HEC MONTRÉAL

The impact of order book and market information on Bitcoin price movements

par

Amin Nejat

Professeur Michèle Breton
HEC Montréal
Directrice de recherche

Sciences de la gestion
(Spécialisation Économie Financière Appliquée)

Mémoire présenté en vue de l'obtention
du grade de maîtrise ès sciences
(M. Sc.)

December 2021

Résumé

L'« impact sur les prix », c'est-à-dire la variation du prix d'un actif résultant d'une transaction, présente un intérêt certain pour tous les participants du marché, notamment l'établissement de stratégies de trading par les investisseurs et les teneurs de marché afin de réduire les coûts de transaction.

Dans ce mémoire, nous étudions empiriquement l'impact de diverses fonctionnalités, dont deux nouvelles variables extraites du carnet d'ordres limités, sur les variations de prix du Bitcoin sur la plateforme d'échange Coinbase, pour la période comprise entre le 1er janvier 2021 et le 3 mars 2021. Nous décomposons le modèle initial en deux modèles distincts selon la direction du changement de prix et vérifions que cette approche améliore la précision des prévisions. Nous proposons une façon d'utiliser cette information à partir d'un modèle de régression logistique pour la direction des mouvements du prix. Ces modèles sont ensuite utilisés pour concevoir une stratégie de trading simple et pour proposer deux méthodes illustrant comment le régulateur pourrait contrôler et surveiller les marchés pour contrer des activités de manipulation illégale des prix. Finalement, nous étudions l'impact du score de sentiment du marché, mesuré par les méthodes VADER et BERT à partir des tweets anglais de Twitter, sur les mouvements de prix du bitcoin.

Selon nos résultats, le volume des cotations du carnet d'ordres et la dispersion des prix par rapport au prix moyen ont un impact significatif sur les mouvements du prix, et ceci pour divers intervalles de temps. Il est intéressant de noter que la décomposition de la régression en deux modèles, l'un pour les variations de prix positives et l'autre pour les variations de prix négatives, améliore considérablement la précision des prévisions. En

ce qui concerne la mesure de l'influence du sentiment du marché sur les changements de prix, nous montrons que notre classificateur BERT affiné fait un très bon travail en catégorisant les tweets en trois classes, « positif », « négatif » et « neutre ». Notre régression avec le résultat de la fonction de sentiment du marché suggère que la variable de score de sentiment mesurée par le classificateur BERT décalé de deux périodes a un impact significatif sur la variation des prix.

Mots-clés

Impact sur les prix, régression linéaire, régression logistique, manipulation du marché, analyse des sentiments.

Abstract

"Price impact", which is the change in asset price that results from a transaction, is of crucial interest for market participants. Understanding this impact would help market makers and investors to optimize their trading strategy and consequently reduce their transaction costs.

In this thesis, we empirically study the impact of various features, including two novel variables extracted from the limit order book, on Bitcoin price changes on the Coinbase exchange, for the period between January 1st 2021 and March 3rd 2021. We then break the initial model into two separate models based on the price change direction and investigate whether this approach improves the forecasting precision. We propose a logistic regression model to predict the direction of price movements. Accordingly, we use these models to devise a simple trading strategy and to propose two illustrations of how the financial regulator could control and monitor markets for suspicious price manipulation activities. Finally, we study the impact of market sentiment score, measured by VADER and BERT methods from Twitter's English tweets, on bitcoin price movements.

According to our result, order book quotes' volume and price dispersion from mid-price have significant impact on price movements for various time intervals. Interestingly, breaking the regression into two models, one for positive price change and one for negative price change, improves the forecasting precision significantly.

Regarding measuring market sentiment influence on price changes, we find that our fine-tuned BERT classifier does a very good job in categorizing tweets in three classes, 'positive', 'negative' and 'neutral'. Our regression results with market sentiment feature

suggest that the sentiment score variable, measured by BERT classifier and 2 lags, has a significant impact on price change.

Keywords

Price Impact, Linear Regression, Logistic Regression, Market Manipulation, Sentiment Analysis

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to my amazing supervisor, Prof. Michèle Breton for her patience, immense knowledge and continuous support during my M.Sc. study.

I am extremely grateful to my parents and my little sister, for their love, support and encouragement. I would like to thank my parents for their countless sacrifices to help me pursue my dream and get the opportunity to study at such a wonderful university.

Last but not the least, my deepest gratitude to my caring, loving and supportive wife, who has stood by me through all my travails, and has always encouraged me to dream big and do my best in everything I set out to do.

Finally, I would like to thank "Centre d'intelligence en surveillance des marchés financiers" and Ramzi Ben-Abdallah for the funding received to support my research subject.

Chapter 1

Introduction

1.1 Objectives

The objective of this thesis is to investigate the price impact function and its possible applications, specifically in electronic cryptocurrency markets. More precisely, this thesis aims are:

(i) introducing new features for the price-impact function, and studying empirically their relations with price change.

(ii) developing a simple trading strategy using the developed price-impact function.

(iii) introducing methods to control market manipulation activities using developed price-impact functions.

(iv) investigating whether market sentiment score has any significant influence on price change.

These objectives are empirically tested and evaluated on the data collected from the Coinbase exchange for Bitcoin transactions.

1.2 Background

"Price-impact", which is the change in asset price that results from a trade, has been a very popular subject for both researchers and financial markets players in the past few years. It is important from both traders and market makers point of view. For traders, the explicit cost of trading consists of bid-ask spreads and commission fees, while price impact is an implicit and much more important cost of trading. Furthermore, it is crucial for market makers to be aware of the price-impact function, for instance, how an order for a large block of stocks affects the asset price, so they can optimize their execution strategy and increase their savings. This topic is referred as "Optimal Execution strategy" in the literature (see for instance Hendershott and Riordan [2013] and O'Hara [2015] for a discussion of how institutions use a price-impact function to develop an optimal trading strategy and minimize their costs).

The topic of price-impact has been investigated by researchers, both theoretically and empirically. One of the first studies in this field is the theoretical model proposed by Kyle [1985], which suggests that price impact of trades is positively related to order volume. Moreover, the theoretical and empirical findings of Keim and Madhavan [1996] and Kraus and Stoll [1972] show that price change is larger for trades that exceed the available market depth. This finding was recently applied in a research by Pham et al. [2020] to develop a novel price impact function that ignores zero-impact trades and only considers large transactions.

Another popular model in the price-impact literature is the function introduced by Lillo et al. [2003], which suggests that price change is a concave function of transactions' volume. This model was used and developed further by many researchers. For instance Almgren et al. [2005] replicated the same model and showed that adding the asset volatility improves its performance. Zhou [2012], applying this function to the Chinese Stock Market, added the average of price change for the trades in a given day in order to improved its precision.

Another interesting work in this field is Cont et al. [2014], which specifically focuses

on the impact of both trade and order imbalance on price movements and empirically finds that order imbalance has more impact on price change, compared to trade imbalance.

With recent technological developments and advances in the Artificial Intelligence (AI) field, many researchers and firms used AI techniques to propose various price-impact models. For instance, Kercheval and Zhang [2015] used Support Vector Machines (SVMs) to propose a classifier to predict the direction of mid-price change using order book data. More recently, Tsantekidis et al. [2020] used Long Short-Term Memory networks (LSTM) and Convolutional Neural Networks (CNN) to predict the direction of mid-price change, having order book quotes' price and size in the feature set and compared the result with a SVM model.

1.3 Outline

In this thesis, we use regression models to analyze the price impact function, using high-frequency data from the Coinbase exchange. We first discuss the empirical results of our regression model and the contribution of new features introduced in our model. Given that the impact function differs according to the price change direction, we propose a logistic regression model to complete the price-impact function. We then show how the price-impact function can be used to devise a trading strategy. Finally we compare the results achieved by our empirical model with the findings in the literature.

In a second part of this thesis, we focus on market manipulation and how our findings could be used to address these activities. After a brief introduction, we suggest an ex-post method to detect market manipulation, using order and trade imbalance. This method is then tested and evaluated on a real sample in our data set. At the end, an ex-ante method to detect manipulators and manipulation activities is proposed.

In last part of thesis, we investigate the idea that market sentiment is significantly related with price change. Two different methods, Valence Aware Dictionary for Sentiment Reasoning (VADER) and Bidirectional Encoder Representations from Transformers

(BERT), are used on Twitter posts to measure the market sentiment. This new feature is then added to the regression model and empirical results are discussed.

1.4 Contribution

In the literature of price-impact, many researchers focused on introducing new models and functions to improve the precision. Moreover, many tried to introduce new features to the model to improves its performance. While many different variables extracted from trades' attributes have been investigated, there has been little effort to include order book data in price-impact studies. Our first contribution is introducing new features to the price-impact model, measured from order book quotes.

Our second contribution is considering market manipulation and developing methods to address this issue. With recent technological advances in algorithmic trading, more control over activities in the market is required to avoid any act that could misprice financial assets and mislead traders. There are a few studies in the literature that try to address spoofing, which is a very important manipulation activity. This thesis specifically considers spoofing and proposes methods to monitor and detect such activities in the market.

Finally, our last contribution consists of introducing a market sentiment score into the price-impact model. Recent sudden movements in financial markets, which were triggered by activities in social media, suggest that financial assets are significantly influenced by market sentiment. We empirically investigate whether market sentiment could be a significant feature in predicting price change.

1.5 Overview of results

Our results show that features extracted from the order book are significant in the price-impact function. We also find that breaking the regression into two models according to the direction of a price change increases the forecasting precision of the price-impact

function significantly and that a logistic regression model does a good job in predicting the direction of price change.

In the second part of the thesis, two different methods are proposed to monitor market actions for possible market manipulation acts. An ex-post method is explained in details and illustrated on a sample data point; however, it is not possible to evaluate it further since we do not have access to data of real spoofing activities.

Finally, the results from the last section of the thesis suggest that our BERT model does a very good job in labeling tweets in comparison with VADER. Moreover, we find that this market sentiment score variable with two lags has significant impact on price change.

1.6 Organization of the thesis

The second chapter of this thesis is a detailed review of the literature on price-impact in details. Chapter 3 describes the data used for this research, the construction of each variable and the regression model used in this thesis. Chapter 4 reports on the regression results and discusses the coefficients measured for each feature in the model. This chapter also proposes a trading strategy based on a logistic regression. The chapter concludes by comparing our results with those of similar studies in the literature. Chapter 5 starts with a brief introduction to market manipulation and definition of some of common methods with this regard, and then moves to spoofing activities and possible ways to detect them. Chapter 6 introduces two different models for Natural Language Processing (NLP) classification and compares their relative performance on our data set. Finally, Chapter 7 is a short conclusion.

Chapter 2

Literature Review

With the latest advances in trading implementation and available computation resources to perform complicated computations, as well as the availability of high frequency history of market trades and quotes, many empirical and theoretical researches have been done to study the impact of order flow and market imbalance (excess of buy or sell orders for a specific security on a trading exchange) on price movements in order-driven markets. An order-driven market is a financial market where all buyers and sellers display the prices at which they wish to buy or sell a particular security, as well as the amounts of the security desired to be bought or sold.

The ability to understand the various aspects of price-impact (for example, the correlation between an incoming order, to buy or to sell, and the subsequent price change Bouchaud [2009]) and how different elements can impact this price change is of a great importance to financial markets participants since it helps them to optimize trades and minimize trading costs. For example, Chakravarty [2001] and Choi et al. [2019] argue that traders usually split trade blocks into smaller trades to mitigate the adverse impact on price. One of the applications of the price-impact function is the "Optimal Execution Problem" (OEP), in which the focus is to design an optimal strategy for an agent who has to execute a large order over a given time period. For instance, Obizhaeva and Wang [2013] propose an optimal trading strategy for the OEP under the assumption of a lin-

ear price impact imposed by trades volume. The findings of price impact studies have been widely used by researchers in institutions in designing algorithmic trading strategies. Algorithmic trading is the use of process- and rules-based algorithms to employ strategies for executing trades. With this regard, O'Hara [2015] and Hendershott and Riordan [2013] discuss that institutions use price impact relations to develop algorithmic trading strategies and reduce their transaction costs.

The previous studies in the literature focus on both the theoretical and empirical sides of price-impact. For instance, the theoretical model provided by Kyle [1985] and Foucault et al. [2013] show that price-impact of trades has positive correlation with order size. Karpoff [1987] explains how trade size and price-impact could correlate with each other. Moreover, the theoretical and empirical evidence provided in Keim and Madhavan [1996] suggest that the price-impact is larger for trades that exceed the opposite side of market's depth. Depth of market (DOM) is a measure for supply and demand of a traded asset. The empirical studies in the literature propose various models for the relation between order imbalance and price-impact; a similar intuition supports all of these models, that is, the price change is driven by the imbalance and inequality of supply and demand side in an Order-driven market. In other words, if the majority of trades are BUY initiated, meaning that there is more demand than supply in the market, while if the majority of trades are SELL initiated, meaning that there is more supply than demand in the market, and both imbalances could be a trigger to the price change.

The early studies in the literature focus on the price impact of trade blocks. For instance, Kraus and Stoll [1972] studies the temporary and permanent impact of large trades on price movements. Furthermore, Keim and Madhavan [1996] focus on the temporary and permanent price impact of large-block trades, using a polynomial regression for price-impact as the dependant variable modelled as third degree polynomial of blocks size. Biais et al. [1995], Coppejans et al. [2003] and Evans and Lyons [2002] argue that temporary price change is a function of trades' size, while the permanent price impact is influenced by information of traders.

Later works in the literature study the impact of various events on price jumps/drops.

For instance, Eisler et al. [2012] studies the impact of market orders, limit orders and orders cancellation on price change, using linear regression. Hopman [2007] focus on the impact of different order types on price, using linear regression for a range of time intervals, using a range of power orders to compute the imbalance variable.

One of the most cited papers in the literature, is the work of Lillo et al. [2003], which study the relation between single trade size and immediate price-impact. The authors show that the logarithm of price change is a concave function of trades' volume in a Quote-Driven Market and validate this theory on the data collected for the 1,000 largest firms on the New York Stock Exchange. The results show that the concavity measure of the price-impact function can be classified on the basis of firms' market capitalization. This study has been one of the core researches and most important models in this field (see Weber and Rosenow* [2005]). Lim and Coggins* [2005] replicate the same study in the context of the Australian order-driven market and get similar results. Both of these studies suggest that price-impact is greater for less liquid stocks and smaller for more liquid stocks.

Torre and Ferrari [1998] measure the price-impact of aggregated trades in 30-minutes intervals and show that price-impact is not only affected by trades size, but is also a function of stock volatility. Using this result, Almgren et al. [2005] modify the concave price-impact function by adding a volatility measure. Zhou [2012] also replicates the model suggested by Lillo et al. [2003] and finds that the measure extracted from firm's market capitalization does not work well for the Chinese stock market. As a result, the author proposes the addition of a new variable, averaging the value of historical price impact of all trades from the beginning of the trading day. Lim and Coggins* [2005] modify the original price-impact function by normalizing daily trades with respect to yearly average and test it on the Australian Stock Exchange.

Another approach proposed in the literature is to include the dollar value of trades in the price-impact function. For instance, Chen et al. [2002] use a Box-Cox transformation for price-impact, as a function of trades dollar value.

Finally, Wilinski et al. [2015] test these models on London-Stock-Exchange data and

show that the model proposed in Zhou [2012] performs better. Most recently, Pham et al. [2017] compares the performance of all of the above-mentioned models on the out-of-sample data from S& P/ASX 200, and also finds that the model provided in Zhou [2012] outperforms the other ones.

Interestingly, Plerou et al. [2002] include the impact of time interval lengths in the price-impact function. They show that price-impact is a concave function of trades size, and that the power is inversely related with time. They also suggest that the price-impact and number of trades relation takes the shape of a $tan(h)$ function.

Another innovation in the price-impact function consists of including an illiquidity proxy measure in the explanatory variables. This liquidity measure is proportional to daily returns and inversely related to daily volume (see Amihud [2002]) and, according to the findings of Marshall et al. [2012], has a significant impact on price change.

Some authors focus on the impact of trade size on stock return volatility. The findings of the study in Jones et al. [1994] suggest that the explanatory power of average trade size on return volatility is not significant. This finding is confirmed by Frino et al. [2009] who however show that the impact of mid-size trades on volatility is significant. This last finding is in line with the theoretical model proposed by Kyle [1985] and Barclay and Warner [1993], which suggests that informed traders will trade in mid-size blocks to hide their identities. This study also shows that the impact of buyer-initiated trades is more significant than that of seller-initiated trades. Chan and Fong [2000] use two-stage regression to study the price volatility-volume relation for a sample of NYSE and Nasdaq stocks, by including number of trades, size of trades and the trades imbalance in the explanatory variables. The empirical results suggest that size of trades impact is more significant compared to number of trades; moreover, the trades imbalance variable has a significant impact on both daily return and volatility-volume relation.

One stream of the price-impact literature uses time-series models, for instance using a Vector Autoregression (VAR) model (see Hasbrouck [1991] and Dufour and Engle [2000]), which is the generalized multivariate Autoregressive model, and specifies that the output variable depends linearly on its own previous values and on a stochastic term.

Hautsch and Huang [2012] proposes a cointegrated VAR system for limit orders short-run and long-run effect on price, by considering the aggressiveness of the trades, their size and the state of variables in the order-book. Most recently, Pham et al. [2020] proposes a Heterogeneous Autoregressive (HAR) structure (originally discussed in Corsi [2009]) for the price-impact relation, which includes a dummy variable for market depth so that only trades with volume greater than the depth of the other side of market are considered. The empirical findings of this research show that including this dummy variable improves the out-of-sample performance of price-impact models, and especially outperforms the structure proposed in Zhou [2012].

While there exist many studies on the subject of the relation between trades imbalance and price-impact, there are relatively few works in the literature that focus on the influence of quotes imbalance on price change. One of the major works in this direction is the research in Cont et al. [2014], which proposes a linear regression with both trades imbalance and quotes imbalance as explanatory variables. Interestingly, the empirical findings of this study shows that quotes imbalance has more explanatory power for price change than trades imbalance.

With the recent advances in computation technologies and the availability of market quotes data in millisecond intervals, the applications of Machine Learning methods in the field of quantitative finance have upsurged noticeably and many researchers have applied these methods to improve the price-impact models by including order book quotes in complicated Machine Learning (ML) model's features. ML is an application of Artificial Intelligence (AI), with the purpose of developing computer algorithms that can learn and improve by learning from experience and data. For instance, Kercheval and Zhang [2015] uses Support Vector Machines (SVM) to propose a classifier predicting the direction of mid-price change, by including a set of variables extracted from the price and volume of quotes in an equity limit order book. Fletcher and Shawe-Taylor [2013] uses similar model in foreign exchange market.

Another ML method commonly used in the literature is artificial neural networks (ANN), which are networks of node layers, containing an input layer, one or more hidden

layers, and an output layer. Each node connects to another and is activated if its output is above a specified threshold value, sending data to the next layer of the network. Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks are some of common ANN methods in the quantitative finance literature. For example, Tsantekidis et al. [2020] uses LSTM and CNN to develop a neural network to predict the direction of mid-price change, having order book quotes' price and size in the feature set, and compares the result with a SVM model.

Finally, many researchers and financial institutions use AI-based approaches to develop innovative models to predict price change. One popular approach in those studies is the inclusion of "market sentiment" as an explanatory variable. Market sentiment refers to an overall consensus and attitude of investors and traders toward a stock or financial markets. The intuition behind these models is that investors' emotion often drives stock markets, so that when "bears" are in control, prices tend to go down, whereas when the market is controlled by "bulls", prices go up. The researchers and professionals in this field apply Natural Language Processing (NLP) approaches to measure the market sentiment. NLP is a branch of AI that helps computers understand, interpret and manipulate human language contents, like corpus and speeches.

Various NLP approaches have been used to study the impact of investors' sentiment and news on price movements. For instance, Niederhoffer [1971] categorizes *New York Times* headlines into 19 categories and studies the relation between them and price movements, and Davis et al. [2006] and Tetlock [2007] study the impact of positiveness and negativeness of news on price changes. One approach applied by researchers to news impact analysis is "bag-of-words" (see Seo et al. [2004] and Schumaker and Chen [2009]). The *Bag-of-words* approach extracts features from text on the basis of the occurrence of words in the data set, mapping those words to outcome labels, and making predictions for out-of-sample data. Another common approach in market sentiment analysis is the use of *rule-based* algorithm. This approach uses a set of pre-defined rules and scores for words or word combinations to compute a total score for a given text. For instance, Li et al. [2014] use a rule-based algorithm to study the impact of news on stock price return.

Their findings show that this approach outperforms bag-of-word models. Considering that many traders and investors express their feelings regarding stocks and financial markets through social media platforms, many researchers analyze the correlation between social media sentiment and price changes. For instance, Chen and Lazer [2013], Mittal and Goel [2012] and Rao et al. [2012] focus on *Twitter* contents and how tweets impact price movements.

In this thesis, we aim to contribute to the literature on price-impact using order-book information from one of the largest Bitcoin Exchanges. We investigate the impact of new variables, namely measuring market imbalance from order-book information, as well as social media sentiment.

Chapter 3

Model Structure, Variables, and Data

3.1 Cryptocurrency and Bitcoin market

3.1.1 What is cryptocurrency?

A cryptocurrency is a form of digital asset or virtual currency based on a network that is distributed across a large number of computers. Cryptocurrencies are secured by cryptography, which makes them almost impossible to counterfeit. This decentralized structure allows them to exist outside the control of governments, thus being used for illegal activities and facing many criticism.

Many cryptocurrencies are based on blockchain technology—a distributed ledger enforced by a disparate network of computers, which is a method for ensuring the integrity of transactional data. The database maintains a secure and decentralized record of transactions in digital format and shares it with the nodes in the network; this method guarantees the fidelity and security of the data and generates trust without the need for a trusted third party.

The first cryptocurrency based on blockchain technology was Bitcoin, which still remains the most popular, valuable and liquid cryptocurrency. Today, there are thousands of alternate cryptocurrencies with various functions and specifications. Some of these digital assets are clones of Bitcoin, while others are new currencies that were built from scratch,

including Ethereum, Cardano, and Solana. By November 2021, the aggregate value of all the cryptocurrencies in existence is over $2.4 trillion with more than $130 billion of daily trades volume (see `https://coinmarketcap.com/charts/`).

3.1.2 Understanding blockchain

Blockchain is a shared database that facilitates the process of recording transactions and tracking assets in a business network. This ledger stores information electronically in digital format and is shared among a network of computer nodes, which makes it nearly immutable so that it is impossible to duplicate a transaction. Basically anything of value can be traded and tracked on a blockchain network, including tangible and intangible assets.

As a transaction happens, it is entered into a fresh block of data, showing the details of movement of an asset. Once the block is filled with data, it is chained onto the previous block, which makes the data chained together in chronological order. Each additional block strengthens the verification of the previous block and hence the entire blockchain. This process makes any tamper in the ledger evident, resulting in a database that everyone in the network can trust.

3.2 Bitcoin, its history and market

3.2.1 History

Bitcoin, also known as BTC, is the first cryptocurrency (created in January 2009) and is presently the most valuable and commonly held digital asset. Bitcoin follows the ideas set out in a white paper by the mysterious and pseudonymous Satoshi Nakamoto, while the identity of the Bitcoin creator is still a mystery. All Bitcoin transactions are verified by a massive amount of computing power via a process known as "mining". Despite it not being legal tender in most parts of the world (El Salvador is the only country that officially adopted Bitcoin as legal tender in September 2021), Bitcoin is very popular and

has triggered the launch of hundreds of other cryptocurrencies. Although Bitcoin is seriously criticized for being used in illegal transactions, the massive electricity consumption of its mining network, price volatility and insecure exchanges, it is widely traded by investors as a profitable asset and can be traded with most currencies and some services and products.

3.2.2 Understanding Bitcoin

Bitcoin is created by a process called Bitcoin mining, which refers to the process of solving a mathematical puzzle generated by Bitcoin's algorithm. Since Bitcoin is based on a blockchain network, by solving this puzzle, Bitcoin miners verify transaction information and make the Bitcoin network trustworthy. The miners verify one megabyte (MB) worth of transactions, which is the size of a single block. Depending on how much data each transaction stores, blocks can be as small as one transaction but more often contain several thousands.

Bitcoin miners assemble valid transactions into a block and, if this block is accepted and verified by other miners, then the miner receives a block reward. The block reward is halved every 210,000 blocks (or roughly every four years). Starting at 50 in 2009, the reward was changed to 6.25 in its most recent halving event.

The other revenue stream for Bitcoin miners who participate in the mining process is transaction fees that are received for any transaction in the verified block. Consequently, when Bitcoin network production reaches its planned limit of 21 million (expected around 2140), miners will only be rewarded by the fee for processing transactions. These fees will ensure that miners still have the incentive to mine and keep the network going.

3.2.3 Bitcoin market

Bitcoin constitutes the largest portion of cryptocurrency market cap. As of November 2021, bitcoin represents approximately 42% of the total value of the crypto market, which is evaluated to more than $1 trillion, with 24h trading volume exceeding $24 billion.

The volatility of Bitcoin price attracted many investors and financial institutions interest toward its market, specifically large and well-known financial institutions' investment in this market upsurged significantly during past two years.

Bitcoin can be traded in cryptocurrency exchanges, also called digital currency exchanges (DCE), which allow customers to trade cryptocurrencies or digital currencies for other assets. As of November 2021, Binance, Coinbase, FTX and Kraken are the four largest cryptocurrency exchanges active in the business (`https://coinmarketcap.com/rankings/exchanges/`).

3.2.4 Bitcoin price and volatility

The price of Bitcoins has gone through cycles of appreciation and depreciation since its creation, especially since 2017. In 2011, the value of one Bitcoin rapidly rose from about $0.30 to $32 before dropping to $2. In the latter half of 2012, the bitcoin price began to rise, reaching a high of $266 on 10 April 2013, before crashing to around $50. In 2017, Bitcoin experienced a historic bullish run, from $900 in March to almost $20,000 by the end of 2017, before returning to $6000 range in April 2018. The latest sharp movement in Bitcoin price happened in the latter half of 2020, from $1000 range in September to $63,000 in April 2021. This bullish rally was followed by a notable depreciation to $30,000 in June 2021. Bitcoin hit an all-time-high on November 2021 when it went above $68,000 for the first time.

3.3 Data

For the purpose of this study, quotes and trades data are collected from the Coinbase exchange. Coinbase is a purely electronic market, and as of November 2021, was the second largest cryptocurrency exchange, with more than $7 billion 24h volume, from which more than $1 billion were generated by bitcoin transactions.

Coinbase supports various types of orders, including market orders (orders that will

be executed immediately at the best offer without price limit), limit orders (orders that will be fulfilled at a specified price) and stop orders (automatic issuing of limit orders or market orders when a given price is reached).

Our data set comprises limit order book data, trades attributes from the Coinbase exchange between January 1st 2021 and March 2nd 2021. Since Bitcoin is a 24/7 market, the data is not limited to any hours like what we observe in traditional markets. The data is collected through the API provided by Coinbase.

Table 3.1 provides the description of variables collected from the Coinbase database.

Table 3.1: Definition of variables collected through Coinbase API

Variables	Description
$v_\tau^{a,i}$	Volume of i^{th} quote in the ask side of order book at time τ
$v_\tau^{b,i}$	Volume of i^{th} quote in the bid side of order book at time τ
$p_\tau^{a,i}$	Price of i^{th} quote in the ask side of order book at time τ
$p_\tau^{b,i}$	Price of i^{th} quote in the bid side of order book at time τ
$v_t^{buy\ or\ sell}$	Size of a buy or sell initiated trade fulfilled at time t^*
$p_t^{buy\ or\ sell}$	Size of a buy or sell initiated trade fulfilled at time t

* subscript t differs from subscript τ since trades and quotes are not necessarily done at the same time

3.3.1 Limit order book (LOB) data

Our data contains information on the accumulated market depth for the top ten quotes on both bid and ask sides of the Coinbase market, and this data was collected every 10 seconds. For instance, the data collected from the limit order book on February 1st at 00:00:00 looks as Table 3.2.

Table 3.3 displays the statistics for all the limit order book quotes collected for the period under investigation in this thesis.

Table 3.2: Limit order book data on February 1, 2021 at 00:00:00 Zulu time

time (τ)	ask-size ($v_\tau^{a,i}$)	ask-price ($p_\tau^{a,i}$)	bid-price ($p_\tau^{b,i}$)	bid-size ($v_\tau^{b,i}$)
2021-02-01T00:00:00.381213000Z	0.500000	33136.07	33133.66	0.089600
2021-02-01T00:00:00.381213000Z	0.065140	33137.70	33130.40	0.030200
2021-02-01T00:00:00.381213000Z	0.380738	33137.71	33130.00	0.045000
2021-02-01T00:00:00.381213000Z	0.499550	33137.75	33124.15	0.001000
2021-02-01T00:00:00.381213000Z	0.065140	33139.24	33122.86	0.001250
2021-02-01T00:00:00.381213000Z	1.500230	33139.25	33122.55	0.015095
2021-02-01T00:00:00.381213000Z	1.150000	33139.29	33118.85	0.603000
2021-02-01T00:00:00.381213000Z	0.100000	33139.82	33118.81	0.268300
2021-02-01T00:00:00.381213000Z	0.787269	33140.06	33112.78	0.080833
2021-02-01T00:00:00.381213000Z	0.750166	33140.30	33111.40	0.157009

Table 3.3: Statistics for limit order book quotes from January 1st 2021 to March 2nd.

		Ask			Bid			Mid-price		
Date of observation	# orders	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
Jan. 1 to Jan. 7	1,020,000	1e-08	0.69	270.34	4e-08	0.76	230.49	27,744.875	32,085.75	36,500.00
Jan. 8 to Jan. 14	1,019,990	1e-08	0.59	228.41	1e-08	0.52	217.93	30,100.500	37,285.31	41,980.62
Jan. 15 to Jan. 21	1,020,000	1e-08	0.48	154.45	1e-08	0.45	140.48	33,406.550	36,338.59	39,695.14
Jan. 22 to Jan. 28	1,020,000	1e-08	0.50	202.81	3.4e-07	0.53	217.77	28,767.415	32,160.68	34,884.28
Jan. 29 to Feb. 4	1,020,000	1e-08	0.52	280.00	1e-08	0.48	114.04	31,990.005	34,593.95	38,646.82
Feb. 5 to Feb. 11	1,020,000	1e-08	0.54	304.22	1e-08	0.46	117.91	36,623.585	41,630.60	48,196.50
Feb. 12 to Feb. 18	1,010,000	1e-08	0.51	205.88	1e-08	0.44	264.44	45,905.660	48,516.31	52,640.35
Feb. 19 to Feb. 25	1,019,990	1e-08	0.49	262.45	1e-08	0.46	216.15	44,937.340	53,313.96	58,363.39
Feb. 26 to March 2	680,000	1e-08	0.41	137.92	1e-08	0.36	197.75	43,025.165	46,561.63	49,813.37

"# orders" display the number of ask (bid) quotes collected; Ask and Bid columns show the statistics for quotes' size and the last three columns report the mid-price statistics for each period

3.3.2 Trades attributes

For each successful transaction completed in the Coinbase exchange, price, size, time and direction (the direction of the initial quote when it was posted) is recorded. Table 3.4 shows 10 trades, closest to the time February 1st at 00:00:00, fulfilled on the Coinbase exchange.

Furthermore, the statistics for all the trades fulfilled during the period from January 1st 2021 to March 2nd are shown in Table 3.5

Table 3.4: Trades attributes of 10 sample transactions fulfilled on Coinbase on February 1st

time (t)	size ($v_t^{buy\ or\ sell}$)	price	direction
2021-02-01T00:00:00.122558000Z	0.000450	33137.75	buy
2021-02-01T00:00:00.442397000Z	0.003822	33130.40	sell
2021-02-01T00:00:00.536390000Z	0.363181	33133.65	buy
2021-02-01T00:00:00.631990000Z	0.018473	33130.40	sell
2021-02-01T00:00:01.064573000Z	0.007437	33130.67	buy
2021-02-01T00:00:01.064573000Z	0.067641	33133.12	buy
2021-02-01T00:00:01.159154000Z	0.003299	33130.40	sell
2021-02-01T00:00:01.507120000Z	0.295698	33133.21	buy
2021-02-01T00:00:01.739332000Z	0.077548	33130.51	sell
2021-02-01T00:00:01.739332000Z	0.004606	33130.40	sell

Table 3.5: Stats for trades fulfilled on Coinbase in the period of January 1st 2021 to March 2nd.

		Trades' size			Total volume	Total volume	% buy initiated	% sell initiated
date of observation	# trades	Min	Mean	Max	(#BTC)	(bn $)	trades	trades
Jan. 1 to Jan. 7	2,181,995	1e-08	0.110	100.000	239,509.01	7.765	60.55%	39.45%
Jan. 8 to Jan. 14	3,032,997	1e-08	0.102	33.802	309,473.34	11.218	59.45%	40.55%
Jan. 15 to Jan. 21	2,603,996	1e-08	0.091	30.000	237,995.71	8.587	61.15%	38.85%
Jan. 22 to Jan. 28	1,612,992	1e-08	0.106	41.517	170,493.40	5.402	59.52%	40.48%
Jan. 29 to Feb. 4	1,867,995	1e-08	0.095	36.119	177,845.95	6.212	61.21%	38.79%
Feb. 5 to Feb. 11	2,259,993	1e-08	0.078	188.556	175,210.23	7.463	61.64%	38.36%
Feb. 12 to Feb. 18	1,884,997	1e-08	0.063	100.000	118,598.21	5.773	61.6%	38.4%
Feb. 19 to Feb. 25	3,706,986	1e-08	0.058	74.000	215,921.65	11.185	61.57%	38.43%
Feb. 26 to March 2	1,807,989	1e-08	0.055	43.476	100,091.41	4.657	62.68%	37.32%

"# trades" display the number of trades, "Trade size" columns show the statistics of trade sizes, "Total Volume" displays the sum of all trade volumes in both BTC and billion $, and the last two columns show the ratio of buy-initiated trades sell-initiated trades.

3.4 Model and variables

The predictors used in the literature for the price impact function usually include volume and volatility. The model used for this study is a linear regression, and for this regression a set of variables for quote and trades price and volume is extracted from the collected data. Table 3.6 gives the description of the variables used in the regression model.

Table 3.6: Definition of features used in the regression model

Feature	Description
ΔP_t	Change in mean of mid-price from time interval t to $t+1$
V_t^{orders}	Difference between bid side and ask side quotes' volume in interval t
P_t^{orders}	Sum of ask and bid quotes' price dispersion from mid-price at interval t
V_t^{trades}	Difference between buy-initiated and sell-initiated trades' volume at interval t
P_t	Distance of mid-price from the moving average price calculated for last q intervals
$\frac{v_t}{\overline{v}_t}$	Ratio of buy (sell) initiated trades volume at t to all buy (sell) trades for the last κ intervals
$\sigma_{t,m}$	Volatility of Bitcoin price return for the last m intervals

We measure volume using three different variables, V_t^{orders}, which is not widely used so far in the literature and is the mean of difference between volume of bid side and ask side of the limit order book during each interval t, V_t^{trades}, which represents the net value difference between buy initiated trades and sell initiated trades fulfilled during an interval, and $v_t/\overline{v}_t$ where v_t is the volume of trades during interval t and $\overline{v}_t$ is the the average volume of all trades in a period prior to (and including) the t-th trade (and have the same direction as that trade). In order to measure volatility, we use σ_t, which is calculated as the standard deviation of the mid-quote returns and P_t, which is the ratio of the mean of mid-price in interval t with respect to the moving average price. Moreover, ΔP_t is the mid-price change from interval t to $t+1$, P_t^{orders} is the mean of difference of all quotes (both bid side and ask side) during an interval from the mid price at the quote instant. Details on the computation of these variables are provided in the following section.

3.4.1 Model features

This section defines the variables used in the regression and described in Table 3.6.

ΔP_t is the dependent variable in the regression and is measured by the following equation,

$$\Delta P_t = \frac{\overline{P}_{t+1}^m}{\overline{P}_t^m}, \tag{3.1}$$

where $\overline{P}_t$ is the average of mid prices during the interval t, and where the mid price is the

average of the best bid and ask quote prices at each instant,

$$p_\tau^m = \frac{p_\tau^a + p_\tau^b}{2}, \tag{3.2}$$

where τ is an instant during t-th interval.

The volume variable V_t^{orders} is computed using

$$V_t^{orders} = \frac{\sum_{\tau=1}^{T} \sum_{i=1}^{N} v_\tau^{b,i} - v_\tau^{a,i}}{T}, \tag{3.3}$$

where $v_\tau^{b,i}$ and $v_\tau^{a,i}$ are the volume of the i-th quotes of bid side and ask side of the limit order book at instant τ respectively.

The price dispersion variable P_t^{orders} is computed using

$$P_t^{orders} = \frac{\sum_{\tau=1}^{T} \sum_{i=1}^{N} [\frac{p_\tau^{b,i}}{p_\tau^m} - 1] + [\frac{p_\tau^{a,i}}{p_\tau^m} - 1]}{T}, \tag{3.4}$$

where $p_\tau^{b,i}$ and $p_\tau^{a,i}$, respectively, are the prices of the i-th quotes of the bid side and of the ask side of the limit order book at instant τ, and p_τ^m is the mid-price at τ. This variable measures how quotes are dispersed from the mid price during each interval.

The difference variable V_t^{trades} is computed using

$$= \sum v_t^{buy} - \sum v_t^{sell}, \tag{3.5}$$

where v_t^{buy} and v_t^{sell} are the volume of buy and sell trades completed during interval t to $t+1$.

The distance variable P_t is computed using

$$P_t = \frac{\overline{p}_t^m}{\frac{1}{q} \sum_{q=0}^{l} \overline{p}_{t-q}^m} - 1, \tag{3.6}$$

where the denominator represents the moving average of bitcoin prices in the q-interval period prior to t.

Moreover, the ratio $\frac{v_t}{\overline{v}_t}$ is computed using

$$\frac{v_t}{\overline{v}_t} = \begin{cases} \dfrac{\sum v_t^{buy}}{\frac{1}{\kappa}\sum_{j=0}^{\kappa} v_{t-j}^{buy}} & \text{if } v_t^{buy} > \sum v_t^{sell} \\ \dfrac{\sum v_t^{sell}}{\frac{1}{\kappa}\sum_{j=0}^{\kappa} v_{t-j}^{sell}} & \text{if } v_t^{sell} > \sum v_t^{buy} c, \end{cases} \tag{3.7}$$

in which κ represents the length of the period prior to t that is used to compute the average trade size.

Finally,

$$\sigma_{t,m} = Standard\ Deviation(R_t, m) \quad \text{where} \quad R_t = ln(\frac{\overline{P}_{t+1}^m}{\overline{P}_t^m}) \tag{3.8}$$

is the standard deviation of the Bitcoin return for the last m intervals.

3.4.2 Price impact model

The regression model proposed in this thesis is described by Equation 3.9, in which ΔP_t is the dependent variable, the features described in Table 3.6 are used as regressors, and η_t is the residual error.

$$\begin{aligned} \Delta P_t = \theta_1 V_t^{orders} + \theta_2 P_t^{orders} + \theta_3 V_t^{trades} + \theta_4 P_t \\ + \theta_5 \Delta P_{t-1} + \theta_6 \frac{v_t}{\overline{v}_t} + \theta_7 \sigma_t + \eta_t \end{aligned} \tag{3.9}$$

Chapter 4

Results and Discussion

In this chapter, we discuss the performance of the regression model described in Chapter 3 on collected data. Two different approaches will be investigated. In the first part, we study the relation between the independent and dependent variables and we examine whether the features included in the regression model are significant in explaining the variations in the dependent variable. In the second part, we investigate the performance of the model in forecasting price movements.

4.1 Independent features and price change

To have a better understanding of the relation between the independent features and the dependent variable, we run the regression in Equation 3.9 using various time windows. Moreover, since features P_t, $\frac{v_t}{\overline{v}_t}$ and σ_t in Equation 3.9 are measured by looking back to previous values, we investigate various look-back windows. More specifically, we used 2-minute, 5-minute and 10-minute windows for the time intervals and look-back period with length of 10, 50, and 100 time intervals (κ, q and m variables introduced in Section 3.4.1).

4.1.1 Empirical results

Table 4.1 shows the regression results (coefficients and p-values) for different windows and look-back periods, along with their R^2 coefficients.

Table 4.1: Regression results for various windows and look-back periods.

	2-minute window (43316 # obs.)			**5-minute window** (17338 # obs.)			**10-minute window** (8678 # obs.)		
	# look-back periods			# look-back periods			# look-back periods		
	10	50	100	10	50	100	10	50	100
θ_1	5.24E-04 (0)	5.30E-04 (0)	5.34E-04 (0)	1.07E-03 (0)	1.10E-03 (0)	1.11E-03 (0)	1.64E-03 (0)	1.71E-03 (0)	1.75E-03 (0)
θ_2	3.43E+00 (0)	3.36E+00 (0)	3.33E+00 (0)	6.94E+00 (0)	6.69E+00 (0)	6.60E+00 (0)	6.93E+00 (4.8E-05)	6.79E+00 (7.2E-05)	7.08E+00 (3.6E-05)
θ_3	1.99E-05 (0)	1.98E-05 (0)	1.98E-05 (0)	1.72E-05 (0)	1.71E-05 (0)	1.72E-05 (0)	1.24E-05 (0)	1.24E-05 (0)	1.25E-05 (0)
θ_4	-2.20E-04 (0)	-2.03E-04 (0)	-1.71E-04 (0)	-5.65E-04 (0)	-4.61E-04 (0)	-4.03E-04 (0)	-8.55E-04 (0)	-7.05E-04 (1.8E-08)	-7.07E-04 (8.1E-08)
θ_5	1.09E-01 (0)	1.12E-01 (0)	1.13E-01 (0)	9.89E-02 (0)	1.03E-01 (0)	1.04E-01 (0)	1.22E-01 (0)	1.24E-01 (0)	1.24E-01 (0)
θ_6	5.40E-04 (2.7E-05)	3.10E-03 (0)	5.74E-03 (0)	2.03E-03 (4.7E-08)	1.03E-02 (0)	1.84E-02 (0)	4.39E-03 (1.0E-08)	1.94E-02 (0)	3.92E-02 (0)
θ_7	1.03E-01 (0)	7.76E-02 (0)	6.21E-02 (0)	1.35E-01 (0)	8.45E-02 (3.1E-08)	7.12E-02 (1.2E-05)	1.01E-01 (4.3E-07)	6.72E-02 (4.1E-03)	6.74E-02 (7.5E-03)
R^2	11.0%	11.0%	11.0%	10.6%	10.0%	10.0%	9.0%	9.0%	9.0%

Each row presents the corresponding feature' coefficient in the regression model and its p-values (in brackets); the last row contains each regression's R^2.
* p-values less than $1e-8$ are reported as 0 for convenience.

These results support the idea that selected features have statistically significant impact in explaining price changes, and this conclusion could be made for all windows and look-back periods. However, the R^2 of regressions are low, which suggests that these features and regression model are not sufficient for prediction purposes. It should be noted that the precision performance of the regression model decreases with the length of of the computation windows.

4.1.2 Interpretation

One important aspect of investigating the relation between independent features and the dependent variable is the economical intuition, that is, whether the coefficients signs can be justified by financial explanations. For easy reference, we recall the regression model:

$$\Delta P_t = \theta_1 V_t^{orders} + \theta_2 P_t^{orders} + \theta_3 V_t^{trades} + \theta_4 P_t + \theta_5 \Delta P_{t-1} + \theta_6 \frac{v_t}{\overline{v}_t} + \theta_7 \sigma_t + \eta_t.$$

θ_1 and θ_3, which are the coefficients related to orders and trades volume impact on price change, are both positive, which intuitively makes sense. A positive sign means that if the volumes on ask side or buy-initiated trades are higher, the price will increase, and this can be justified by the law of demand.

θ_2 is the coefficient for dispersion of order book quotes' price from mid price. If P_t^{orders} has a positive value, it means the ask quotes are further from mid-price and sellers are less interested in selling at mid-price; as a result we expect the price to increase, and this is confirmed by the positive value of θ_2.

P_t represents the distance of the mid-price from the moving average price. If this feature is positive, the price is above the moving average and we expect the price to decrease to the average level, and vice versa for negative values. The negative sign of θ_4 coefficient supports this intuition as well.

θ_5 represents the impact of the price change in the last period (ΔP_{t-1}) on the dependent variable. We expect this impact to be positive, since when the market is bullish investors are more attracted in buying the underlying and the price increases, and when the market is bearish the investors are more willing to sell and the price decreases. This is supported by the negative sign of θ_5.

$\frac{v_t}{\overline{v}_t}$ always has a positive value and the sign of its coefficient, θ_6, is also positive, which suggests that increase in market activity and trading volume will increase the price. The regression results suggest that this variable has a significant impact on the price change.

Finally, the coefficient θ_7 is positive and significant. The corresponding volatility feature is also always positive and the positive coefficient suggests that higher uncertainty

Table 4.2: OLS Regression result for 5-minute window and 10 look-back period

Dep. Variable:	Y	**R-squared (uncentered):**	0.106
Model:	OLS	**Adj. R-squared (uncentered):**	0.106
Method:	Least Squares	**F-statistic:**	293.4
Date:	Sat, 25 Dec 2021	**Prob (F-statistic):**	0.00
Time:	18:35:31	**Log-Likelihood:**	75008.
No. Observations:	17338	**AIC:**	-1.500e+05
Df Residuals:	17331	**BIC:**	-1.499e+05
Df Model:	7		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0011	7.15e-05	15.004	0.000	0.001	0.001
θ_2	6.9424	0.704	9.857	0.000	5.562	8.323
θ_3	1.723e-05	6.28e-07	27.418	0.000	1.6e-05	1.85e-05
θ_4	-0.0006	6.36e-05	-8.878	0.000	-0.001	-0.000
θ_5	0.0989	0.009	11.562	0.000	0.082	0.116
θ_6	0.0020	0.000	5.466	0.000	0.001	0.003
θ_7	0.1349	0.013	10.176	0.000	0.109	0.161

and volatility in the market tend to increase the price.

4.1.3 Breaking the regression model

A common methodology in the literature to improve the forecasting precision of the price-impact model is to break the regression into two models. The intuition is that price change in a bullish/bearish rally, or when there is sell/buy imbalance, could behave differently and it is plausible that impact of features on price change in these different market conditions could vary. We will apply the same idea on our data set, dividing it into two parts (positive and negative price change).

In this section we focus on a 5-minute window and 10 look-back periods. Table 4.2 displays the regression results on the complete data set.

We now present the results using two different regression models, one for positive price changes and one for negative price changes, in Tables 4.3 and 4.4, respectively.

Table 4.3: OLS Regression Results for positive price changes over 5-minute windows and 10 look-back period

Dep. Variable:	Y	**R-squared (uncentered):**	0.581
Model:	OLS	**Adj. R-squared (uncentered):**	0.580
Method:	Least Squares	**F-statistic:**	2437.
Date:	Sat, 25 Dec 2021	**Prob (F-statistic):**	0.00
Time:	19:45:08	**Log-Likelihood:**	41531.
No. Observations:	8812	**AIC:**	-8.305e+04
Df Residuals:	8807	**BIC:**	-8.302e+04
Df Model:	5		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0004	6.46e-05	5.992	0.000	0.000	0.001
θ_3	5.685e-06	5.04e-07	11.280	0.000	4.7e-06	6.67e-06
θ_4	-0.0003	5.93e-05	-5.671	0.000	-0.000	-0.000
θ_6	0.0076	0.000	21.287	0.000	0.007	0.008
θ_7	0.5644	0.012	48.019	0.000	0.541	0.587

* Coefficients θ_2 and θ_5 do not appear and the corresponding variables were removed from this regression since these were not statistically significant.

Table 4.4: OLS Regression Results for negative price changes over 5-minute windows and 10 look-back period

Dep. Variable:	Y	**R-squared (uncentered):**	0.567
Model:	OLS	**Adj. R-squared (uncentered):**	0.567
Method:	Least Squares	**F-statistic:**	1597.
Date:	Sat, 25 Dec 2021	**Prob (F-statistic):**	0.00
Time:	19:45:08	**Log-Likelihood:**	39901.
No. Observations:	8526	**AIC:**	-7.979e+04
Df Residuals:	8519	**BIC:**	-7.974e+04
Df Model:	7		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0005	7.55e-05	6.758	0.000	0.000	0.001
θ_2	6.7385	0.714	9.440	0.000	5.339	8.138
θ_3	1.122e-05	6.69e-07	16.782	0.000	9.91e-06	1.25e-05
θ_4	-0.0002	6.76e-05	-2.476	0.013	-0.000	-3.49e-05
θ_5	0.0589	0.009	6.405	0.000	0.041	0.077
θ_6	-0.0043	0.000	-10.966	0.000	-0.005	-0.004
θ_7	-0.5070	0.014	-35.157	0.000	-0.535	-0.479

Comparing the regression results in Tables 4.3 and 4.4 with the ones in Table 4.2 lead to the following observations:

1. The explanatory power of the model significantly increased when considering the direction in the price change, going from 10.6% value of R^2 for the global data set to 58.1% and 56.7% for positive and negative price changes, respectively.

2. Statistically significant features in positive/negative regression are not necessarily the same.

3. While signs are the same, some of the estimated coefficients in the three regressions are noticeably different.

4. This seems to confirm the original assumption of different sensitivities to independent features in different market condition.

To conclude, we note that there is a well-known problem in the literature, referred as "Too big to fail", which arises when the sample size is very large, resulting in small p-values so that any feature appears to be significant. This could be the case in our regression results, since sample sizes are relatively large. We did test the robustness of our results on a smaller sample, using the period between January 10 to February 8 (8523 observations), obtaining similar results for the coefficients and p-values. Results of this experiment are presented in Table 6.7 of Chapter 6.

4.2 Forecasting price change

In the previous section, we showed that selected features in single linear regression have significant impact on the price change, while the R^2 of regression model is low when the direction of the price change is not taken into account. This suggests that the regression model does not show a good forecasting performance. However, when the data set is split

Table 4.5: OLS Regression Results for positive price changes

Dep. Variable:	Y	**R-squared (uncentered):**	0.582
Model:	OLS	**Adj. R-squared (uncentered):**	0.582
Method:	Least Squares	**F-statistic:**	1543.
Date:	Fri, 17 Dec 2021	**Prob (F-statistic):**	0.00
Time:	20:37:50	**Log-Likelihood:**	31228.
No. Observations:	6648	**AIC:**	-6.244e+04
Df Residuals:	6642	**BIC:**	-6.240e+04
Df Model:	6		

	coef*	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0004	7.57e-05	4.693	0.000	0.000	0.001
θ_3	5.803e-06	6.72e-07	8.640	0.000	4.49e-06	7.12e-06
θ_4	-0.0005	7e-05	-6.538	0.000	-0.001	-0.000
θ_5	0.0248	0.009	2.785	0.005	0.007	0.042
θ_6	0.0084	0.000	19.719	0.000	0.008	0.009
θ_7	0.5753	0.014	41.121	0.000	0.548	0.603

* Coefficient θ_2 does not appear as the corresponding variable was removed from this regression since it was not statistically significant.

into positive and negative price changes, the explanatory power of the model improves significantly. In this section, we investigate the model more closely to determine whether it is possible to improve its prediction performance.

In order to do so, the collected data has been split into training and test data sets. First, the model is trained on in-sample data (75% of collected data) and then its forecasting performance is evaluated on the out-of-sample set (remaining 25% of collected data). In the following sections we focus on 5-minute window and 10 look-back periods.

4.2.1 Breaking the regression model

As mentioned in previous section, we use two separate regressions for positive and negative price change values to improve the forecasting precision of model. We start by dividing the training set into positive and negative price changes. Table 4.5 displays the OLS regression result for positive price changes and Table 4.6 shows the regression results for negative price changes over training data set.

Table 4.6: OLS Regression Results for negative price changes

Dep. Variable:	Y	**R-squared (uncentered):**	0.563
Model:	OLS	**Adj. R-squared (uncentered):**	0.563
Method:	Least Squares	**F-statistic:**	1169.
Date:	Fri, 17 Dec 2021	**Prob (F-statistic):**	0.00
Time:	20:53:15	**Log-Likelihood:**	29649.
No. Observations:	6355	**AIC:**	-5.928e+04
Df Residuals:	6348	**BIC:**	-5.924e+04
Df Model:	7		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0005	9.01e-05	5.415	0.000	0.000	0.001
θ_2	6.8377	0.820	8.336	0.000	5.230	8.446
θ_3	1.01e-05	7.72e-07	13.085	0.000	8.59e-06	1.16e-05
θ_4	-0.0002	8.01e-05	-2.428	0.015	-0.000	-3.75e-05
θ_5	0.0613	0.011	5.828	0.000	0.041	0.082
θ_6	-0.0045	0.000	-9.684	0.000	-0.005	-0.004
θ_7	-0.4925	0.017	-29.762	0.000	-0.525	-0.460

As expected, the result of regressions over the training data sets is promising. However, another problem arises, which is predicting which regressing model that data point lies into, i.e. whether the price likely to increase or decrease.

4.2.2 Price change direction classification

In this section, we propose a method to predict price change direction. By doing so, one will be able to choose which regression model should be used to predict the price change.

This task lies into binary classification problems, which evaluate the probability of various possible outcomes for an event. In our case, the event is an increase or a decrease in the mid-price; as a result we should develop a classifier model to predict the direction of change in mid-price. Various methods exist to address classification problem, including logistic regression, support vector machines (SVM), decision trees, and random forests.

In order to compare the performance of these various approaches for our classification problem, we run them on our training data set. The results for classification methods'

metrics can be found in Table 4.7.

Table 4.7: Classification metrics for each method used.

method	**Accuracy**	**Precision**	**Recall**	**F1**
Logistic regression	61.61%	60.64%	65.80%	63.12%
SVM	62.23%	61.9%	63.3%	62.59%
Decision tree	59.07%	58.75%	60.48%	59.6%
Random forest	60.32%	60.6%	58.64%	59.6%

Accuracy is the percentage of correctly predicted observation; Precision is the ratio of correctly predicted positive observations to the total predicted positive observations; Recall is the ratio of correctly predicted positive observations to all the observations in actual positive class; F1 is the weighted average of Precision and Recall

As these results table suggest, SVM and logistic regression have a slightly better performance compared to the other two models, and since the recall score of logistic regression is better than that of SVM, we focus on this model going forward.

Table 4.8 displays how the logistic regression model is doing in each of the positive and negative classes.

Table 4.8: Confusion matrix for logistic regression

		Predicted labels		
		Negative	Positive	Total
True labels	Negative	1247	924	2171
	Positive	740	1424	2164
	Total	1987	2348	4335

Note that the default threshold for the logistic regression model is 0.5, but this discrimination threshold can be optimized to improve the classifier accuracy. Figure 4.1 displays the receiver operating characteristic (ROC) curve for the logistic regression model. This plot illustrates the diagnostic ability of the logistic regression model as a function of its threshold. The closer is the curve to the top left corner, the better the classifier is working. Optimizing this plot, we found that the value 0.500286 is the best threshold, which is very close to the default setting.

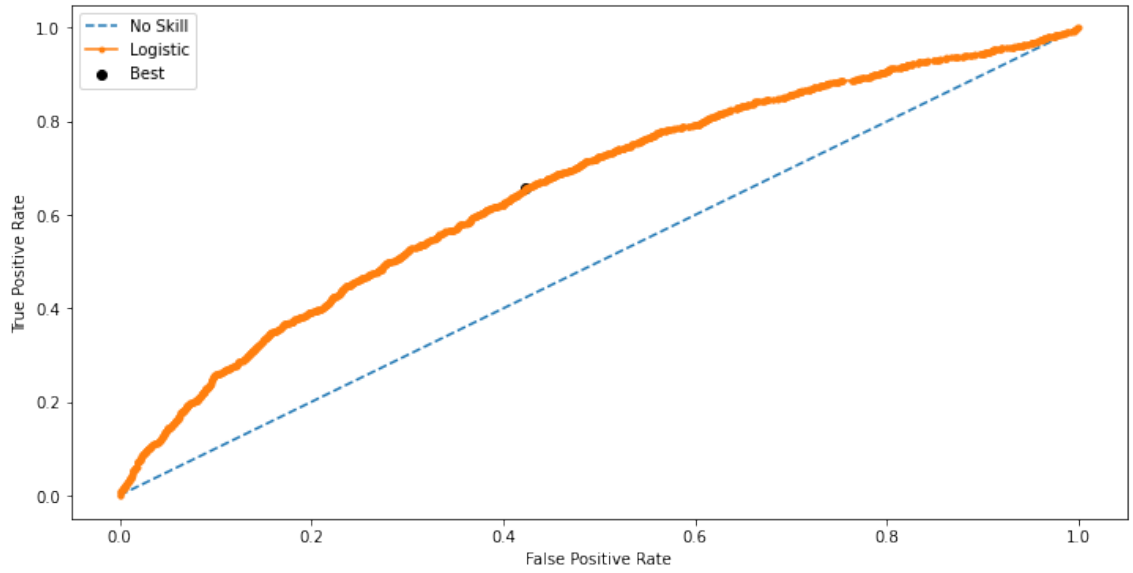


Figure 4.1: ROC curve for logistic regression classifier

4.2.3 Forecasting model

We now introduce two 2-step forecasting algorithms, using the results of the positive-negative linear regressions and the logistic regression model:

1. **Weighting algorithm:** The logistic classifier gives a probability for lying in each regime; these probabilities are multiplied by the price change respectively forecasted by the positive/negative regressions. These values are added up to yield the predicted price change for the next interval. Figure 4.2 illustrates the process details for the Weighted 2-step model.

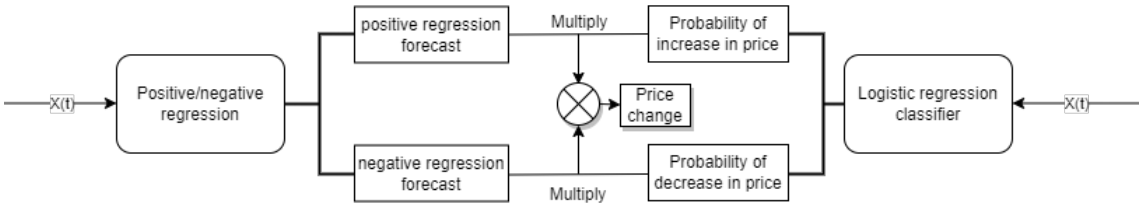


Figure 4.2: Flowchart of weighting algorithm

2. **Selection algorithm:**If the logistic regression predicts an increase in price, then the price change forecast is obtained from the output of the positive regression model; in the other case, the forecast for price change is the output of the negative regression model. Figure 4.3 displays the working of this algorithm.

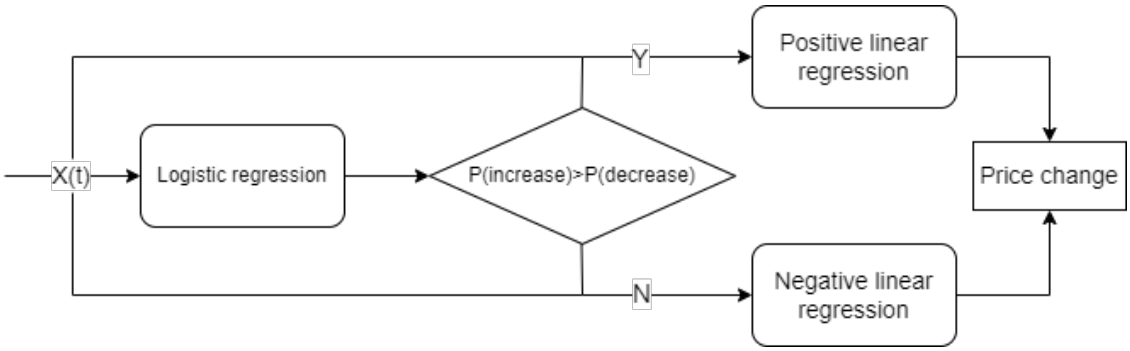


Figure 4.3: Flowchart of selection algorithm

4.2.4 Comparing forecasting models

So far three different models have been introduced to forecast the price change value: a single regression for all price changes, and two 2-step models to predict the direction of change and then its value. In this section we compare the forecasting performance of these three models.

Table 4.9 reports the OLS regression result for the single step model on the training set.

We now compute the mean absolute error (MAE) on the out-of-sample data for each of the three models; results are reported in Figure 4.4).

These results suggest that the weighting algorithm's performance is very close to that of the single regression model, while the selection algorithm is performing poorly compared to the other two.

This finding is counter-intuitive, given that both the logistic regression and the positive/negative linear regression models had relatively good prediction performance. How-

Table 4.9: Single step OLS regression results on the training data set

Dep. Variable:	Y	**R-squared (uncentered):**	0.108
Model:	OLS	**Adj. R-squared (uncentered):**	0.108
Method:	Least Squares	**F-statistic:**	224.9
Date:	Sun, 19 Dec 2021	**Prob (F-statistic):**	8.01e-317
Time:	02:46:04	**Log-Likelihood:**	56088.
No. Observations:	13003	**AIC:**	-1.122e+05
Df Residuals:	12996	**BIC:**	-1.121e+05
Df Model:	7		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0010	8.35e-05	12.327	0.000	0.001	0.001
θ_2	5.6100	0.808	6.946	0.000	4.027	7.193
θ_3	1.694e-05	7.17e-07	23.628	0.000	1.55e-05	1.83e-05
θ_4	-0.0006	7.48e-05	-7.975	0.000	-0.001	-0.000
θ_5	0.1093	0.010	11.155	0.000	0.090	0.129
θ_6	0.0022	0.000	4.990	0.000	0.001	0.003
θ_7	0.1306	0.015	8.463	0.000	0.100	0.161

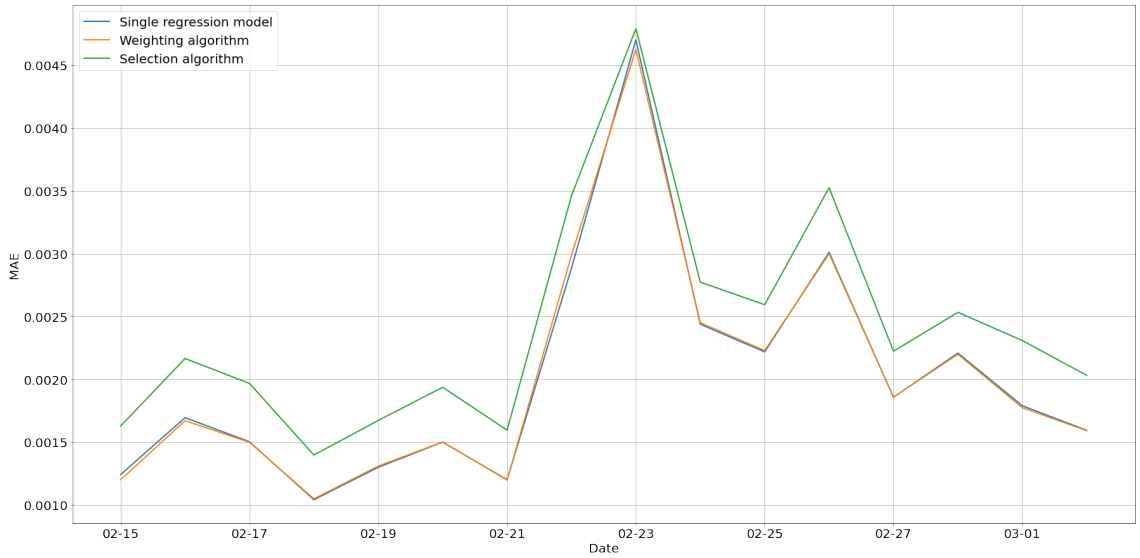


Figure 4.4: MAE of out-of-sample data set for three forecasting models

ever, this poor performance can be explained by looking at the MAE corresponding to each element in the confusion matrix.

Examination of Table 4.10 shows that the 2-step selection algorithm performance in the "true positive" and "true negative" data points is significantly better than that of the

Table 4.10: MAE corresponding to different elements of the confusion matrix

labels	**single regression**	**2-step selection algorithm**
True positive	0.00188	0.00142
False Positive	0.00198	0.00369
True Negative	0.00251	0.00154
False Negative	0.002	0.00438

single regression. However, when the classification is wrong and data is fed to the wrong regression model, the result is very far from the real value. As a result, the total MAE of the 2-step selection algorithm is worse than that of the single regression.

4.2.5 Graphing mid-price forecasts

We now examine the mid-price forecasts for different models on the out-of-sample data set. Since the single regression model and the 2-step weighted model have similar performances, we only report the forecasts of the single regression and the 2-step selection algorithm. Figure 4.5 displays the results on the test data set, but obviously this graph is not clear, so we are going to zoom in on three different periods of the test data set.

Figure 4.5: Predictions of mid-price over out-of-sample data set for different models

Figure 4.6 illustrates the prediction for mid-price over 3 different periods using differ-

ent methods, along with the real data from the market over the same period.

It should be noted the plots in Figure 4.6 are representing the prediction for mid-price for each interval, and the direction displayed in the plot is not the same as the direction predicted by each model, since the plot is connecting predicted values and the first point of each line connecting two intervals is not correct.

To compare the predictions of each model with real values, the difference between each model's forecast and the real market values are displayed in Figure 4.7. It can be seen in Figure 4.7 that the predictions made by the single regression model are less volatile then those made by the 2-step algorithms.

Finally, the prediction in mid-price change and real mid-price change at each interval are displayed in Figure 4.8. The findings illustrated by these graphs suggest the following:

1. The volatility of the single regression predictions are smaller than its true value, while the 2-step algorithms forecasts are more volatile and at some points very close to real values.

2. Figure 4.8 reveals an interesting point, that is, the three models, and specifically the single regression, do a good job in correctly forecasting the direction of a price change.

3. The 2-step selection algorithm does a better job in predicting large price change values, while at some points the prediction is very off since the logistic regression binomial classifier did not correctly categorize the direction of move.

4.2.6 A simple trading strategy

In previous sections, we provided a classification model that could predict the direction of a price change with acceptable accuracy. However, acceptable accuracy may not be enough to devise a trading strategy. The reason being that most trading exchanges charge traders with commission fees that could range from 0.1% to 2.5%. So, upon using classifier method to devise a direction trading strategy, even though the model could predict

the direction correctly, it is likely that the appreciation in the price is not enough to cover the commission cost. Having the positive/negative regressions with high forecasting precision as one of the steps in the model is beneficial since it could predict whether the price increase/decrease value would be enough to cover the commission fees, so the trader can decide about entering into a long/short position.

In this section, we propose a simple algorithmic trading strategy using the models developed in the previous sections. This strategy consists of the following steps:

1. **Classifying price direction move:** In the first step, the algorithm uses the logistic regression model to predict whether the price is going to increase or decrease. The confusion matrix for the proposed logistic regression is shown in Table 4.8. Recall that, if the classifier predicts a wrong direction, the positive/negative regression model might poorly forecast the value of the price change. Consequently we should modify the classifier in a way to decrease the probability of this poor performance, which could lead to a bad trading decision. In order to do so, we should modify the classifier to make it more conservative. By default, the threshold for logistic regression is set at 0.5, which means that the classifier will predict an increase (decrease) if its probability is even slightly higher than 50%. As a result, if the increase/decrease probability predicted by logistic regression is close to 50%, it is very likely that the model will guess the direction incorrectly. To improve the classifier performance at those critical points, we make the following modifications:

    - If the dominant predicted probability (either increase or decrease) is less than 60%, the algorithm will not enter into a trade.
    - If the probability is more than 60%, the trader will do a trade in the predicted direction.

    This steps will make the classifier more conservative. Table 4.11 shows the confusion matrix using this new strategy.

Table 4.11: Confusion matrix for conservative logistic regression

		Predicted labels		
		Negative	Positive	Total
True labels	Negative	221	111	332
	Positive	122	314	436
	Total	343	425	768

As expected, less predictions are made, compared to the original logistic regression model (768 vs. 4335). Moreover, the classification accuracy is improved over each class (from 62.75% to 64.43% in the negative class and from 60.64% to 73.88% in the positive class).

2. **Forecasting price change:** After classifying with the conservative logistic regression, the price change is forecast using the relevant linear regression model. If the predicted price change is greater than the commission fee, the trader will enter into a long/short position.

3. **Closing the position:** The price should monitored during each trading interval. One of below actions should be taken based on market condition:

   - If the price reaches the predicted level, the trader should close the position.
   - If the price declines to entering price level, trader should exit the trade.
   - If none of the above situations happens, the trader should hold the asset to the end of the interval and repeat the whole process again to evaluate the situation and decide accordingly.

This trading strategy was evaluated over the test data set, with 0.2% commission fee and no short selling assumptions. Figure 4.9 displays the gain of proposed trading strategy for a 1$ investment over the test period, compared to Bitcoin value.

4.3 Interpretation and conclusion

In this chapter, we introduced a novel price impact function by including new explanatory variables, namely two features extracted from the order book, "volume imbalance" and "price dispersion". Our results show that these variables are significant in the proposed price impact function. In the following sections, we will interpret our results and compare them with results from similar research in the literature.

4.3.1 Zhou universal price impact functions vs. single regression model

There is an important literature pertaining to the estimation of price-impact functions, and various forms of price-impact have been proposed by researchers. We will apply one of the most popular model to our data set and investigate whether our single regression model is able to show better performance.

The concave price-impact function introduced in Zhou [2012] is one of the most famous and cited models in the literature. This model suggests that price-impact is a concave function of trades' volume, according to the following equation:

$$\Delta P_t = A\,\omega^{\alpha}\,|\overline{\Delta P}_{i,t}| + \eta_t,$$

where ω represents the volume of buy or sell initiated trades, α is the concavity order, which is empirically measured for many stocks and found to be very close to 0.66, $\overline{\Delta P}_{i,t}$ is the average price change in the last i periods, and A is the regression coefficient, positive for buy-initiated trades and negative for sell-initiated trades. Tables 4.12 and 4.13 display the polynomial regression results using Zhou universal price impact model on our data set for $i = 10$ periods.

As these results suggest, the value found for coefficient A is acceptable referring to the assumptions of Zhou model. Note that the R^2 values are lower that those obtained using our single regression model.

Table 4.12: Zhou concave model regression for buy-initiated trades

Dep. Variable:	Y	**R-squared (uncentered):**	0.081
Model:	OLS	**Adj. R-squared (uncentered):**	0.081
Method:	Least Squares	**F-statistic:**	704.3
Date:	Thu, 23 Dec 2021	**Prob (F-statistic):**	8.51e-149
Time:	06:24:07	**Log-Likelihood:**	34786.
No. Observations:	7991	**AIC:**	-6.957e+04
Df Residuals:	7990	**BIC:**	-6.956e+04
Df Model:	1		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
A	9.483e-05	3.57e-06	26.538	0.000	8.78e-05	0.000

Table 4.13: Zhou concave model regression results for sell-initiated trades

Dep. Variable:	Y	**R-squared (uncentered):**	0.068
Model:	OLS	**Adj. R-squared (uncentered):**	0.068
Method:	Least Squares	**F-statistic:**	365.4
Date:	Thu, 23 Dec 2021	**Prob (F-statistic):**	1.13e-78
Time:	06:24:07	**Log-Likelihood:**	21126.
No. Observations:	5012	**AIC:**	-4.225e+04
Df Residuals:	5011	**BIC:**	-4.224e+04
Df Model:	1		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
A	-9.295e-05	4.86e-06	-19.115	0.000	-0.000	-8.34e-05

We now compare the performance of Zhou model with the single linear regression model we developed on the out-of-sample data set. Figure 4.10 illustrates the MAE for both models over the out-of-sample data set.

This plot clearly shows that the model proposed in this thesis performs a better job in predicting out-of-sample data, compared to the Zhou universal price impact function.

4.3.2 Comparing regression result with findings of the literature

As discussed in section 4.1.2, the signs of the coefficients in the linear regression model are justified by financial and economical intuitions. In this section, we focus on comparing these results with similar studies in the literature.

The impact of order volume on price has not been widely investigated in the literature. Hautsch and Huang [2012] show that small order quotes do not affect the market price, while aggressive orders can push market in the same direction. These results are supported by the sign of the coefficient for order imbalance measured in a regression model. We also find a positive coefficient for order imbalance, which is in line with the findings in Hautsch and Huang [2012].

Many researchers focus on the impact of trades' volume on price change. This feature is included in our regression model as V_t^{trades}. Lillo et al. [2003], Lim and Coggins* [2005], Almgren et al. [2005] and Wilinski et al. [2015] investigating on various stock data, show that the trade imbalance positively correlates with price change. We also find a positive sign for our trade imbalance feature.

The variable $\frac{v_t}{\overline{v}_t}$ was first introduced in the price impact functions by Lim and Coggins* [2005]. Their empirical investigation show that this feature is positively correlated with price change. This finding was later validated by Zhou [2012] on China stock market data. More recently, Pham et al. [2020] includes this feature in various price impact models and shows that it has a significant positive impact on price change. We also find a similar behaviour using our Bitcoin market data.

Another important feature in the study of price change prediction is asset volatility. This idea was first proposed by Torre and Ferrari [1998], then validated later by Almgren et al. [2005] empirically, which suggests that volatility is positively correlated with price change. This feature is used by Pham et al. [2020] for various asset groups in the Australia stock exchange market. Their findings show that the correlation of asset volatility and price change is not the same for buy-initiated and sell-initiated trades. Our investigation rather suggests that Bitcoin price change is positively correlated with its volatility.

Pham et al. [2020] include the ΔP_{t-1} feature in the price impact function and show that it positively correlates with price change. Our regression results also suggests that last interval price change has positive correlation with the price change of coming interval.

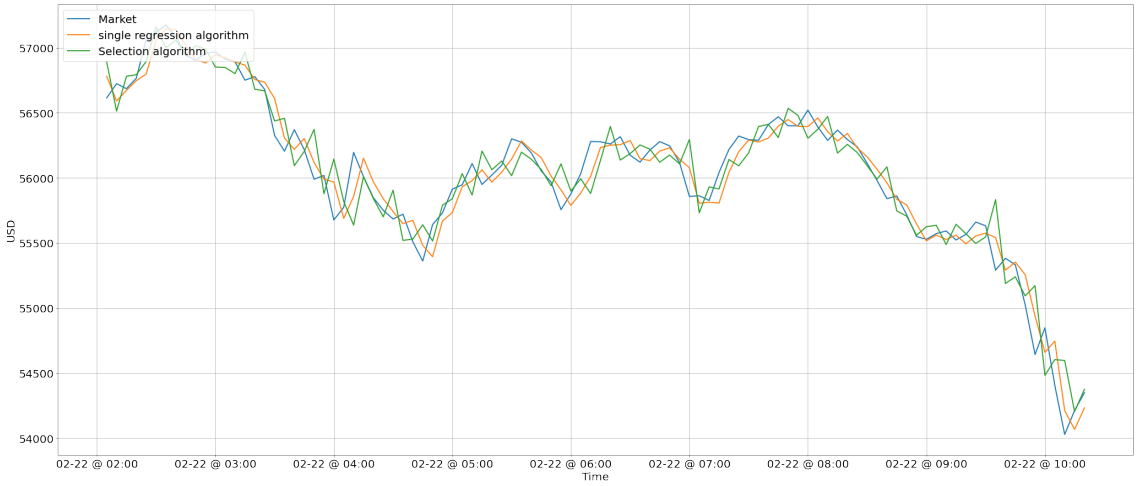
4.3.3 Logistic regression vs. other classifier methods

In this thesis, a logistic regression model is proposed to predict the price change direction. Many researchers focused on developing models to forecast this categorical variable. For instance, a recent study by Tsantekidis et al. [2020] uses similar order book features and proposes various deep learning models to predict price change direction. Table 4.14 provides a comparison of the performance of these models vs. the logistic regression classifier introduced in this thesis in predicting price move direction. We specifically focus on comparing our result with this paper's findings since it has used similar input variables.

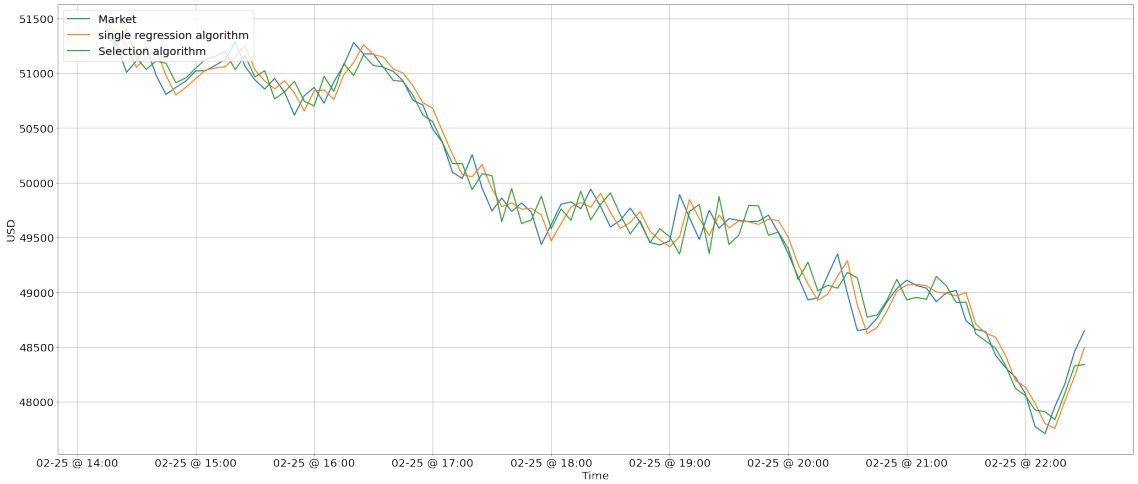
Table 4.14: Comparison of Tsantekidis et al. [2020] results and our logistic regression

Model	**Recall**	**Precision**	**F1**
SVM	0.33	0.46	0.30
MLP	0.34	0.35	0.09
CNN	0.53	0.46	0.46
LSTM	0.55	0.46	0.43
CNN-LSTM	0.55	0.46	0.47
Logistic regression	**0.61**	**0.66**	**0.63**

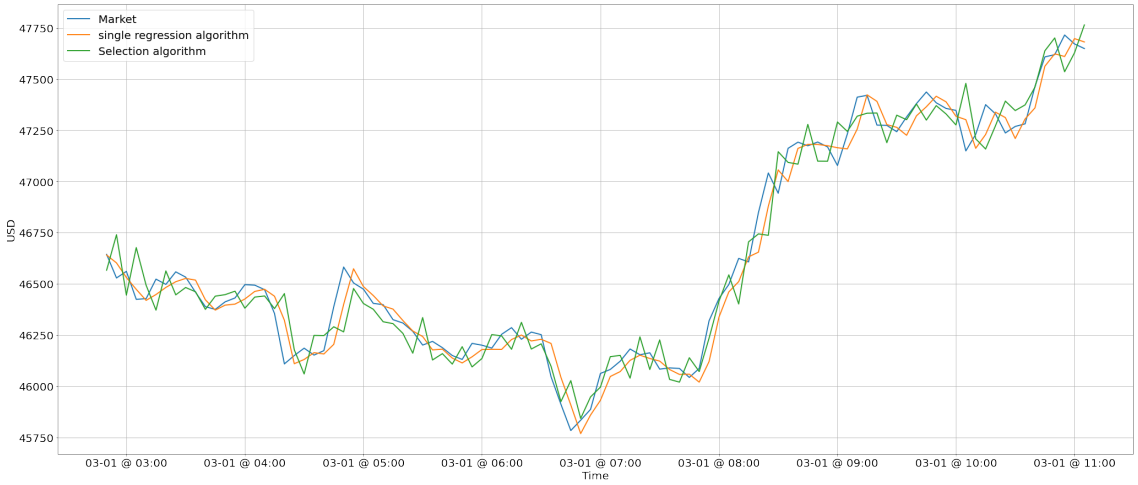
As Table 4.14 suggests, our logistic regression model outperforms Tsantekidis et al. [2020] models' metrics on Bitcoin market data. Since features extracted from order book are similar in both studies, we can conclude that including other features in the logistic regression model have improved the classifier performance.



(a) Period 1

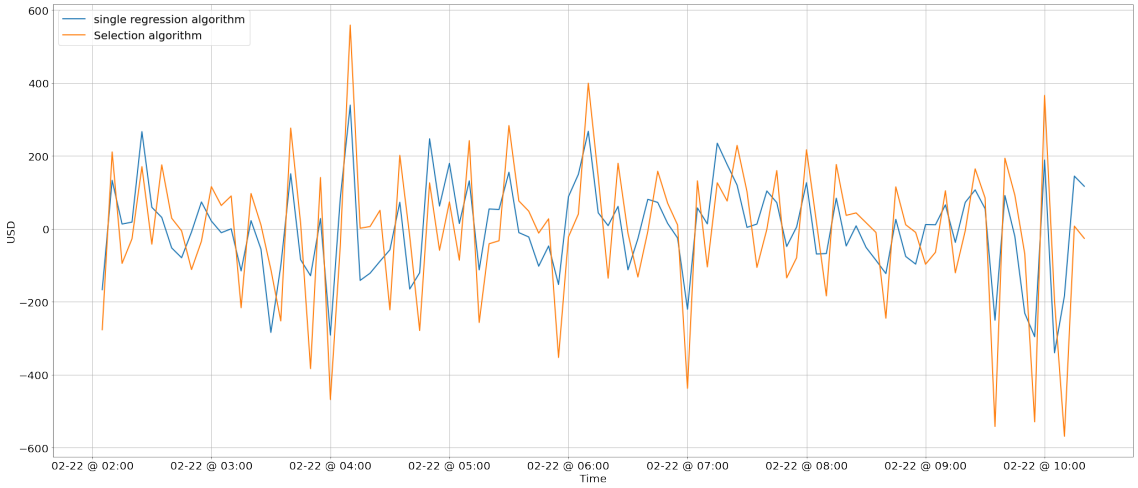


(b) Period 2

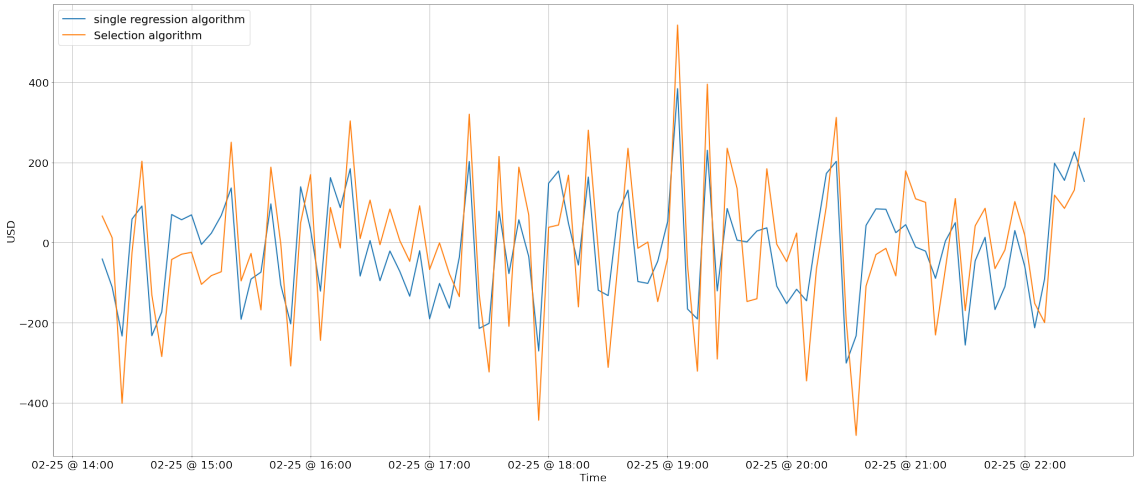


(c) Period 3

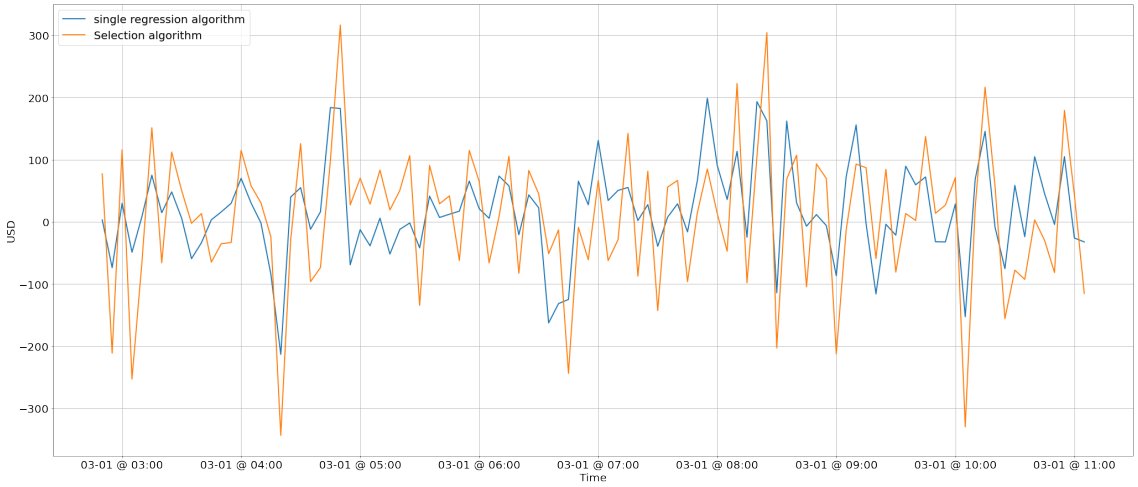
Figure 4.6: Mid-price forecast over 3 different periods for different models



(a) Period 1

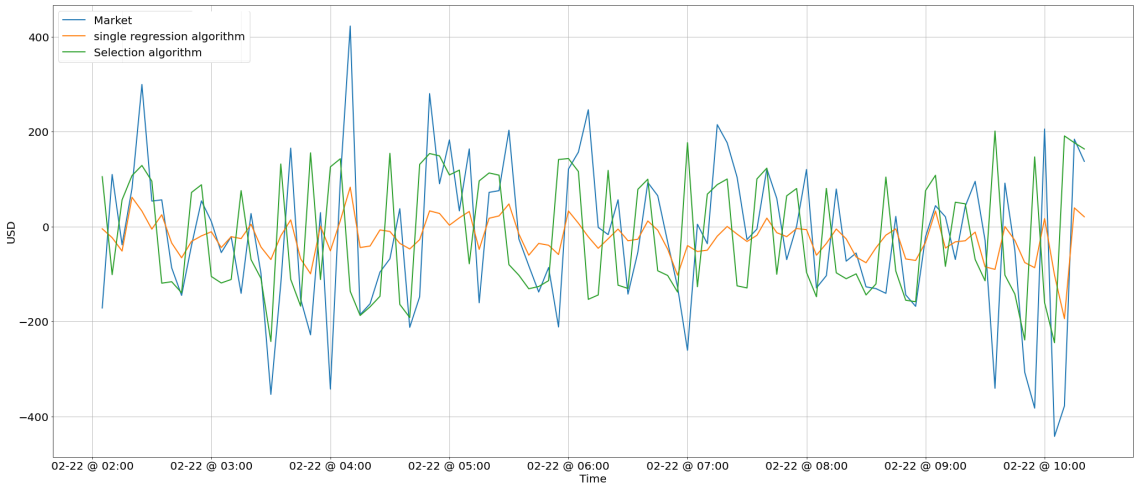


(b) Period 2

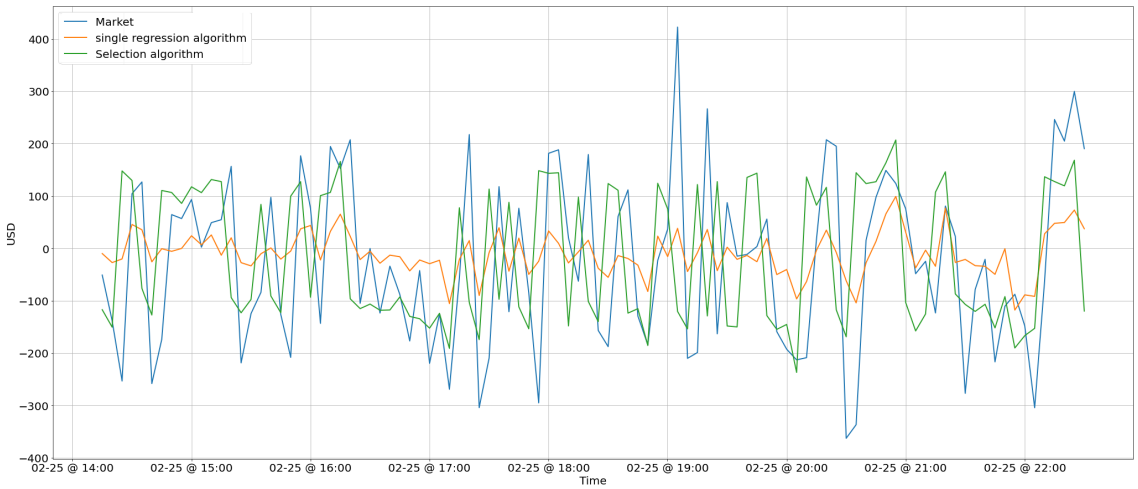


(c) Period 3

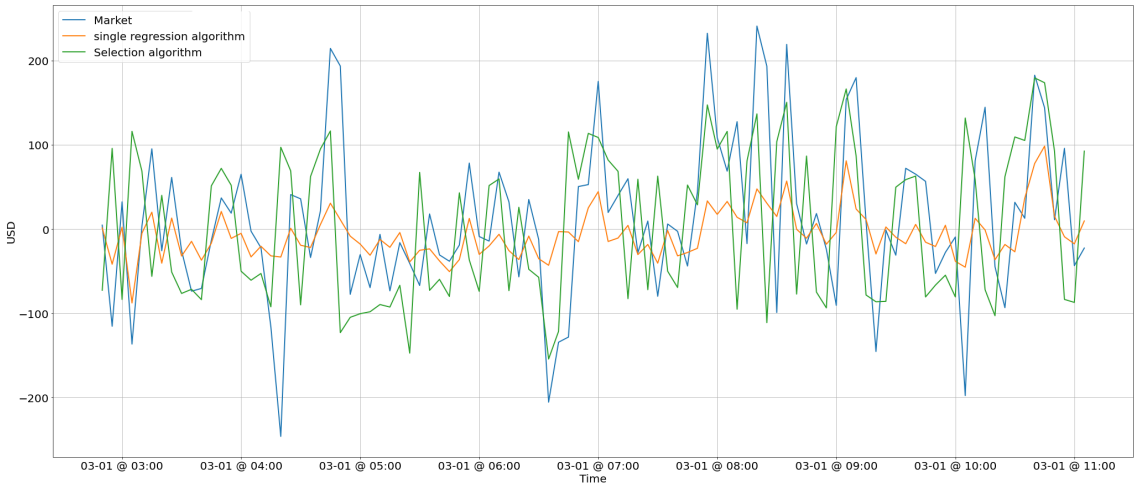
Figure 4.7: Difference of market mid-price and models' mid-price forecast



(a) Period 1



(b) Period 2



(c) Period 3

Figure 4.8: Mid-price change, real values and values forecasted by models

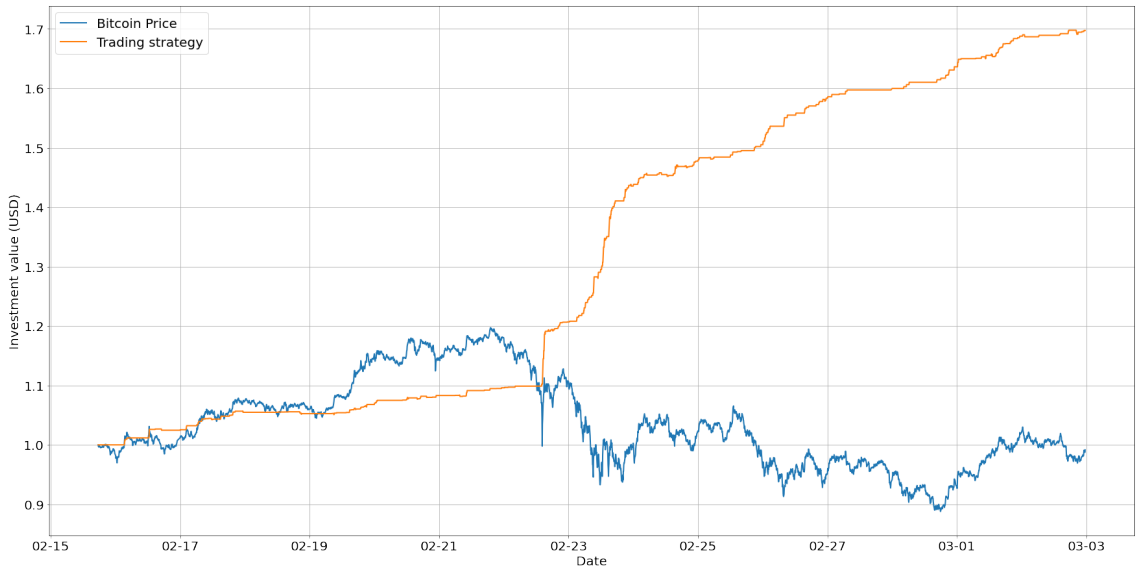


Figure 4.9: Trading strategy gain over the test period

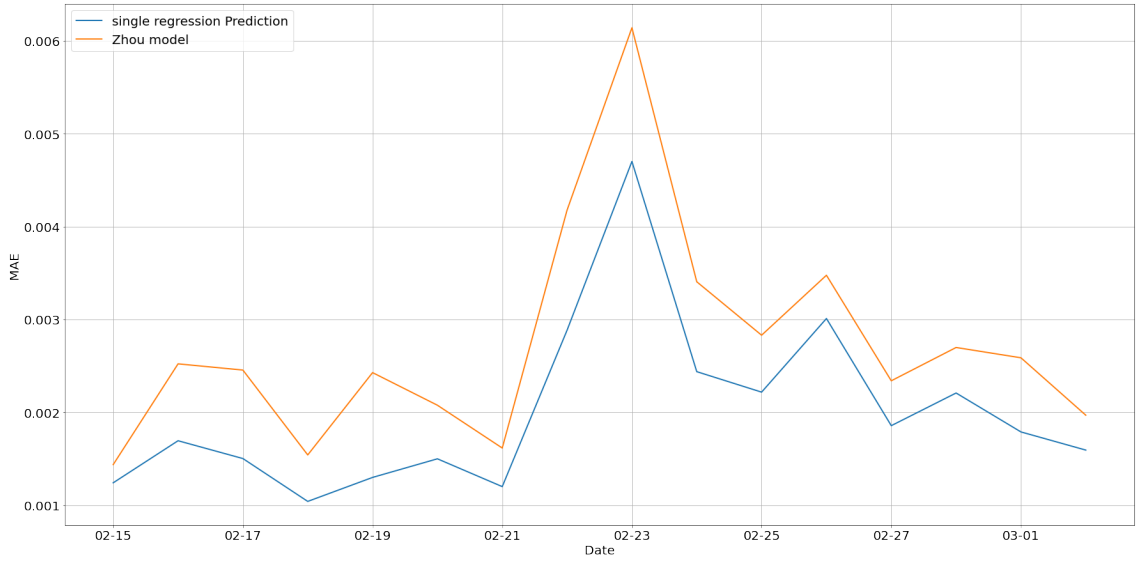


Figure 4.10: MAE comparison for Zhou and single regression model over out-of-sample data set

Chapter 5

Price Impact and Market Manipulation

With the recent expansions in financial markets and the increasing number of traders, manipulation has taken a new display. Forgers try to manipulate the stock market and provide false information in order to mislead the investors for their own benefits. As a result, controls and regulations have been posed by authorities to mitigate market manipulation activities.

In this chapter, we focus on market manipulation and its variations, and we discuss how our price-impact models such as the one presented in the previous chapter could be used in controlling market manipulation activities.

5.1 Market manipulation

The first part of this chapter is dedicated to the definition of market manipulation and its various forms.

5.1.1 What is market manipulation?

Market manipulation is the act of artificially raising or lowering the price of a security so that its price differs from its true value, or otherwise influencing the behavior of the market for personal gain. Manipulation is illegal in most cases, but it can be difficult for

regulators and other authorities to detect (https://www.investopedia.com/terms/m/manipulation.asp).

Manipulation also gets more difficult for the manipulator as the size and number of participants in a market increases. It is much easier to manipulate the share price of smaller companies because analysts and other market participants do not watch them as closely as the medium and large-cap firms.

The goal of market manipulating is to deceive other market participants in order to create a situation where assets are mispriced, so that the manipulator (who knows better) can then profit from the situation. The manipulator thus profits at the expense of other market participants, whom the manipulator has deceived. Because the manipulators themselves create (and subsequently reverse) the mispricing in the first place, market manipulation, unlike honest investing strategies, does not improve market efficiency or benefit society.

5.1.2 Manipulation forms

Market manipulation takes many forms, some of common methods are (see https://en.wikipedia.org/wiki/Market_manipulation):

- Churning: When a trader places both buy and sell orders at about the same price. The increase in activity is intended to attract additional investors, and increase the price.

- Pump and dump: A manipulative scheme that attempts to boost the price of a stock or security through fake recommendations. These recommendations are based on false, misleading, or greatly exaggerated statements. The perpetrators of a pump-and-dump scheme already have an established position in the company's stock and will sell their positions after the hype has led to a higher share price.

- Runs: When a group of traders create activity or rumours in order to drive the price of a security up.

- Wash trade: In such case, the manipulator takes both the buy and the sell side of a trade, often using a third party as a proxy to trade on his or her behalf, for the purpose of generating activity and increasing the price. This is more involved than churning because the orders are actually fulfilled.

- Bear raid: This consists of an attempt to push the price of a stock down by heavy selling or short selling.

- Price-fixing :A very simple type of fraud where the principals who publish a price or indicator conspire to set it falsely and benefit their own interests. The Libor scandal for example, involved bankers setting the Libor rate to benefit their trader's portfolios or to make certain entities appear more creditworthy than they were.

- Spoofing (Layering): Spoofing involves placing bids to buy or offers to sell assets, and canceling the bids or offers prior to the deal's execution. The practice intends to create a false picture of demand or false pessimism in the market.

Since manipulators spoof the market by posting fake quotes in the orderbook, spoofing is closely related to our study and on the relation between order book and price movements. Consequently, in the following section we specifically focus on this methods and how we can connect it to our findings.

5.2 Spoofing or layering

5.2.1 What is Spoofing?

Spoofing, or interchangeably layering, refers to posting a relatively large number of limit orders on one side of the limit order book to make other market participants believe that there is pressure to sell (limit orders are posted on the offer side of the book) or to buy (limit orders are posted on the bid side of the book) the asset (see `https://en.wikipedia.org/wiki/Spoofing_(finance)`).

The Dodd-Frank Act describes spoofing as 'bidding or offering with the intent to cancel the bid or offer before execution (see Reform and Act [2010]). Spoofing may cause prices to change because the market interprets the one-sided pressure in the limit order book as a shift in the balance of the number of investors who wish to purchase or sell the asset, which causes prices to increase (more buyers than sellers) or prices to decline (more sellers than buyers). Spoofers bid or offer with intent to cancel before the orders are filled. The flurry of activity around the buy or sell orders is intended to attract other traders to induce a particular market reaction (`https://en.wikipedia.org/wiki/Spoofing_(finance)`).

Layering is a similar strategy, which consists of posting several large limit orders at different prices on one side of the book. The goal is to move the price because other market participants interpret the one-sided pressure in the LOB as a signal of a price move and trade in anticipation of expected change in price.

Spoofing and layering are very difficult to detect, since these activities are hidden behind the huge number of updates in the order book and use sophisticated automated algorithms to avoid detection. However, regulators have been able to detect and prosecute many spoofers for market abuse and price manipulation.

5.2.2 Review of related literature

While there is an extensive theoretical literature, there is comparatively little empirical research regarding manipulative stock trading. . Allen and Gale [1992] propose a simple model for trade-based stock manipulation and show that this method is profitable.

Lee et al. [2013] define spoof order as a quote at least 6 ticks away from the market price (in our study we focus on the top 10 quotes of each market side, so any spoofing act at top 10 quotes' level could be monitored) with a volume at least twice as large as the average volume of the orders posted on the previous day. The authors use a proprietary data set with account information from the Korea Exchange, and show empirically that the spoof orders create imbalance in the order book, which moves the price. They also

show empirically that spoofing achieves substantial extra profits and that spoofing tends to target stocks with: higher volatility of returns, lower market capitalization and lower price level.

Wang [2015] use data from the Taiwan Futures Exchange and show that spoofers manipulate the order book in the stocks that have high volumes of trading, high volatility and high prices. Their findings also suggest that spoofing increase trading volume, asset price volatility and bid-ask spread.

Recently, Cartea et al. [2020] derives an optimal trading strategy for spoofers, which trades off between the benefits from spoofing and the potential fine the investor may receive from the financial authorities. They show that spoofing deviates the price of the asset from its fundamental value. This deviation is highest when market participants believe the information conveyed by the order book and the fine for spoofing is zero.

On the other hand, there exist studies that use machine learning methods to detect manipulation activities in markets. For instance, Cao et al. [2014] use k_Nearest Neighbour (KNN) and One Class Support Vector Machine (OCSVM) on order book data to detect price manipulation.

5.3 Detecting spoofing

In this section, we start by defining a method to detect potential spoofing activities ex-post. We then try to apply it on a sample data point in our data set. Finally, we propose a method to detect spoofing ex-ante

5.3.1 Detecting spoofing ex-post

We start by defining two imbalance variables at a given time interval t. The "Order imbalance" variable is defined by

$$\rho_t = \frac{\sum_{\tau=1}^{T}\sum_{i=1}^{N} v_\tau^{b,i} - \sum_{\tau=1}^{T}\sum_{i=1}^{N} v_\tau^{a,i}}{\sum_{\tau=1}^{T}\sum_{i=1}^{N} v_\tau^{b,i} + \sum_{\tau=1}^{T}\sum_{i=1}^{N} v_\tau^{a,i}} \quad \in (-1,1), \tag{5.1}$$

and the "Trade imbalance" variable is defined by

$$\lambda_t = \frac{\sum v_t^{buy} - \sum v_t^{sell}}{\sum v_t^{buy} + \sum v_t^{sell}} \quad \in (-1,1), \tag{5.2}$$

at time interval t.

The intuition behind defining these variables is more clear assuming a healthy and non-spoofing order book. If the order imbalance variable is high and close to 1 (resp. low and close to -1), we expect the trade imbalance for next 1 or 2 intervals to increase (resp. decrease), since the traders are more interested in buying (resp. selling), and we expect the mid-price to increase (resp. decrease). In the same scenario, if the trade imbalance is negative, then one may suspect a spoofing activity since the trade activities in the market are against the direction of the order book and price change.

We use Figure 5.1 to illustrate this intuition:

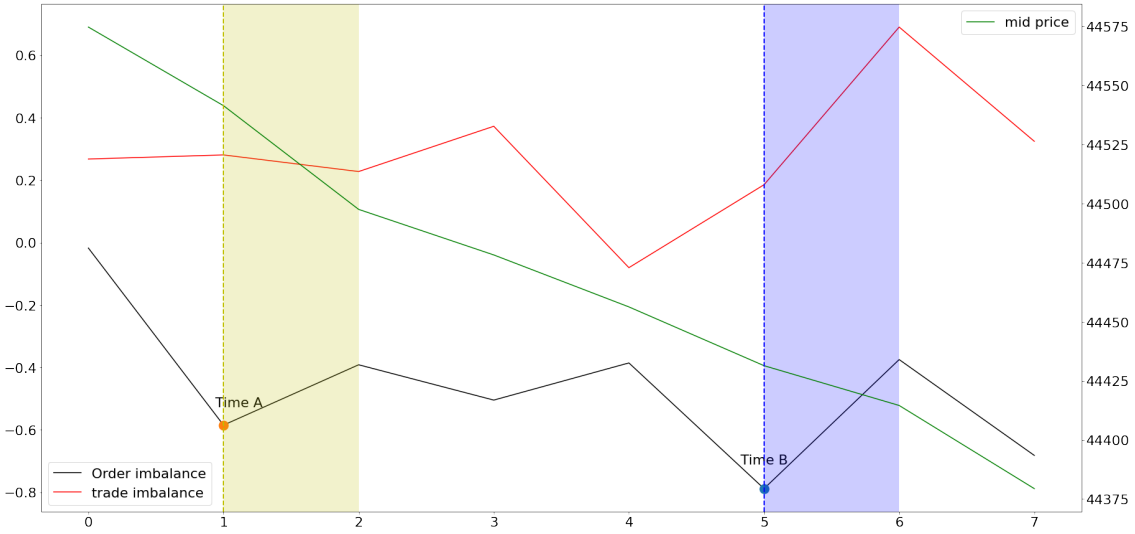


Figure 5.1: Ex-post method for detecting spoofing

At time labeled as "Time A" in Figure 5.1, the order imbalance is negative, so traders are more willing to sell and we expect the price and trade imbalance to decrease, which

has happened in the coming interval (shaded in yellow). While at "Time B", where the order imbalance is very low, which suggests that the bid side volume of the order book is much stronger and traders are willing to sell, the price decreases as expected, whereas the trade imbalance suddenly jumps, which shows that the trade imbalance is more toward buy-initiated trades during the blue-shaded interval, which is against the expectation. As a result, one can suspect interval B to host spoofing activities in it.

Under this assumption, the trades and quotes during the interval following the suspicious point should be monitored. If one observes a sudden cancellation in the order book and large trades fulfilled in the opposite direction, this suggests that there is an intention to inflate one side of the order book to manipulate the price and trade in the opposite direction, and trading accounts causing this chain of events are potential spoofers.

5.3.2 Applying ex-post method on a sample point in the data set

In this section, we apply the "ex-post" method on one sample data point. Figure 5.2 displays the imbalance variables and Bitcoin mid-price for 1-minute windows on February 28 during the time period from 14:40 to 15:15.

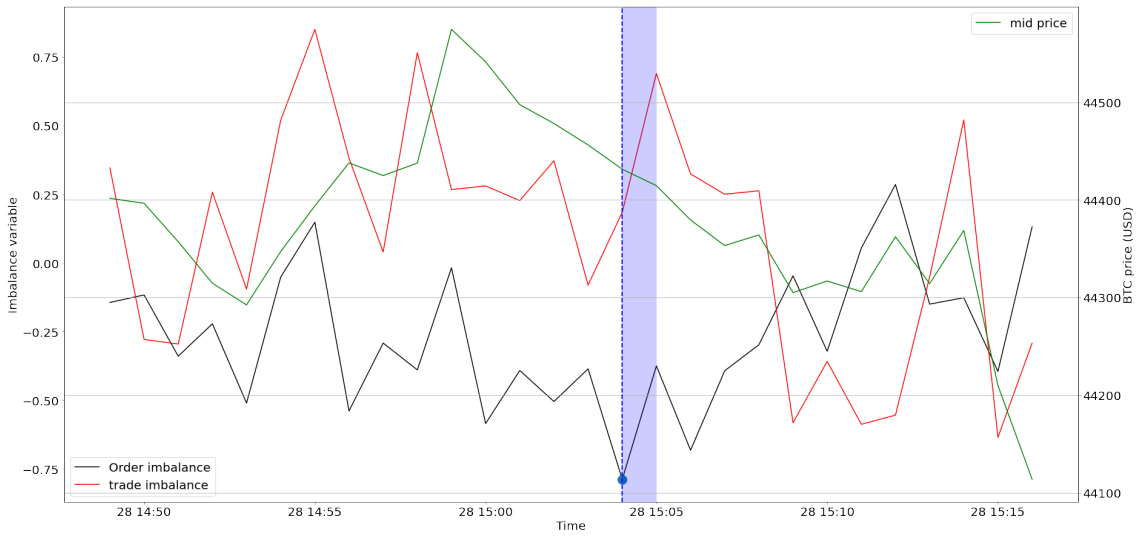


Figure 5.2: Ex-post method for a sample data point

Based on the method explained in the previous section, the shaded part of Figure 5.2 at

15:04 is suspicious for spoofing. As a result, we will investigate the trades in the next one-minute interval. The reason we use short time interval and graphed this plot for 1-minute windows is that spoofing usually takes place very quickly by algorithmic trading agents, considering the fact that there are other algorithmic trading participants who implement trading strategies that take into account the smallest updates in the order book.

We examine the buy-initiated trades fulfilled in the next interval (we only consider buy-initiated trades since the order imbalance is very low while there are buying activities, so the potential spoofer is buying the asset), from 15:04 to 15:05. Figure 5.3 displays the volume distribution of buy-initiated trades during that time interval.

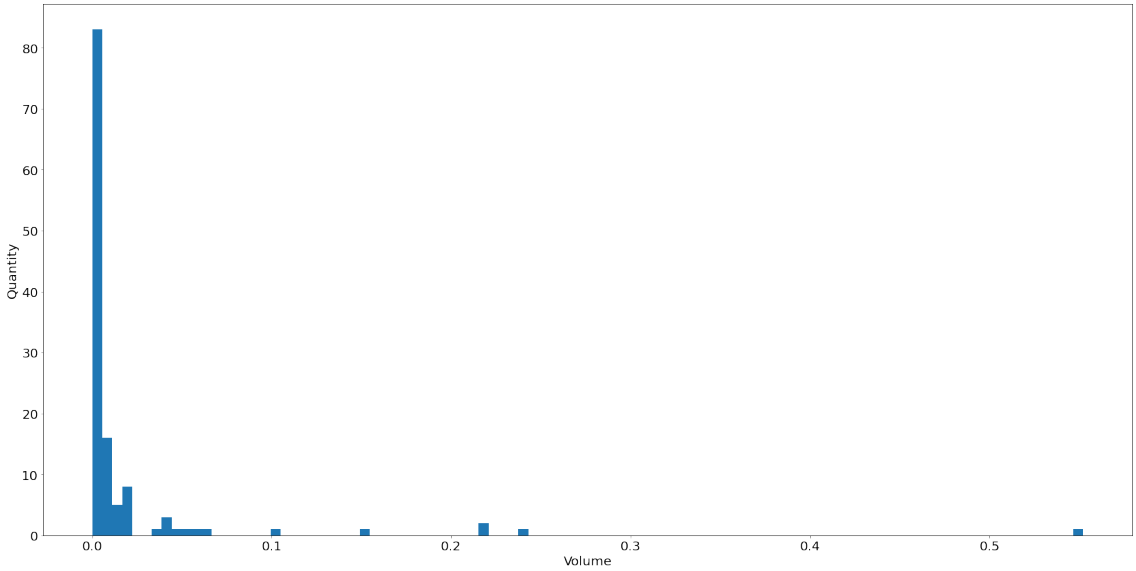


Figure 5.3: Volume distribution of buy-initiated trades from 15:04 to 15:05

As Figure 5.3 suggests, there is an outlier trade, on which we are going to focus. Table 5.1 represents the detailed attributes of this trade, which shows that it has been fulfilled at 15:04:08. As a result, we are going to examine the order book data just before this instant.

Figure 5.4 displays the updates in the order book prior to the outlier trade. The green lines display bid quotes and red lines display ask quotes in the order book.

Figure 5.4 shows that, at 15:03:43, the order imbalance is toward bid quotes, whereas at 15:03:53 the volume of ask quotes increases and the order imbalance decreases to a negative value, showing more supply in the order book, which results in the fall in mid

Table 5.1: Detail attributes of outlier trade

attribute	value
time	2021-02-28 15:04:08
size	0.552009
price	44442.8
direction	buy

price at 15:04:03. This process could be a good justification for spoofer intention. In other words, at 15:03:43 the best ask price is high and the spoofer is looking for a lower price, so the volume on the ask side is inflated to decrease the price; as soon as the price reaches the desired level, a buy trade is fulfilled by the spoofer (at 15:04:03).

Note that it is possible that this particular trade is due to spoofing (or layering), but the following modifications are necessary to make a more confident conclusion:

- The granularity of the order book data is vital in this model, since most spoofing activities happen in very short periods, so that instantaneous monitoring is required from regulators.

- It is necessary to know the account information of traders, since one needs to know which trader has posted which quotes so if they are cancelled, their following trading activities can be monitored by financial authorities.

Because of the limitations in collecting order book data and trading account information, applying the above conditions was not possible in this thesis.

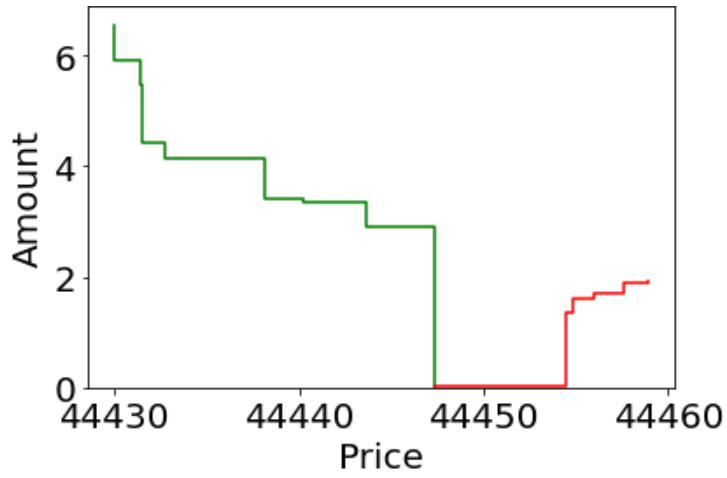
5.3.3 Detecting spoofing ex-ante

In the previous chapter, we showed that market information, including quotes price and volume, has significant impact on price change. We also showed that the forecasting error noticeably shrinks if we split the regression data according to the price change direction. In this section, we propose a method to avoid spoofing ex-ante, according to the following observations:

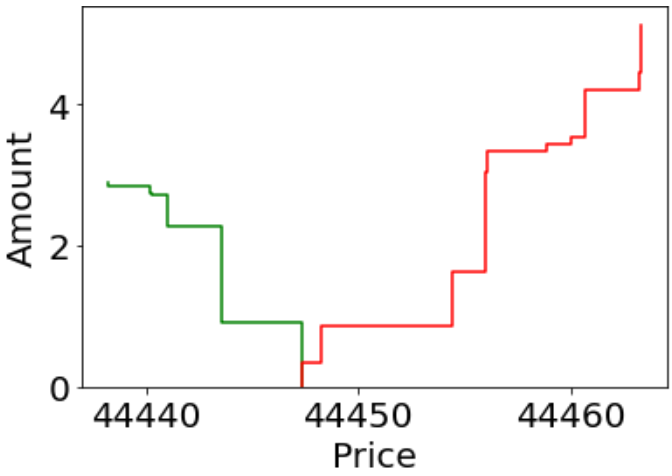
- Since spoofers manipulate and trade in shorter periods, regression should be performed over shorter time windows.
- The regulators should define a limit for price change during each interval.
- All the variables in the regression except for quotes' volume should be collected from the market information.
- Every coming order should be checked using the regression models and if the predicted price change is beyond the defined limit, that quote should be declined or activities by that trading account should be monitored.

Combining proposed methods give the regulator the ability to actively monitor order book updates and reduce the risk of manipulation in the order book.

(a) Order book visualization at 15:03:43



(b) Order book visualization at 15:03:53



(c) Order book visualization at 15:04:03

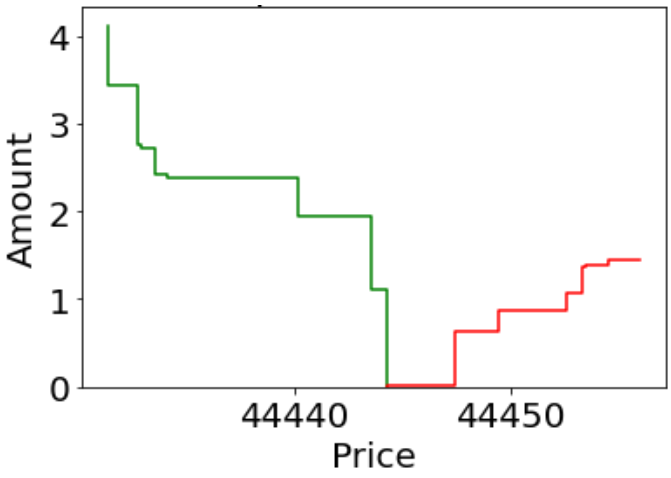


Figure 5.4: Order book updates before the outlier trade

Chapter 6

Market Sentiment Effect in Price Impact Function

An interesting aspect of Bitcoin price volatility is the impact of social media sentiment on sudden price jumps and drops. This fact has attracted attention to this subject and many researchers investigated the relation between market sentiment and price movements. For instance, Kristoufek [2015] shows that Google Trends is one of the factors affecting the price of Bitcoin and Karalevicius et al. [2018] uses sentiment analysis of social media forums to predict intraday Bitcoin prices.

Furthermore, recent activities in the Bitcoin market show that some public figures have significant influence on the Bitcoin price. On 19 January 2021, Elon Musk's positive tweet about Bitcoin caused the price to briefly rise by around $5000 in an hour to $37,299. Later on February 8, Tesla's purchase of $1.5 billion worth of Bitcoin and announcement of the plan to start accepting Bitcoin as payment for vehicles pushed the Bitcoin price to $44,141. Interestingly, later on May 13, Tesla announced that it will not accept bitcoin anymore since its mining process is not environment friendly, which resulted in the price of Bitcoin dropping by approximately 12% on the same day.

In this chapter, we use two methods to measure market sentiment score in order to include market sentiment as an explanatory variable in our linear regression model and

investigate whether or not it has a significant impact on the Bitcoin price.

To measure market sentiment, we focus on social media text, more specifically `www.twitter.com` users' tweets. The reason we choose this platform is that many small and large Crypto traders post their financial view on this media and we believe that the users' sentiment on Twitter.com might be a good indicator of market sentiment.

6.1 Measuring sentiment feature

6.1.1 Data

In this part of the thesis, we study the price impact function for the period spanning January 10 to February 8 2021. For this time period, all English language tweets related to Bitcoin or cryptocurrency were collected through Twitter API, which counts to more than 6,980,000 tweets. Along with the tweet text, some other attributes[1] regarding the content and popularity of the tweets' authors were collected. These attributes are not used in this thesis but can definitely be used for further research. Table 6.1.1 displays some of these tweets and their influence attributes.

[1]These attributes include 'retweet count', 'reply count', 'like count', 'quote count', 'followers count', 'following count'

created_at	text	retweet_count	reply_count	like_count	quote_count	followers_count	following_count
2021-02-08 T18:04:01.000Z	[1D] Bitcoin market is weakly trending up current momentum suggests the market is overbought. update investor foreignexchange visit: https://t.co/yuSljvM9Fq for more!	6	0	0	0	1728	4161
2021-02-08 T14:28:38.000Z	RT @verge: Tesla to accept bitcoin as payment in "near future" after $1.5 billion investment https://t.co/NGwdOaGcxp https://t.co/QTxtUajiqZ	276	0	0	0	146	124
2021-02-01 T00:00:37.000Z	RT @ByzGeneral: I've never seen a triple top play out in crypto, so I'm gonna go ahead and say that this isn't the top for $ETH	19	0	0	0	0	91
2021-02-01 T00:00:29.000Z	the total number of Bitcoin transactions in the past 24 hours was: 288893	0	0	0	0	10	0
2021-02-01 T00:00:29.000Z	#Bitcoin is just nice	1	0	0	1	251	261

Table 6.1: Tweets and influence attributes collected from Twitter.com API

6.1.2 Market sentiment score feature

We define a new variable, MSS_t, which is the market sentiment score during interval t and is measured using

$$MSS_t = \sum_{k=1}^{N} SS_t^k. \tag{6.1}$$

In this equation, SS_t^k is the sentiment score label (which is either -1, 0 or 1) of the k-th tweet posted during the t-th time interval. Note that we have not considered influence attribute variables in our model since all retweets were included in the computations, so if a post is popular and important, its significance was taken into account by the fact that the sentiment score label of all retweets are included in the market sentiment score variable.

6.1.3 Sentiment labeling methods

For the purpose of this study, we use two different methods to measure a tweet's direction score. These methods classify tweets in 3 categories, positive, neutral and negative. The positive class includes tweets that show optimistic and bullish view toward Bitcoin and are scored as +1; neutral tweets don't have any clear sentiment about Bitcoin price change and are labeled as 0; finally, the negative tweets have a pessimistic and bearish sentiment toward Bitcoin and are scored as -1.

The methods used to score the tweets are VADER (Hutto and Gilbert [2014]), which is a rule-based sentiment analysis method, and BERT (Devlin et al. [2018]), which is a transformer-based machine learning technique. The following sections elaborate on how these models work.

VADER

VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis method that is specifically trained for the sentiments expressed in social media. This model proposes a compound score for a social media text, which is computed by summing the valence scores of each word in the lexicon, adjusted according

to the rules. This compound score is then is normalized to be between -1 (most extreme negative) and +1 (most extreme positive). Compared to other rule-based models, VADER is popular among researchers because of the possibility of detecting sentiment of complicated texts, such as (Hutto and Gilbert [2014]):

- Typical negations (e.g., "not good")
- Use of contractions as negations (e.g., "wasn't very good")
- Conventional use of punctuation to signal increased sentiment intensity (e.g., "Good!!!")
- Conventional use of word-shape to signal emphasis (e.g., using ALL CAPS for words/phrases)
- Using degree modifiers to alter sentiment intensity (e.g., intensity boosters such as "very" and intensity dampeners such as "kind of")
- Understanding many sentiment-laden slang words (e.g., 'sux')
- Understanding many sentiment-laden slang words as modifiers such as 'uber' or 'friggin' or 'kinda'
- Understanding many sentiment-laden emoticons such as :) and :D
- Translating utf-8 encoded emojis
- Understanding sentiment-laden initialisms and acronyms (for example: 'lol')

The VADER model output determines three score for "positive", "negative" and "neutral" classes, which are the probabilities of the text sentiment lying in each category, and obviously add up to 1. Table 6.2 are some of the examples labeled by the researcher who developed the VADER model.

The compound score could also be used to specify standardized threshold and classify text into three mentioned categories. A typical use of threshold value (used in Hutto and

Table 6.2: Examples labeled by VADER

Sentence	Probability of lying in each category
The book was good.	'pos':0.492, 'compound':0.440, 'neu':0.508, 'neg':0.0
The book was only kind of good	'pos':0.303, 'compound':0.383, 'neu':0.697, 'neg':0.0
Today SUX!	'pos':0.0, 'compound':-0.546, 'neu':0.22, 'neg':0.779
Make sure you :) or :D today!	'pos':0.706, 'compound':0.863, 'neu':0.294, 'neg':0.0
Not bad at all	'pos':0.487, 'compound':0.431, 'neu':0.513, 'neg':0.0

Gilbert [2014]) is:

$$\begin{cases} \text{Positive Sentiment} & \text{if} \quad \textit{compound score} \geq 0.05 \\ \text{Neutral Sentiment} & \text{if} \quad -0.5 < \textit{compound score} < 0.05 \\ \text{Negative Sentiment} & \text{if} \quad \textit{compound score} \leq -0.05 \end{cases} \tag{6.2}$$

BERT

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google (Devlin et al. [2018]). Transformer is a deep-learning method that relies on a self-attention mechanism. The self-attention model allows inputs to interact with each other, i.e calculate attention of all other inputs with respect to one input. The steps described below explain how BERT uses a self-attention mechanism to contextualize tokens and develop a state-of-the-art model in the NLP literature.

1. Tokenizing:

   The first step of BERT is transforming words into tokens (tokenizing), that is, transferring a word to its id in the vocabulary. By default, BERT will add a token to the beginning and the end of a sentence ([CLS] and [SEP]) and pad or truncate the sentence to the maximum length allowed. For instance, the sentence `"Let's learn deep learning!"`, is tokenized to `['[CLS]', 'Let', "'", 's', 'learn', 'deep', 'learning', '!', '[SEP]', '[PAD]']` if the maximum allowed length is 10.

Then each token will be converted to its ID in the vocabulary. For the same sentence, the converted to ID list would be `[101, 2421, 112, 188, 3858, 1996, 3776, 106, 102, 0]`. By default, BERT converts `'[CLS]'`, `'[SEP]'` and `'[PAD]'` to IDs `101`, `102` and `0` respectively.

2. Word embedding:

   Each word or ID in the BERT vocabulary has a constant vector. These word embeddings are vector of real, continuous values with length of 768, and are constant in all sentences regardless of the context of corpus.

3. Positional embedding:

   BERT combines the word embeddings with positional embedding, which takes into account the information about the order of input tokens.

4. Token relationships

   The key to the state-of-the-art performance in the BERT method is that it transforms the embeddings to create the right numerical picture from the tokens in any given sentence through the scaled dot-product of self-attention mechanism. The words in a sentence sometimes relate to each other, like `deep` and `learning` in the previous example, and sometimes they don't. To determine how related two tokens are, attention simply computes the scalar product of their embeddings.

   However, there could be important relations in a text that are not necessarily the relation between the meaning of words. Two words can be completely different and have grammatical relations, like a subject and a verb, a preposition and a complement, etc. To address this issue, the embeddings go through different linear projections so that one embedding creates a key, a query, and a value vector. This projection gives more freedom to the BERT model to select the components of the embeddings so that the scalar products between the keys and the queries represent the relationships that matter. This scalar product, which characterizes the level of relation between the query's token and every other token, is then given to a softmax

activation function. The results of the softmax activation function of input tokens are combined to find the embedding specific to the context. For instance, in the above example, the value of `learning` will be added to context embedding of `deep` with high significance.

5. Multi-head attention

   The self-attention module projects query, key and value vectors in *n* different ways and repeats its computations multiple times in parallel. This is what is called THE multi-head attention method in transformers. All of these calculations are then combined together to produce a final attention score embedding.

BERT, and basically all other self-attention models, are pre-trained on plain text corpus. For instance, BERT is pre-trained on the entirety of the English Wikipedia and on the Brown Corpus[2] and BERTweet is pre-trained on a dataset containing 850M English tweets.

BERT models are designed to solve various NLP tasks, and their setting can be adjusted based on task specifications.

For classification tasks, we should append the special [CLS] token to the beginning of every sentence. This token has special significance in BERT classification problems. BERT consists of 12 Transformer layers. Each transformer takes in a list of token embeddings, and produces the same number of embeddings on the output, but with the feature values changed. Figure 6.1 demonstrates the structure of BERT classifier model.

On the output of the final 12^{th} transformer, only the [CLS] token embedding is used by the classifier following it.

[2]The Brown University Standard Corpus of Present-Day American English (or just Brown Corpus) is an electronic collection of text samples of American English, the first major structured corpus of varied genres. This corpus first set the bar for the scientific study of the frequency and distribution of word categories in everyday language use. Compiled by Henry Kučera and W. Nelson Francis at Brown University, in Rhode Island, it is a general language corpus containing 500 samples of English, totaling roughly one million words, compiled from works published in the United States in 1961.

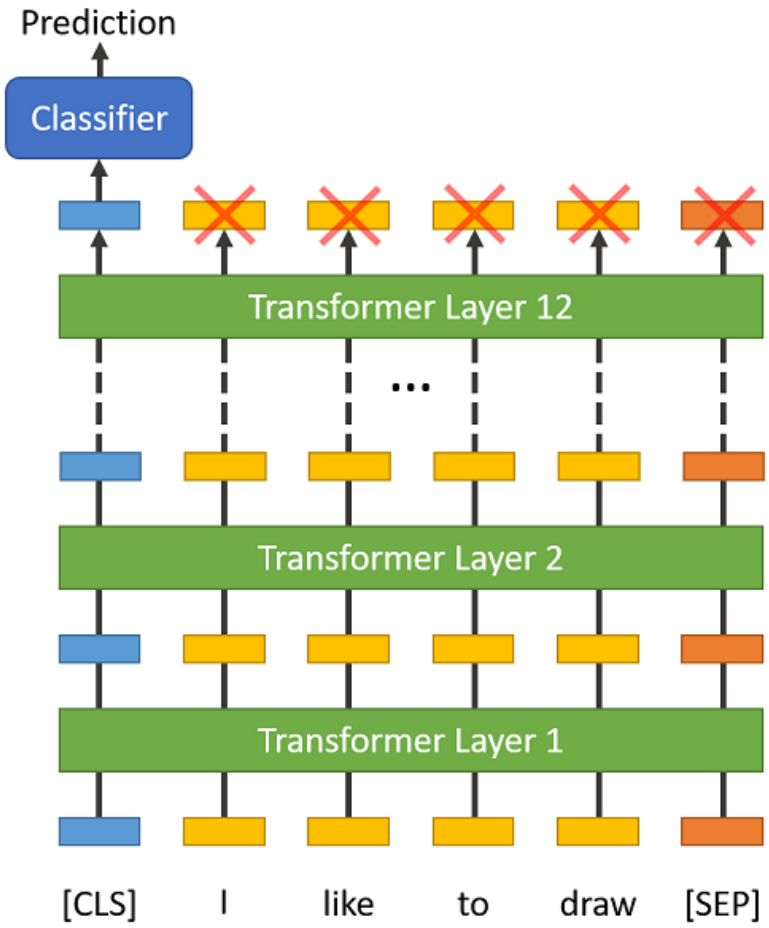


Figure 6.1: Structure of "BERT for classification"

"The first token of every sequence is always a special classification token ([CLS]). The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks." (from Devlin et al. [2018])

The crucial part of training a model with transformers is the combination of pre-training and fine-tuning. The pre-training step provides information that is task-independent, while the fine-tuning step provides task-specific information to the model. The fine-tuning step modifies the higher layers of the base model so that it fits the task at hand better.

For the purpose of this study, we fine-tune a pre-trained transformer model for the task of classifying tweets into the three categories mentioned before, that is, 'Positive', 'negative' and 'neutral'.

6.2 Measuring the market sentiment score (MSS_t) variable

This section shows how we measure the MSS_t feature using the VADER and BERT methods.

6.2.1 Train, validation and test data

In order to evaluate the accuracy of VADER and BERT classifiers, a set of 1961 tweets, randomly selected from the collected data, was manually labeled. In the labeling step, any tweet that included positive news related to cryptocurency or an optimistic view on its market was labeled as positive; tweets posting negative news or a bearish view on the market were labeled as negative, and all other posts that were not showing specific sentiment toward the market were labeled as neutral.

This data set was then divided into three subsets, training, validation and test, according to the ratios (0.68, 0.17, 0.15).[3]

Figure 6.2 displays the distribution of the labeled tweets among the three categories. As this pie chart suggests, our tweet data set is noticeably imbalanced and we should take this point into account going forward.

6.2.2 Using VADER on test data

The VADER method was defined in the previous section, and we did not modify its default setting. We apply VADER on the test data and see how well it can label these tweets. We use VADER on test data in order to be able to compare its accuracy with BERT's on the same data set.

Table 6.3 displays the confusion matrix of the VADER classifier on the 295 samples in the test data set.

[3]BERT as a deep learning method requires a validation set.

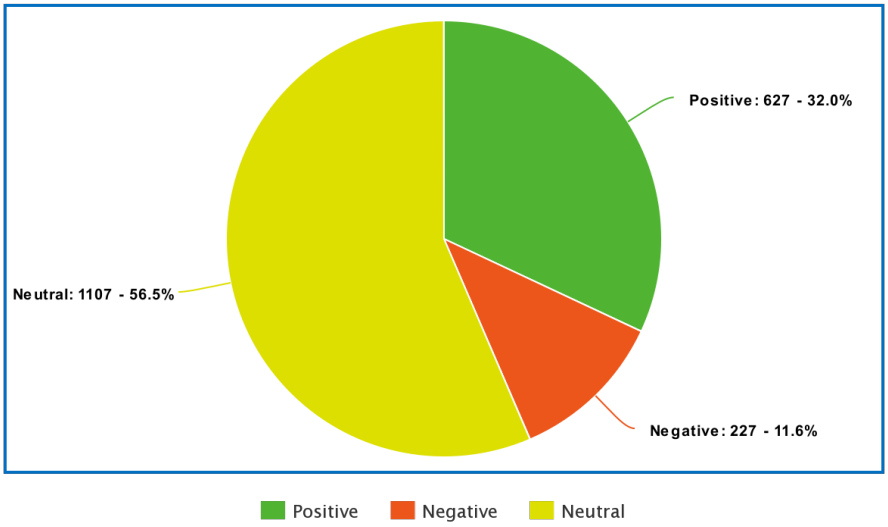


Figure 6.2: Labeled tweets distribution among three categories

Table 6.3: Confusion matrix for VADER classifier on test data set

		Predicted labels		
		Negative	Neutral	Positive
True labels	Negative	13	6	9
	Neutral	27	56	94
	Positive	11	32	47

6.2.3 Fine-tuning a transformer model

As explained above, to train a BERT classifier, a pre-trained model should be selected and be set as the default settings of the transformer, which is independent of the task. This model should then be fine-tuned using training data, which is task specific. For our purpose, three pre-trained transformer models, which could be related to our task specifications, have been chosen as candidates for the pre-trained model.

- **bert-base-uncased**: This is the most common BERT model, trained on the BooksCorpus (800M words) and Wikipedia (2,500M words) for general NLP tasks. It was first introduced in paper Devlin et al. [2018].

- **BERTweet**: This is the first public large-scale language model pre-trained for English Tweets. The corpus used to pre-train BERTweet consists of 850M English

Tweets (16B word tokens 80GB). The general architecture can be found in paper Nguyen et al. [2020].

- **FinBERT**: This is a pre-trained NLP model to analyze sentiment of financial text. It is built by further training the BERT language model in the finance domain, using a large financial corpus and thereby fine-tuning it for financial sentiment classification (see details in Araci [2019]).

These three models were tested separately on our test data set. As the 'bert-based-uncased' showed a better accuracy, we use this pre-trained model going forward.

Fine-tuning the BERT classifier

As explained above, a set of training data should be used to fine-tune the BERT classifier; however there is an issue with our training data set, being that it is seriously imbalanced. The default loss function in the BERT classification models is *"Cross Entropy Loss"*[4], which does not consider the weight of each class and the fact that our training data is imbalanced in its computations. As a result, if the model is fine-tuned on our actual training data set, it will predict mostly *Neutral* and *Positive* labels since by doing so it will have a high accuracy[5]. We need to address this issue before training our BERT classifier.

Balancing the training data set

The issue of imbalanced data set is common in the literature and industry and there are various ways to address it, including random oversampling, random under-sampling and the Synthetic Minority Oversampling Technique (SMOTE). In this thesis, we use a simple random oversampling technique, which basically consists of randomly duplicating examples in the minority classes to generate a balanced data set. We applied this technique to

[4]Cross Entropy Loss measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label.

[5]This investigation has been done but the results are not reported since they were poor.

the "Positive" and "Negative" classes to balance the training data set. The distribution of the training data set after oversampling is shown in Figure 6.3.

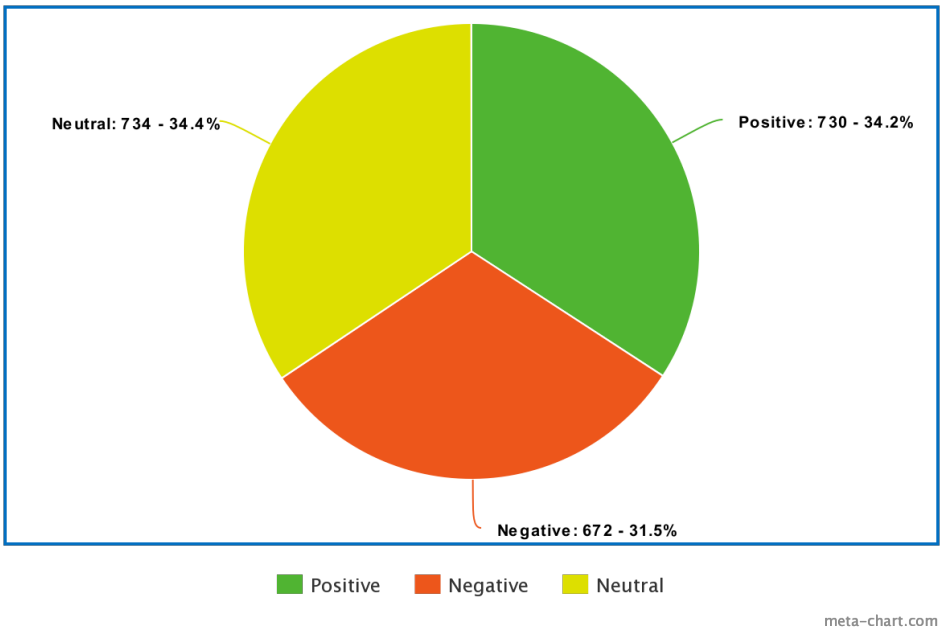


Figure 6.3: Labeled tweets distribution among three categories after oversampling

Hyperparameter tuning

The fine-tuning step requires tuning the hyperparameters. In machine learning, a hyperparameter is a parameter whose value is used to control the learning process. We use the values reported in Table 6.4 (taken from BERT original paper Devlin et al. [2018]) to tune the hyperparameters in the model.

Table 6.4: Hyperparameters used to tune the model

Hyperparameter	tested values
# of epochs	2,3,4,5
Learning rate	2e-5, 5e-5, 1e-4
Batch size	16, 32

We trained the model using these values[6], and compared them on the basis of their

[6]The validation data is not seen during training

weighted accuracy on the test data set. The best combination for our data is *# of epochs: 4, learning rate: 1e-4, batch size:32* yielding the best performance among all hyperparameter sets. Table 6.5 displays the confusion matrix of the resulting BERT model on the test data set.

Table 6.5: Confusion matrix for BERT classifier on the test data set

		Predicted labels		
		Negative	Neutral	Positive
True labels	Negative	14	4	10
	Neutral	6	139	32
	Positive	2	20	68

6.2.4 Comparing BERT and VADER models

Examination of Tables 6.3 and 6.5 shows that the developed BERT model has a better performance, both in overall and individual classes' accuracy. Table 6.6 displays the label predicted by BERT and VADER for some tweets that were randomly selected from our data set.

Table 6.6 shows that both models successfully label the first tweet; the third entry is labeled correctly only by VADER , while the second, fourth and fifth labels are correctly labeled only by BERT.

6.3 Price impact function with sentiment variable

6.3.1 Regression model

In this section, we investigate the contribution of a sentiment variable to the linear regression model introduced in Chapter 3. Accordingly, the price impact function becomes

$$\begin{aligned} \Delta P_t = \theta_1 V_t^{orders} + \theta_2 P_t^{orders} + \theta_3 V_t^{trades} + \theta_4 P_t \\ + \theta_5 \Delta P_{t-1} + \theta_6 \frac{v_t}{\overline{v}_t} + \theta_7 \sigma_t + \theta_8 MSS_t + \eta_t, \end{aligned} \tag{6.3}$$

Table 6.6: Sample labeled tweets by VADER and BERT

Tweet	VADER label	BERT label
RT @business: Elon Musk's Bitcoin support helps the market value of crypto tokens climb to a record https://t.co/lb6Sm9XxQK	1	1
@tyler Let's pump that bitcoin dogecoin	0	1
@SimonDingle Wait until he contemplates whether Tesla shares are backed by Bitcoin or if Bitcoin is backed by Tesla earnings.	0	1
@adetolaov The CBN just TODAYdeclared crypto as prohibited but wants the banks to close accts of people/companies that transacted in it when it wasnt prohibited.... What kind of retroactive enforcement is this..?	0	-1
BITCOIN WILL PEAK IN A FEW MONTHS!!! Today changed everything... [TIME TO PREPARE] https://t.co/L8gL0pL2ry	0	1

in which variable MSS_t is defined by Equation 6.1.

6.3.2 Regression results

Table 6.7 displays the regression result for the period January 10 to February 8, 2021, without the sentiment feature. All regressions are performed for 5-minute windows and 10 look-back periods.

Tables 6.8 and 6.9 display the corresponding results for regressions including a sentiment feature, measured by the VADER and BERT models, respectively.

Table 6.7: Regression result for January 10 to February 8 without sentiment score

Dep. Variable:	Y	**R-squared (uncentered):**	0.112
Model:	OLS	**Adj. R-squared (uncentered):**	0.112
Method:	Least Squares	**F-statistic:**	153.9
Date:	Wed, 29 Dec 2021	**Prob (F-statistic):**	7.19e-215
Time:	03:51:05	**Log-Likelihood:**	36553.
No. Observations:	8523	**AIC:**	-7.309e+04
Df Residuals:	8516	**BIC:**	-7.304e+04
Df Model:	7		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0010	0.000	8.938	0.000	0.001	0.001
θ_2	5.3038	1.017	5.216	0.000	3.311	7.297
θ_3	1.52e-05	8.77e-07	17.338	0.000	1.35e-05	1.69e-05
θ_4	-0.0007	9.47e-05	-6.873	0.000	-0.001	-0.000
θ_5	0.1331	0.012	10.848	0.000	0.109	0.157
θ_6	0.0026	0.001	4.806	0.000	0.002	0.004
θ_7	0.1325	0.019	6.863	0.000	0.095	0.170

Table 6.8: Regression result for January 10 to February 8 including the sentiment feature (VADER)

Dep. Variable:	Y	**R-squared (uncentered):**	0.112
Model:	OLS	**Adj. R-squared (uncentered):**	0.112
Method:	Least Squares	**F-statistic:**	134.8
Date:	Wed, 29 Dec 2021	**Prob (F-statistic):**	6.76e-214
Time:	03:58:10	**Log-Likelihood:**	36554.
No. Observations:	8523	**AIC:**	-7.309e+04
Df Residuals:	8515	**BIC:**	-7.304e+04
Df Model:	8		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0010	0.000	8.944	0.000	0.001	0.001
θ_2	5.3115	1.017	5.223	0.000	3.318	7.305
θ_3	1.526e-05	8.8e-07	17.344	0.000	1.35e-05	1.7e-05
θ_4	-0.0006	0.000	-5.419	0.000	-0.001	-0.000
θ_5	0.1327	0.012	10.808	0.000	0.109	0.157
θ_6	0.0026	0.001	4.824	0.000	0.002	0.004
θ_7	0.1347	0.020	6.906	0.000	0.096	0.173
θ_8 **(VADER)**	-2.154e-07	2.71e-07	-0.794	0.427	-7.47e-07	3.17e-07

Table 6.9: Regression result for January 10 to February 8 including a sentiment feature (BERT)

Dep. Variable:	Y	**R-squared (uncentered):**	0.112
Model:	OLS	**Adj. R-squared (uncentered):**	0.112
Method:	Least Squares	**F-statistic:**	134.9
Date:	Wed, 29 Dec 2021	**Prob (F-statistic):**	4.32e-214
Time:	03:53:20	**Log-Likelihood:**	36554.
No. Observations:	8523	**AIC:**	-7.309e+04
Df Residuals:	8515	**BIC:**	-7.304e+04
Df Model:	8		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0010	0.000	8.972	0.000	0.001	0.001
θ_2	5.2951	1.017	5.208	0.000	3.302	7.288
θ_3	1.533e-05	8.83e-07	17.361	0.000	1.36e-05	1.71e-05
θ_4	-0.0006	9.61e-05	-6.557	0.000	-0.001	-0.000
θ_5	0.1325	0.012	10.791	0.000	0.108	0.157
θ_6	0.0026	0.001	4.849	0.000	0.002	0.004
θ_7	0.1411	0.021	6.875	0.000	0.101	0.181
θ_8 **(BERT)**	-2.335e-07	1.89e-07	-1.237	0.216	-6.04e-07	1.36e-07

In both cases, the sentiment feature has no statistically significant contribution to the price-impact function.

Note however that the regression model described by Equation (6.3) assumes that the market sentiment immediately impacts the price change, which may not be realistic. It makes more sense to assume that there is a lag between the instant where investors and traders realize there is an update in the market sentiment and the moment where they start trading and affecting the price. Modifying Equation 6.3 to account for a lag of two periods yields

$$\begin{aligned} \Delta P_t = \theta_1 V_t^{orders} + \theta_2 P_t^{orders} + \theta_3 V_t^{trades} + \theta_4 P_t \\ + \theta_5 \Delta P_{t-1} + \theta_6 \frac{v_t}{\overline{v}_t} + \theta_7 \sigma_t + \theta_8 MSS_{t-2} + \eta_t. \end{aligned} \tag{6.4}$$

Table 6.10 displays the regression results corresponding to Equation 6.4 including a sentiment feature measured by the VADER model, while Table 6.11 reports the results of the same regression model using a sentiment feature measured by the BERT model. While the VADER variable is still not significant, the BERT sentiment variable with a lag

Table 6.10: Regression result for January 10 to February 8 including a sentiment feature (VADER) with a lag of two periods

Dep. Variable:	Y	**R-squared (uncentered):**	0.112
Model:	OLS	**Adj. R-squared (uncentered):**	0.112
Method:	Least Squares	**F-statistic:**	134.8
Date:	Wed, 29 Dec 2021	**Prob (F-statistic):**	6.70e-214
Time:	04:38:09	**Log-Likelihood:**	36554.
No. Observations:	8523	**AIC:**	-7.309e+04
Df Residuals:	8515	**BIC:**	-7.304e+04
Df Model:	8		

	coef	**std err**	**t**	**P$>\|$t$\|$**	**[0.025**	**0.975]**
θ_1	0.0010	0.000	8.947	0.000	0.001	0.001
θ_2	5.3148	1.017	5.226	0.000	3.321	7.308
θ_3	1.524e-05	8.78e-07	17.355	0.000	1.35e-05	1.7e-05
θ_4	-0.0006	0.000	-5.347	0.000	-0.001	-0.000
θ_5	0.1328	0.012	10.813	0.000	0.109	0.157
θ_6	0.0026	0.001	4.805	0.000	0.002	0.004
θ_7	0.1346	0.019	6.909	0.000	0.096	0.173
θ_8 **(VADER)**	-2.18e-07	2.71e-07	-0.805	0.421	-7.49e-07	3.13e-07

of two periods has a statistically significant impact on the price change, slightly improving the R^2 of the regression model.

6.4 Conclusion

In this chapter we introduced two methods to measure a market sentiment score from Twitter posts, VADER and BERT. Our investigations suggest that:

- BERT shows a better classifying performance in comparison with VADER
- A market sentiment variable, measured by a BERT classifier, and lagged by two periods, has a significant impact on price change.

Table 6.11: Regression result for January 10 to February 8 including a sentiment feature (BERT) with a lag of two periods

Dep. Variable:	Y	**R-squared (uncentered):**	0.113
Model:	OLS	**Adj. R-squared (uncentered):**	0.112
Method:	Least Squares	**F-statistic:**	135.3
Date:	Wed, 29 Dec 2021	**Prob (F-statistic):**	1.15e-214
Time:	04:47:18	**Log-Likelihood:**	36556.
No. Observations:	8523	**AIC:**	-7.310e+04
Df Residuals:	8515	**BIC:**	-7.304e+04
Df Model:	8		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
θ_1	0.0010	0.000	9.009	0.000	0.001	0.001
θ_2	5.3120	1.017	5.225	0.000	3.319	7.305
θ_3	1.538e-05	8.81e-07	17.459	0.000	1.37e-05	1.71e-05
θ_4	-0.0006	9.67e-05	-6.311	0.000	-0.001	-0.000
θ_5	0.1318	0.012	10.723	0.000	0.108	0.156
θ_6	0.0026	0.001	4.798	0.000	0.002	0.004
θ_7	0.1465	0.020	7.154	0.000	0.106	0.187
$\theta_8(BERT)$	-3.85e-07	1.88e-07	-2.047	0.041	-7.54e-07	-1.62e-08

The impact of market sentiment could be investigated further, for instance by including some of influence variables (popularity of the tweet content or of the person posting it). Another promising avenue would be to investigate different measures for the market sentiment score. For instance, instead of labelling tweets by (-1, 0 and 1), one could use a more continuous measure, such as the probability of lying in each class.

Chapter 7

Conclusion

This short conclusion reviews the results achieved by this research and reported in this thesis and suggest some avenues for further research.

The first part of the thesis proposes a linear regression model for the the Bitcoin price-impact function, using features extracted from the order book data.

- The empirical results show that all features in the regression model have significant impact on price change value. More specifically, the features extracted from order book, V_t^{orders} and P_t^{orders} are significant variables in the price-impact functions. This finding applies to regression for different windows (2, 5 and 10 minutes) and look-back periods (10, 50 and 100 periods). The signs of all coefficients are justified by economical and financial intuitions and in line with findings in the literature.

- We find that the sensitivity of price changes to the value of independent variables differs according to the direction of the price change. Breaking the single regression to two separate models improved the forecasting precision from 11% to approximately 57%.

- We propose a logistic regression model to predict the direction of price movements. This model was applied to out-of-sample data with 66% precision and 63% F1

scores. Our logistic regression model performance outperforms other classifier methods in similar studies.

- Comparing the out-of-sample MAE of single regression and 2-step algorithms suggests that a single regression is more precise in predicting price change, while its R^2 is significantly lower. Further investigation shows that a 2-step algorithm can forecast the dependent variable with noticeably lower error if the direction is predicted correctly by the logistic regression model.

- The single regression model outperforms Zhou [2012] universal impact function in forecasting out-of-sample price change values.

In the second part, we propose two methods for detecting spoofers in financial markets. The ex-post method was tested on a sample trade that could be a candidate for spoofing. It is not possible to investigate this trade any further since the data for trading accounts and their activities were not available. An ex-ante method was also proposed to detect spoofers prior to an illegal activity based on their predicted impact on market price measured by the results of a 2-step regression.

In the last part of thesis, we studied the impact of market sentiment score on price movements. We measured market sentiment score using two methods, VADER and BERT. We find that

- The accuracy of the BERT classification model, fine-tuned on 'bert-base-uncased', was significantly higher than VADER sentiment analyzer on our test data set (75% vs. 35%).

- The sentiment score for an interval, measured by both VADER and BERT methods, does not have an immediate significant impact on price change.

- However, the sentiment score measured by the BERT classifier method has a significant lagged impact on price change.

This research could be developed further in the following directions:

- The model proposed for the price-impact function in this research is a simple linear regression. More complicated models, like time series, polynomials, and deep learning models could be investigated for improvement in forecasting precision.
- The classifier method used in this study is a logistic regression model. More advanced ML methods, like LSTM and CNN could be used to improve the prediction accuracy.
- The methods proposed for spoofing detection should be tested on real spoofing data so their accuracy, strengths and weaknesses be determined.
- To improve BERT classifier precision, more tweets should be manually labeled. Moreover, the pre-trained model used in this thesis is 'bert-base-uncased', which is not exactly related to our task. A domain related BERT model could be pre-trained on Crypto related tweets to improve the accuracy of the BERT classifier.
- The content of tweets and popularity of tweeter have not been considered in our model. These variables could be added to sentiment score computations to make it closer to real market sentiment.

Bibliography

F. Allen and D. Gale. Stock-price manipulation. *The Review of Financial Studies*, 5(3): 503–529, 1992.

R. Almgren, C. Thum, E. Hauptmann, and H. Li. Direct estimation of equity market impact. *Risk*, 18(7):58–62, 2005.

Y. Amihud. Illiquidity and stock returns: cross-section and time-series effects. *Journal of financial markets*, 5(1):31–56, 2002.

D. Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

M. J. Barclay and J. B. Warner. Stealth trading and volatility: Which trades move prices? *Journal of financial Economics*, 34(3):281–305, 1993.

B. Biais, P. Hillion, and C. Spatt. An empirical analysis of the limit order book and the order flow in the paris bourse. *the Journal of Finance*, 50(5):1655–1689, 1995.

J.-P. Bouchaud. Price impact. *arXiv preprint arXiv:0903.2428*, 2009.

Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity. Detecting price manipulation in the financial market. In *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 77–84. IEEE, 2014.

Á. Cartea, S. Jaimungal, and Y. Wang. Spoofing and price manipulation in order-driven markets. *Applied Mathematical Finance*, 27(1-2):67–98, 2020.

S. Chakravarty. Stealth-trading: Which traders' trades move stock prices? *Journal of Financial Economics*, 61(2):289–307, 2001.

K. Chan and W.-M. Fong. Trade size, order imbalance, and the volatility–volume relation. *Journal of Financial Economics*, 57(2):247–273, 2000.

R. Chen and M. Lazer. Sentiment analysis of twitter feeds for the prediction of stock market movement. *stanford edu Retrieved January*, 25:2013, 2013.

Z. Chen, W. Stanzl, and M. Watanabe. Price impact costs and the limit of arbitrage. In *EFA 2002 Berlin Meetings Presented Paper*, pages 00–66, 2002.

J. H. Choi, K. Larsen, and D. J. Seppi. Information and trading targets in a dynamic market equilibrium. *Journal of Financial Economics*, 132(3):22–49, 2019.

R. Cont, A. Kukanov, and S. Stoikov. The price impact of order book events. *Journal of financial econometrics*, 12(1):47–88, 2014.

M. Coppejans, I. Domowitz, and A. Madhavan. Dynamics of liquidity in an electronic limit order book market. *Unpublished working paper, Duke University*, 2003.

F. Corsi. A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196, 2009.

A. K. Davis, J. M. Piger, L. M. Sedor, et al. *Beyond the numbers: An analysis of optimistic and pessimistic language in earnings press releases*, volume 5. Citeseer, 2006.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

A. Dufour and R. F. Engle. Time and the price impact of a trade. *The Journal of Finance*, 55(6):2467–2498, 2000.

Z. Eisler, J.-P. Bouchaud, and J. Kockelkoren. The price impact of order book events: market orders, limit orders and cancellations. *Quantitative Finance*, 12(9):1395–1419, 2012.

M. D. Evans and R. K. Lyons. Order flow and exchange rate dynamics. *Journal of political economy*, 110(1):170–180, 2002.

T. Fletcher and J. Shawe-Taylor. Multiple kernel learning with fisher kernels for high frequency currency prediction. *Computational Economics*, 42(2):217–240, 2013.

T. Foucault, O. Kadan, and E. Kandel. Liquidity cycles and make/take fees in electronic markets. *The Journal of Finance*, 68(1):299–341, 2013.

A. Frino, R. Segara, and H. Zheng. The impact of trade characteristics on stock return volatility: Evidence from the australian stock exchange. *Asia-Pacific Journal of Financial Studies*, 38(2):163–186, 2009.

J. Hasbrouck. Measuring the information content of stock trades. *The Journal of Finance*, 46(1):179–207, 1991.

N. Hautsch and R. Huang. The market impact of a limit order. *Journal of Economic Dynamics and Control*, 36(4):501–522, 2012.

T. Hendershott and R. Riordan. Algorithmic trading and the market for liquidity. *Journal of Financial and Quantitative Analysis*, 48(4):1001–1024, 2013.

C. Hopman. Do supply and demand drive stock prices? *Quantitative Finance*, 7(1): 37–53, 2007.

C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014.

C. M. Jones, G. Kaul, and M. L. Lipson. Transactions, volume, and volatility. *The Review of Financial Studies*, 7(4):631–651, 1994.

V. Karalevicius, N. Degrande, and J. De Weerdt. Using sentiment analysis to predict interday bitcoin price movements. *The Journal of Risk Finance*, 2018.

J. M. Karpoff. The relation between price changes and trading volume: A survey. *Journal of Financial and quantitative Analysis*, 22(1):109–126, 1987.

D. B. Keim and A. Madhavan. The upstairs market for large-block transactions: Analysis and measurement of price effects. *The Review of Financial Studies*, 9(1):1–36, 1996.

A. N. Kercheval and Y. Zhang. Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, 15(8):1315–1329, 2015.

A. Kraus and H. R. Stoll. Price impacts of block trading on the new york stock exchange. *The Journal of Finance*, 27(3):569–588, 1972.

L. Kristoufek. What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. *PloS one*, 10(4):e0123923, 2015.

A. S. Kyle. Continuous auctions and insider trading. *Econometrica: Journal of the Econometric Society*, pages 1315–1335, 1985.

E. J. Lee, K. S. Eom, and K. S. Park. Microstructure-based manipulation: Strategic behavior and performance of spoofing traders. *Journal of Financial Markets*, 16(2): 227–252, 2013.

X. Li, H. Xie, L. Chen, J. Wang, and X. Deng. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23, 2014.

F. Lillo, J. D. Farmer, and R. N. Mantegna. Master curve for price-impact function. *Nature*, 421(6919):129–130, 2003.

M. Lim and R. Coggins*. The immediate price impact of trades on the australian stock exchange. *Quantitative Finance*, 5(4):365–377, 2005.

B. R. Marshall, N. H. Nguyen, and N. Visaltanachoti. Commodity liquidity measurement and transaction costs. *The Review of Financial Studies*, 25(2):599–638, 2012.

A. Mittal and A. Goel. Stock prediction using twitter sentiment analysis. *Stanford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)*, 15, 2012.

D. Q. Nguyen, T. Vu, and A. T. Nguyen. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, 2020.

V. Niederhoffer. The analysis of world events and stock prices. *The Journal of Business*, 44(2):193–219, 1971.

A. A. Obizhaeva and J. Wang. Optimal trading strategy and supply/demand dynamics. *Journal of Financial Markets*, 16(1):1–32, 2013.

M. O'Hara. High frequency market microstructure. *Journal of Financial Economics*, 116 (2):257–270, 2015.

M. C. Pham, H. N. Duong, and P. Lajbcygier. A comparison of the forecasting ability of immediate price impact models. *Journal of Forecasting*, 36(8):898–918, 2017.

M. C. Pham, H. M. Anderson, H. N. Duong, and P. Lajbcygier. The effects of trade size and market depth on immediate price impact in a limit order book market. *Journal of Economic Dynamics and Control*, 120:103992, 2020.

V. Plerou, P. Gopikrishnan, X. Gabaix, and H. E. Stanley. Quantifying stock-price response to demand fluctuations. *Physical review E*, 66(2):027104, 2002.

T. Rao, S. Srivastava, et al. Analyzing stock market movements using twitter sentiment analysis. 2012.

D.-F. W. S. Reform and C. P. Act. Public law 111-203. *US Statutes at Large*, 124(2010): 1376, 2010.

R. P. Schumaker and H. Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):1–19, 2009.

Y.-W. Seo, J. Giampapa, and K. Sycara. Financial news analysis for intelligent portfolio management. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 2004.

P. C. Tetlock. Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3):1139–1168, 2007.

N. Torre and M. J. Ferrari. The market impact model. *Horizons, The Barra Newsletter*, 165, 1998.

A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis. Using deep learning for price prediction by exploiting stationary limit order book features. *Applied Soft Computing*, 93:106401, 2020.

Y. Wang. Strategic spoofing order trading by different types of investors in the futures markets. *Wall Street Journal*, 2015.

P. Weber and B. Rosenow*. Order book approach to price impact. *Quantitative Finance*, 5(4):357–364, 2005.

M. Wilinski, W. Cui, A. Brabazon, and P. Hamill. An analysis of price impact functions of individual trades on the london stock exchange. *Quantitative Finance*, 15(10):1727–1735, 2015.

W.-X. Zhou. Universal price impact functions of individual trades in an order-driven market. *Quantitative Finance*, 12(8):1253–1263, 2012.