanges, as they have passed numerous tests for artificial volume (Bitwise Asset Management, 2019) and score highly on Coinmarketcap's ranking system that takes into consideration the legitimacy of reported trading volume[3]. Market manipulation such as artificial inflation of trading volume has been a persistent problem in the cryptocurrency market due to the decentralization and lack of regulation (Alexander & Heck, 2020). This type of manipulation intuitively imposes instability in the LOBs (and the market overall), which consequently may affect return predictions. Secondly, as the most liquid reliable exchanges, their prices and LOBs are presumably more frequently updated than less liquid exchanges', which can facilitate the short-term return predictions. Thirdly, the chosen exchanges are of different sizes, which can potentially assist the identification of lead-lag relationships, since Alexander & Heck (2020) found exchanges with the highest trading volumes to be the leaders. During the period studied, Coinbase had the largest daily trading volume at around 67.8 MUSD, followed by Bitfinex at around 52.7 MUSD, and Gemini at approximately 4.6 MUSD.[4]

3.2 Data

The data was collected from a common source. Due to relatively low trading and order activity with few updates on ultra-high frequencies, I chose to use 10-second sampling frequencies for both the trade data and the LOB data. The complete data set consists of six subsets, one for each cryptocurrency pair at each exchange. All six subsets contain the top five price levels of the LOB (an example is shown in table 1) and prices from the trade data, for the respective exchange and cryptocurrency pair.

[2] https://coinmarketcap.com/charts/

[3] https://coinmarketcap.com/

[4] https://coinpaprika.com/exchanges/

Table 1: Snapshot of the LOB from Bitfinex BTC/USD 01-09-2019

Time	ask-price (p_{ask})	ask-size (Q_{ask})	bid-price (p_{bid})	bid-size (Q_{bid})
00:00:30	9616.6	5.573	916.5	0.501876
00:00:30	9616.9	2	9616.2	0.0533998
00:00:30	9618.9	0.00560217	9614	0.2
00:00:30	9619.2	0.01	9612.9	0.00040643
00:00:30	9619.5	0.243029	9612.2	0.1896
00:00:40	9614.2	1.99	9613.1	1.127
00:00:40	9616	0.2	9613	0.245387
00:00:40	9616.6	0.5749	9612.9	0.0004063
00:00:40	9617.3	0.366072	9612.1	0.0207961
00:00:40	9617.4	0.1178	9611.1	0.5293

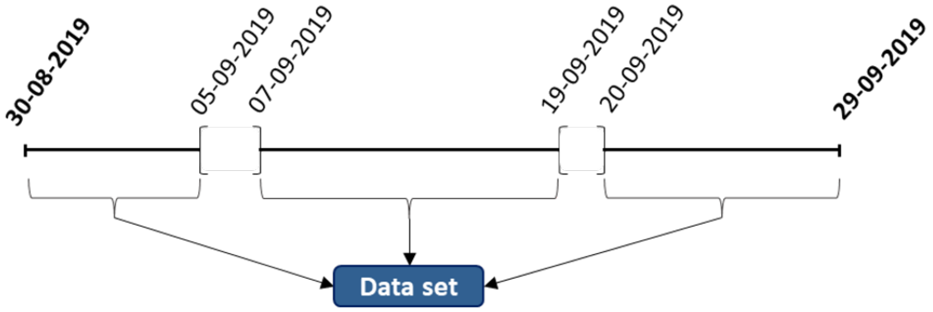


Figure 1: Approach for splitting and merging the data due to periods of missing data

The data spans from 30-08-2019 to 29-09-2019, but data for Gemini was completely missing during two sub-periods within the month, as shown in figure 1. The presence of these missing periods would prevent cross-exchange predictions, which require data from two exchanges to be analyzed together. Furthermore, the sub-periods with missing data were positioned in such a way within the month that using only data preceding, succeeding, or in between the gaps would result in a sample size much smaller than anticipated. Using too little data could lead to poor performance of the Random forest classifier, since it is quite 'data-hungry' and class imbalances (which will be discussed later) could be more severe. Therefore, I filtered away the missing sub-periods from all of the six data sets with the approach shown in figure 1. While I acknowledge that there may be potential issues related to the splitting and merging of the data due to its temporal nature, I believe these issues to be minor. To prevent confusion in the methodology section, I will refer to data from a single exchange for a single crypto as an *individual data set*.

After the six *individual data sets* had undergone the previously described process, there were still occasional missing values present. However, the number of missing values was negligible and therefore these observations were replaced with the values from the preceding observation (so-called forward filling). The imputation will be described in more detail in section 4.2.2.

3.3 Features

To reduce the dimensionality of the LOB data and extract meaningful information from it, it is advantageous to specify a set of features (predictor variables). The process of identifying important features from the LOB is beyond the scope of this paper, and therefore the selection is based on findings in previous research. More specifically, the choice of features is influenced by Petrova & Vilhelmsson (2023), who extract a large set of features from the LOB to predict fiat currency returns. To better suit the methodology used in this thesis, only a subset of those features were selected. The 16 chosen features are shown in table 2. While the characteristics of crypto- and fiat currencies differ, most of the chosen features are commonly used when dealing with LOB data, regardless of asset type. For example, the bid-ask spread and the slope of the bid/ask side are also used in the crypto prediction paper by Fang et al. (2021).

Table 2 includes various features that capture important information in the LOB. The LOB imbalance and slope measures quantify the level of buying and selling demand, while the price change measures identify patterns of autocorrelation in returns. In the methodology section, I will refer back to table 2 when addressing specific aspects of the analysis. For now, it is important to note that features 13-16 are temporal, in the sense that they require information from the current and previous periods. I also want to highlight that the buy/sell indicator, which identifies the order as buyer- or seller initiated, takes binary values.

Table 2: Feature set for each cryptocurrency at each exchange

	Feature	Description
f_1	$p_M = \frac{p^{(1)}_{ask}(t) + p^{(1)}_{bid}(t)}{2}$	Mid-price
f_2	$Spread = max(0, p^{(1)}_{ask}(t) - p^{(1)}_{bid}(t))$	Bid-ask spread
$f_3 - f_7$	$Imb^i = \frac{Q^i_{ask}(t)}{Q^i_{bid}(t)}$	LOB imbalance on the ith level
f_8	$Slope_{ask} = \frac{P^{(1)}_{ask}(t) - P^{(2)}_{ask}(t)}{Q^{(1)}_{ask}(t)}$	Slope ask side
f_9	$Slope_{bid} = \frac{P^{(1)}_{bid}(t) - P^{(2)}_{bid}(t)}{Q^{(1)}_{bid}(t)}$	Slope bid side
f_{10}	$Ind = \begin{cases} 0 & \text{if buyer initiated} \\ 1 & \text{if seller initiated} \end{cases}$	Buy/sell indicator
f_{11}	$log(Q^1_{ask})$	Natural logarithm of best ask quantity
f_{12}	$log(Q^1_{bid})$	Natural logarithm of best bid quantity
f_{13}	$\Delta p_{ask} = p^{(1)}_{ask}(t) - p^{(1)}_{ask}(t-1)$	Change in best ask price
f_{14}	$\Delta p_{bid} = p^{(1)}_{bid}(t) - p^{(1)}_{bid}(t-1)$	Change in best bid price
f_{15}	$\Delta log(Q^{(1)}_{ask}) = \begin{cases} log(Q^{(1)}_{ask}(t)) - log(Q^{(1)}_{ask}(t-1)) & \text{if} \quad \Delta p_{ask} = 0 \\ 0 & \text{otherwise} \end{cases}$	Market depth ask side
f_{16}	$\Delta log(Q^{(1)}_{bid}) = \begin{cases} log(Q^{(1)}_{bid}(t)) - log(Q^{(1)}_{bid}(t-1)) & \text{if} \quad \Delta p_{bid} = 0 \\ 0 & \text{otherwise} \end{cases}$	Market depth bid side

Note: Superscript refers to the price level of the LOB, $i \in \{1, 2, 3, 4, 5\}$. Index is referred to as t.

3.4 Target variable

The returns are calculated from the trade data prices as follows:

$$r(t) = ln(\frac{p(t)}{p(t-1)})$$

The target variable (dependent variable) is the sign of the returns, defined as follows: positive (P) if $r(t) > 0$, negative (N) if $r(t) < 0$, and stable (S) if $r(t) = 0$. This creates a multiclass classification problem.

4 Methodology

To perform the predictions of directional returns, I use the Random forest (Rf) classifier. The Rf can effectively handle multiclass classification problems and manage large sets of possibly highly correlated features, while also accounting for non-linear relationships. The scikit-learn library in Python is used, as it provides a built-in function for the Rf classifier, as well as tools for data preprocessing and performance evaluation.

In this section, I provide a walk-through of how the Rf is implemented by first describing a background to Rf, and then continue on to explain how it is used for predictions for a single exchange, followed by cross-exchange predictions. It will become apparent that the methodology includes some concepts specific to machine learning and data science. As I do not expect the general reader to be familiar with such concepts, I try to avoid complicated mathematical expressions, and instead use figures and plain language to make the section comprehensible. Additionally, it is necessary to discuss some choices more in depth because, in machine learning, there is often no right or wrong approach, and the choices have to be motivated based on trial and error. At the end of this methodology section, an overview is provided to facilitate the understanding of the full prediction process.

4.1 A background to Random forest

The Random forest (Rf) is a supervised machine learning method developed by Breiman (2001). The Rf builds upon multiple individual decision trees, bagging and random selection of features. These will now be described out of a classification viewpoint.

A *decision tree* is essentially a set of conditions that split the data sample recursively, to partition observations into homogeneous groups (in this case 'positive', 'negative' or 'stable'). The algorithm starts with the full data sample and chooses one feature as the first condition to perform a binary split of the data. This creates two new subsamples. In each of these subsamples, an unused feature is selected to generate another binary split. Each split should maximize the homogeneity in the new subsamples. Since every split is performed based on a feature, the algorithm calculates the information gain (or increase in homogeneity) that would have been created by splitting the mth subsample using that feature. In this study, the widely-used Gini index is adopted as a measure of information gain:

$$G(m) = \sum_{k=1}^{K=3} p_k(1 - p_k) \qquad \in [0, 1]$$

where p_k is the fraction of observations from the kth class in the mth subsample. The Gini index is calculated for each unused feature for a particular split, and the feature that is ultimately chosen is the one generating the lowest Gini index (highest information gain). This process of evaluating splits and creating new subsample continues until a stopping criteria is met, or until no further splits generate any information gain. When no more subsample are being created, the tree has reached its leaf node. At this point, the algorithm provides the decision based on the conditions throughout the tree.

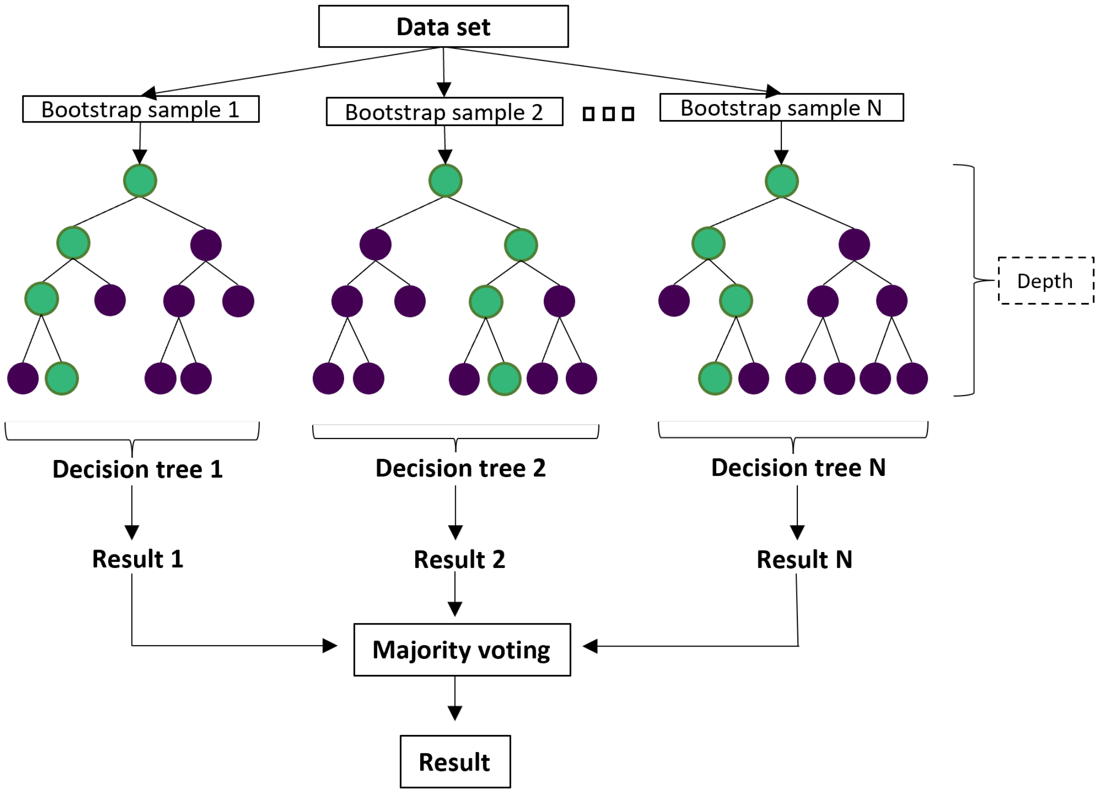


Figure 2: The structure of a Random forest classifier

Bagging is an ensemble procedure which creates B bootstrapped samples from the data set and trains one decision tree to each of these samples. The decisions made by the B trees are then aggregated into a single decision, which comprises the most commonly occurring class among the B decisions (so called majority voting). Bagging enables for more robust predictions compared to a single decision tree, because it limits the variance and overfitting of a single tree through the use of multiple random samples of the data set.

Random selection of features is a procedure by which Rf improves upon decision trees and bagging. It refers to the use of a random sample of features as candidates for each split within a tree, instead of the full set of features. When creating trees from bootstrapped samples, the random feature selection decorrelates the trees by limiting the similarities in their decision structures, thus possibly creating more robust predictions. The final prediction of the random forest is, as in bagged trees, the majority voting. Figure 2 provides a general overview of a Rf classifier's structure.

To use the Rf for predictions, it is common to beforehand split the data into separate training and test sets. The training set is used to build the Rf model, by allowing the algorithm to learn patterns and relationships between the features and the target variable. The choice of features and splits is decided based on the training data. Then, the Rf is fitted to the test set, while maintaining the structure of the trees from the learning phase. This means that the test observations are sent through the model to predict which class each observation belongs to. As a final step, the accuracy of the predictions is determined by the use of performance measures.

4.2 Random forest for single-exchange predictions

This subsection describes how the Rf is implemented to answer the first research question of how accurately the directional BTC and ETH returns can be predicted with the use of LOB data. Given that the aim of this research question is to assess the accuracy of predictions for a single exchange and cryptocurrency, the analysis utilizes the *individual data sets* separately.

In essence, the analysis involves utilizing the features listed in table 2 at index $t-1$ to predict the sign of the 10-second return at index t. As the feature set used is from index $t-1$, I refer to this as lagged features. To deepen the analysis, additional predictions are performed by lagging the features by up to 9 steps, $\delta \in (1, 2, ..., 9)$. This means that I incorporate feature values from as far back as 90 seconds ago (lag 9, or $t-9$) to predict the sign of the 10-second return at t. The aim is to investigate for how long the LOB features stay informative with regards to the returns.

To ensure readers can follow the methodology used to implement the Rf, the process is broken down into several subsections. A reappearing term in the subsections is 'trial run'. This refers to the testing of multiple different specifications on a subset of the full data, to decide appropriate final specifications.

4.2.1 Hyperparameter tuning and overfitting

Hyperparameters are pre-determined parameters set to configure the structure and behavior of the Rf in the learning phase. As an analogy, hyperparameters can be viewed as a blueprint for a building. A blueprint is handed to the contractor as a guide for the design and dimensions of the building. Similarly, hyperparameters are handed to the algorithm to define the overall shape of the forest, such as the depth, the number of splits et cetera. The algorithm must then follow these pre-defined guidelines when building the Rf model. As a widely used tool, the scikit-learn package provides default hyperparameters for the Rf classifier[5]. These include building 100 decision trees, and for each split within the trees a maximum of $4 = \sqrt{16}$ features are considered, where 16 represents the number of features in table 2. The default settings also allow the algorithm to grow each tree until the leaf node contains only a single observation or a single class. It is important to investigate the consequences of the latter settings, because allowing the algorithm to determine the sizes of the trees is a common cause for overfitting (Zhou & Mentch, 2023). Overfitting transpires when the Rf grows overly complex and learns the patterns in the training data almost perfectly, but struggles to generalize to unseen data. To mitigate this issue, I chose to set the hyperparameter regulating the depth of the trees to 11. This number was chosen after comparing the Rf's performances in trial runs. For all remaining hyperparameters, the default settings were used.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

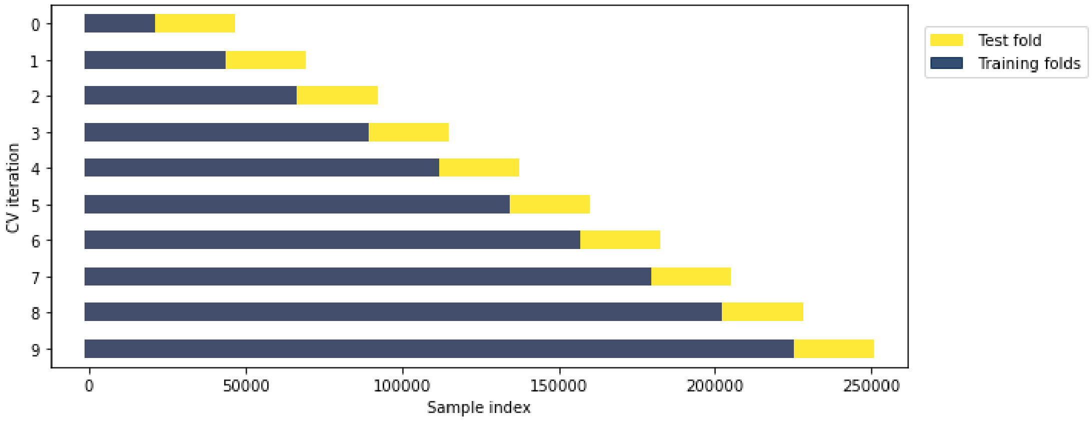


Figure 3: The behavior of 10-fold time-series cross-validation

4.2.2 Time-series cross-validation and data leakage

Continuing the analogy from before, it is important in any construction to not only consider the design of the building but also the ground on which a building is placed, to ensure the stability and durability. Similarly, the Rf needs to be trained on representative data to ensure the generalizability and robustness of the results. K-fold cross-validation (CV) is a tool which enables for this. More specifically, k-fold CV is a resampling technique which can be used to evaluate the model's performances on new data and optimize the use of a data set (James et al., 2013). I use a version commonly referred to as time-series CV (henceforth TSCV), or walk-forward CV, for each *individual data set*.

To understand the workings of TSCV, I will provide an explanation associated with figure 3, which shows the 10-fold TSCV used. The full data set contains approximately 250 000 observations, as shown on the x-axis. There are 10 rows, representing the iterations the algorithm performs. The top one contains the first 45 300 observations in the full data set. These observations are split into folds (subsets), where the blue/dark are *training* folds with 22 650 observations, on which the Rf is trained. The yellow/light fold is a *test* fold with 22 650 observations, on which the Rf is fitted and evaluated. The Rf is then discarded. In the next iteration, the previous folds are merged into training folds, and the next 22 650 observations in the full data set are added as a test fold. As shown in the figure, this means that the data is walking forward in time, towards the right. The procedure with Rf is repeated on this new setting. This process continues for each iteration shown in the figure. Since one Rf model is fitted to each test fold, there are 10 Rf models to evaluate. Obtaining 10 models is the rationale for using TSCV, as it allows for accessing information regarding the predictive accuracy of the Rf on different data. An alternative to TSCV is to simply split the full data set into a single training set and test set, similar to the last iteration of the TSCV. However, in that case only 1 Rf would be fitted to the data and less information would be obtained in the analysis. Furthermore, I chose to use 10 folds because it is commonly used in practice and generally has a reasonable balance between bias and variance (James et al., 2013).

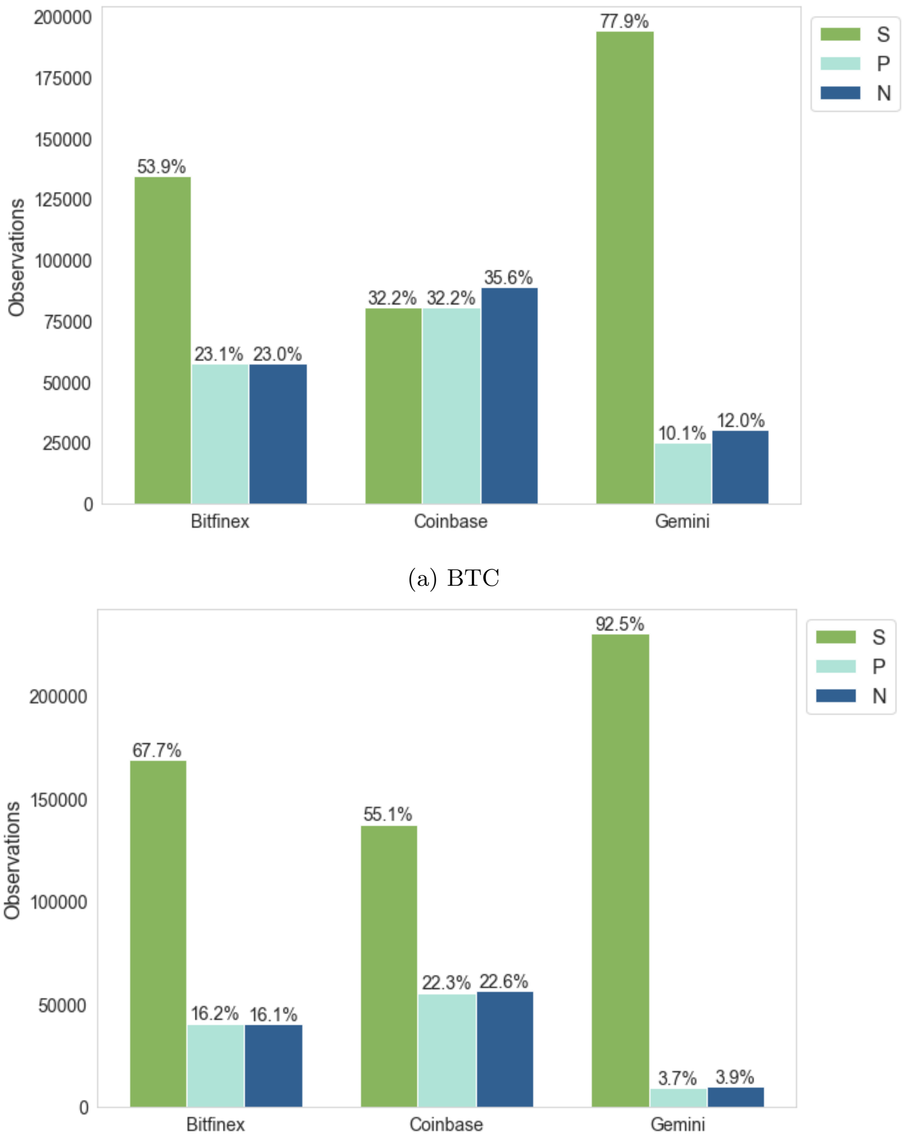
It is worth noting that the prediction task is not a time-series per se, as there is no autoregressive term involved. One might then wonder why the time-series version of k-fold CV is used rather than 'ordinary' k-fold CV. The reason is that TSCV better suits the data structure at hand. K-fold CV *randomly* shuffles the 250 000 observations into folds instead of walking forward in time and preserving the order of the observations, as TSCV does. The random shuffling of observations can cause problems when there is a temporal structure in the data. Recall from section 3.3 that the feature set contains some temporal features. These features were calculated from the full raw data due to constraints related to data accessibility. When randomly shuffling all observations as in k-fold CV, the temporal structure is disrupted, which can become an issue when the test fold and training folds directly or indirectly share some information. The Rf learns the patterns in the training folds, but those patterns may, to some degree, be influenced by the patterns in the test fold. Once the Rf is fitted to the test fold, it may already have considered some information in that fold and will thus perform better than it should. It is subjectively probable that the possible information sharing is very subtle; however, since I cannot estimate or know if it occurs, I chose not to use k-fold CV. When using TSCV, the order of the observations is preserved and any information sharing should only be possible on the border between the test fold and training folds. To address this, a gap is incorporated between the folds such that they do not overlap.

The issue just described is similar to *data leakage*, which in machine learning refers to the scenario when a model is trained on data which it should not have access to (Huyen, 2022). There are numerous ways, in any step of the analysis, in which data leakage can occur and it can lead to overly optimistic results (Huyen, 2022). I include the topic in this section because preventing data leakage is related to the folds in the TSCV. The data leakage phenomena might appear as an abstract small detail. However, it is actually an intuitive and important aspect in the process of performing reliable predictions. To understand it better, let us revisit the analogy of constructing a building again. Remember that a blueprint has already been created and a suitable ground to build on has been found. When starting the actual building process, it is important to thoroughly construct the foundation and anchoring to the ground. If the foundation is shoddy workmanship, the walls may crack once more floors are added to the building. These cracks may not be visible at the first floor, but as the building becomes higher, the cracks expand and may lead to an unstable building. In machine learning, one can view the training folds as the foundation and the first floor as the test fold. Remember that the Rf is trained on training folds. If this data contains leaked information, it may not be an appropriate foundation. Thus, once the test fold is introduced, the Rf may appear to perform well. What is not visible is the bias in the results (the cracks). If one were to gather even more data outside the original data (build additional floors), the issue would become more apparent as the Rf would perform worse. Since all the data is used in the TSCV and there is no data left to 'build additional floors', it would be difficult to determine whether the accuracy scores obtained are deceptively high. As I aim to measure the accuracy of predictions, it is necessary to ensure that the predictions are reliable. Therefore, the method includes identifying possible causes for leakage and prevent it.

One way in which data leakage can occur is when missing values are replaced with previous values (forward-filling), as described in section 3.2. If this imputation is performed on the full data set, observations in the TSCV test folds may be copied into the training folds. Thus, some data in the test folds is no longer 'unseen' when the Rf is evaluated on it. To prevent this, I perform forward-filling within each fold separately, in each iteration of the TSCV. Furthermore, it will become apparent in the next subsection that it is advantageous to transform the features by scaling their values. This is performed with a MinMaxScaler, which finds the maximum and minimum value of a feature in the data set, and use those values to scale the feature to values between 0 and 1. Again, if the full data set is used, the scaling procedure is influenced by both training and test folds, which can cause data leakage. To avoid this, the scaler is 'trained' and applied to the training folds, and then fit to the test fold, in each TSCV iteration.

4.2.3 Imbalanced data and SMOTE-NC

To describe the concept of imbalanced data, the analogy is revisited one last time. Even after ensuring a thorough foundation and anchoring to the ground, it might still be the case that the ground is very uneven. As a consequence, the building will become tilted or leaning, already at the first floor (the test fold). Therefore, the ground should be leveled beforehand. In machine learning, the uneven ground corresponds to imbalanced data and data rebalancing can be seen as the leveling of the ground. A data set is imbalanced when the class distribution is significantly unequal, meaning that one or multiple classes are greatly overrepresented (Fernández et al., 2018). When applying classifier models to such data it can lead to misclassifications of the underrepresented (minority) classes, since models are usually biased toward the overrepresented (majority) classes (Fernández et al., 2018). Figures 4a and 4b show the distribution of the classes in each *individual data set* for BTC and ETH respectively. It is evident from these figures that there is an uneven number of observations among the classes, and that it is usually the 'stable' class that is overrepresented. While the BTC data for Coinbase is almost balanced, the asymmetry is profound for the two other exchanges. The imbalancedness is more severe for the ETH data than the BTC data, across all three exchanges.



(a) BTC

(b) ETH

Figure 4: Class imbalances in the data for the three exchanges

Note: This figure shows the class distribution, where the x-axis indicates the exchange. Figure a) represents the data for BTC, while figure b) represents the data for ETH. 'S' stands for stable, 'P' for positive and 'N' for negative.

Since the Rf can be biased towards the majority class in the learning phase, and was so in trial runs, I use over-sampling to rebalance the training folds in the TSCV. Specifically, I adopt a version of the widely-used sampling technique called 'Synthetic Minority Over-sampling TEchnique' (SMOTE), introduced by Chawla et al. (2002). SMOTE balances the data by increasing the number of minority class observations, such that all classes have approximately the same number of observations. It is important to understand how this happens, because it can affect the final predictions. To clarify how SMOTE operates, figure 5 is used as aid.

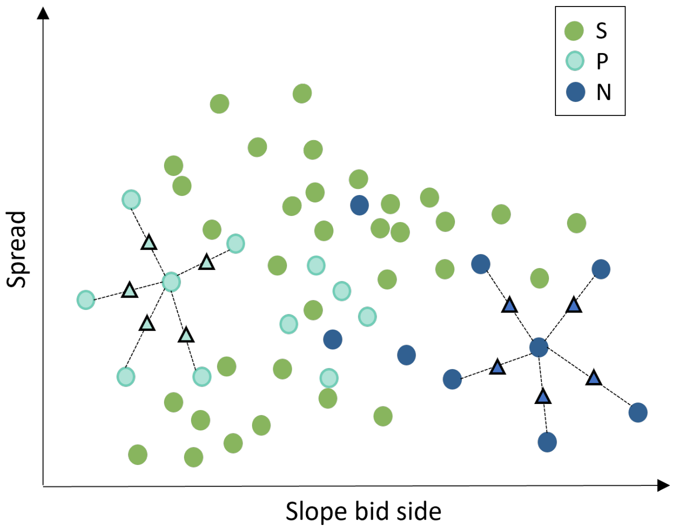


Figure 5: Simplified structure of the SMOTE algorithm

Note that the figure shows a very simplified overview in a 2-dimensional feature space, and is not representative of the actual data used. In the figure, the 'stable' class is clearly the majority class, whereas 'positive' and 'negative' are minority classes with much fewer observations. For the 'positive' class, the algorithm randomly selects an observation belonging to that class. Then, it measures the (Euclidean) distance to other 'positive' observations and finds the k nearest neighbors. I use the default setting of 5 nearest neighbors, which was used in the paper by Chawla et al. (2002). Next, the algorithm performs a randomized interpolation to create new synthetic 'positive' observations (the triangles in the figure). The process is performed for the 'negative' class too, as it is also underrepresented. In the actual data used, the SMOTE operates in a 16 dimensional feature space, since there are 16 features. Additionally, the number of synthetic observations (triangles) per randomly selected observation depends on the imbalance in the data.

As mentioned above, I use a variation of SMOTE, known as SMOTE-Nominal Continuous (SMOTE-NC) which was also introduced by Chawla et al. (2002). Recall from section 3.3 that one feature, the buy/sell indicator, is binary. This characteristics means that there is no meaningful distance between observations, and there is no actual direction. Since 'ordinary' SMOTE by default calculates Euclidean distances, it typically requires continuous features where distance is meaningful. The SMOTE-NC treats nominal and continuous features differently, which better suits the data at hand and therefore SMOTE-NC was chosen. Furthermore, as described in the previous subsection, the features were transformed with the MinMaxScaler. The purpose is to facilitate the calculations of distances in the SMOTE-NC process. However, since the buy/sell indicator is binary, it was not transformed.

Moreover, there are several limitations with the SMOTE-NC (and SMOTE). One of them is over-generalization, where the procedure leads to increased class overlap (Maldonado et al., 2019). As shown in the simplified version in figure 5, there is no clear boundary between the classes. For example, some ’positive’ observations appear in the middle of a cloud of ’stable’ observations. Since SMOTE-NC randomly selects a ’positive’ observation as base point, it might select observations in such clouds. Consequently, the synthetic observations will appear in the same cloud of ’stable’ observations. In other words, the classes become more overlapping, which imposes a greater challenge for the Rf to distinguish them. This can lead to misclassifications and poor prediction performance. Importantly, the issue can be more severe if the classes are not well separated to start with. There exists other more advanced resampling techniques and combinations accounting for this issue, many of which are discussed by Fernández et al. (2018). In trial runs, I tried versions of cost-sensitive learning and under-sampling methods, but they did not perform as well as SMOTE-NC. However, the SMOTE-NC could perhaps be boosted by combining it with under-sampling of the majority class, for example. Due to limited processing power, I decided to only use SMOTE-NC to keep the computational time at a reasonable level.

Furthermore, it should be noted that SMOTE-NC only solves the class imbalances in the training folds, on which it is applied. If a specific class has too few or no observations in those folds, SMOTE-NC cannot oversample that class. As a result, the Rf may not learn the patterns of that class, leading to misclassification of observations belonging to that class when the Rf is fitted to the test fold. To ensure that this issue did not occur, I tracked the number of observations in each class in each TSCV fold. Indeed, all three classes had large number of observations.

4.2.4 Dummy classifier

For each iteration in the TSCV, a dummy classifier is also fit to the test fold. A dummy classifier is a model which ignores the features when it predicts the observations’ classes. The version of dummy classifier used in this thesis considers all three classes equally probable and randomly assigns each observation to one of the three classes. Thus, the dummy classifier serves as a baseline for the Rf predictions. It is expected that the Rf will perform better than random chance, since the Rf considers the LOB features. Yet, the dummy classifier’s performance can help quantifying the significance of the Rf’s predictions.

4.2.5 Prediction accuracy measure

Throughout this section on single-exchange predictions, several important aspects have been discussed. Behind the scenes, a total of 540 Rf models have been created, highlighting the need for a comprehensive performance measure to evaluate and compare the models’ prediction accuracies. One widely used performance metric for multiclass classification problems with data imbalances is the F1 score (Fernández et al., 2018). The F1 score measures the accuracy of the predictions by combining the precision and recall metrics. For example, for the ’stable’ class, precision measures the proportion of observations the classifier predicts as ’stable’ that truly belong to the ’stable’ class. Recall measures the proportion of all ’stable’ observations that are correctly classified as ’stable’. These measures may seem similar, but they measure the accuracy differently. To obtain the F1 score, both the precision and recall are calculated for each class (k) and then averaged across the three classes, which is commonly referred to as macro averaging (Grandini et al., 2020):

$$Macro\ Precision = \frac{\sum_{k=1}^{3} Precision_k}{3}$$

$$Macro\ Recall = \frac{\sum_{k=1}^{3} Recall_k}{3}$$

The macro F1 score is the harmonic mean of the precision and recall:

$$Macro\ F1\ Score = 2 \times \frac{Macro\ Precision \times Macro\ Recall}{Macro\ Precision + Macro\ Recall} \quad \in [0, 1]$$

The macro averaging implies that the scores of the three classes are assigned equal weights, regardless of the number of observations in each class. As a result, the effect of the majority class is the same as that of the minority classes. This means that a high macro F1 score indicates a good performance across all three classes, while a low score indicates a poor performance across the classes.

There exists alternative aggregation methods that can yield significantly different results, for which reason it is suitable to motivate why the macro averaging was used. To do so, it is important to consider the data structure and the purpose of the analysis. Recall that the over-sampling was performed in the training folds, while the test fold remained imbalanced. As explained previously, it is important to recognize that SMOTE-NC, while effective, is not a flawless tool, and the sample of synthetic observations it generates may be of lower quality compared to the original observations due to class overlap. Consequently, even after over-sampling, the Rf is likely to exhibit a greater tendency to misclassify minority classes compared to the majority class, which was observed in trial runs. By assigning each class equal weights in the macro averaging, the macro F1 score is likely to be pulled down by the minority classes. On the other hand, other aggregation techniques like micro averaging assign weights to each class based on their frequency in the data (Grandini et al., 2020). As a result, the minority classes, due to their lower representation in the data set, are less influential and thus affect the F1 score less than in macro averaging (Grandini et al., 2020).

Micro averaging may appear suitable based on the data structure, but it is necessary to also consider the purpose of the analysis. In my study, the goal is to investigate the overall predictability without favoring any specific class. Therefore, using micro averaging could lead to overly optimistic results. While the minority classes might lower the macro average, it is crucial to remember that this aspect contributes to capturing the comprehensive picture of overall predictability. Understanding the performance across all classes is essential, as it reveals any challenges or patterns that investors would also face.

4.3 Random forest for cross-exchange predictions

This subsection describes how the Rf is implemented to answer the second research question, which is whether the prediction accuracy can be increased by including cross-exchange information. Given that the aim of this research question is to assess the accuracy of predictions for a pair of exchanges, the analysis utilizes two *individual data sets* at the time. Specifically, data sets for ETH and BTC are kept completely separate, as the purpose is not to investigate any cross-currency predictability. There are therefore three *individual data sets* to analyze pairwise for each cryptocurrency.

Table 3: Cross-exchange prediction settings

Input data		Prediction
Own Features	Cross Features	
Bitfinex	Coinbase	Bitfinex
Bitfinex	Gemini	Bitfinex
Coinbase	Bitfinex	Coinbase
Coinbase	Gemini	Coinbase
Gemini	Bitfinex	Gemini
Gemini	Coinbase	Gemini

To facilitate the understanding of cross-exchange predictions, consider the information in table 3. The predicted exchange is the exchange whose return (sign) is being predicted. The input data to the Rf is the predicted exchange's own features, just like in single exchange predictions. The addition is another exchange's features. For instance, Bitfinex's features at index $t-1$ and Coinbase's features at index $t-1$ are used to predict the sign of Bitfinex's returns at index t. Figure 6 demonstrates the single- and cross-exchange prediction processes and how they differ, with X representing a set of features and y the sign of returns.

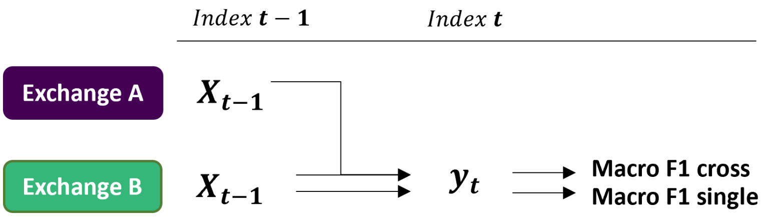


Figure 6: The difference in the single- and cross-exchange prediction processes

By comparing the prediction accuracy of single- and cross-exchange predictions my methodology can be viewed as an identification of potential lead-lag relationships. Recall that the lead-lag phenomena refers to the situation when one exchange (leader) leads the price/return movement over another (lagging) exchange with a time delay. Intuitively, this means that the price information at the leading exchange is ahead of the lagger's, for which reason the leader's information should be informative in predicting the lagger's returns.

In my setting, the information from the leading exchange is contained in its LOB features 10 seconds ago. If the leading exchange is indeed a leader, its LOB features 10 seconds ago should significantly improve the predictions of the lagger's returns. Additionally, this relationship should be unidirectional, meaning that predicting the leader's returns with features from the lagging exchange should not provide significant improvements.

To simplify the explanation with the help of figure 6, let us assume that exchange A leads over exchange B. In that case, using A's features to predict B's returns should yield a much higher F1 score than when only using B's features. However, the opposite should not hold, meaning that B's features should not significantly improve A's predictions.

If the predictions of both A's and B's returns are significantly improved by each other's features, then the relationship is bidirectional. This suggests that neither exchange is a leader or lagger over the other.

If a lead-lag relationship is found, the comparison of the single- and cross-exchange F1 scores can serve as a measure of the predictive power of the lead-lag relationship.

4.3.1 Hyperparameters, time-series cross-validation and SMOTE-NC

Regarding the specifications of the TSCV and SMOTE-NC, the methodology for cross-exchange predictions remains the same as for single-exchange predictions. The cross-exchange prediction only introduces more features to the process. Therefore, the imbalancedness of the data is unchanged. However, there is a possibility that SMOTE-NC may not achieve the same level of performance as for single-exchange, in the sense that overlapping classes may be more of an issue than in the single-exchange case. The short and simplified explanation to this is that the Euclidean distance is less appropriate the higher the dimensionality of the data. For the more elaborate explanation, I refer the reader to for example Maldonado et al. (2019). There are various approaches to address this potential issue, but for simplicity, comparability and computational reasons I use SMOTE-NC in the same manner as for single-exchange predictions.

The hyperparameters for the cross-exchange predictions are the same as those used in the single-exchange predictions. However, this requires a motivation. As discussed in section 4.2.1, the default hyperparameters allow the algorithm to consider a maximum of $\sqrt{16}$ features for single-exchange predictions, where 16 represents the total number of features. For cross-exchange predictions, an additional 16 features are introduced, implying that the default setting would allow the algorithm to consider a maximum of $\sqrt{32}$ features. This means that in the cross-exchange predictions, the Rf has the opportunity to choose from additional important features at each split, which may affect the final predictive accuracy. Consequently, when comparing single- and cross-exchange predictions, the analysis may be unfair because potential differences can be driven by both the new cross-exchange data and the model itself. To ensure a fair comparison, I use $\sqrt{16}$ for cross-exchange predictions as well.

However, by only allowing for a maximum of $\sqrt{16}$ features, the dilution of features in cross-exchange predictions is implicitly overlooked. This can potentially limit the Rf's ability to sufficiently leverage the information in the new data, which may be counterproductive. Clearly, there is a trade-off between $\sqrt{16}$ and $\sqrt{32}$ maximum features, where neither alternative is ideal. For that reason, cross-exchange predictions will be performed with both alternatives.

Furthermore, it is worth mentioning that the dummy classifier is not included in the cross-exchange analysis. This is because the dummy classifier simply ignores all features and would produce the same result as in the single-exchange predictions. Hence, including it would not add any value to the analysis.

4.4 Overview

Throughout the methodology section, I have provided explanations as to why and how each step in the analysis is performed. It has purposely been detailed, because I want the reader to understand what tools have been used to increase the reliability of the predictions. Nevertheless, when every step needs careful consideration it is easy to lose sight of the bigger picture. For that reason, the whole methodology is summarized in figure 7.

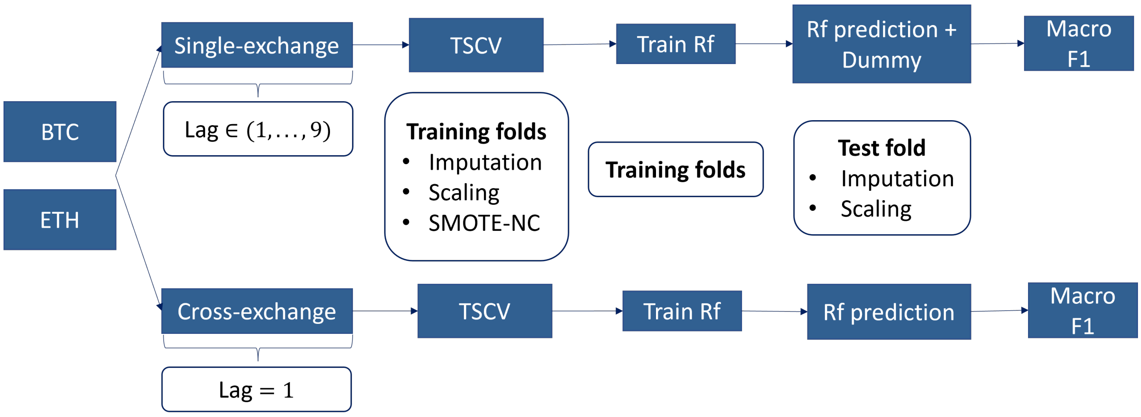


Figure 7: Overview of the methodology

5 Results

The prediction results for BTC and ETH will be presented separately in subsections, as no predictions rely on data crossed over cryptocurrency pairs. Within each subsection the single-exchange and cross-exchange predictions are separated to ensure clarity. It is worth noting that the Macro F1 score is referred to as simply F1 score throughout this section.

5.1 BTC predictions

5.1.1 Single-exchange predictions

For single-exchange predictions the sign of returns was predicted with features of up to 9 lags. Firstly, I present predictions with 1 lag. Then, more results are presented with the additional lags.

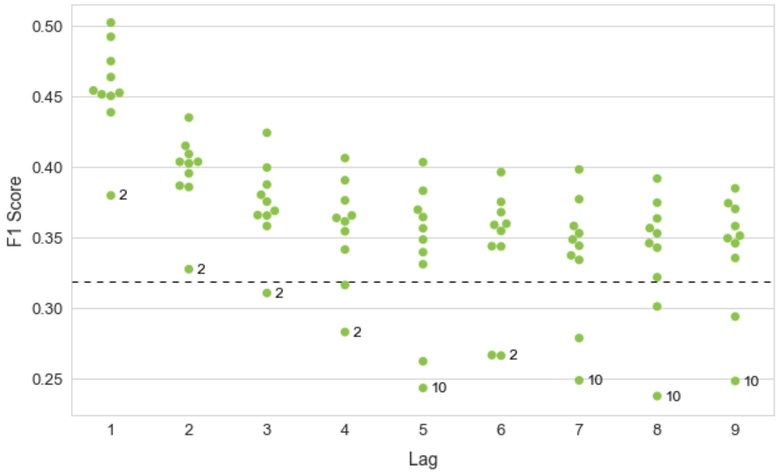
Table 4: Average F1 scores of single exchange predictions for BTC

Exchange	Rf (1lag)	Dummy	Difference
Bitfinex	0.456	0.319	0.137
Coinbase	0.600	0.331	0.269
Gemini	0.396	0.268	0.128

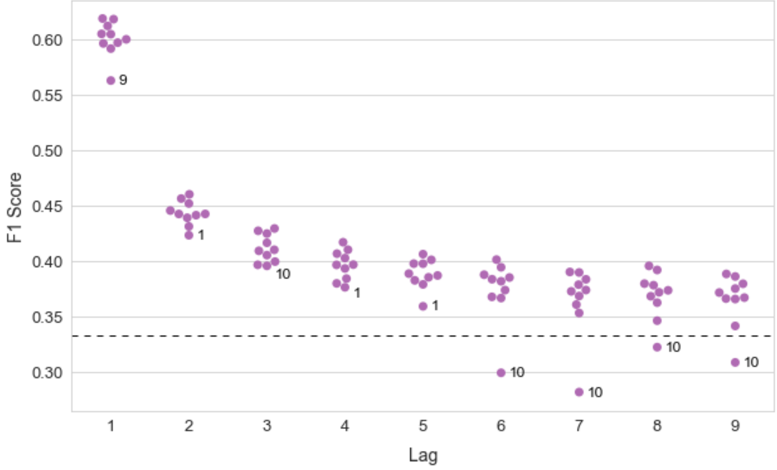
Note: This table shows the arithmetic averages of the Rf models' and dummy classifiers' F1 scores. It also shows average of the difference between them. The averages are based on the 10 cross-validation test folds, for the respective exchange.

The average Rf F1 score for each of the three exchanges with 1 lag is shown in table 4. The table also presents the F1 score of the dummy classifier, and the difference calculated as the Rf's F1 score less the dummy classifier's. The Rf's prediction accuracy varies significantly across the three exchanges, with Coinbase achieving the highest F1 score. The F1 score is the second highest for Bitfinex, and the lowest for Gemini. Additionally, the Rf outperformed the dummy classifier for all three exchanges. This means that, on average, the Rf's predictive power is beyond that of random chance.

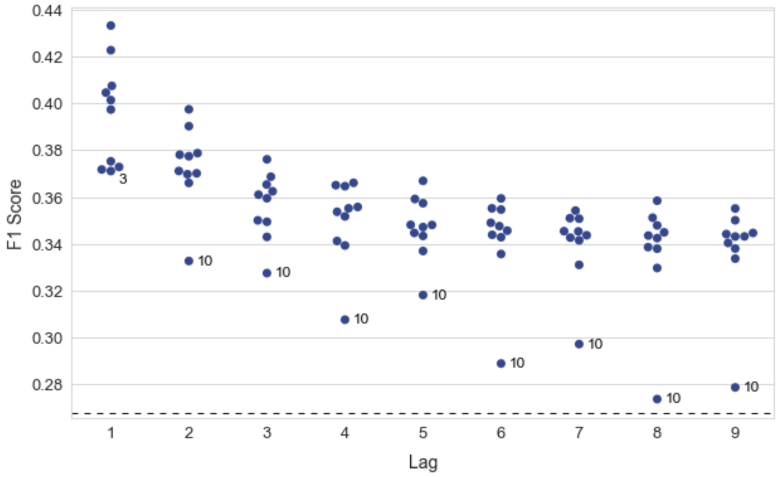
While averages can provide a useful overview of the results, they do not reveal the variability in the F1 scores. Table A1 in the appendix shows the F1 scores in every fold, for all 9 lags and all three exchanges. It also contains the averages and standard deviations of the F1 scores. The table is quite overwhelming, and to obtain a more intuitive understanding of the F1 scores, they are visualized with swarmplots in figure 8, where subplot 8a, 8b, and 8c represent Bitfinex, Coinbase and Gemini respectively. These swarmplots show the F1 score on the y-axis. The x-axis is divided into categories, representing the number of periods the feature set was lagged for the predictions. The points (dots) above the lag number represent the test folds from the 10-fold TSCV. The digit next to the lowest point represents the test fold number corresponding to that point. The horizontal dashed line represents the average F1 score of the dummy classifier.



(a) Bitfinex



(b) Coinbase



(c) Gemini

Figure 8: BTC swarmplots for the three exchanges

Note: This figure shows the F1 scores for exchange separately. Each data point per x-axis category represents a test fold from the 10-fold cross-validation. The x-axis shows how many periods the feature set was lagged. The digit next to the lowest data point represents the corresponding test fold. The horizontal dashed line represents the average F1 score of the dummy classifier.

The initial focus in the figure will be on lag 1, as a continuation of the previous discussion. For Bitfinex (figure 8a), most points are clustered around the average of 0.456, but there is a considerable difference of 0.122 between the minimum and maximum F1 score. For Coinbase (figure 8b), the scores are more clustered, with the corresponding difference of 0.056. The difference for Gemini (figure 8c) is 0.062. A similar indication is apparent in appendix table A1, where the F1 score's standard deviation is the largest for Bitfinex. These results suggest that the prediction accuracies vary depending on the test fold, and that the variability is greater for Bitfinex than the others. The figures also illustrate that for 1 lag, the accuracy across all folds is greater than the average score of the dummy classifier; meaning that the accuracies are higher than those of random chance.

When analyzing the F1 scores across all 9 lags in figure 8, it can be observed that the scores tend to gradually decrease as the lags increase. This indicates that using older LOB data results in lower prediction accuracy. For Bitfinex and Coinbase, the clusters of points drop sharply downwards when moving from lag 1 to lag 2, indicating that using LOB data older than 10 seconds causes the accuracy to fall significantly. The drop is more subtle for Gemini, suggesting that slightly older LOB data may still be effective, but that the accuracy is the highest for the first lag. Overall, utilizing the most recent LOB data appears to be more critical for the prediction accuracy for Bitfinex and Coinbase than for Gemini.

Apparent in all subplots is that there is often one test fold (dot) that represents a much lower F1 score than the rest, which reduces the average score. In some cases, these dots represent F1 scores that are even below the average F1 score of the dummy classifier, for the respective exchange; meaning that some test fold prediction accuracies are close to or below that of random chance. In the appendix, table A1 presents the folds with the lowest F1 scores in bold. These test fold numbers are also mapped in figures 8a, 8b, and 8c. For Bitfinex, the 2nd and 10th fold yield the lowest F1 score for the nine lags. For Coinbase, it is mostly the 1st and 10th fold, with the exception of the 9th fold for lag 1. For Gemini, it is mostly the 10th fold, except for lag 1 where the 3rd fold has the lowest F1 score.

Evidently, the 10th fold is reappearing to be yielding the lowest accuracy for large lags, across all exchanges. When the Rf performs predictions in this fold, it has considered the previous 226 000 observations in its learning phase. Intuitively, one would think that if the Rf has been trained on the most data, it should be able to identify most patterns. However, more data can also increase the risk of overfitting. Although overfitting is one possible reason for the poor performance, it is peculiar that the 10th fold is not the worst performing fold for smaller lags. For example, for Bitfinex (figure 8a), the 2nd fold yields the lowest accuracy for most lags. In that case, the Rf has only been trained on 45 000 observations, for which reason overfitting is less likely to be the issue. When I performed trial runs, I tracked the Rf's performance in the training and test fold and did not observe signs of overfitting. However, I only used subsets of data in the trial runs.

Even though overfitting may partly contribute to the poor performance in some folds, it is probably not the sole cause. Another potential reason is that the quality of the data may be poor in some folds. This could happen if the LOB data and/or return data contains more noise in some folds. Noise is likely to be an issue when there is more trading activity or in highly volatile periods. To examine whether volatility can be a cause for the low accuracy in some folds, I will analyze figure 9, which displays the returns for the respective exchange against the index, with a top axis showing the corresponding test fold locations. It should be noted that the Rf only classifies observations and does not measure the magnitude of the returns, for which reason the figure may be somewhat misleading. However, during highly volatile periods it is likely that the LOB and/or return data is noisier, making it more difficult for the Rf to interpret the patterns in the features and classify correctly.

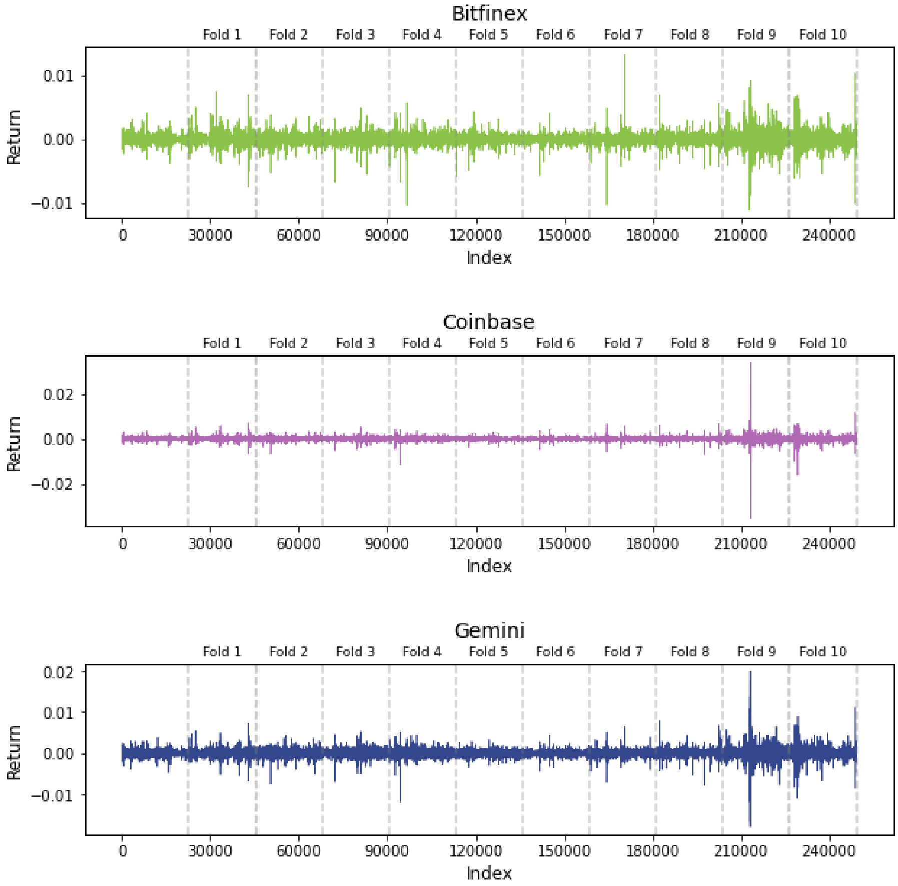


Figure 9: Stacked line plot showing the BTC returns at the three exchanges

Note: This figure shows the BTC 10-second returns for Bitfinex, Coinbase and Gemini respectively. The bottom axis represents the time index. The top axis shows the 10 cross-validation test folds, with dashed vertical lines separating the folds.

In figure 9, it is clear that the 9th and 10th fold encapsulate periods of relatively high volatility for all three exchanges. For Coinbase, the 9th yielded the lowest F1 score for lag 1 (see figure 8b). As shown in figure 9, the period corresponding to the 9th fold displays higher volatility than the previous folds, on which the Rf was trained. It may therefore be the case that the 9th fold yields low accuracy because the Rf is introduced to noisy new patterns that were not represented in the training data. For lag 2, 4 and 5, the 1st fold yielded the lowest accuracy for Coinbase. Again, this fold encapsulates a period of higher volatility than the data the Rf was trained on. For the larger lags (6, 7, 8 and 9), the 10th fold yielded the lowest accuracy for Coinbase. In this case, the Rf was trained on data including the volatile period in fold 9; thus it should be familiar with volatile patterns. Still, the performance in the 10th fold is poor. It may therefore be the case that the data is too noisy to find predictability further into the future than 50 seconds. For some of the larger lags, the Rf in the 10th fold actually yields lower accuracy than the average score from the dummy classifier which predicts at random, as displayed in figure 8b.

For Bitfinex, the 2nd fold consistently yielded the lowest accuracy for the first 4 lags. In figure 9 the returns in the 2nd fold do not necessarily show higher volatility than in the folds the Rf was trained on. Similarly for Gemini, the 3rd fold yielded the lowest accuracy for lag 1, yet that period was not more volatile than the preceding periods on which the Rf was trained. Neither of the two exchanges showed the lowest accuracy for the 9th fold, even though that fold corresponds to much higher volatility than the preceding folds on which the Rf is trained. This is peculiar, since Coinbase's accuracies showed clearer associations to volatility.

It could be the case that the BTC LOB information at Gemini and Bitfinex is generally less informative than at Coinbase. Another reason is related to over-sampling. In section 4.2.3 I demonstrated that the data from Coinbase was almost balanced, while the class imbalancedness was severe for Gemini and Bitfinex. Hence, SMOTE-NC had to over-sample considerably more for the latter two exchanges. It is possible that the synthetic observations generated the classes to overlap more in the training folds; causing the Rf to perform poorly in some test folds, even though the test data was not necessarily particularly noisy. In the test folds with high volatility, such as the 9th fold, it is possible that the SMOTE-NC had already introduced noise in the training folds that was similar to the noise imposed by the high volatility. Thus, the Rf had already been trained on challenging data, which enabled it to perform adequately in the volatile period. This is, however, an hypothesis that was not rigorously tested, and is only intended to highlight the potential challenges when comparing data sets with varying degrees of imbalances and their respective application of SMOTE-NC.

5.1.2 Cross-exchange predictions

In the analysis of cross-exchange predictions, the aim is to identify significant differences between the F1 scores of single- and cross-exchange predictions. As outlined in section 4.3, a significant difference can indicate a lead-lag relationship if the difference is unidirectional.

Table 5: Average F1 scores of cross-exchange predictions for BTC

Predicted exchange	Cross-exchange	Rf cross	Difference
Bitfinex	Coinbase	0.475	0.0181
Bitfinex	Gemini	0.469	0.0132
Coinbase	Bitfinex	0.601	0.0012
Coinbase	Gemini	0.601	0.0007
Gemini	Bitfinex	0.402	0.0062
Gemini	Coinbase	0.409	0.0136

Note: This table shows the arithmetic average of the Rf models' F1 scores from cross-exchange BTC predictions, when using $\sqrt{16}$ as the maximum number of features for each split in cross-exchange predictions. It also shows the average of the difference in F1 score between single- and cross-exchange predictions. For each exchange pair there are ten Rf models, fitted from 10-fold cross-validation.

Table 5 presents the average F1 scores of cross-exchange predictions in column 3, when using $\sqrt{16}$ as the maximum number of features for each split. Column 4 contains the average difference, calculated as the average of the difference between cross- and single-exchange F1 scores across the 10 folds. As shown in column 4, all differences are positive, meaning that all exchanges' predictions are improved by including another exchange's features.

What is noteworthy is that when Coinbase's returns are predicted, the improvement from the other exchanges' features is very small. At the same time, the other two exchanges' predictions show much larger improvement from including Coinbase's features. This can be an indication that Coinbase leads Bitfinex and Gemini. Additionally, Coinbase leads Gemini to a larger extent than Bitfinex, since Gemini's improvement is higher than Bitfinex's. Table A2 in the appendix shows the results in each single fold, and it is apparent in this table that in some folds, Coinbase's prediction accuracies are actually reduced by the inclusion of another exchange's features. The indication that Coinbase is the leader corresponds to to the fact that Coinbase had the largest trading volume among the three exchanges.

The overall results are similar to those in appendix table A3, which reports the difference in F1 score for cross-exchange predictions with a maximum of $\sqrt{32}$ features at each split. Predictions of Coinbase's sign still had the lowest average difference, and negative differences in multiple folds. When comparing table 5 and appendix table A3, all averages increased when allowing for a maximum of $\sqrt{32}$ features, but the increase was the smallest for predictions of Coinbase's sign. This indicates that the tendency of Coinbase being the leader is not associated with the Rf specifications, further supporting that it is the data that drives the lead-lag results.

When only considering Bitfinex and Gemini, it appears in both table 5 and appendix table A3 that Gemini leads over Bitfinex. The reason is that Bitfinex's improvement is higher when Gemini's features are added, than vice versa. This result contradicts the expected relationship between the two exchanges based on their trading volumes. Gemini had substantially lower daily trading volume at 4.6 MUSD, compared to Bitfinex with 52.7 MUSD. As discussed for the single-exchange predictions, there may be muddled patterns from the over-sampling. Therefore, it is appropriate to exercise caution when drawing conclusions from these findings.

5.2 ETH predictions

5.2.1 Single-exchange predictions

Moving on to the analysis of ETH, the results will be presented in a similar manner as for BTC. Firstly, I present predictions with 1 lag. Then, more results are presented with the additional lags.

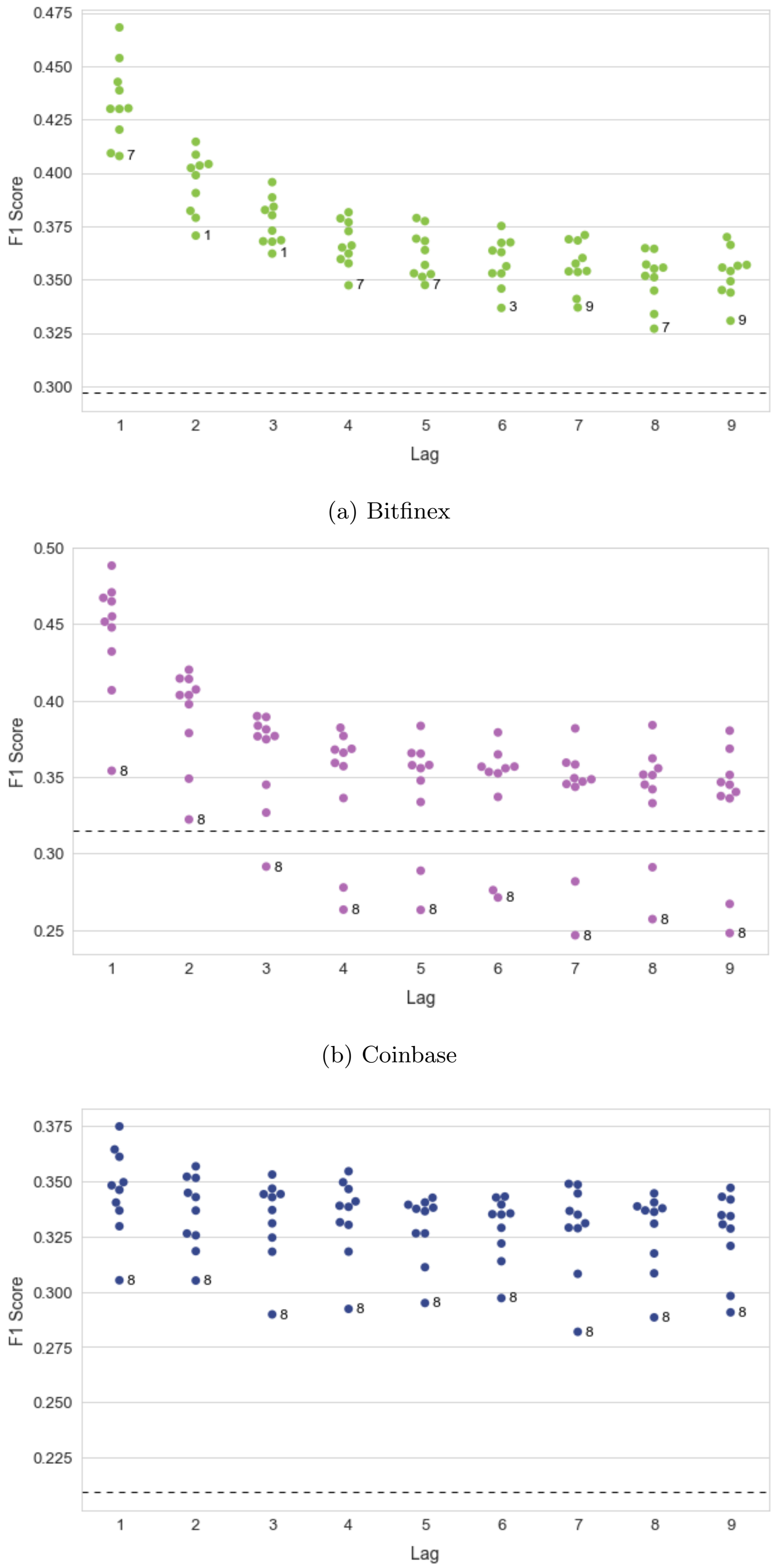
Table 6: Average F1 scores of single exchange predictions for ETH

Exchange	Rf (1lag)	Dummy	Difference
Bitfinex	0.433	0.297	0.136
Coinbase	0.444	0.315	0.129
Gemini	0.346	0.209	0.137

Note: This table shows the arithmetic averages of the Rf models' and dummy classifiers' F1 scores. It also shows average of the difference between them. The averages are based on the 10 cross-validation test folds, for the respective exchange.

The average Rf F1 score for each of the three exchanges with 1 lag are shown in table 6. The table also presents the average F1 score of the dummy classifier, and the difference calculated as the Rf's F1 score minus the dummy classifier's. The Rf's prediction accuracy varies across the three exchanges, with Coinbase achieving the highest F1 score. The F1 score is the second highest for Bitfinex, and the lowest for Gemini. Additionally, the Rf outperformed the dummy classifier for all three exchanges. This means that the Rf's predictive power is on average beyond that of random chance. The scores across all exchanges are notably lower than those of BTC that have been previously presented. This indicates that it is more difficult to predict the sign of ETH returns than the sign of BTC returns.

Table A5 in the appendix shows the F1 scores in every fold, for all 9 lags and all three exchanges. It also contains the averages and standard deviations of the F1 scores. The general results are presented with swarmplots in figure 10, where subplot 10a, 10b, and 10c represent Bitfinex, Coinbase and Gemini respectively.



(a) Bitfinex

(b) Coinbase

(c) Gemini

Figure 10: ETH swarmplots for the three exchanges

Note: This figure shows the F1 scores for exchange separately. Each data point per x-axis category represents a test fold from the 10-fold cross-validation. The x-axis shows how many periods the feature set was lagged. The digit next to the lowest data point represents the corresponding test fold. The horizontal dashed line represents the average F1 score of the dummy classifier.

For lag 1 in figure 10, it can be observed that there is a small cluster of points around the average for each exchange, but the remaining points are dispersed. The range between the maximum and minimum points is the largest for Coinbase, followed by Bitfinex and Gemini. These results suggest that the prediction accuracies vary depending on the test fold, and that the variability is greater for Coinbase than the other two exchanges. Compared to BTC, the F1 scores for Bitfinex are more gathered, Coinbase are much more scattered and for Gemini it is approximately the same.

When considering the F1 scores across all 9 lags in figure 10, the scores tend to gradually decrease as the lags increase for Bitfinex and Coinbase (figures 10a and 10b). In other words, using older LOB data tends to result in lower prediction accuracy. This is particularly evident when moving from lag 1 to lag 2, indicating that using the most recent LOB data is critical for the prediction accuracy. In contrast, the decrease in F1 scores for Gemini (figure 10c) is much more subtle. For example, in fold 7, predictions with features 20 or 30 seconds ago yield the exact same F1 score (see table A4 in the appendix). This is quite interesting, given that one would typically expect older data to be less informative than more recent data. One potential reason may be that ETH is traded infrequently at Gemini. In figure 4b, which illustrates the class imbalances for Gemini, 92.5% of the observations in the full data set belonged to the 'stable' class. This means that approximately 230 000 out of 250 000 returns are equal to zero. It is likely that the features take on similar values for these observations. Consequently, the information in the test folds' features from 10 seconds ago or 30 seconds ago may be almost equally useful in predicting the sign of returns 10 seconds ahead.

Furthermore, the data points in figure 10a (Bitfinex) and 10c (Gemini) are never lower than the average F1 score of the dummy classifier. Recall that the imbalanced nature of the data sets from Bitfinex and Gemini was more severe than that of Coinbase (see figure 4b). The imbalance is likely to result in poorer performance for the dummy, as it regards the classes equally likely and tends to misclassify more frequently when the class distribution is highly asymmetric.

The digits next to the bottom 'dot' for each lag in figure 10 represents the test fold number. An interesting aspect of figures 10b and 10c is that the 8$^{\text{th}}$ test fold yields the lowest F1 score across all lags. Table A in the appendix shows that the 7$^{\text{th}}$ test fold has the second lowest accuracy across most lags. For Bitfinex in figure 10a, mainly the 7$^{\text{th}}$, 1$^{\text{st}}$ and 9$^{\text{th}}$ yield the lowest accuracies.

As for BTC, I will analyze whether the volatility of returns can explain why some folds present lower accuracy than others. In figure 11, the ETH returns for the respective exchange are plotted against the index, with a top axis showing the corresponding test fold locations.

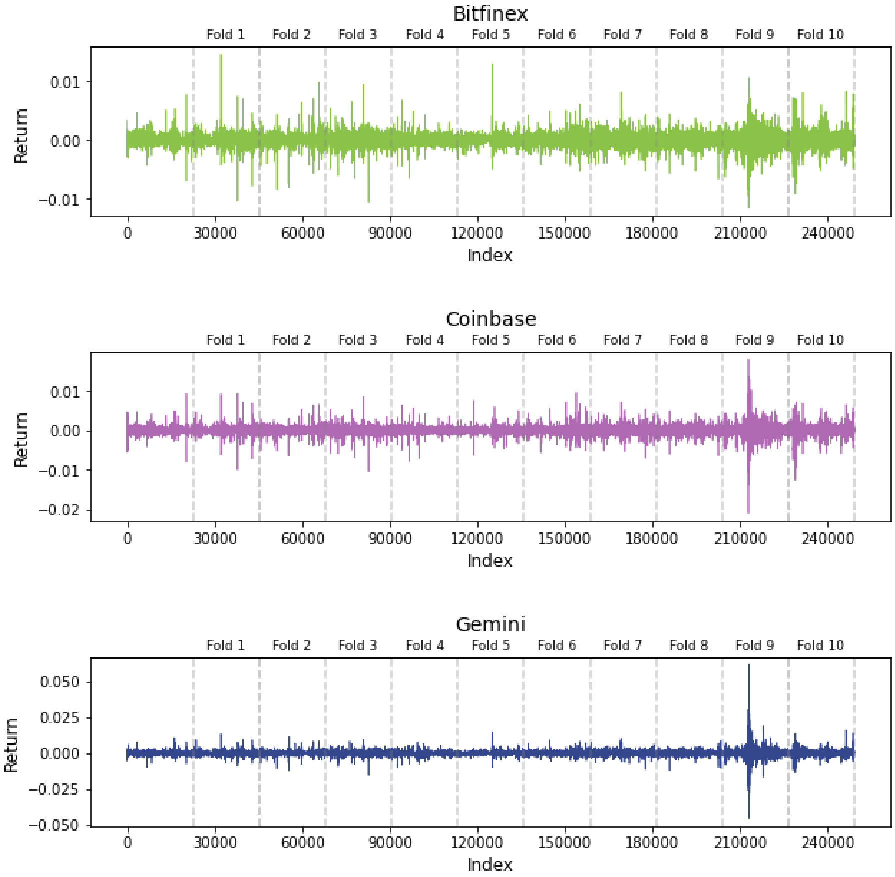


Figure 11: Stacked line plot showing the ETH returns at the three exchanges

Note: This figure shows the ETH 10-second returns for Bitfinex, Coinbase and Gemini respectively. The bottom axis represents the time index. The top axis shows the 10 cross-validation test folds, with dashed vertical lines separating the folds.

For Coinbase and Gemini, the 8th and 7th test folds are of interest. The periods encapsulated in these folds do not exhibit any significantly higher or lower volatility than what is contained in the preceding training data. It is noticeable that the 9th fold contains a period of higher volatility than the folds before it; yet, the F1 scores for Coinbase and Gemini in the 9th fold are not the lowest. On the contrary, this fold yields among the highest F1 score for both exchanges (see table A4 in the appendix) across all lags. Overall, the poor performing test folds do not appear to be associated with especially volatile periods.

Regarding Bitfinex, the 7th fold performed the worst for lag 1, despite the corresponding period in figure 11 not being more volatile than the preceding periods on which the Rf was trained. However, the 1st fold yielded the lowest accuracy for lags 2 and 3, and the corresponding period had slightly more variable returns than the training data. A similar argument could be made for the 9th fold. These observations suggest that volatility may partially explain the varying results in the test folds.

It is worth noting that the exchanges' data for ETH was much more imbalanced than for BTC. Consequently, the ETH data had to be over-sampled to a larger extent. This can muddle the patterns, as discussed in section 5.1.1 for the BTC predictions. It is possible that this could explain why the poorly performing test folds do not consistently correspond to highly volatile periods.

5.2.2 Cross-exchange predictions

Similar to the analysis for BTC, I also examine the differences in F1 scores for single- and cross-exchange predictions of ETH returns.

Table 7: Average F1 scores of cross-exchange predictions for ETH

Predicted exchange	Cross-exchange	Rf cross	Difference
Bitfinex	Coinbase	0.447	0.0141
Bitfinex	Gemini	0.472	0.0390
Coinbase	Bitfinex	0.458	0.0145
Coinbase	Gemini	0.489	0.0448
Gemini	Bitfinex	0.350	0.0042
Gemini	Coinbase	0.354	0.0080

Note: This table shows the arithmetic average of the Rf models' F1 scores from cross-exchange ETH predictions, when using $\sqrt{16}$ as the maximum number of features for each split in cross-exchange predictions. It also shows the average of the difference in F1 score between single- and cross-exchange predictions. For each exchange pair there are ten Rf models, fitted from 10-fold cross-validation.

Table 7 presents the average F1 scores of cross-exchange predictions in column 3, when using $\sqrt{16}$ as the maximum number of features for each split. Column 4 contains the average difference, calculated as the average of the difference between cross- and single-exchange F1 scores across all folds. As shown in column 4, all differences are positive, meaning that all exchanges' predictions are improved by including another exchange's features.

In the presented table, it is observed that Bitfinex and Coinbase exhibit almost equal improvement in predictive accuracy upon inclusion of each other's features. Interestingly, Gemini's features improve the other two exchanges' accuracies significantly, while the reverse is not true. Additionally, there are two folds in appendix table A5 in which Gemini's prediction accuracies are reduced by the inclusion of Bitfinex's features. The results could suggest that Gemini leads over Bitfinex and Coinbase, which is unexpected considering Gemini had the smallest trading volume.

However, the results are quite different when extending the maximum number of features to $\sqrt{32}$, shown in appendix table A6. There are no folds in which Gemini's results are negative. Gemini's average F1 scores are still significantly higher than the other two exchanges', upon inclusion of each other's features. However, when comparing the averages in table 7 and appendix table A6, it becomes apparent that Gemini's averages increased the most when allowing for $\sqrt{32}$ maximum features. This is opposed to the BTC case, where the leader's (Coinbase's) averages increased the least. This is possibly an indication of Gemini's leadership not being driven by the data, but rather model specifications and noise, or muddled patterns from the over-sampling.

6 Discussion

In this section, I will compare and analyze the results with respect to previous literature.

One of the key findings of this thesis is the presence of predictability beyond random chance for all three exchanges and both cryptocurrencies. These findings can be considered as counterexamples to the weak and semi-strong forms of the Efficient Market Hypothesis. Interestingly, the findings of predictability are in contrast to those of Petrova & Vilhelmsson (2023), as they did not identify predictability of returns using LOB features in their study on the fiat currency market. Although there are notable differences in data and methodology between our studies, the results indicate that the crypto market is less efficient than the more mature fiat currency market.

Even though my results show that it is possible to predict crypto returns, the single-exchange predictions show F1 scores notably lower than those found by Fang et al. (2021) in their study on crypto prediction, which used LOB data from 2018. They achieved F1 scores of 70.2-81.4% for BTC and 55.5-73.6% for ETH when predicting $mid-prices$, depending on the amount of training data used. In comparison, my F1 scores for BTC (with 1 lag) range between 37.11-61.83%, and 30.5-48.8% for ETH, across all three exchanges when predicting $returns$. There are multiple possible explanations as to why the results differ so greatly. Firstly, it is unclear what aggregation technique Fang et al. (2021) used for the F1 scores, and it may not have been the macro averaging. Secondly, the target variable in my thesis is different from the study by Fang et al. (2021), and it may be easier to predict the sign of mid-prices than actual returns, which are separate to the LOB. Thirdly, Fang et al. (2021) used data exclusively from the GDAX exchange (later rebranded as Coinbase Pro[6]). It is possible that the characteristics of traders on the exchanges used in my study and GDAX are different, which affects the LOB characteristics and return patterns, thereby influencing the predictability. Fourthly, it is possible that the general predictability of the crypto market decreased from 2018 to 2019. Fifthly, Fang et al. (2021) used a more advanced model and a much shorter sampling period with higher-frequency data. The list of possible causes for the differences in prediction accuracies can be extensive, and it is essential to acknowledge that my results are not directly comparable to those reported by Fang et al. (2021).

Moreover, Gradojevic et al. (2023) demonstrated that during periods of high volatility, the predictability of crypto returns decreased significantly, supporting the idea of adaptive markets. Although they did not use LOB data and examined longer time periods, some of the results in this thesis are consistent with their findings. For BTC Coinbase single-exchange predictions, the folds with the lowest accuracies corresponded to periods of high volatility, indicating that volatility may have contributed to the poorer predictions. However, this pattern was less noticeable for the other exchanges and for the ETH predictions. If volatility indeed is a cause for decreased predictability, which was not tested formally, then a possible cause for not observing this pattern could be the over-sampling, which may have obscured the relationship during the Rf's learning phase.

[6] https://www.investopedia.com/terms/g/gdax.asp

Furthermore, the results show that all exchanges' prediction accuracies on average favored from including another exchange's features. Specifically, the results indicate that for BTC, Coinbase was the leading exchange over Bitfinex and Gemini. Since Coinbase had the largest trading volume of the three exchanges, this result is consistent with previous research on the BTC spot market (Alexander & Heck, 2020; Blasco & Corredor, 2022; Brandvold et al., 2015; Schei Norheim, 2019). However, for ETH, the results suggest that Gemini might have been the leading exchange. This result is questionable since Gemini has been shown to lag behind the other two exchanges in the BTC market (Alexander & Heck, 2020) and had the lowest trading volume among the three considered exchanges. There are several potential reasons for this ambiguity. On the one hand, the lead-lag relationship has been shown to change over time (Brandvold et al., 2015), and the period I studied may have been one where Gemini indeed lead the other two exchanges. On the other hand, the results could be influenced by methodological choices. I showed that the F1 scores varied based on a hyperparameter setting and I discussed the possible effect of noise resulting from over-sampling. This implies that the lead-lag results for ETH are comparatively less robust than those for BTC.

7 Conclusion

The main aim of this study was to investigate the use of LOB data in crypto spot return predictions. I used the Random forest classifier and 10-second frequency LOB data to predict the sign of the 10-second return, and found that the accuracy, measured by the macro F1 score, varied significantly across exchanges, cryptocurrencies and time periods in the test data. When predicting 10 seconds ahead, the highest F1 scores were observed for Coinbase, with scores of 56.24-61.83% for BTC and 35.40-48.82% for ETH. The second highest F1 scores were observed for Bitfinex, ranging between 38.00-50.24% for BTC and 40.79-46.82% for ETH. The lowest F1 scores were found for Gemini, ranging between 37.11-43.34% for BTC and 30.52-37.50% for ETH. This order of the exchanges is consistent with their daily trading volume rank, indicating that predicting returns is more difficult for smaller exchanges. Additionally, ETH predictions generally yielded significantly lower accuracies than BTC predictions, likely due to its lower market capitalization.

The results also underscore the importance of using recent LOB data to achieve more accurate predictions. When predicting the sign of the 10-second returns further than 10 seconds ahead, the accuracy generally dropped sharply for the 20-second lag and gradually decreased thereafter.

I investigated why certain test data yielded lower prediction accuracy for each exchange and cryptocurrency. Overfitting was identified as a potential contributing factor. Further investigation revealed that, at least for BTC Coinbase, the poorer accuracy scores corresponded to periods of high return volatility.

Furthermore, I compared single- and cross-exchange predictions, discovering that the accuracy for all exchanges increased, on average, in cross-exchange predictions. The results suggest that Coinbase was the leading exchange for BTC predictions among the exchanges considered. For ETH, the results indicated that Gemini was the leader, however this result could be driven by model specifications and noise.

Overall, interpreting the ETH predictions was more challenging due to imbalanced data and possibly muddled patterns following over-sampling.

For investors, accurately predicting the sign of returns is clearly advantageous in making informed decisions regarding their positions and hedging strategies. This study has shown that generating crypto return predictions using the most recent LOB data yields accuracies far beyond that of random chance. Given the variations in prediction accuracy observed in this study, it is possible that alternative approaches may yield higher accuracies. Therefore, further research is needed to explore other models and techniques that can help improve the accuracy of crypto return predictions with LOB data.

References

Abergel, F., Anane, M., Chakraborti, A., Jedidi, A., & Toke, I. M. (2016). *Limit Order Books*. Cambridge University Press.

Akyildirim, E., Goncu, A., & Sensoy, A. (2021). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*, 297, 3–36.

Albers, J., Cucuringu, M., Howison, S., & Shestopaloff, A. Y. (2021). Fragmentation, price formation and cross-impact in bitcoin markets. *Applied Mathematical Finance*, 28(5), 395–448.

Alexander, C. & Heck, D. F. (2020). Price discovery in bitcoin: The impact of unregulated markets. *Journal of Financial Stability*, 50, 100776.

Bitwise Asset Management (2019). Presentation to the U.S. securities and exchange commission. Available at: https://www.sec.gov/comments/sr-nysearca-2019-01/srnysearca201901-5164833-183434.pdf.

Blasco, N. & Corredor, P. (2022). If the bitcoin market grows, size matters. *Applied Economics Letters*, 29(11), 983–987.

Brandvold, M., Molnár, P., Vagstad, K., & Valstad, O. C. A. (2015). Price discovery on bitcoin exchanges. *Journal of International Financial Markets, Institutions and Money*, 36, 18–35.

Breiman, L. (2001). Random forests. *Machine learning*, 45, 5–32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321–357.

Chu, J., Zhang, Y., & Chan, S. (2019). The adaptive market hypothesis in the high frequency cryptocurrency market. *International Review of Financial Analysis*, 64, 221–231.

Crépellière, T., Pelster, M., & Zeisberger, S. (2023). Arbitrage in the market for cryptocurrencies. (In press). https://doi.org/10.1016/j.finmar.2023.100817.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383–417.

Fang, F., Chung, W., Ventre, C., Basios, M., Kanthan, L., Li, L., & Wu, F. (2021). Ascertaining price formation in cryptocurrency markets with machine learning. *The European Journal of Finance*, (pp. 1–23).

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets*. Springer.

Fischer, T. G., Krauss, C., & Deinert, A. (2019). Statistical arbitrage in cryptocurrency markets. *Journal of Risk and Financial Management*, 12(1), 31.

Gradojevic, N., Kukolj, D., Adcock, R., & Djakovic, V. (2023). Forecasting bitcoin with technical analysis: A not-so-random forest? *International Journal of Forecasting*, 39(1), 1–17.

Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

Guo, H., Zhang, D., Liu, S., Wang, L., & Ding, Y. (2021). Bitcoin price forecasting: A perspective of underlying blockchain transactions. *Decision Support Systems*, 151, 113650.

Huang, J.-Z., Huang, W., & Ni, J. (2019). Predicting bitcoin returns using high-dimensional technical indicators. *The Journal of Finance and Data Science*, 5(3), 140–155.

Huyen, C. (2022). *Designing Machine Learning Systems*. O'Reilly Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer, 2$^{\text{nd}}$ edition.

Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. *The journal of finance and data science*, 7, 45–66.

Jha, R., De Paepe, M., Holt, S., West, J., & Ng, S. (2020). Deep learning for digital asset limit order books. *arXiv preprint arXiv:2010.01241*.

Kang, H.-J., Lee, S.-G., & Park, S.-Y. (2022). Information efficiency in the cryptocurrency market: The efficient-market hypothesis. *Journal of Computer Information Systems*, 62(3), 622–631.

Khuntia, S. & Pattanayak, J. (2018). Adaptive market hypothesis and evolving predictability of bitcoin. *Economics Letters*, 167, 26–28.

Kumar, A. (2021). Short-term prediction of crypto-currencies using machine learning. *Available at SSRN 3890338*.

Kyriazis, N. A. (2019). A survey on efficiency and profitable trading opportunities in cryptocurrency markets. *Journal of Risk and Financial Management*, 12(2), 67.

Lim, Y. (2022). *Deep Learning of the Order Flow for Modelling Price Formation*. PhD thesis, University College London.

Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management*.

Makarov, I. & Schoar, A. (2020). Trading and arbitrage in cryptocurrency markets. *Journal of Financial Economics*, 135(2), 293–319.

Maldonado, S., López, J., & Vairetti, C. (2019). An alternative smote oversampling strategy for high-dimensional datasets. *Applied Soft Computing*, 76, 380–389.

Nejat, A. (2021). The impact of order book and market information on bitcoin price movements. Master's thesis, HEC Montréal.

Petrova, Y. & Vilhelmsson, A. (2023). The information content in fx orderbook data. Unpublished manuscript.

Qureshi, F. (2018). Investigating limit order book features for short-term price prediction: A machine learning approach. *Available at SSRN 3305277*.

Rathore, R. K., Mishra, D., Mehra, P. S., Pal, O., Hashim, A. S., Shapi'i, A., Ciano, T., & Shutaywi, M. (2022). Real-world model for bitcoin price prediction. *Information Processing & Management*, 59(4), 102968.

Schei Norheim, B. (2019). High frequency lead-lag relationships in the bitcoin market. Master's thesis, Copenhagen Business School.

Silantyev, E. (2019). Order flow analysis of cryptocurrency markets. *Digital Finance*, 1(1-4), 191–218.

Tran, V. L. & Leirvik, T. (2020). Efficiency in the markets of crypto-currencies. *Finance Research Letters*, 35, 101382.

Valencia, F., Gómez-Espinosa, A., & Valdés-Aguirre, B. (2019). Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. *Entropy*, 21(6), 589.

Zaznov, I., Kunkel, J., Dufour, A., & Badii, A. (2022). Predicting stock price changes based on the limit order book: a survey. *Mathematics*, 10(8), 1234.

Zhou, S. & Mentch, L. (2023). Trees, forests, chickens, and eggs: when and why to prune trees in a random forest. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 16(1), 45–64.

A Appendix

Tables for BTC

Table A1: BTC single-exchange F1 scores from each fold for each exchange

Lag	Exchange	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	std	mean
1	Bitfinex	0.4751	**0.3800**	0.4389	0.4638	0.4543	0.4527	0.4516	0.4505	0.5024	0.4924	0.033	0.4177
2	Bitfinex	0.4039	**0.3278**	0.3869	0.4038	0.4027	0.3956	0.4093	0.4151	0.4351	0.3859	0.028	0.3631
3	Bitfinex	0.3657	**0.3109**	0.3661	0.3805	0.3756	0.3691	0.3877	0.3997	0.4244	0.3583	0.029	0.3425
4	Bitfinex	0.3417	**0.2833**	0.3546	0.3658	0.3641	0.3616	0.3765	0.3907	0.4064	0.3166	0.036	0.3270
5	Bitfinex	0.3313	0.2625	0.3398	0.3647	0.3567	0.3487	0.3699	0.3833	0.4035	**0.2436**	0.051	0.3141
6	Bitfinex	0.3440	**0.2666**	0.3439	0.3591	0.3600	0.3549	0.3681	0.3754	0.3965	0.2669	0.043	0.3162
7	Bitfinex	0.3344	0.2790	0.3375	0.3488	0.3532	0.3444	0.3584	0.3773	0.3984	**0.2490**	0.044	0.3113
8	Bitfinex	0.3221	0.3014	0.3462	0.3569	0.3531	0.3430	0.3637	0.3748	0.3919	**0.2379**	0.044	0.3122
9	Bitfinex	0.3745	0.2942	0.3357	0.3515	0.3497	0.3461	0.3583	0.3704	0.3850	**0.2486**	0.041	0.3141
1	Coinbase	0.5966	0.5996	0.5912	0.6045	0.6183	0.6117	0.6178	0.6042	**0.5624**	0.5959	0.016	0.5471
2	Coinbase	**0.4228**	0.4421	0.4308	0.4420	0.4515	0.4409	0.4559	0.4598	0.4386	0.4451	0.011	0.4037
3	Coinbase	0.3962	0.4088	0.3990	0.4097	0.4243	0.4049	0.4268	0.4289	0.4160	**0.3952**	0.013	0.3748
4	Coinbase	0.3759	0.3929	0.3837	0.3961	0.4165	0.3964	0.4098	0.4062	0.4022	**0.3794**	0.013	0.3611
5	Coinbase	**0.3587**	0.3821	0.3849	0.3882	0.4058	0.3865	0.3972	0.3972	0.4007	0.3785	0.013	0.3539
6	Coinbase	0.3672	0.3662	0.3733	0.3832	0.4008	0.3813	0.3847	0.3872	0.3939	**0.2987**	0.029	0.3423
7	Coinbase	0.3527	0.3603	0.3680	0.3722	0.3897	0.3734	0.3783	0.3831	0.3892	**0.2813**	0.032	0.3345
8	Coinbase	0.3458	0.3620	0.3678	0.3731	0.3952	0.3714	0.3778	0.3792	0.3915	**0.3219**	0.022	0.3370
9	Coinbase	0.3653	0.3410	0.3657	0.3666	0.3856	0.3711	0.3791	0.3747	0.3879	**0.3081**	0.024	0.3335
1	Gemini	0.3974	0.3718	**0.3711**	0.3753	0.4015	0.4076	0.4334	0.4047	0.4229	0.3729	0.022	0.3619
2	Gemini	0.3788	0.3698	0.3660	0.3712	0.3781	0.3774	0.3975	0.3701	0.3903	**0.3326**	0.017	0.3408
3	Gemini	0.3624	0.3429	0.3500	0.3494	0.3610	0.3653	0.3687	0.3595	0.3761	**0.3274**	0.014	0.3252
4	Gemini	0.3552	0.3392	0.3412	0.3557	0.3517	0.3646	0.3651	0.3537	0.3661	**0.3075**	0.018	0.3198
5	Gemini	0.3471	0.3369	0.3433	0.3480	0.3446	0.3573	0.3591	0.3481	0.3669	**0.3180**	0.013	0.3166
6	Gemini	0.3475	0.3356	0.3428	0.3456	0.3490	0.3551	0.3594	0.3438	0.3546	**0.2887**	0.020	0.3129
7	Gemini	0.3453	0.3309	0.3414	0.3454	0.3436	0.3507	0.3509	0.3426	0.3542	**0.2970**	0.016	0.3108
8	Gemini	0.3449	0.3296	0.3385	0.3379	0.3478	0.3435	0.3511	0.3424	0.3584	**0.2735**	0.024	0.3083
9	Gemini	0.3403	0.3336	0.3379	0.3441	0.3431	0.3446	0.3551	0.3431	0.3500	**0.2785**	0.021	0.3083

Note: This table shows the BTC F1 scores in each of the ten folds from the 10-fold cross-validation separately, for each lag. The last two columns show the arithmetic mean and standard deviation for each lag. Bold numbers represent the lowest F1 score among all 10 folds on that row.

Table A2: Difference in F1 scores from BTC single- and cross-exchange predictions

Predicted	Cross	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10
Bitfinex	Coinbase	0.0012	0.0317	0.0259	0.0197	0.0165	0.0107	0.0240	0.0230	0.0163	0.0157
Bitfinex	Gemini	0.0049	0.0215	0.0128	0.0123	0.0187	0.0102	0.0198	0.0217	0.0110	**-0.0011**
Coinbase	Bitfinex	0.0036	0.0016	0.0030	0.0000	**-0.0019**	0.0034	0.0017	**-0.0001**	0.0021	**-0.0016**
Coinbase	Gemini	**-0.0011**	0.0012	0.0022	0.0021	0.0001	0.0031	0.0001	**-0.0010**	0.0020	**-0.0018**
Gemini	Bitfinex	0.0126	0.0024	0.0100	0.0162	0.0018	0.0027	0.0053	0.0034	0.0048	0.0024
Gemini	Coinbase	0.0060	0.0175	0.0127	0.0220	0.0130	0.0028	0.0015	0.0074	0.0163	0.0369

Note: This table shows the difference in BTC F1 score for each fold in the 10-fold cross-validation, when using $\sqrt{16}$ **as the maximum number of features for each split in cross-exchange predictions**. The difference is calculated as the F1 score from cross-exchange prediction less the F1 score from single-exchange prediction. Bold numbers represent negative F1 score differences.

Table A3: Difference in F1 scores from BTC single- and cross-exchange predictions with maximum features of $\sqrt{32}$

Predicted	Cross	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	mean
Bitfinex	Coinbase	0.0050	0.0278	0.0247	0.0173	0.0167	0.0118	0.0222	0.0221	0.0160	0.0177	0.0181
Bitfinex	Gemini	**-0.0041**	0.0333	0.0168	0.0132	0.0168	0.0128	0.0134	0.0245	0.0091	0.0144	0.0150
Coinbase	Bitfinex	0.0019	0.0010	0.0014	0.0013	0.0008	0.0006	0.0013	0.0007	0.0014	**-0.0009**	0.0009
Coinbase	Gemini	**-0.0028**	0.0028	0.0021	0.0027	**-0.0003**	0.0042	0.0003	0.0008	0.0016	**-0.0025**	0.0009
Gemini	Bitfinex	0.0149	0.0048	0.0125	0.0151	0.0063	0.0063	0.0071	0.0055	0.0069	0.0181	0.0097
Gemini	Coinbase	0.0058	0.0170	0.0160	0.0222	0.0131	0.0041	0.0050	0.0140	0.0204	0.0436	0.0161

Note: This table shows the difference in BTC F1 score for each fold in the 10-fold cross-validation, when using $\sqrt{32}$ **as the maximum number of features for each split in cross-exchange predictions**. The difference is calculated as the F1 score from cross-exchange prediction less the F1 score from single-exchange prediction. Bold numbers represent negative F1 score differences.

Tables for ETH

Table A4: ETH single-exchange F1 scores from each fold for each exchange

Lag	Exchange	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	std	mean
1	Bitfinex	0.4300	0.4387	0.4203	0.4300	0.4303	0.4426	**0.4079**	0.4092	0.4682	0.4538	0.0188	0.4331
2	Bitfinex	**0.3706**	0.3989	0.3906	0.4023	0.4034	0.4085	0.3790	0.3822	0.4041	0.4146	0.0142	0.3954
3	Bitfinex	**0.3622**	0.3802	0.3680	0.3826	0.3842	0.3957	0.3678	0.3730	0.3685	0.3885	0.0108	0.3771
4	Bitfinex	0.3576	0.3769	0.3596	0.3660	0.3727	0.3816	**0.3473**	0.3651	0.3622	0.3787	0.0107	0.3668
5	Bitfinex	0.3530	0.3681	0.3526	0.3638	0.3692	0.3774	**0.3475**	0.3569	0.3513	0.3789	0.0112	0.3619
6	Bitfinex	0.3530	0.3637	**0.3367**	0.3675	0.3629	0.3673	0.3458	0.3529	0.3563	0.3751	0.0114	0.3581
7	Bitfinex	0.3539	0.3689	0.3536	0.3575	0.3602	0.3709	0.3410	0.3540	**0.3370**	0.3684	0.0114	0.3565
8	Bitfinex	0.3551	0.3644	0.3511	0.3518	0.3556	0.3571	**0.3270**	0.3449	0.3339	0.3647	0.0122	0.3506
9	Bitfinex	0.3557	0.3663	0.3439	0.3569	0.3565	0.3540	0.3451	0.3493	**0.3307**	0.3700	0.0113	0.3528
1	Coinbase	0.4648	0.4515	0.4067	0.4478	0.4550	0.4882	0.4320	**0.3540**	0.4707	0.4671	0.0387	0.4438
2	Coinbase	0.4035	0.4035	0.3787	0.4072	0.3975	0.4201	0.3488	**0.3221**	0.4144	0.4140	0.0320	0.3910
3	Coinbase	0.3746	0.3767	0.3449	0.3765	0.3810	0.3897	0.3266	**0.2913**	0.3835	0.3892	0.0324	0.3634
4	Coinbase	0.3678	0.3570	0.3361	0.3591	0.3658	0.3768	0.2776	**0.2631**	0.3684	0.3822	0.0416	0.3454
5	Coinbase	0.3652	0.3557	0.3335	0.3576	0.3577	0.3656	0.2886	**0.2629**	0.3476	0.3834	0.0375	0.3418
6	Coinbase	0.3647	0.3556	0.3369	0.3567	0.3532	0.3567	0.2759	**0.2710**	0.3523	0.3791	0.0367	0.3402
7	Coinbase	0.3581	0.3453	0.3435	0.3484	0.3469	0.3593	0.2816	**0.2464**	0.3492	0.3817	0.0404	0.3360
8	Coinbase	0.3621	0.3449	0.3328	0.3515	0.3511	0.3556	0.2909	**0.2569**	0.3419	0.3838	0.0369	0.3371
9	Coinbase	0.3684	0.3401	0.3374	0.3449	0.3465	0.3513	0.2669	**0.2478**	0.3359	0.3802	0.0420	0.3319
1	Gemini	0.3497	0.3462	0.3405	0.3297	0.3483	0.3750	0.3369	**0.3052**	0.3645	0.3612	0.0196	0.3457
2	Gemini	0.3369	0.3429	0.3256	0.3264	0.3449	0.3569	0.3185	**0.3051**	0.3517	0.3522	0.0168	0.3361
3	Gemini	0.3443	0.3371	0.3246	0.3311	0.3468	0.3532	0.3182	**0.2898**	0.3429	0.3443	0.0186	0.3332
4	Gemini	0.3385	0.3410	0.3315	0.3304	0.3390	0.3546	0.3182	**0.2923**	0.3466	0.3497	0.0181	0.3342
5	Gemini	0.3395	0.3405	0.3265	0.3265	0.3381	0.3376	0.3112	**0.2949**	0.3365	0.3427	0.0154	0.3294
6	Gemini	0.3352	0.3350	0.3219	0.3290	0.3356	0.3428	0.3139	**0.2972**	0.3396	0.3432	0.0146	0.3293
7	Gemini	0.3311	0.3350	0.3288	0.3291	0.3446	0.3486	0.3082	**0.2819**	0.3367	0.3489	0.0205	0.3293
8	Gemini	0.3362	0.3369	0.3174	0.3309	0.3387	0.3405	0.3084	**0.2884**	0.3379	0.3446	0.0178	0.3280
9	Gemini	0.3348	0.3418	0.3208	0.3306	0.3287	0.3431	0.2982	**0.2907**	0.3343	0.3471	0.0189	0.3270

Note: This table shows the ETH F1 scores in each of the ten folds from the 10-fold cross-validation separately, for each lag. The last two columns show the arithmetic mean and standard deviation for each lag. Bold numbers represent the lowest F1 score among all 10 folds on that row.

Table A5: Difference in F1 scores from ETH single- and cross-exchange predictions

Predicted	Cross	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10
Bitfinex	Coinbase	0.0161	0.0094	0.0059	0.0145	0.0095	0.0294	0.0171	0.0216	0.0116	0.0055
Bitfinex	Gemini	0.0063	0.0226	0.0271	0.0310	0.0365	0.0527	0.0511	0.0431	0.0590	0.0602
Coinbase	Bitfinex	0.0100	0.0059	0.0234	0.0129	0.0212	0.0061	0.0161	0.0311	0.0150	0.0031
Coinbase	Gemini	0.0074	0.0107	0.0301	0.0365	0.0385	0.0441	0.0584	0.0883	0.0773	0.0563
Gemini	Bitfinex	0.0029	0.0017	0.0052	0.0167	0.0063	0.0048	**-0.0147**	**-0.0004**	0.0126	0.0073
Gemini	Coinbase	0.0089	0.0111	0.0079	0.0176	0.0039	0.0057	0.0051	0.0018	0.0110	0.0074

Note: This table shows the difference in ETH F1 score for each fold in the 10-fold cross-validation, when using $\sqrt{16}$ **as the maximum number of features for each split in cross-exchange predictions**. The difference is calculated as the F1 score from cross-exchange prediction less the F1 score from single-exchange prediction. Bold numbers represent negative F1 score differences.

Table A6: Difference in F1 scores from ETH single- and cross-exchange predictions with maximum features of $\sqrt{32}$

Predicted	Cross	fold1	fold2	fold3	fold4	fold5	fold6	fold7	fold8	fold9	fold10	mean
Bitfinex	Coinbase	0.0202	0.0109	0.0166	0.0135	0.0048	0.0284	0.0271	0.0258	0.0181	0.0104	0.0176
Bitfinex	Gemini	0.0112	0.0229	0.0351	0.0315	0.0321	0.0562	0.0568	0.0556	0.0667	0.0663	0.0435
Coinbase	Bitfinex	0.0079	0.0054	0.0274	0.0154	0.0196	0.0047	0.0155	0.0432	0.0146	0.0051	0.0159
Coinbase	Gemini	0.0092	0.0102	0.0489	0.0388	0.0388	0.0425	0.0666	0.1010	0.0782	0.0608	0.0495
Gemini	Bitfinex	0.0036	0.0061	0.0115	0.0157	0.0101	0.0087	0.0021	0.0157	0.0170	0.0158	0.0106
Gemini	Coinbase	0.0100	0.0172	0.0156	0.0204	0.0107	0.0109	0.0131	0.0168	0.0158	0.0152	0.0146

Note: This table shows the difference in ETH F1 score for each fold in the 10-fold cross-validation, when using $\sqrt{32}$ **as the maximum number of features for each split in cross-exchange predictions**. The difference is calculated as the F1 score from cross-exchange prediction less the F1 score from single-exchange prediction.