Ca' Foscari
University
of Venice

Master's Degree

in

Economics and Finance

Final Thesis

Cryptocurrency markets microstructure

with a machine learning application to the Binance bitcoin market

Supervisor
Ch.ma Prof.ssa Antonella Basso

Graduand
Christian Bozzetto
Matriculation Number 893488

Academic Year
2022 / 2023

Table of Contents

Introduction

The world of financial markets is an extremely fertile ground where people, firms and institution can interact to allocate capital following the widest array of interests. This thesis will explore how these interactions can take place, who the players actually are and how prices end up moving as a result of the intricate relationships that come and go every day in the markets. This microstructure perspective will be present throughout: adopting a top-down approach, we will start from a bird's eye view of how financial markets function and gradually delve deeper into how and why microstructural features are considered to be at the very basis of price movements. Specifically, the relatively young world of cryptocurrency will be explored to appreciate the effects an ever-evolving microstructural framework can have on market efficiency.

Chapter I introduces the concept of market microstructure: market participants, types of orders, liquidity and market efficiency make a great primer to understand further, more complex considerations presented in later chapters.

Chapter II focuses on microstructural aspects that are unique to cryptocurrency markets and represent an evolution compared to traditional market structure. New players, non-stop trading, peculiar trading venues and regulatory concerns are analyzed as distinctive traits of a new phenomenon that has yet to reach its maturity stage.

Chapter III delves deeper into the technical aspects of market microstructure, proposing a practical way to measure how this seemingly abstract concept can impact prices and other market characteristics such as liquidity. The chosen analysis approach focuses on order flow, a field studying the price impact of market players posting, executing and cancelling orders during trading hours. An original analysis exercise is presented in the form of regressing mid-price changes on order flow and trade flow imbalances, with the aim of finding potential explanatory power. Moreover, an exogenous shock that characterized bitcoin trading on the Binance exchange is analyzed to gauge how changes in market characteristics can impact the explanatory power of order flow. Findings point at imbalances having significant explanatory power with respect to contemporaneous mid-price changes, and Binance's exogenous shock consisting in a sudden change in fee structure proves to have impacted such explanatory power.

Finally, Chapter IV explores the innovative world of Artificial Intelligence and Machine Learning and potential applications to order flow analysis. Advancements in Machine Learning and some further considerations are laid out, with the addition of an

original application of Machine Learning to the regression problem considered in Chapter III. Model training, feature engineering and a comparison of out-of-sample predictions with traditional OLS regression are presented, with the aim of assessing the potential improvements brought by novel complex algorithms in the field of order flow analysis. Future trends and drivers are outlined in the last section, reiterating how vast an ocean of opportunities lays ahead of us.

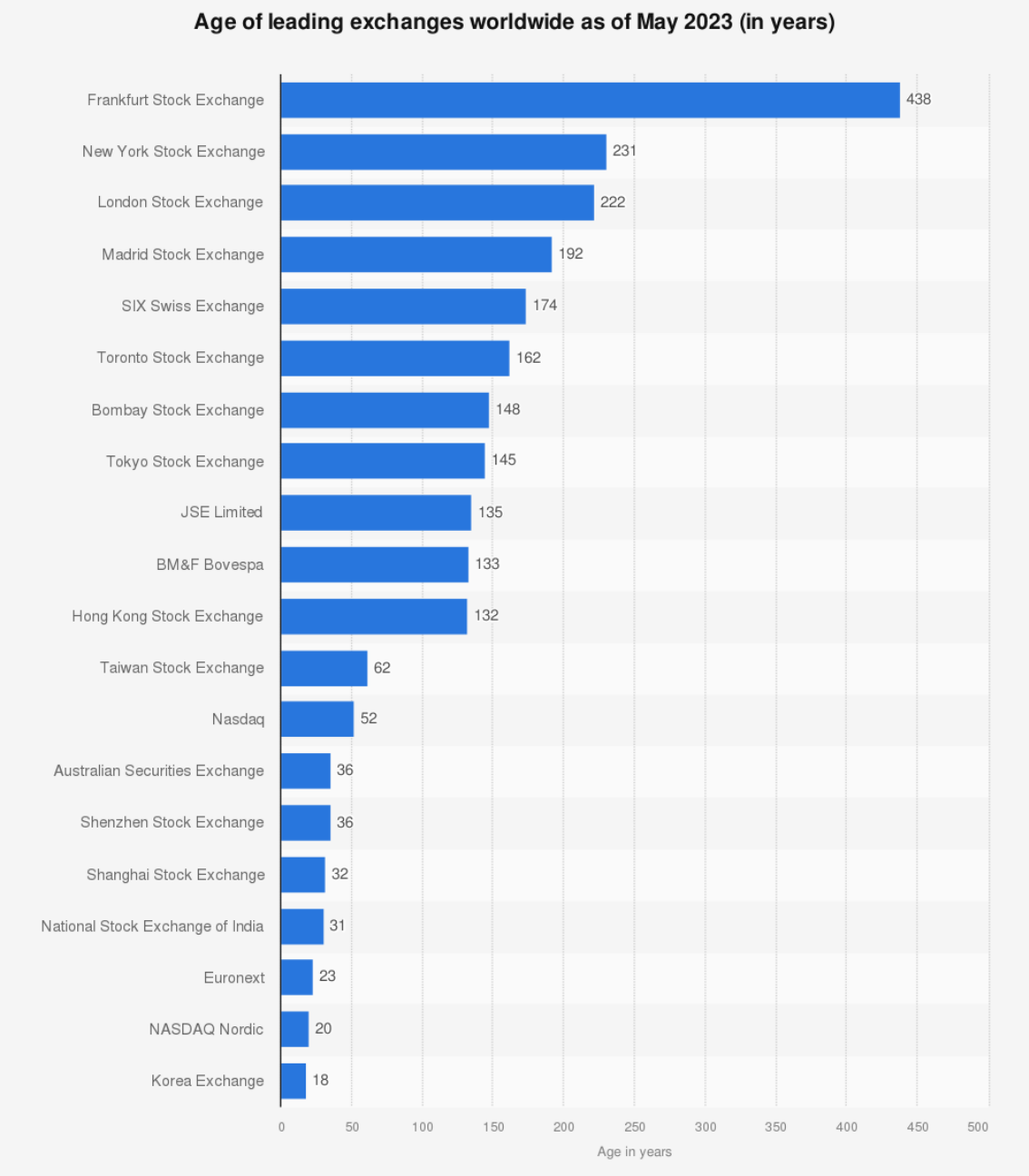
Chapter I Introducing Market Microstructure

1.1. Popularity of Financial Markets Over Time

Financial Markets are among the most essential elements in our global economy. They constitute a network of platforms enabling the transfer and allocation of capital: from liquid savings to productive investments, market participants can defer or anticipate capital availability for themselves or others. On the two ends of the spectrum, participants can conservatively allocate liquidity to low-risk assets or risk it all in an extremely incautious bet, maybe only based on a "gut feeling". One could find oneself in the markets in search of an investment for the future (e.g., in the equities market), a short-term transfer of liquidity (or a conversion thereof, e.g. from one currency to another in the foreign exchange market), protection from a risk stemming from one's activities, be it in private life (personal insurance) or during business endeavors (hedging with derivatives). Financial markets have come to offer a significant variety of assets, liabilities and instruments to channel liquidity into: from simple, easy-to-understand equity shares to extremely complex synthetic derivatives, everyone's taste and needs can potentially be fulfilled. The vast democratization of access to financial markets we have been experiencing during the last two decades has brought to the birth of platforms with extremely diversified offerings that are accessible to anyone, almost regardless of financial literacy. This is undoubtedly a double-edged sword: diversification and tailor-made solutions for those who know what they are doing (be it individual investment professionals or entities, such as investment funds and governments) on the one hand and, at the same time, unnecessary risks for the average Joe, hidden behind shiny and complexly named instruments that can constitute great opportunities on paper, while in reality they can also result in investor capitulation when paired with lack of adequate financial knowledge.

Financial Markets today are the result of an evolution that has lasted centuries and will surely continue onwards, based on one main, crucial driver: technological advancements. The advent of electronic trading in the 1970s set the first step for the democratization of the markets we are seeing today, however it is important to keep in mind that exchanges and trading venues alike already existed far before that period, although not offering the same variety of instruments (see Figure 1 for the age of the main exchanges worldwide).

Figure 1 - Age of leading exchanges worldwide as of May 2023

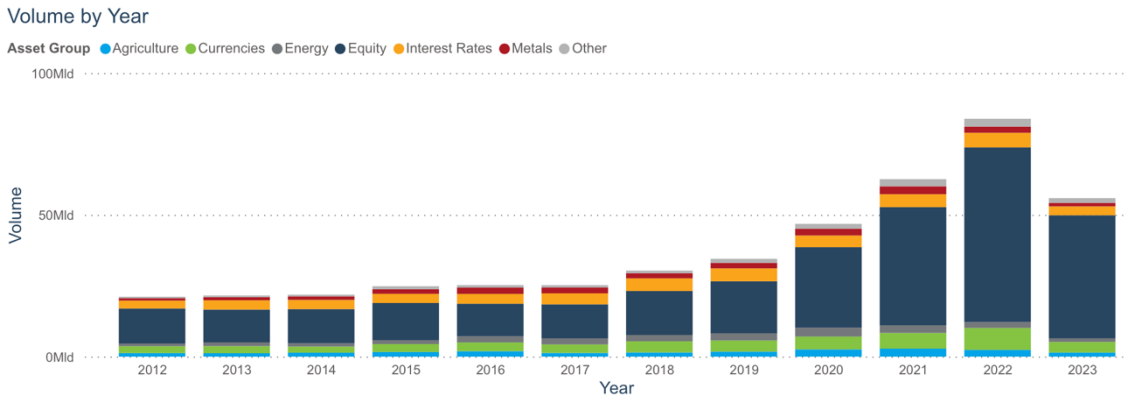


Source: Visual Capitalist, © Statista (Statista, 2023c)

Since the introduction of electronic trading, equity shares have been the most popular instruments among market participants. As time went by, new instruments, particularly derivatives contracts such as futures and options, have caught the eye of many traders for their dual purpose: hedging risk and leveraging positions. Although riskier, these instruments have lately taken the markets by storm becoming more and more popular, as Figure 2 shows.

The 21st century has seen great technological advancements in financial markets, most notably High Frequency Trading, Algorithmic Trading, the advent of Cloud Computing with the possibility of crunching vast amounts of data to train Artificial Intelligence implementations and, last but not least, Blockchain Technology. Each of these novel technologies brought disruption to the markets, birthing new participants, seeing the capitulation of others, reshaping the way people and institutions approach trading. Segmentation of markets by instruments traded and technology used has created diversified ecosystems, all referring to the financial markets umbrella, all with great commonalities but, nevertheless, all slightly different from one another in their functioning and their dynamics. One way to understand how a market actually works, why prices move in that market, what incentives move participants to buy, sell or hold and instrument, is analyzing a market's own microstructure. This master's thesis will focus on the microstructure of the youngest, most technologically inclined ecosystem: Cryptocurrency Markets. Before delving into the inner working of crypto markets, though, a general exploration of market microstructure and the possible configurations of markets is in order, and will be touched on starting from the following Section, 1.2.

Figure 2 - Worldwide volume of derivatives trading, by asset group



Source: Futures Industry Association

1.2. Market Microstructure: Making Sense of "Noise"

Market microstructure is a relatively new field of study for academic finance (Madhavan, 2000). The term was coined by Prof. Mark B. Garman at UC Berkeley in his paper titled "Market Microstructure" (Garman, 1976). Garman's work set off to analyze the markets with a different viewpoint compared to the usual theory. Specifically, he put a lens on temporal

microstructure, as he named the frenetic sequence of moment-to-moment behavior characterizing price, volume, market participants and market states.

Market microstructure is mainly (albeit not only – see Subrahmanyam, 2009) concerned with the minute aspects of market transactions: prices are believed to move around a "fair" or "intrinsic" value due to "noise". This noise is believed to be the aggregate result of many different participants interacting with each other while defending different interests or carrying different views. Market microstructure studies seek to analyze transaction dynamics and aggregate behavior between market participants, identifying the sources of noise and effectively making sense of them, analyzing various "micro-level" drivers and their impact on market efficiency. Market microstructure expands the scope of price analysis and seeks to give a more comprehensive and detailed answer to the secular question "why do prices move the way they do?". If the nature of the markets is in fact an ensemble of decision-making individuals with potentially different views and interests, observing their interaction at each instant could give some insights on the actual causal process guiding price movements (or, as it is often called, "price discovery").

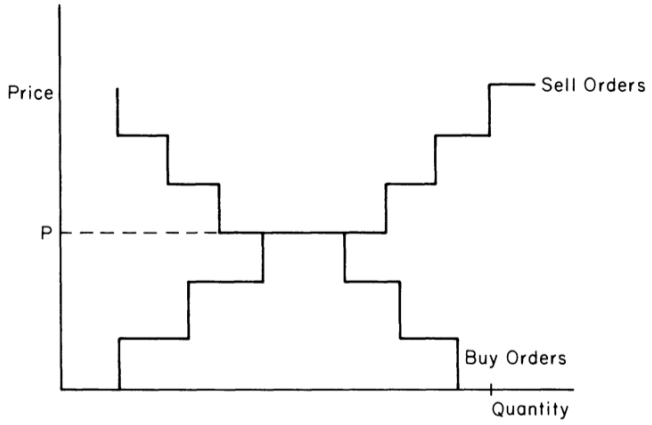
One practical example of Market microstructure analysis can be found in the 1987 work published by Y. Amihud and H. Mendelson (Amihud & Mendelson, 1987). They found significant differences between open-to-open and close-to-close returns on NYSE stocks: the former tended in fact to have relatively higher variance than the latter; they also deviated from normality more frequently and showcased more negative and significant autocorrelation. This difference was traced back by the authors to the different execution methods applied to opening and closing transactions. The opening price of NYSE-listed securities was set following a process that resembled a "clearing house" procedure, as described by Mendelson (1982): in this procedure, participants submit their orders (be it buy or sell, limit or market) and those orders are not worked until a specified clearing time. Come this clearing time, buy and sell orders are sorted and matched at a single market-clearing price, as depicted in Figure 3.

This procedure characterized the opening auction for NYSE-listed stocks at the time and is described as a "single price auction" in Larry Harris' book "Trading and Exchanges" (Harris, 2003). Steps are as follows:

a. Participants submit their orders. Some will submit limit orders to buy or sell the listed security at a specific price, others will submit market orders if they are interested in being filled at market open regardless of the price.

b. Orders are collected and organized by the exchange. Supply and demand, in the form of buy and sell orders, are initially compared to work out an equilibrium price.
c. The sorted orders are crossed by the exchange to determine the clearing price. This price level is the one maximizing the number of traded shares.
d. Orders are filled at the single clearing price, based on priority: market orders are filled first, then eligible limit orders are filled on a first-in, first-out basis.

Figure 3 - Price setting in a clearing house



Source: Amihud and Mendelson, 1987

The closing price, on the other hand, was formed like all other price levels during trading hours: that is, following what Amihud and Mendelson call a "dealership auction". A dealership auction sees participants continuously trade at price quotes set by designated market makers[1] on the trading floor. The closing price, then, was simply the last transacted quote before the bell.

This difference in price-setting mechanisms caused significant differences in the distribution of open-to-open returns compared to close-to-close returns, as outlined by the authors, observing that trading the opening exposed participants to greater volatility compared to trading the close. This analysis and others alike are the core of market microstructure studies, aimed at finding the cause of price behavior based on the functioning and organization of a specific market. If a difference in the price-setting methodology significantly affects returns, it is intuitive to infer that differences in market participants, type of assets traded, level of (de)centralization of trading venues, etc. are potentially additional drivers of price fluctuations.

[1] Market makers will be introduced in section 1.3.

Nowadays, modern microstructure studies have the aim of finding significant explanatory power in extremely short-lived patterns that can only be observed in extremely granular datasets, i.e. millisecond price and volume data. This also explains the increasing developments observed in closer years, as dealing with such data requires considerable computing power that has only become available during the last 50 years.

Three main components characterize a market's microstructure:

a. Market Participants
b. Types of order participants can submit
c. Liquidity and order flow characterizing the market

The next three sections will each be dedicated to one of the components above.

1.3. Financial Markets Participants

Price movements are ultimately the aggregate result of an ensemble of actions carried out by a large number of market participants with different views and interests driving their behavior, resulting in supply and demand forces driving quotes over time.

Transactions in the financial markets are also called "trades", and as a result those who participate in the markets are called "traders". Traders can be grouped into many categories based on their role at a specific time. The easiest, most high-level categorization would be between buyers and sellers: buyers are traders who are seeking to acquire an asset, while sellers are seeking to sell an asset. What moves buyers and sellers in their actions may differ from the classic "buy low, sell high" paradigm, since traders may be looking for diversification or risk protection regardless of asset price, as outlined below. These different objectives can be used to organize buyers and sellers in different sets.

Harris' (2003) book offers various trader categorizations, based on different aspects. The most general one would be "buy side" vs "sell side". This distinction should not be confused with buyers vs sellers, as buy side and sell side refer respectively to users and providers of exchange services.

1.3.1. The Buy Side

The "Buy Side" encompasses all traders finding themselves in the markets in search of the possibility to complete a transaction on a specific asset (that could both be a buy or a sell transaction). Typical Buy Side traders, their objectives and usual instruments are presented in Table 1.

The Buy Side includes both individuals and institutions such as investment funds. Somewhere in between Investors and Gamblers one could also add the Speculator species, usually referred to as focused on a lower time horizon compared to Investors, but surely more informed and determined to make a profit compared to Gamblers.

Table 1 - Buy Side Traders

Trader type	**Objective**	**Preferred Instrument**
Investors	Transfer of wealth from the present to the future, for themselves or their clients	Stocks, Bonds
Borrowers	Transfer of wealth from the future to the present	Mortgages, Bonds, Notes
Hedgers	Reduction of operative risk stemming from their business activities	Derivative Contracts (Futures, Forwards, Swaps)
Asset Exchangers	Acquisition of assets that they value more than the ones they offer for exchange	Currencies, Commodities
Gamblers	Entertainment via risk-taking	Various

Source: Harris (2003), p. 33, adapted

1.3.2. The Sell Side

The sell side is mainly focused on providing the buy side with liquidity and general access to instruments and trading venues. Harris identifies Dealers, Brokers and hybrid Broker-Dealers as the main sell side players.

Sell side traders only exist as a function of the buy side trader's willingness to transact without the burden of having to personally find a counterparty to their trades.

One additional key component of financial markets are what Harris calls Trade Facilitators: institution allowing orderly trading and facilitating operations linked to filled transactions.

Table 2 – Sell Side Traders

Trader type	**Objective**
Dealers	Earn trading profits by offering to take the opposite side of a Buy Side trader's transaction. Dealers profit when they buy from impatient sellers at low prices and sell to impatient buyers at high prices. The difference in prices compensates them for providing immediacy.
Brokers	Earn a commission, be it a flat fee or a spread on offered quotes, for facilitating a transaction between two traders
Broker-Dealers	Earn both trading profits and commissions

Source: Harris (2003), p. 34, adapted

1.3.3. Trade Facilitators

These are exchanges, clearing and settlement agents, depositories and custodians as outlined in Table 3.

Both clearing and settlement operations in many markets are carried out by specific entities called clearinghouses (Harris, 2003). Clearinghouses add value by guaranteeing that both parties in the transaction will perform their respective obligations: they do so by acting as a third transaction party, a mutual insurance company of sorts (Harris, 2003), selling to the buyer (guaranteeing delivery) and buying from the seller (guaranteeing payment). This can be a considerable source of risk for the clearinghouse which, as a precaution, employs various cautionary practices to limit the risk of failed trade settlement. These measures include posting collateral, monitoring credit risk, requesting periodic updates on the parties' financial situation and trading activities, and defining maximum position limits.

It is important to note that organized exchanges, although being one of the most convenient venues for the average trader, are not the only venues where transactions can occur. Specific instruments like some derivatives contracts (e.g., forwards) and corporate bonds are not traded on organized exchanges. Rather, they change hands over the counter, as a result of transactions arranged by brokers and dealers outside of exchanges bounds and rules. This will in turn intuitively affect the relative grade of liquidity for specific markets, as organized exchanges are place to far more traders and thus host more transactions than OTC arrangements. As will be outlined in section 1.4, liquidity is a crucial component to market efficiency, thus the structure and organization of a market is well able to influence the efficiency of prices quoted on that same market.

Table 3 – Main Trade Facilitators

Trader type	**Role**
Exchanges	Provide venues where traders can interact and conclude transactions. Only members of the exchange can trade there personally. Non-members need to rely on member-brokers to execute trades for them.
Clearing Agents	Before being settled, a trade must contemplate the same terms as agreed and recorded by buyer and seller. Clearing Agents ensure that the buy and sell parts of the trade are matching and clears the transaction. This third party confirmation adds security to trades by eliminating mismatches and ensuring correct execution. Clearing systems have been automated, for the most part, since the 80s (Domowitz & Steil, 1999) (Domowitz & Wang, 1994).
Settlement Agents	Ensure that the buyer receives the agreed asset and the seller receives the agreed amount of money, acting as an intermediary. Like clearing operations, also settlement operations have been automated since the 80s (Domowitz & Steil, 1999) (Domowitz & Wang, 1994).
Depositories, Custodians	Ensure the security of their client's assets and the quick availability of cash and security certificates when clients conclude a trade involving held assets. The American Depository Trust Company (DTC), owned by the Depository Trust and Clearing Corporation, is one of the largest depositories in the world with $87 trillion in held assets as of 2023[2].

Source: Harris (2003), pp. 35-38, adapted

One additional type of trade facilitators exists, although less known and often veiled by a shroud of mistery: dark pools (a different, less ominous name could be Block Trading Facility – Hayes, 2022).

Dark pools started emerging in the 80s and, as technological advancements evolved along markets and traders needs, they became progressively more popular among traders willing

[2] Source: https://www.dtcc.com/about/businesses-and-subsidiaries/dtc. Assets in custody have roughly tripled in 20 years, after accounting for inflation (see Harris, 2003: assets held amounted to $20 trillion, which would be worth some $30 trillion today)

to carry out large transactions. Banks (2010) describes dark pools as "A […] venue or mechanism containing anonymous, nondisplayed trading liquidity that is available for execution". The flow of orders submitted to a dark pool is not publicly displayed as it is in public exchanges. Apart from this lack of public display, dark pools have a structure similar to public exchanges, with similar rules (Banks, 2010). Traders wanting to settle large orders in a small number of transactions (also known as block trades - an activity that would constitute considerable price displacement, was it carried out in a public exchange where orders are visible to the public before execution[3]) while staying anonymous and without disrupting the markets can turn to dark pools, where trades are matched with anonymous counterparties and transactions are settled without public disclosure prior to execution (Zhu, 2014). The matter of block trades significantly impacting price is an evergreen topic in the markets. In fact, the number of block trades in the marketplace may be small in absolute value but they make up the largest share of traded volume in equity markets (Banks, 2010), and often absorb more liquidity than is available on a normal exchange or through a dealer network (mind, this does not mean that all block trades are internalized by dark pools. For reference, Boulton and Braga-Alves (2020) estimate that dark pools internalize roughly 16% of the dollar trading volume in the US equity market. These types of trades can also be settled in different OTC venues or on public exchanges using iceberg orders). Along with this aim to reduce market impact, Banks identifies other drivers to the rise of dark pools such as confidentiality, cost savings and profit opportunities with related price improvements. These four main points can be seen as advantages brought by dark pools to both buy side and sell side traders. Although the main application of dark pool traces back to equity markets, dark pools (specifically decentralized dark pools) also bring advantages to the specific microstructure of cryptocurrency markets, as will be outlined in later chapters.

After defining the Buy Side, the Sell Side and Trade Facilitators in the markets, some additional trader categorizations are in order and will be outlined starting with the next section.

[3] Price displacement would be the result of traders becoming aware of the intention to post a large block trade by a deep-pocketed trader. After hearing about, say, a large potential buy block trade, traders would try to jump ahead of the trade by posting market orders and effectively running the price up, creating a price displacement that would be unfavorable to the block trader and would worsen the entry price compared to the scenario where the block trade information remained undisclosed.

1.3.4. Informed and Uninformed Traders

The secular debate in the field of Finance and Economics revolves around market efficiency. Market efficiency deals with the degree of information available to market participants and the degree to which prices internalize such information. By construction, the field of market microstructure cannot be aligned with full informational efficiency, as microstructure focuses on the very frictions causing prices not being 100% informationally efficient at all times. One of the most popular assumptions in basic economic models is agent rationality and perfect knowledge: this is unfortunately too simplistic for financial markets where valuable information can be said to be the key to alpha generation. In a world where alpha generation is possible, albeit difficult, discrepancies in information among market participants must exist. We can then distinguish between informed and uninformed traders, where the former enter trades in lieu of their information actually being relatively valuable and not priced in yet (an intuitive example would be traders having inside information regarding sensible topics such as company earnings, interest rate decisions, etc); and the latter enter trades either knowing of having sub-par information (it's the case of hedgers that only seek protection from a risk stemming from activities that outscope the markets and, intuitively, of gamblers, who ideally trade just for the sake of taking on risk), or believing to possess superior information that is actually, unfortunately, either already priced in or effectively irrelevant. Considering that some uninformed traders may enter the profitable side of a trade out of sheer chance (with their counterparty most likely being another uninformed trader), informed traders have the opportunity of generating alpha (that is, generating excess returns) by taking uninformed traders as their counterparty. The market would then fulfill its feature of wealth transfer from uninformed to informed traders. Uninformed traders are considered the largest contributor to "noise" in price discovery, as they don't tend to trade based on fundamentals or any relevant news or analysis related to the "fair" asset price. Because of this, they are usually referred to as "noise traders". Mind: informed traders would not have any possibility of settling a transaction if uninformed traders did not exist, and this brings directly to the next categorization: liquidity providers (or makers) and liquidity takers.

1.3.5. Liquidity Makers and Liquidity Takers

The vast majority of global financial markets are structured as double auction markets. A double auction market is characterized by the matching of a buyer's bid price and a seller's ask price, with the trade proceeding at that common price. For a buyer to buy, a seller willing to sell needs to exist. Vis-a-vi, for a seller to sell, a willing buyer must exist at the same price. The availability of one or multiple counterparties willing to enter a transaction for a specified number of instruments at a specific price influences the grade of liquidity in a market. A market is relatively more liquid when a trader can place an order in the market and that order easily finds a matching counterparty. Liquidity, as will also be touched upon later, is crucial for the orderly functioning of the markets and for price efficiency. An illiquid market, where trades are difficultly matched, will deliver more volatile and less "accurate" or "fair" prices. Thus, for a market to function as intended, liquidity makers are crucial. Markets can envisage dedicated liquidity makers, such as the previously presented market makers, whose objective is to provide liquidity to traders so that the price follows an orderly evolution. However, market makers aren't necessarily the most important liquidity makers: they sure are during brief periods of volatility and illiquidity, when either no sellers or no buyers are willing to take on any transaction, but still the "usual" liquidity makers are normal traders, when they place limit orders[4]. In fact, most stock exchanges and financial markets today employ limit order book technology, where no designated market maker is needed, and limit orders make up most of the liquidity provision (Brogaard et al., 2019). When looking at "normal" traders, that is, traders that do not belong to the previously presented sell side, liquidity makers and takers can be seen, respectively, as patient and impatient traders. Liquidity makers (patient traders) do not seek immediate execution and as a result post limit orders at price levels they're interested in, waiting for the price to eventually reach those levels and at the same time accepting the risk of not getting filled and thus not being able to participate in the market with an open position. Liquidity takers (impatient traders) do seek immediate execution and as a result post market orders at the current price level, getting immediately filled at that price if the appropriate amount of limit orders were waiting to be filled. If this does not happen, price will creep up or down (depending on wether the impatient trader is buying or selling, respectively) towards the next price level at which limit orders can satisfy the liquidity taker. Further focus on order types and what they mean for traders will be tackled in section 1.4.

[4] Types of orders will be presented in section 1.4.

1.3.6. Investors, Speculators, Arbitrageurs

Leaving aside the figure of gamblers presented above, as it configures an extreme case more frequent in other markets such as sports betting (but still being present, although to a lesser degree, in financial markets), traders can be split into three main categories based both on their objectives and their time horizon.

Investors tend to have the longest time horizon among the three. They usually stick to buying assets they believe are undervalued, with the aim of selling them for a higher price in the future, assuming that prices will converge to the true "fair" value of the asset (and that the fair value is close to the one resulting from the investor's analysis). Investors do not trade often, they might rebalance their portfolio once or twice a year and add/remove assets from their portfolio throughout the year based on their analysis. Their time horizon is usually measured in decades.

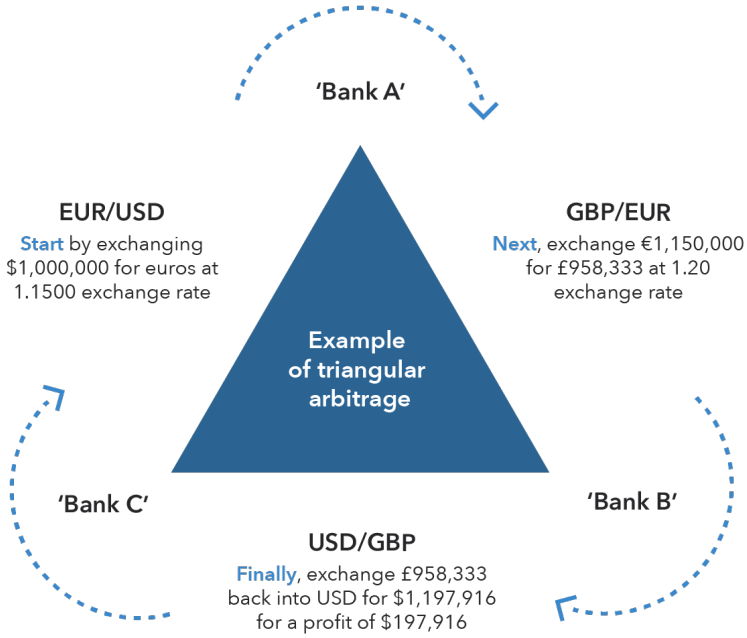
Speculators are focused on shorter term (albeit still directional) profits and can take both the long (buying) side and the short (selling short) side of a trade. Investors will only sell an asset that is already in their portfolio, while speculators, in search of profits, might as well "sell short" assets that are not in their possession by borrowing them and promising to buy them back and return them, pocketing the difference if the price decreased in the meantime, or suffering a loss otherwise. Regardless of long or short positions, speculators can hold a trade for weeks or months (position trading), some days to a week at most (swing trading), some hours to one day at most (day trading) or seconds to minutes (scalping).

Arbitrageurs, differently from investors and speculators, do not take single directional trades. Their objective is pocketing a profit measured by the price difference between similar assets or between the same asset traded on different exchanges. Arbitrageurs are particularly active in the Foreign Exchange market (and in turn in the cryptocurrency market, due to the similarities in nature of the instruments traded) and deploy a variety of strategies, among which (Biais et al., 2016):

- Classic arbitrage: exploits the differences between market prices and prices implied by "no arbitrage" conditions. One example is cross currency arbitrage (also called triangular arbitrage), which takes advantage of discrepancies between the actual current quote for a currency pair and the value implied by the current quote of two pairs from which the initial currency pair can be backed out. An example of triangular arbitrage, both in the FX and crypto market, is presented in Figure 4.

- Latency arbitrage: taking advantage of time lags between execution of market-moving trades and the update of FX quotes by market-makers. This intuitively requires strong hardware architecture with fast internet connections and considerable computing power, as many other arbitrage strategies actually do too.
- Liquidity imbalance strategies: detecting order book[5] imbalances in currency pairs and pricing discrepancies between different trading platforms.
- Complex event processing: exploiting properties of foreign exchange rates such as momentum, mean-reversion, correlation with other exchange rates or reaction to news releases.

Figure 4 – Example of triangular arbitrage when the exchange rates for EUR, GBP and USD are not perfectly equivalent



Source: IG Australia

These arbitrage strategies cannot intuitively be carried out successfully by humans, as arbitrage opportunities are extremely short lived and often complex to identify in short amounts of time. Technological advancements have broadened the capabilities of computers up to the point of being able to create and automate complex strategies, effectively eliminating human error and impressively improving timeliness of execution. This brings to a further classification that is mostly peculiar to our days.

[5] The order book will be introduced in section 1.5.

1.3.7. Discretionary and Systematic Traders: from Manual to Algorithmic

Traders usually join the markets multiple times during their life. They might trade based on different motives each time ("gut" feelings might drive uninformed traders to join the markets; informed traders might join the markets a first time because they had insider information on a company's earnings, and a second time they could join based on an anticipated leak they received regarding a possible M&A transaction carried out by a different company: all of these are examples of discretionary behavior), or they might follow a well-defined, systematic checklist that will make or break their entry in the market based on how many points of the checklist are satisfied. This latter systematic approach is typical in technical traders employing graphical and technical analysis to identify market trends and possible price levels of interest. It would not be exceptional to hear a technical trader that to enter a long position on an asset, "the RSI should be in the oversold region, price should be touching the lower Bollinger Band and we should be at the 61.8% Fibonacci retracement level considering the latest bullish impulse". This is a perfect example of a checklist that a systematic trader swears by and would never break. Now, since the checklist is always the same, and the matter is to identify assets currently ticking all the boxes, little sense remains on trading manually. The advent of computers and the ever-increasing computing power available to individuals have enabled systematic traders to automate their strategies via algorithms, configuring a new frontier called Algorithmic Trading, of which High Frequency Trading (HFT) has been one of the first and probably best known implementations (Debelle et al., 2011). HFT employs computers and fast internet connections to identify entry points for trades that could last only milliseconds. Arbitrage strategies have benefited the most from HFT because of the very nature of those arrangements, extremely difficult for humans to carry out successfully.

1.4. Types of Orders and Price Discovery

Price discovery is the dynamic process by which market prices incorporate new information (Yan & Zivot, 2010), and its efficacy is strongly dependent on factors such as trading mechanisms, market liquidity, and the prevalence of asymmetric information. The canonic view of price discovery is that trades reveal investors' private information, while market maker quotes reflect public information (Glosten & Milgrom, 1985). This section will focus on one crucial component, determining how traders can enter the markets: types

of orders. The importance of having multiple types of orders is evident for the efficiency of price discovery, as traders indirectly signal their intentions (and a result, the directionality stemming from their information) via the type of orders they submit, at least on public exchanges, where this information is available. A brief exploration of the possible impact of different order types on the price discovery process will also be provided.

1.4.1. Market Orders

As already discussed in section 1.3, market orders are preferred by impatient traders that want to enter the market with immediacy. In fact, market orders guarantee a fill to those who submit them, without however guaranteeing a specific execution price. Thus, market orders prioritize execution speed (and execution certainty) over price certainty. Market orders come in two types: market buy orders and market sell orders. The former allow traders to buy an asset immediately, the latter allow traders to immediately sell (or sell short, if the asset is not owned before the sell order is submitted).

The potential difference between the current price quote and a market order's execution price depends strongly on liquidity. In markets employing bid and ask quotes, the difference between these two quotes (also called bid-ask spread) is one of the most telling indicators of liquidity. A wider difference between the current best bid (the highest price a buyer is willing to pay for an asset at a given time) and best ask (the lowest price at which a seller is willing to sell an asset at a given time) indicates a higher degree of illiquidity (narrower bid-ask spreads, on the other hand, typically indicate higher liquidity). If a trader wants to buy 1000 units of a specific instrument trading at a price of, say $10, the execution price of a market order submitted now would be $10 if and only if at least 1000 units of the instrument are available for purchase at that price level on the ask side. If, as it often happens for relatively large orders, the number of counterparties willing to take the opposite position is insufficient, the order will be filled at progressively higher prices (lower prices in case of a sell order). Assume, for instance, that the current situation sees a total of 1000 instrument units worth of limit sell orders resting at various ask price levels: 500 units at $10.00 (the current best ask), 300 units at $10.05, 150 units at $10.07 and 50 units at $10.10 (price in this imaginary market moves in 1 cent increments, with some price level evidently hosting no sell orders). Assuming our 1000-unit buying trader sweeps all orders at all prices levels before any other trader has the chance to do so, the final execution price will be:

$$\frac{500}{1000}10.00 + \frac{300}{1000}10.05 + \frac{150}{1000}10.07 + \frac{50}{1000}10.10 = 10.03$$

$0.03 more than the price at the time of submission. This cost ($30 for the whole order), technically called slippage, is paid by our impatient trader in exchange of entering the market immediately and is a function of liquidity. Very liquid markets tend to host more impatient traders since the cost of market orders in terms of slippage can be negligible or even nill. However, one cost will always be paid by impatient traders submitting market orders: the bid-ask spread. In very liquid markets this spread can be very narrow, but will still exist and consist in a cost for submitting market orders (Brogaard et al., 2019).

A particular type of market order is the market-if-touched order. Roughly resembling a limit order (see below), market-if-touched orders are placed away from the current quote. This allows flexibility to traders that might not have the physical possibility of being ready to trade when price reaches their desired level. These orders become market orders and are filled exactly as a normal market order (with all the pros and cons) if and only if price reaches the level specified by the trader at the time of submission.

1.1.2. Limit Orders

Opposite from market orders paying the bid-ask spread, limit orders, submitted by patient traders (liquidity makers/providers) receive that very bid-ask spread. Hence, the bid-ask spread can be seen as a cost for immediate execution and a revenue for liquidity provision (Brogaard et al., 2014; Brogaard et al., 2019).

Limit orders allow traders to enter the market at a specified price level, without the need of physically waiting in front of a computer for the price to reach it. Limit orders are diametrically opposed to market orders, as they prioritize price certainty to execution certainty. Limit buy orders can only be placed at or below the current quote, while limit sells can only be placed at or above the current quote. Once placed, a limit order will be filled only if price reaches the level where the limit order lies. Limit orders can stay in force indefinitely unless manually cancelled if submitted as Good-Til-Cancelled (GTC) orders, or can have a pre-defined lifespan beyond which they are automatically cancelled (it's the case, for instance, of Good-Til-Day orders, which are automatically cancelled at the end of the trading day if they don't get filled). Limit orders thus allow the flexibility of getting filled when the trader is "away from the charts" or simply not paying attention when price reaches the level of interest. Since limit orders give certainty regarding the execution price, many

traders use them in place of market orders and place them at the current market price to avoid paying the bid-ask spread and being filled far from the desired price level. Limit orders are usually matched by contingent market orders, that may be placed for the same number of instruments, more, or less. In the first two cases, the limit order gets fully matched and the fill amounts exactly to the number of instruments the patient trader offered to buy and sell at the desired price. In the third case, assuming only one market order touches the limit order's level before seeing price retrace lower, a "partial fill" would materialize: the limit order will be matched at the desired price for the number of instruments included in the market order, leaving a portion of the limit order unfilled. This scenario is included in execution uncertainty: the patient trader in this case will enter the market at the desired price, however with a smaller order than anticipated. The unfilled portion of the limit order will remain in force until cancelled or filled by other market orders if price was to rally back to that price level. To avoid the risk of partial fills, traders can use All Or None (AON) limit orders, which explicitly require a full fill at the specified price level or no fill at all in case the opposite order's size is insufficient.

1.4.2. Impact of Market and Limit Orders on Price Discovery

Brogaard, Hendershott and Riordan (Brogaard et al., 2019) discuss and analyze the impact on both price and the price discovery process by market and limit orders, observing IIROC data on trading activity in recognized Canadian equity markets. They find that market orders tend to have relatively higher impact on price compared to limit orders. However, since limit orders are far more numerous than market orders (only 5% of the analyzed messages were market orders), the former contribute more to the price discovery process. Interestingly, the authors find significant difference in price impact and discovery contribution when classifying orders as HFT and non-HFT (as in "(not) submitted by high frequency traders"). Specifically, they find that HFT limit orders contribute up to twice as much to price discovery compared to HFT market orders. HFT limit orders are also responsible for twice as much price discovery than non-HFT limit orders (30% of price discovery is attributable to HFT limit orders, 15% to non-HFT limit orders). On the opposite, non-HFT limit orders contribute less to price discovery compared to non-HFT market orders. Non-HFT market orders also contribute more to price discovery than HFT market orders. Overall, HFT limit orders contribute the most to price discovery, while HFT market orders contribute the least.

1.4.3. Stop orders

Stop orders function similarly to market-if-touched orders. The main difference is that while market-if-touched buys can only be placed at or below the current quote, buy stops can only be placed at or above the current quote; while market-if-touched sells can only be placed at or above the current quote, sell stops can only be placed at or below the current quote. This feature makes stop orders extremely popular with traders that want to manage risk related to open positions. If a long (buy) position is currently open and the trader wanted to set a maximum amount for the potential loss this position will suffer, a sell stop order could be placed some levels below the entry price. This way, if price was to fall lower, the trader's position would automatically close as a result of the sell stop being triggered, limiting further losses if price was to continue lower. A buy stop above a short entry point would serve the same purpose for short positions. This use gave stop orders the well-known designation of stop loss orders. Mind, stop orders can also be used as entry orders, rather than exits. Traders wanting to buy at a price level that is above the current quote can use a buy stop, while traders wanting to sell below the current quote can use a sell stop. However, it is important to highlight that stop orders do not guarantee price but guarantee execution. This, together with what outlined above, makes them a stand-alone type with components of both market-if-touched and limit orders.

1.4.4. Bracket Orders: Coexistence of Stop and Limit Orders

Limit and/or market-if-touched orders, differently from stop orders, can be used as "take profit" orders, capping the maximum amount of profit a trader can get from a position. A sell limit would be the take profit for a long position, a buy limit would be the take profit for a short position. Limit orders used as take profit guarantee price, but don't guarantee full execution. To make sure that the position will be fully closed at a given price, traders might want to use a market-if-touched order, being wary of slippage in case price moves against the position right after hitting the take profit price. Composite orders that already include either a market or a limit order as an entry order, plus a stop and a take profit, are referred to as "bracket orders". Most brokers only allow the use of limit orders as take profits and don't offer the possibility of using market-if-touched orders instead. Figure 5 depicts a limit buy bracket order positioned below the current quote for the S&P 500 futures contract with September expiration, ticker ESU2023.

Intuitively, since many traders approach position risk management in the way depicted by Figure 5, some price areas will be particularly heavy either in stop or limit orders. This is crucial for price discovery, as sudden spikes in volatility and price excursions can happen at specific price levels solely because of the stop or limit orders resting there. Since many traders look at price charts as part of their decision-making process, relative lows and relative highs historically painted by price tend to have relative more weight in traders' decisions compared to other areas. Relative lows can be seen in Figure 5 for example around the 4405 or the 4360 price levels, respectively in early July and late June, while relative highs can be seen around 4480 (early June, start of July) or 4640 (late July).

Figure 5 - A Limit Buy bracket order. In green, the take profit (a sell limit); in orange, the stop loss (a sell stop). Based on the portfolio's equity, risk has been managed by limiting the potential loss to a maximum of $2750. This is a simulated bracket order on August 7th, 2023

Source: tradingview.com

These levels tend to host large numbers of stop loss and/or take profit orders. Looking at the price structure in Figure 5, one could imagine many traders being long at the moment, since price has been rising substantially over the last couple of months. It is then reasonable to expect a large amount of stop losses right around the orange line or right below it, as a risk management measure adopted by traders. If price were to slip down at or below that area, many sell stop orders will be triggered and, if buying pressure won't be sufficient to offset all the sell stops, price will inevitably explore lower levels. Those areas, called "liquidity

pockets" by some, can also be useful to large traders wanting to post large orders. A very deep-pocketed trader might be willing to buy a large number of contracts around 4400, effectively offsetting the selling pressure from stop losses and mitigating the downward price impact.

1.5. Liquidity, Order Flow and Market Efficiency

After looking at who populates financial markets and how traders can interact, we now turn to one of the most crucial relationships analyzed in microstructure studies: the intertwining of liquidity and order flow, with relative implications on price and market efficiency.

Since this thesis will further delve into cryptocurrency markets, this topic will be presented for electronic exchanges, where order matching and settlement are automated and transactions can happen millisecond to millisecond.

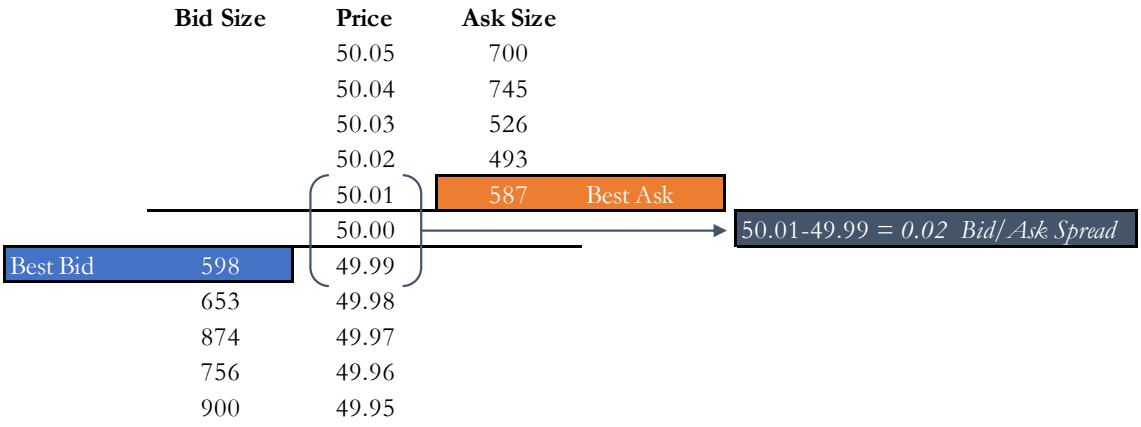
1.5.1. Order Flow and the Order Book

Order Books today are mostly electronic databases created and updated by brokers and exchanges to keep track of unfilled orders in real time (Harris, 2003). With access to financial markets being democratized, some exchanges receive orders in volumes that would be impossible to keep track of manually. This brought the greater part of public exchanges to employ an Electronic Open Limit Order Book (EOLOB): a real-time database accessible (open) to all exchange traders, hosting all unfilled limit orders. Open books have been found to bring significant microstructural differences compared to closed books, where the set of limit order is not publicly available (Brown et al., 1999). "Opening the book" tends to benefit impatient traders and remove the informational advantage patient traders have as a result of being the market's liquidity providers, overall resulting in more informative prices (Baruch, 2005; Chaboud et al., 2021). Unfortunately, limit orders displayed on the open book are not necessarily exhaustive of the activities carried out in a specific venue: some public exchanges, by hosting separate dark pools (one example would be Euronext-owned SmartPool), allow for posting of undisclosed limit orders, which clearly won't be displayed on the LOB; other exchanges might on the other hand restrict access to the order book (Harris, 2003).

As displayed in Figure 6, LOB activity is bound by a predefined price and volume grid with two "resolution parameters": tick size (the smallest possible increment in the quote, e.g. $0.01) and lot size (the smallest tradable amount, e.g. one unit) (Bouchaud et al., 2018).

LOBs display a range of price levels together with bid and ask sizes. These sizes represent the amount of active limit orders at each level. As highlighted earlier, limit orders can only be placed above the current price if short and below if long. This is why the two columns are only half complete. The Bid column shows the amount of active limit buy orders, while the Ask column shows the active limit sell orders. A market buy order would then be executed at the lowest available ask price (matching the market buy order with a limit sell order), while a market sell order would "hit the bid" at the highest bid price available (matching the market sell order with a limit buy order). The difference between the highest bid and the lowest ask is the previously mentioned bid-ask spread.

Figure 6 - A Limit Order Book



Original visualization via MS Excel

When matched, limit orders are immediately removed from the order book, updating the available bid and ask volume. Market orders are not effectively displayed on the LOB, as they match directly upon arrival. By their own nature, the same can be said of stop and market-if-touched orders. Active limit orders are held in a queue until they are either cancelled by their respective trader or executed against an opposite order. Order matching mainly follows two precedence rules (Harris, 2003): price precedence first, then time precedence. Meaning, orders submitted at the current price are executed first. Out of two orders submitted at the same price, the one submitted first is executed first, in a first-in, first-out fashion.

The stream of orders hitting the order book is called order flow. Order flow dynamics have important effects on price discovery and overall market efficiency (Bouchaud et al.,

2018; Lee et al., 2004). One significant difference between LOB systems and dealer systems concerns, in fact, the price formation process: in absence of designated dealers, price discovery and liquidity provision in LOB systems are self-organized processes driven by the submission and cancellation of orders by traders, with patient market participants (liquidity providers) acting in place of traditional market makers[6]. Order flow then is the backbone of price discovery: with the same basis of supply and demand, a flow of orders at the current price characterized by relatively more buyers than sellers will push the price upwards, vice versa for scenarios where sellers outweigh buyers. Intuitively, order flow from informed traders is the instrument through which price internalizes new information. Sometimes, price also internalizes sentiment, expectations, emotions and other dimensions that can still be traced back to order flow. Not only price, order flow also directly influences liquidity and is itself the source of liquidity. Imbalances in the LOB between buy and sell orders (also called order flow imbalances) can be measured by observing market orders hitting both sides of the LOB (the manifestation of market orders hitting the LOB would be a decrease in the bid or ask sizes): an imbalance is detected whenever the number of market orders hitting the bid differ significantly compared to the number of orders hitting the ask. This type of order flow imbalance has been time and time again found to have significant predictive power in explaining short term returns right after the imbalance appears - see (K. Xu et al., 2018) for a review of the literature on order flow imbalances.

Order flow is deeply tied with market liquidity and price discovery. Market microstructure could go as far as saying that order flow is the very archetypal source of liquidity and price movements, as it is in essence the continuous series of buy and sell orders flooding the markets at every instant during the trading day. Bouchaud (2009), however, warns about this last point of view, as it could easily bring to the internalization of information in prices being "a mere self-fulfilling prophecy which would occur even if the fraction of informed traders is zero".

1.5.2. Market Depth and Other Liquidity Measures

Liquidity, commonly described as the degree of ease with which an asset can be bought or sold, is of great interest to traders that want to appropriately assess the risks of trading a

[6] This dichotomy between patient and impatient traders, when referring to order flow, can be renamed as Passive traders (posting limit orders) vs Aggressive traders (hitting the book with market orders).

specific asset and want to avoid bad pricing, spikes in volatility or, at worse, lock-in situations where closing a position is virtually impossible.

Harris (Harris, 2003) describes three main dimensions of liquidity:

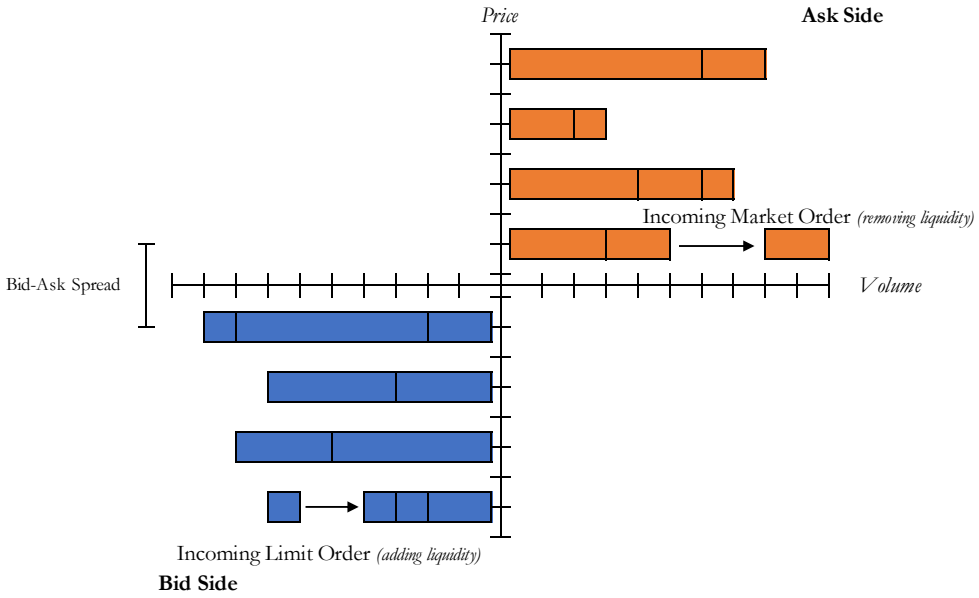
- Immediacy: most commonly used to describe the concept. It refers to how quickly a trade can be arranged, holding size and transaction costs equal (mainly focusing on market spreads rather than commissions in this case).
- Width (market breadth): the cost of arranging a trade of a given size. The most popular measure of market width is the bid-ask spread.
- Depth: the size of a trade that can be arranged at a given cost, measured in units available per price level at a specific time.

In his influential 1985 paper, among the three dimensions above, A. Kyle also highlighted Resiliency (Kyle, 1985): the speed at which prices recover from an uninformative shock.

Many brokers, platforms and exchanges offer visualization tools to better understand the order book, often focusing on visualizing market depth. Figure 7 shows a stylized visualization of market depth by using bars indicating the number of units available for trading at each price level. Figure 8 shows two real world examples: Binance's order book for the BTC/USDT pair and Webull's order book for Gamestop stock (ticker: GME). Some platforms may offer only "top of book" data, referring only to the current best bid and best ask, and may ask subscriptions for order book data at more than one price level (usually referred to as either "Level 2" data or "Depth of Market" (DOM) data).

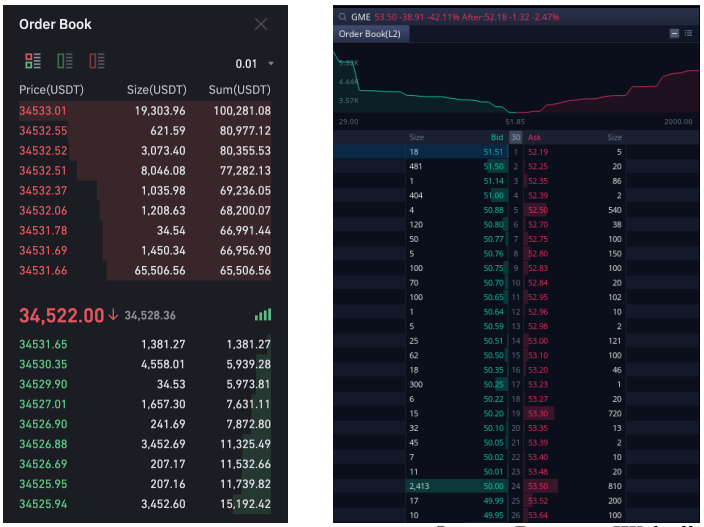
Another popular indicator used to gauge the liquidity of an asset is trading volume, representing the number of units traded in a given timeframe. Intuitively, trading volume and liquidity ought to be related. Counterintuitively, however, Johnson (T. C. Johnson, 2008) finds that the two do not appear to be related at all (and this finding is in line with observations from the US bond and stock markets). Rather, volume shows some degree of positive correlation to liquidity variance (liquidity risk). Thus, trading volume does not appear to be an appropriate indicator of liquidity. It can, on the other hand, indicate the general degree of liquidity risk for an asset, as liquidity risk increases with trading activity (i.e., trading volume).

Figure 7 - Limit Order Book. A market buy order is about to hit the ask and a limit buy order is being added to the last visible bid level



Original visualization via MS Excel

Figure 8 – Two examples of LOBs



Source: Binance, Webull

One direct consequence of increased or decreased liquidity is price impact. Price impact refers to the correlation of price changes to incoming orders (Bouchaud et al., 2018). One of the most known measures of price impact is Kyle's λ coefficient (Kyle, 1985), with 1/λ measuring the order flow necessary to induce a one dollar increase or decrease in prices. As Bouchaud et al. highlight, λ is inversely proportional to market liquidity. Thus, thinner (less

liquid) markets tend to showcase higher price impact in each executed trade, hindering price efficiency and increasing trading costs in the form of slippage.

1.5.3. Liquidity and Market Efficiency

Fama, (1970) famously outlined three main forms of efficiency in capital markets: weak form efficiency sees prices only reflecting data on historical information, with historical prices as the only information set incorporated, effectively negating the possibility for alpha generation via graphical/technical analysis alone; semi-strong form efficiency sees prices incorporating all publicly available information, leaving small room for excess returns, as neither technical nor fundamental analysis would be successful in generating information that is not already priced in – only insiders can systematically succeed; finally, strong form efficiency sees prices already incorporating all historical, public and private (insider) information, effectively negating the possibility of alpha generation.

Fama's concept of efficient markets focuses on the lack of return predictability as a criterion for efficiency. However, when it comes to market efficiency, microstructure literature is more concerned with the amount of private information represented in prices. Intuitively, increased liquidity in markets due to secular events (such as NYSE's reduction in the minimum tick size from one eighth to one sixteenth in 1997, resulting in narrower bid-ask spreads), may incorporate private information more efficiently: in fact, related reductions in trading costs may stimulate trading based on information about fundamentals - see Admati and Pfleiderer, 1988 in Chordia et al. (2008); Goldstein and Kavajecz (2000).

Chordia et al. (2008) explored whether fluctuations in liquidity occurred between the 90s and the early 2000s were related to changes in the degree of market efficiency for NYSE-listed stocks, and in particular whether the systematic increase in liquidity observed brought improvements in terms of efficiency. They found that order flow's forecasting power on stock returns diminished on days characterized by higher degree of market liquidity. Moreover, they observed that the reduction in the minimum tick size from one eighth to one sixteenth arranged by NYSE in 1997 improved efficiency through a reduction in bid-ask spreads. Goldstein and Kavajecz (2000), however, specify that this reduction favored specifically impatient traders trading relatively smaller orders. The resulting generalized decrease in market depth resulting from the increased order book granularity hindered traders submitting large orders in stocks with less trading activity, even more so if those stocks where low priced. Chordia et al. also found that prices behaved closer to a random walk after the minimum tick size decrease.

Liquidity fosters price efficiency also by reducing transaction costs in the form of narrower bid-ask spreads and lower slippage.

This section concludes the introductory tour in the realm of market microstructure. Although the principles and concepts presented are indeed generally applicable to a variety of markets, each one has specific peculiarities that make each market's microstructure unique. This thesis will focus on cryptocurrency market microstructure and how it can be exploited to generate alpha, hence the next chapter will expand on Chapter I by introducing microstructure specificities that characterize cryptocurrency markets.

Chapter II Cryptocurrency Market Microstructure

2.1. Introducing Cryptocurrencies

Cryptocurrency refers to a type of digital asset specifically designed to be used as a medium of exchange. Unlike traditional (physical and/or electronic[7]) currency, cryptocurrency is not controlled by any central institutional authority like a central bank. Rather, cryptocurrencies employ various types of decentralized transaction validation procedures based on cryptography. Bitcoin, the most well-known cryptocurrency, relies on a decentralized network of computers to validate transactions posted on a publicly accessible, immutable database called distributed "ledger". The use of these ledgers has become the defining criterion for "blockchain" technology, the scope of which may go well beyond transaction validation. The value of each cryptocurrency is mainly set through transactions, with the contribution of algorithms managing the available supply (Tredinnick, 2019).

Bitcoin is the first modern cryptocurrency as we know it, effectively dating the birth of cryptocurrency on January 3, 2008, when the pseudonymous Satoshi Nakamoto published Bitcoin's white paper[8] titled "Bitcoin: A Peer-to-Peer Electronic Cash System" (Nakamoto, 2008). One could argue, though, that the origins of cryptocurrency and blockchain technology date back to the 80s and 90s. In fact, 1983 saw the publishing by David Chaum of a paper titled "Blind Signatures for Untraceable Payments", outlining what he defined blind signature cryptosystems (Chaum, 1983). Then, 1990 saw the birth of a digital currency that could be seen as analogous to Bitcoin in some way: eCash, developed by company DigiCash, founded by Chaum himself (Reiff, 2022). Other digital solutions were created before the advent of Bitcoin, namely E-gold (1996), Hashcash (1997), Bit Gold (1998), B-Money (1998) and RipplePay (2004). None of them survived, including eCash. As of today, the oldest surviving cryptocurrency is in fact Bitcoin (Reiff, 2022).

[7] Electronic and virtual currency are two different concepts: electronic currency refers to the non-physical version of fiat money, having legal tender (bank wires, debit/credit card payments, etc), while virtual currencies are usually unregulated forms of currency not having legal tender, but used and accepted in the scope of a specific virtual community (an example could be cryptocurrency). Together with other concepts, electronic and virtual currencies tend to be hosted under the "digital currency" umbrella.

[8] "White paper" is a general designation for informational documents published by companies or organizations to present the features of a solution, product, or service that they plan to offer. DAOs (Decentralized Autonomous Organizations) make extensive use of white papers to present the technology underlying their new cryptocurrencies or tokens.

By nature, cryptocurrency sets out to eliminate the "middle man" in financial transactions (both payments and micropayments) by enabling borderless, immediate, cheap and secure payments verified by peer-to-peer mechanisms (Härdle et al., 2020). As a result, cryptocurrency is also a valuable asset to foster financial inclusion for the 1.4 billion unbanked adults worldwide (Demirgüç-Kunt et al., 2022). This disintermediation potential can also result in gains in terms of market efficiency (OECD, 2020).

Being decentralized, cryptocurrency eliminates the need for banks and traditional intermediaries. Coins are stored in wallets, software or hardware tools holding cryptographic information related to one's holdings. Hardware wallets use USB interfaces to interact with computers and connect to the internet, while software wallets can either be installed or accessed via a web browser. Software wallets are usually classified as custodial and non-custodial: the former is usually offered by online platforms and is responsible for safeguarding user funds, as the private key (more on this later) is held by the wallet provider; the latter gives full control of the private key to the user, rendering the owner of the funds the sole responsible for their integrity (quite like an actual wallet holding physical money). Wallets can interact with each other like bank accounts, and interaction is made possible by public and private keys. Keys are alphanumeric sequences (25-36 characters long) uniquely identifying a wallet. The public key is used to generate a wallet address that is meant to be shared with transaction counterparties, akin to a bank account number, while the private key is used to complete the transaction "signing" process and can't clearly be shared with anyone if the wallet owner's objective includes maintaining control over the funds. The public key is derived from the private itself through complex and difficultly reversable computational operations.

Developments in the world of crypto and blockchain has brought to a phenomenon of asset tokenization via distributed ledger technologies or DLTs (OECD, 2020). OECD's report on asset tokenization highlights some key features of tokens that are also characteristic of cryptocurrencies[9]:

- Immutability: recorded blockchain transactions are virtually impossible to alter, delete or forge. This is achieved through cryptographic hashing and consensus mechanisms that safeguard the integrity of the transaction history. A direct result of this system is the lack of a single-point-of-failure and thus relatively stronger security.

[9] Tokens are slightly different from cryptocurrency coins: coins function on top of their dedicated blockchain, while tokens rely on blockchains that were not necessarily developed and deployed for them in particular.

- Transparency: as outlined above, transactions are recorded on public ledgers. These ledgers can be accessed by anyone to verify and audit transactions. One can independently track the flow of funds, enhancing trust and reducing the risk of misinformation.
- Programmability: Smart contracts are self-executing agreements with predefined conditions and outcomes that introduce programmability for those agreements that include quantifiable requirements/goals/covenants/etc. They automatically execute and enforce terms when specific conditions are met. Decentralized applications (DApps) can automatically transfer funds, verifying identities and more, without the need of intermediaries.

Financial markets can benefit from the data integrity, immutability, and security inherent in many blockchain-based solutions, as well as automated auditability. Furthermore, DLT-based security registries may provide improved transparency and a clear record of beneficial ownership at any moment in time. Registars/transfer agents may therefore be rendered obsolete or redundant (OECD, 2020).

2.2. New Market Participants Shaping Market Structure

Disintermediation and the peer-to-peer paradigm introduce the need for new roles, mainly related to the audit and verification of transactions. Ledger transactions, in the realm of blockchain technology, are verified through so-called consensus mechanisms. The consensus mechanism used may vary from blockchain to blockchain and to this day new consensus mechanisms continue to be proposed, surrounded by active debates on which are the most appropriate. Table 4 showcases a list of the currently existing consensus mechanisms. These mechanisms rely on network participants being incentivized to validate transactions ("adding blocks to the ledger") by assigning rewards.

Based on the consensus mechanisms in Table 4, at least three new market participants make their appearance:

- Miners: the word "miner" can refer to both the computer carrying out the mining computation and the owners of those computers. Miners (individuals or entities) invest in computing power, usually in the form of hardware such as external graphic cards or GPUs to maximize their chance of solving hashing problems and being assigned the possibility of adding blocks of transactions to the ledger. They are mainly active in PoW networks. Bitcoin, specifically, features a deflationary mechanism that

strongly relies on miners called halving. Halving consists in cutting in half the number of bitcoins generated for each solved block. This lowers supply, effectively increasing one bitcoin's price and reducing its inflation rate, assuming demand remains steady or increases. Halvings happens once every 210,000 validated blocks[10]. Thus, miners indirectly control Bitcoin's supply and inflation rate, and their participation to the network relies on the rewards being economically viable even after each halving, considering hardware and power costs. If Bitcoin's price were to decrease, due to market conditions, to the point where rewards aren't incentivizing miners anymore, the whole consensus mechanism could come to a halt, stopping block validation and effectively blocking transactions from occurring. The worst that has happened up until now are decreases in the hash rate (rate of block validations) in periods of low prices and two downtimes related to errors in block validation, however no overall stop as described above has been experienced in Bitcoin's blockchain so far.

- Validators: in a PoS blockchain environment, validators take the place of miners. Validators need not invest in hardware or incur in costs other than the ones related to acquiring the blockchain's native cryptocurrency. Once acquired, the cryptocurrency is staked on the network and used as proof that the network participant has some "stake" in the project, or "skin in the game". Based on the size of their stake, validators become eligible for adding transaction blocks to the ledger.
- Stakers: all validators are stakers, however not all stakers are validators. A staker is, in general, a network participant holding all or part of their currency on the network, usually in exchange for interest payments. Staking is somewhat akin to owning certificates of deposit in traditional banking: network participants can decide the staking term (which can go from days to years) and collecting the staked amount before the term results in the loss of the interest component matured up until that moment. These checks can be implemented in automated smart contracts, without the need for an intermediary checking whether the network participant has withdrawn the funds before or after the term specified in the contract.

Additionally, other participants not necessarily related to consensus mechanisms come to mind. First, "whales" are entities or individuals holding enormous amounts of (fiat equivalent) cryptocurrencies. The presence of whales in cryptocurrency markets sees as a result volatility spikes when large orders are submitted. This is mainly because dark pools do exist in cryptocurrency markets, but function in a quite different fashion from traditional

[10] Source: https://buybitcoinworldwide.com/halving/

dark pools and some whales might prefer entering in the open market as a result. Lack of regulation, as will be outlined later, contributes to this intuitive source of inefficiency.

Table 4 - Consensus Mechanisms

Mechanism	**Description**	**Related Blockchains**
Proof of Work (PoW)	Lets miners add a new block to the ledger based on the computation (work) done to correctly solve a mathematical hashing problem	Bitcoin, Ethereum (until 2022), Litecoin
Proof of Stake (PoS)	Lets validators add a new block to the ledger based on them locking up (staking) an amount of currency in the network. Only staking participants can be selected as validators. Requires less computing power compared to PoW.	Ethereum (since 2022), Polkadot, Cardano
Delegated Proof of Stake (DPoS)	Additional layer to PoS, where staking participants vote for delegates that will in turn be randomly selected to add a block.	EOS, Tron
Proof of Importance (PoI)	Assigns blocks based on an importance score, computed considering reputation in the network and transaction quality. Avoids richer participants monopolizing the verification process as in PoS.	NEM
Proof of Capacity (PoC)	Assigns blocks based on proof of disk storage capacity used. Blocks are mined using disk storage instead of CPU power. It takes less than half the time of PoW to validate and add a new block to the ledger.	Signum, Spacemint
Proof of Elapsed Time (PoET)	Based on a time-lottery-based mechanism, network participants are assigned random waiting times. The first running out of waiting time gets the chance to add a block to the ledger.	Hyperledger Sawtooth
Proof of Activity (PoA)	Combines PoW and PoS: adding blocks to the ledger is based on the PoW mechanism, while added blocks are verified through PoS	Decred, Espers
Proof of Authority (PoA)	Based on reputation in the network, usually used in private or permissioned blockchains.	VeChain
Proof of Burn	Blocks can be added by users sending coins to a "dead-end" account from which they won't be able to retrieve them. Similar to PoS with the difference that PoS allows de-staking, while PoB involves burning tokens (de facto paying to add a block to the ledger).	Slimcoin
Byzantine Fault Tolerance (BFT)	Based on the Bizantine's general problem, avoids network capitulation in case of failure by a single participant, ensuring constant communication among all participants.	Hyperledger Sawtooth, Hyperledger Fabric

Source: (Garg, 2023)

Second, one of the newest figures to enter the markets (specifically crypto markets, although not exclusively) are so-called finfluencers. Finfluencers, or finance influencers, are

public figures active on social media covering various fields of finance. Crypto influencers have a particular role in creating a sense of community around a specific coin or token, sometimes making or breaking its success since intrinsic value is difficult if not impossible to determine for these assets. More often than one can imagine, however, these influencers also have more gruesome roles in so-called pump-and-dumps. Pump-and-dumps are schemes arranged by informed traders at the expense of uninformed traders, using some mean of influence to get the most uninformed traders possible on board before running the scheme. Usually, a very cheap token is created from scratch by informed traders and advertised through campaigns and influencers (that may or may not be in themselves, as informed traders[11]). Once the project has gained traction and a large number of uninformed traders have bought the token, making the price inflate considerably as a result of the usually relatively small available supply, the creators "pull the rag" under uninformed traders and sell their whole stake, creating a considerable slump in price and leaving behind an awfully illiquid asset with uninformed traders being stuck in their positions, as no one would want to buy the token after the "rag pull", rendering selling virtually impossible. Again, lack of regulation can be seen as an enabling factor for these manipulatory endeavors contributing to overall market inefficiency.

Cryptocurrencies are a novel addition to the realm of financial markets and as such features some structural peculiarities in how and when coins are traded. The next sections will cover these features, from 24/7 trading to issues with order flow centralization given the decentralized nature of these assets.

2.3. 24/7 Trading

One of the most unique features of cryptocurrency markets is their 24/7 activity. These markets are fully automated, contemplate no closing times and no vacation days. Be it Christmas or New Year's Eve, traders are always welcome in crypto markets.

There is no specifically recognized time at which crypto trading starts and ends for the day. The conventional time zone used tends to coincide with New York's local time, as that is the official time of reference for the major US equity exchanges. Midnight New York time would then be the reference time to stop price returns computations for the current day and start computations for the next day. Since patterns in volume, volatility and other indicators throughout the day and the week are a well-known phenomenon in equity markets (Amihud and Mendelson, 1987), one is left to wonder if cryptocurrency showcases similar patterns as

[11] See for istance the CryptoZoo scandal with Youtuber and influencer Logan Paul

well, given the difference in trading time. One first, interesting finding by Caporale and Plastun (2019) and Aharon and Qadan (2018) is that bitcoin returns are significantly higher on Mondays compared to the rest of the week. The same analysis on different cryptocurrencies showed rather inconclusive, hinting at the fact that Bitcoin is probably even more peculiar than the average cryptocurrency. After all, Bitcoin accounts for almost 50% of the whole crypto market cap as of August 2023, according to coinmarketcap.com data. Kurihara and Fukushima (2017), analyzing Bitcoin data from 2010 to 2016, noticed considerable volume inefficiencies in the first half of the observation period, as volumes systematically decreased during weekends. Baur et al. (2018) noticed increased volume in Bitcoin trading at times coinciding with US stock markets being open and decreased volume around midnight to early morning (NY time). They also went as far as concluding that the Bitcoin market is weak-form efficient, as they found no persistent patterns for the time-of-day, day-of-week or month-of-year Bitcoin returns, adding that increased trading volume during weekdays may be a result of institutional traders not being active during weekends. In fact, weekends tend to be characterized by fewer large orders (Johnson, 2019). Johnson conducted similar investigations as the ones above and obtained mixed results: there still is support for a statistically significant reduction in trading activity during weekends, however that depends on the cryptocurrency analyzed, the exchange from which data is collected[12] and the time period considered. The issue remains that if lower trading activity during weekends is actually a systematic occurrence, relatively illiquid weekends would increase transaction costs and volatility for impatient traders operating during Saturdays and Sundays.

Although quite mixed, it appears that cryptocurrency markets showcase higher degrees of inefficiency compared to traditional ones. Whether this is linked to the 24/7 trading arrangement or to the sheer novelty of these markets (less than 20 years old) is still premature to conclude.

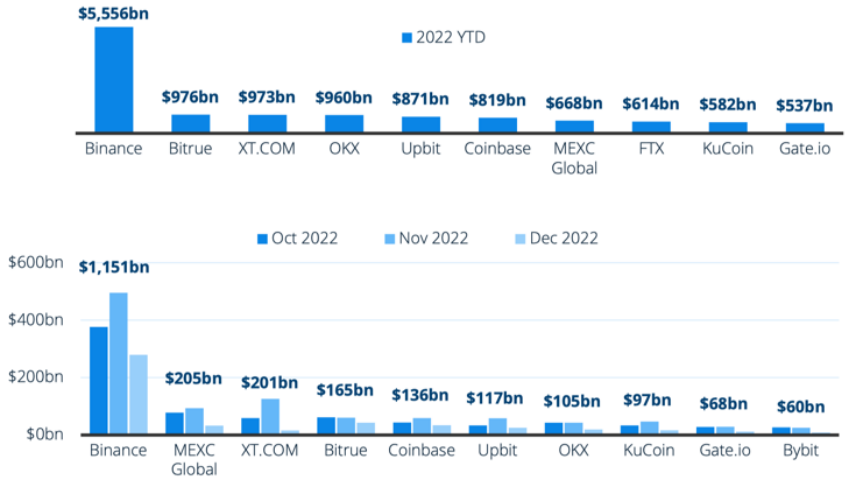
2.4. Venue Fragmentation and Liquidity

The decentralized nature of cryptocurrency gets translated in a variety of separated trading venues of different sizes that don't usually share or aggregate order flow data. This results in a lack of information for traders, as observing order flow on one single exchange

[12] Cryptocurrency decentralization brings to considerable fragmentation in trading venues, causing difficulties in collecting data regarding the "entire market" (more on this in the next section). As a result, authors tend to focus on a limited number of main exchanges, that still don't capture the full picture.

does not equate to gauging the situation for the entire market. Some exchanges even have legally autonomous subsidiaries based on state regulation (see, for instance, Binance US), possibly with a lack of intra-exchange order flow aggregation as a result as well. Venue fragmentation thus results in more difficult information retrieval for traders. However, venue fragmentation has both downsides and upsides, explored in the next subsections.

Figure 9 - Top 10 crypto exchanges by trading volume. First chart: YTD 2022, second chart: full month Oct, Nov, Dec 2022



Source: CryptoCompare, © Statista (Statista, 2023a)

2.4.1. Advantages of Venue Fragmentation

The main pro of having a vast array of trading venues is ensuring decentralization and limiting the dominance by single exchanges. Moreover, the coexistence of many exchanges can both stem competition and product specialization: those exchanges offering the same products will probably end up adopting more competitive pricing in the form of lower transaction costs, or could turn to innovation to diversify their product portfolio, still benefitting traders by introducing new technologies and instruments to the market. In a pool of over 1600 centralized and decentralized exchanges[13], some will choose to specialize their product offering, tailoring it to a specific niche of traders requesting particular services. One example would be EDX Markets, an exchange exclusively dedicated to institutional investors

[13] Source: https://blockspot.io/exchange/. More on centralized and decentralized exchanges in section 2.4.3.

launched in late 2022 and backed by Citadel Securities, Fidelity Digital Assets, Charles Schwab, Virtu Financial and Sequoia Capital[14].

Notably, however, not many pros are related to order flow, volume or liquidity. As it turns out, fragmentation might hinder these components, resulting in price inefficiencies. The type of trader that would probably benefit most from fragmentation are arbitrageurs and cross-arbitrageurs, it is important to consider however the higher complexity these players may face when operating on less liquid exchanges with unstable spreads.

2.4.2. Disadvantages of Venue Fragmentation

Jeon, Samarbakhsh and Hewitt (Jeon et al., 2021) analyzed liquidity gains obtained by aggregating order book data from 5 major crypto exchanges from 2017 to 2019. They found a relatively informed trader using the 5-exchange-consolidated order book can significantly benefit from additional liquidity gain. Moreover, they found that market fragmentation for Bitcoin did not improve over the analyzed period, highlighting the current lack of consolidation tools such as the popular smart order routers used in US equity exchanges to provide the national best bid and offer prices (NBBO). Clearly, relative lack of liquidity and limited order book depth are amongst the major cons of venue fragmentation. Liquidity aggregators do exist in cryptocurrency markets, particularly in decentralized exchanges, however they tend to be hard to come by. Some solutions are being proposed, such as Takemiya's Aggregate Liquidity Technology (ALT) (Takemiya, 2023).

Lack of order flow consolidation can bring to a series of additional inefficiencies. Price disparities among exchanges, not necessarily due to data directly related to cryptocurrency prices, can create unfair advantages and disadvantages between traders operating on relatively more transparent exchanges (offering "fairer" bid-ask spreads, for instance) and those trading in more opaque venues. Isolated venues and price disparities together strongly hinder price discovery as a result. Different venues might have different costs, execution speeds and product offerings and might showcase an uneven distribution of liquidity. Such a heterogeneous landscape would result in possible adverse selection regarding new traders that might feel overwhelmed by the complexity of the crypto field, limiting potential for adoption and market participation. Not only that: heterogeneity in trading venues is a perfect source of market inefficiencies that can be exacerbated by opportunistic behavior and possible manipulation by rogue traders capitalizing on the lack of regulation and the resulting

[14] Source: https://edxmarkets.com/about/.

limited market surveillance solutions in place. All these complexities can also contribute to an overall less stable market.

2.4.3. Types of Trading Venues in Cryptocurrency Markets

The structure of cryptocurrency markets is also characterized by peculiarities in the venues used by traders to carry out transactions. What follows is a brief overview of the main venues one should expect to find in the fabric of cryptocurrency markets.

Centralized Exchanges (CEXs)

Centralized exchanges are what most traders use to buy and sell crypto assets, from currencies, to NFTs, to derivatives such as futures and options on cryptocurrency. They are online trading platforms where users can create accounts, deposit funds, and trade. CEXs match buy and sell orders from traders, updating their order books. One example is Binance[15].

Decentralized Exchanges (DEXs)

DEXs are part of the ever-evolving decentralized finance (DeFi) landscape. Differently from CEXs, DEXs allow true peer-to-peer trading and don't require deposits in centrally controlled wallets. Traders will trade directly from their personal wallets, adequately connected to the decentralized platform. DEXs tend to leave more control to traders compared to CEXs, resulting in potentially higher operational risk for those inexperienced with such platforms. An example is dYdX[16].

OTC Venues

Typically reserved for institutional and high-volume traders, OTC exchanges are usually used to execute large orders needing significant liquidity and a higher degree of tailoring. They contribute to reducing market impact stemming from large trades. One example is Genesis Global Trading[17].

[15] https://www.binance.com/
[16] https://trade.dydx.exchange/
[17] https://genesistrading.com/

Dark Pools

Similarly to equity dark pools, crypto dark pools are private venues where traders can transact without disclosing their position before execution (orders are only visible after execution). However, crypto dark pools process trades quite differently: they still allow large zero-slippage trades and do not have public order books, however the way in which pre and post-trade anonymity is guaranteed is based on cryptographic solutions such as multi-party computation (MPC) and zero-knowledge proofs (ZKP)[18]. An example of dark pool is Renegade[19].

Peer-to-peer (P2P) Platforms

P2P platforms specialize in escrow services for transactions between buyers and sellers. They constitute a sort of third-party guarantee supporting an agreement between two private parties. An example of P2P platform is Paxful[20].

Security Token Exchanges

Security tokens are a novel digital form of funding based on blockchain technology that fall under the broader "digital asset" umbrella. Investment contracts and ownership are automatically verified through smart contracts. Security tokens, among other things, can be the result of equity tokenization, through which companies issue equity in the form of coins or tokens instead of shares. Security Token Exchanges allow trading of these novel digital assets. One example is INX[21].

Swap Protocols

Swap protocols are usually DApps hosted on dedicated platforms that allow swapping one cryptocurrency for another. The value in swap protocols is that they allow the exchange of cryptocurrencies hosted on different blockchains: without swap protocols, if a trader wanted to convert, say, a coin or a token based on the Bitcoin blockchain with a coin or token based on the Ethereum blockchain, the only way to do so would be to initiate a crypto-to-fiat transaction (e.g., BTC to USD) and then a fiat-to-crypto one (e.g. USD to ETH). Swaps make this conversion direct and bring transaction costs savings. Swap protocols make extensive use of liquidity pools (large portions of cryptocurrency locked in a smart contract,

[18] Source: https://docs.renegade.fi/core-concepts/dark-pool-explainer/
[19] https://renegade.fi/
[20] https://paxful.com/
[21] https://www.inx.co/

ready to provide liquidity to various networks when needed). A rather famous example is Uniswap[22].

Liquidity Aggregators

Liquidity Aggregators are not trading venues per se, however they play an important role in making markets more efficient by consolidating order book data, usually from various DEXs. Liquidity aggregators are another example of DApp (Decentralized Application) and can be accessed through dedicated websites, making them a venue regardless. Examples are 1inch[23], which also acts as a swap protocol, and OpenOcean[24], a full-fledged DEX aggregating order flow from other DEXs.

Staking Platforms

Staking platforms allow users to stake their coins or tokens for predefined periods of time in exchange of rewards. Rewards might be interest, newly minted tokens, opportunity to validate transactions, etc. Many exchanges offer staking services: an example is Gemini Staking[25].

Cryptocurrency markets offer a vast array of products and services. On the one hand, individuals can satisfy their specific needs with the most diverse crypto products; on the other hand, however, investor protection and market surveillance are severely lacking. Although regulators across the globe are moving fast and proposing dedicated laws, parts of the crypto landscape (such as DeFi) still remain vastly unregulated, with far fewer safeguards compared to traditional markets.

2.5. Transparency and (Lack of) Regulation

The extremely fast-paced innovation brought by cryptocurrency sees regulators playing "catch up" in the field (Narain and Moretti, 2022). Moreover, international harmonization of laws is a serious concern: the definition and actual or intended use of "crypto assets" can attract at the same time multiple domestic regulators, having different frameworks and objectives but still wanting to regulate the same case (a crypto asset used as a security might

[22] https://app.uniswap.org/
[23] https://app.1inch.io/
[24] https://app.openocean.finance/
[25] https://www.gemini.com/en-us/staking/

attract a security and exchange regulator, prioritizing financial integrity, while a crypto asset used as currency might attract a payments regulator, prioritizing consumer protection, and so on) (Narain and Moretti, 2022). Even crypto market participants like miners and validators do not have a clear place in current regulatory frameworks. Calls for the application of "same activity, same risk, same rule" principles are increasing, however that can be projected into the future and one could imagine that integration of traditional finance and the crypto world might in the end bring to the forceful application to crypto of the same central bank facilities and safety nets that today characterize traditional finance (Narain and Moretti, 2022), defying crypto's intrinsic decentralization purpose.

Japan and Switzerland have already introduced comprehensive crypto asset legislations, while other countries (US, UK, EU, UAE) are still at preliminary stages. Global standard setters like the Financial Stability Board and the Basel Committee on Banking Supervision are issuing and working on standards based on their scope and jurisdiction. The FSB published its global regulatory framework for crypto-asset activities[26] on July 17, 2023, while the BCBS published standards regarding the prudential treatment by banks of crypto-asset exposures[27]. The EU has seen the proposal of MiCA, or Markets in Crypto-Assets regulation in 2020, while in the US Congress has left the task of addressing issues created by digital assets to regulatory agencies. The IMF is specifically calling for a globally coordinated regulatory response, that shall be consistent with mainstream regulatory approaches and as comprehensive as possible (Narain & Moretti, 2022).

Figure 10 - Regulation of cryptocurrency by country. Regulation includes preliminary, exploratory rules

Source: Buchholz (2022), © Statista 2023

[26] https://www.fsb.org/2023/07/fsb-finalises-global-regulatory-framework-for-crypto-asset-activities/
[27] https://www.bis.org/bcbs/publ/d545.pdf

A properly harmonized regulation of the crypto landscape could help in standardizing part of the markets and products, rendering information retrieval easier and possibly avoiding the far too frequent pricing and spread differences across exchanges. It is apparent that regulatory bodies will need to dedicate more resources to the crypto case if effective regulation of the space is among their objectives, particularly now that central banks are starting to pave the way for digital fiat currency (not to be confused with the already existing electronic cash used by debit and credit cards).

This chapter has expanded on Chapter 1 with some peculiarities of cryptocurrency market structure. The next chapter will introduce a somewhat traditional approach used to identify patterns in one of the most telling microstructural data: order flow. The final chapter will then explore the realm of Artificial Intelligence and Machine Learning implementations applied to order flow analysis.

Chapter III Order Flow Analysis In Practice: The BTC/USDT Pair

Order flow analysis examines the sequence of buy and sell orders, as well as their size, timing, and execution, to give insight on market state, participant behavior, and potential price movements. Studying the complexity of order flow is critical to understand the underlying mechanics that determine market movements in the cryptocurrency landscape. As a tool, traditional order flow analysis can be revisited in the context of the ever-evolving world of cryptocurrency. This chapter will delve into the complexities of crypto order flow analysis, discovering how it affects liquidity, trading methods, and provides insights into the core of decentralized market dynamics.

3.1. Introduction to Order Flow Analysis

Order Book and Order Flow analysis are often explored by academic research when a market is "on-book", i.e. trades are arranged through systems matching orders from a Limit Order Book (LOB). One of the main reasons for the success attributed to this type of analysis in the field of market microstructure are the multiple findings of persistency or "long memory" in order flow (Jaisson, 2015; Tóth et al., 2015; Cohen, 2022) across various markets at short timescales. Tóth et al. trace back this persistence to two main phenomena: order splitting and herding behavior. They conclude that one of the main drivers of persistence in equity order flow, at least in time frames of less than a few hours, is activity clustering in the form of order splitting, i.e. traders splitting large orders into multiple smaller ones and executing them sequentially. They find in fact that bursts of orders having the same sign (the main source of autocorrelation in order flow) usually come from a single individual rather than being the result of trader interaction. Quite surprisingly, they find that herding behavior resulting from trader interaction might bring negative autocorrelation in order flow, for one very specific reason: they observe that if a market order placed by one trader changes the price, other traders are less likely to place market orders in the same direction. Oppositely, if the original market order does not change the price, traders tend to place orders in the same direction. Although the true underlying reason requires further investigation, the suggested explanation for this behavior lies in the level of adaptation of traders to the microstructural mechanisms of the LOB. If, for instance, price rises and causes a momentaneous increase in the bid-ask spread, traders who were previously buying at market may switch to buying with limit orders placed into the enlarged spread. This would in turn increase the best bid price,

making the use of market orders more favorable for sellers. The combined effect of market buyers decreasing and market seller potentially increasing might explain the negative autocorrelation stemming from interactions among traders, opposed to the positive autocorrelation in order flow caused by order splitting. Still, in hourly time spans, the authors find order splitting to have a much higher and significant impact. This makes Tóth et al. argue that the market might very often be out of equilibrium as a result of split-traders revealing their intentions only gradually, partial order by partial order. If split traders revealed their intentions immediately, price dynamics would intuitively be different.

3.1.1. Imbalance in the Limit Order Book

One of the most telling order book metrics is the so-called order flow imbalance. It is based on the concept that the arrival and the cancellation of limit orders, together with market orders hitting the book, are reliable indicators of changes in market supply and demand.

Cont et al. (2014) propose a stylized analytical description of order flow imbalances, describing the bid price P_n^B and bid size[28] q_n^B as measures of demand and the ask price P_n^A and ask size q_n^A as measures of supply, where n indicates the current observation in time. A "level-1" order book state (i.e. looking at only one price level) at observation n would then be represented by the quadruple $(P_n^B, q_n^B, P_n^A, q_n^A)$. If one were to compare the current single-level order book state $(P_n^B, q_n^B, P_n^A, q_n^A)$ with the immediately preceding state $(P_{n-1}^B, q_{n-1}^B, P_{n-1}^A, q_{n-1}^A)$, only a set number of events can occur:

- Either $P_n^B > P_{n-1}^B$ or $q_n^B > q_{n-1}^B$ representing an increase in demand
- Either $P_n^B < P_{n-1}^B$ or $q_n^B < q_{n-1}^B$ representing a decrease in demand
- Either $P_n^A < P_{n-1}^A$ or $q_n^A > q_{n-1}^A$ representing an increase in supply
- Either $P_n^A > P_{n-1}^A$ or $q_n^A < q_{n-1}^A$ representing a decrease in supply

Focusing on sizes, when the n-th order book event modifies the bid or ask size, its contribution can be then defined as

$$e_n = I_{\{P_n^B \geq P_{n-1}^B\}} q_n^B - I_{\{P_n^B \leq P_{n-1}^B\}} q_{n-1}^B - I_{\{P_n^A \leq P_{n-1}^A\}} q_n^A + I_{\{P_n^A \geq P_{n-1}^A\}} q_{n-1}^A \quad (1)$$

[28] Bid (ask) size is the number of limit buy (sell) orders resting at a given price level, waiting to be filled.

Where I's are indicator functions equal to 1 when the relative inequality between curly braces is satisfied and equal to 0 when the inequality is not satisfied. Events e_n can either be limit order submissions, limit order cancellations or market orders hitting either side of the book. Notably, if a number of limit buy orders was added at the current bid level without a subsequent bid price increase (i.e., $q_n^B > q_{n-1}^B \wedge P_n^B = P_{n-1}^B$), the contribution of this event would be $e_n = q_n^B - q_{n-1}^B$ effectively equal to the number of limit orders added at the current level. The case $q_n^B < q_{n-1}^B$ with $P_n^B = P_{n-1}^B$ can be caused either by the cancellation of buy limit orders already resting at P_{n-1}^B or by market sell orders hitting the bid at P_{n-1}^B. In this case $e_n = q_n^B - q_{n-1}^B$ would be negative and represent the number of units sold at market or the amount of cancelled limit buys. Order flow imbalances can be defined as the sum over a set time period $[t, t + \mathrm{k}]$ of all events e. Assuming, for simplicity, that at most one order book event can happen at each time instant t:

$$OFI_k = \sum_{n=t}^{t+k} e_n$$

The resulting imbalance will be a net measurement of all the contributions by bid and ask order book events to the current order book state in the specified time interval. Cont et al. demonstrate that order flow imbalances as defined above are linearly correlated to price changes in 50 NYSE TAQ stocks.

In the cryptocurrency realm, Silantyev, (2019) analyzed perpetual futures contracts on the BitStamp exchange, discriminating between order flow imbalances (OFI) and trade flow imbalances (TFI). OFIs were defined following Cont et al.'s model, while TFIs were defined as the difference between market buy and market sell orders over a given period. Using linear regression, the author analyzed the effect of OFIs and TFIs on short-term price changes. The main finding was that TFIs (only based on market orders) have higher impact on prices compared to OFIs, which showcased poorer model fit further exacerbated by the irregular depth of the LOB. Silantyev makes an extremely interesting case for insufficient regulation being part of the reason why TFIs are more predictive than OFIs as far as price changes go:

the weak presence of regulation, particularly in relation to spoofing[29] and market manipulation, contributes to the submission of low-information limit orders. Market orders, on the other hand, requiring the immediate payment of commissions and bid-ask spreads, act as a filter for high-information orders signaling trader intent and thus impacting prices with a higher degree of significance.

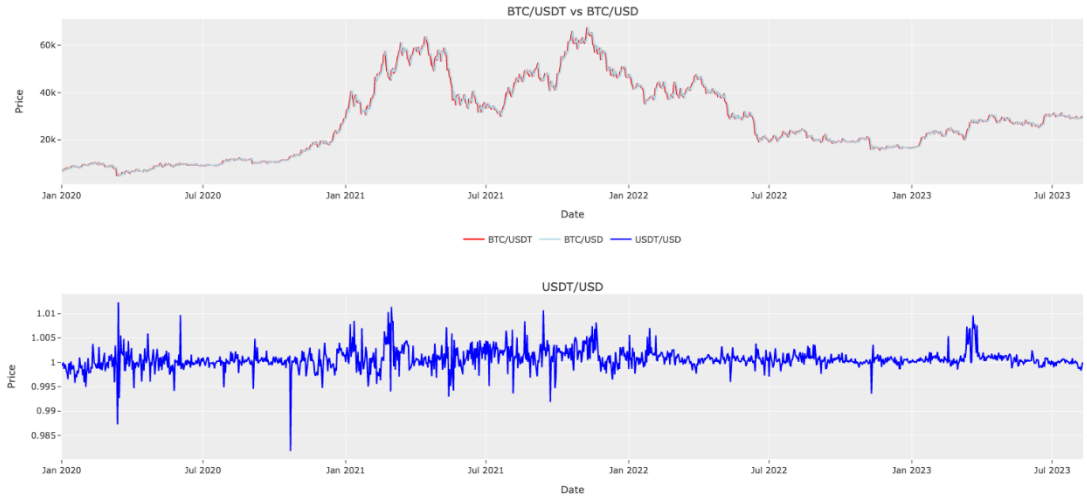
Silantyev relied on traditional linear models to carry out his analysis. The relevance of order flow and at the same time the complexity entailed by analyzing large amounts of order book data over extended periods of time calls for the application of novel techniques, expanding beyond the scope of traditional statistical models. Artificial Intelligence (AI) and Machine Learning (ML) models have been deployed in various declinations to recognize hardly detectable patterns in the intricate and extremely granular fabric of order book data, and will be explored in Chapter 4.

3.2. A Practical Application: the BTC/USDT Pair

The analysis presented in this section is an independent contribution and represents original research. It serves as a practical example describing how one would screen for order book imbalances in one of the most liquid cryptocurrency pairs as of currently: Bitcoin (BTC) vs Tether US (USDT). USDT is a so-called "stablecoin", a cryptocurrency pegged to fiat currency through a specific algorithm managing its supply. In this case, USDT is pegged to the US Dollar. Figure 11 shows that during the last three years the peg has remained relatively stable, making BTC/USDT a good proxy for the actual dollar value of Bitcoin. USDT will be used instead of USD because data for this application will be retrieved from Binance, the largest exchange worldwide by trading volume (as already highlighted by Figure 9), and the most liquid pair on Binance is specifically BTC/USDT. Binance allows calls to its API that can be used to retrieve salient data for this analysis. Order book aggregation utilizing data from other large exchanges is outside the scope of this application. With Binance having a high degree of spot market liquidity, and the BTC/USDT being the most traded and liquid spot pair on the exchange, data availability is sufficient for this close-environment demonstration.

[29] Spoofing is a usually illegal market manipulation practice that involves placing deceptive limit orders in the market, often cancelled before execution, to create a false impression of supply or demand. Traders may spoof to manipulate prices, induce others to trade, or gain advantageous positions, exploiting market psychology for potential profit. See Navinder Sarao's story for further anecdotes on spoofing and its potential consequences.

Figure 11 - Price charts for the pairs BTC/USD, BTC/USDT and USDT/USD. (a) BTC/USDT is a good proxy for BTC/USD thanks to the stability of the peg in (b): USDT/USD

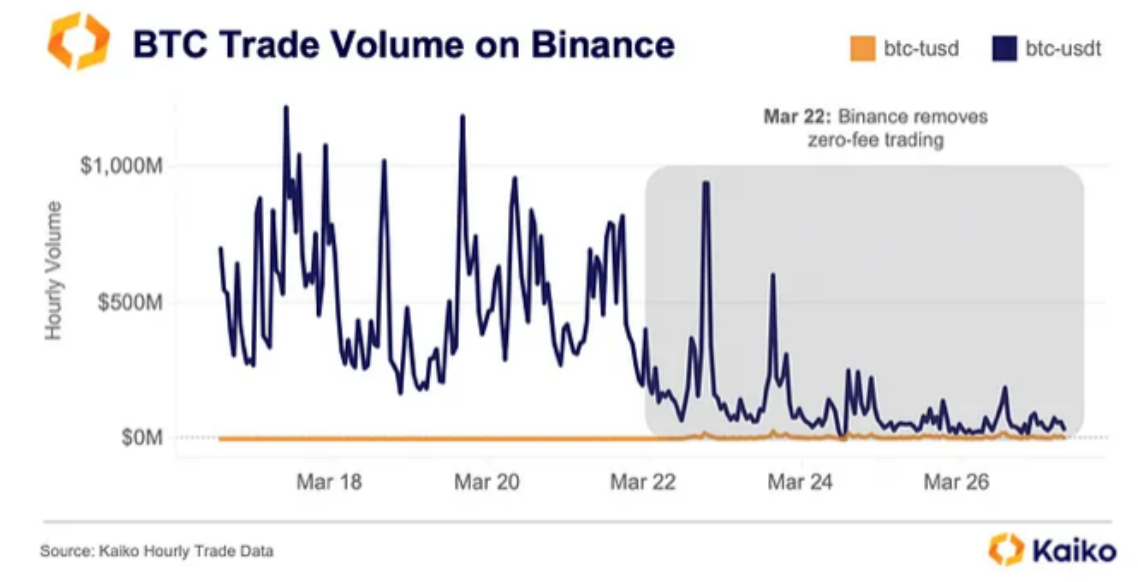


Data source: Coingecko, Binance. Original graphical representation via Python.

This exploratory analysis has two main focuses: the first concerns verifying Silantyev's (Silantyev, 2019) findings using a larger dataset, focusing on an exchange that was not considered in the paper. The choice of focusing on Binance is related to the second aim of the analysis: gauging the possible impact of the change in fee structure that was announced on March 15th, 2023 and has taken effect on March 22nd. Binance has in fact shifted from a zero-fee structure for all BTC pairs (started in July 2022 as a celebration of their 5th anniversary) to the application of maker and taker fees on March 22nd, allowing the continuation of zero-fee trading only on BTC/TUSD, not to be confused with BTC/USDT, as USDT (Tether) and TUSD (True USD) are different stablecoins. This analysis will focus on BTC/USDT, impacted by the fee structure change. The end of generalized commission-free trading has already impacted overall trading volume on Binance over the last few months, as Figure 12 shows. The still difficult to understand choice of maintaining commission-free trading on the low-volume BTC/TUSD pair does not seem to have had any effect on retaining users, who are systematically moving to other exchanges where BTC trading is still commission free (clearly, with an application of a spread between bids and asks) or where the fee structure is cheaper.

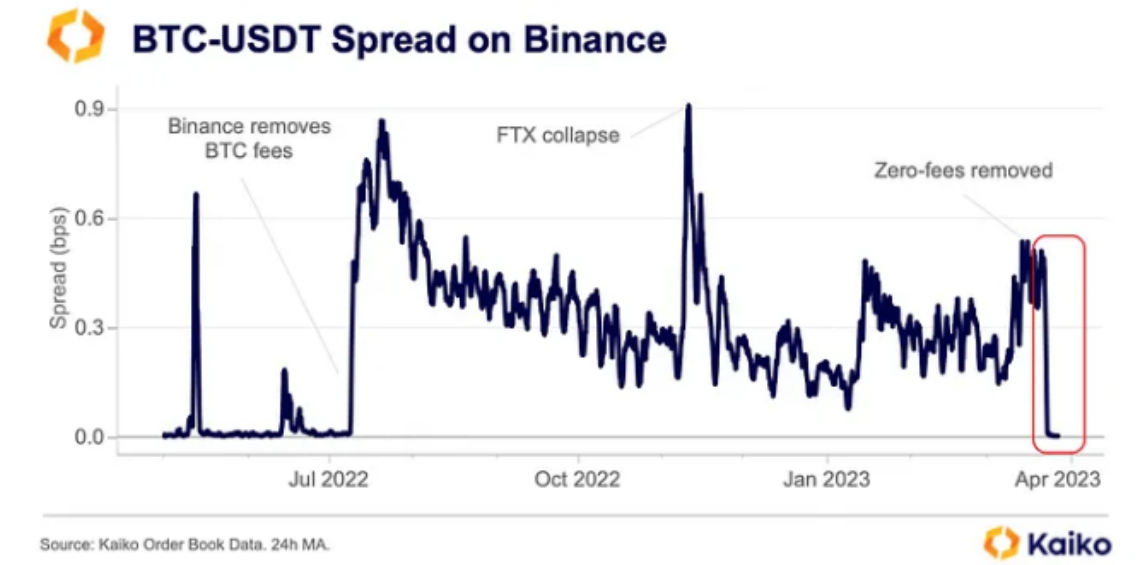
Intuitively, the cost of a market order in a commission-free trading structure depends on the contingent width of the bid-ask spread. Adding commissions should see a subsequent decrease in the bid-ask spread, as confirmed by Figure 13.

Figure 12 - BTC/USDT trading volume (in blue) before and after removing generalized zero-fee trading



Source: Kaiko

Figure 13 - Spread close to 0 after applying a fee structure to BTC/USDT trading

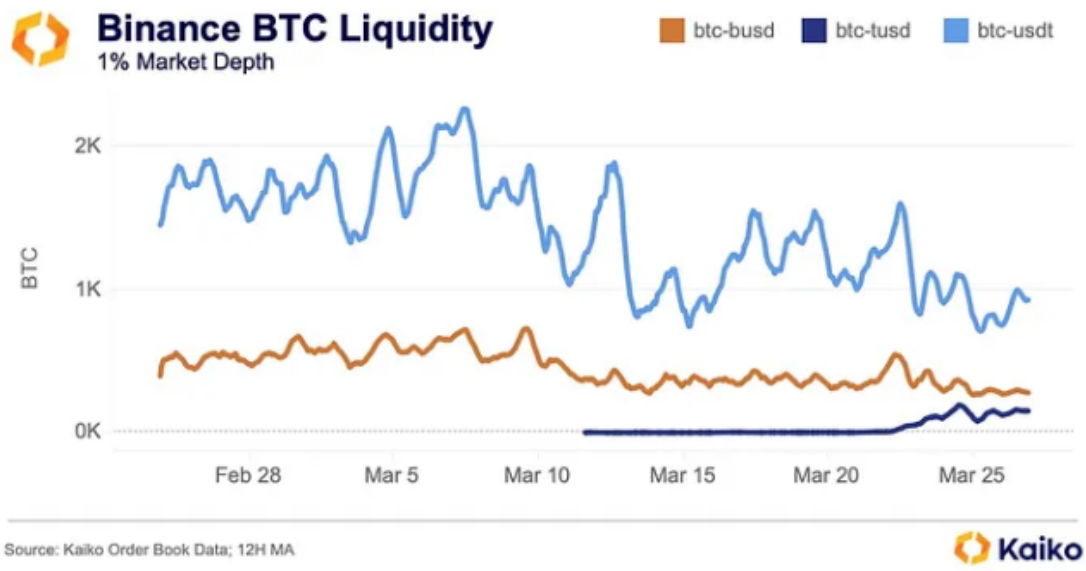


Source: Kaiko

Fee-based crypto exchanges apply both maker and taker fees to orders, hence both traders posting market orders and traders posting limit orders will face a fee at execution (makers will not thus enjoy the advantage given by the bid-ask spread, as it is often insignificant in these fee structures). To compensate and favor the influx of liquidity,

exchanges tend to charge higher taker fees (for those posting market orders) and lower maker fees (for those posting limit orders). This should in turn favor the posting of limit over market orders, simply based on the cost at execution, thus granting adequate amounts of liquidity in the market. In a fragmented landscape where different exchanges are structured differently, however, Binance's move of allowing commission-free trading for a period and stopping after a while has not revealed wise from a liquidity stand point (Figure 14), as users have started turning to different exchanges, regardless of commission-free trading still being in force on a minority of pairs.

Figure 14 - Commission trading harms BTC/USDT liquidity



Source: Kaiko

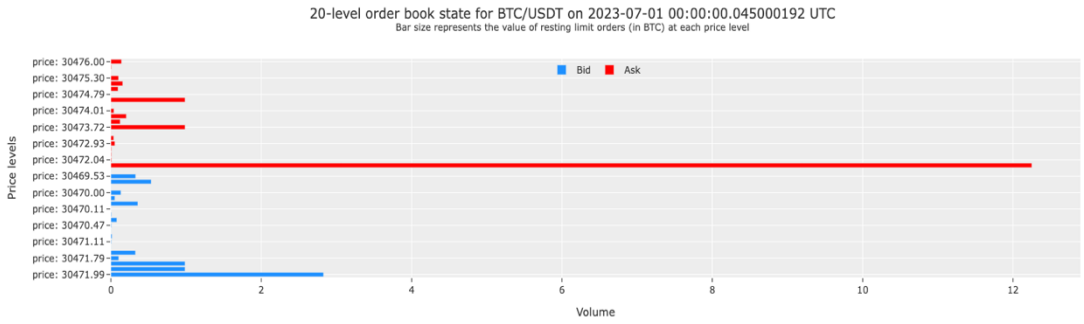
3.2.1. Data Collection

This analysis will make use of nanosecond order book data streamed from Binance starting on July 31st, 2022 and ending on July 31st, 2023 that will be upsampled at different frequencies for computational concerns. This year's worth of data (amounting to approximately 143 million order book states at nanosecond granularity) is provided by the Crypto Lake API and will be used to compute the Order Flow Imbalance measure. The Trade Flow Imbalance will be computed using executed trade data from March 15th 2023 to July 31st 2023 for reasons related to the available computational power. Data has been collected with Python and will be both manipulated and visualized leveraging the same programming language. RAM constraints mentioned by Silantyev in 2019 have been partially

resolved by using Apache Parquet files, specifically designed for table management and efficient data storage/retrieval. Confirming significant linear correlation between OFIs and contemporaneous price changes (and TFIs and contemporaneous price changes) at two years from Silantyev's paper is not guaranteed, as automated trading strategies exploiting order flow imbalances have been flourishing for some time now.

Crypto Lake offers up to 20 price levels per state at a millisecond granularity, showcased in Table 5 and visualized in Figure 15. The focus will however be on level-1 order book states (focusing on best bid and best ask quotes and sizes).

Figure 15 - Order Book representation at the start of July 1st, 2023. 34,700USDT seems to be a high level of interest, with a total worth of more than 12BTC in limit orders resting there



Data source: Crypto Lake data, original visualization created via Python

Some higher-level trading strategies rely exclusively on sequential order book states, as the one depicted in Figure 15: one apparent price level is around 34,700USDT at midnight on July 1st, 2023. This level showcases a large ask size, this means that a large amount of BTC in the form of limit sell orders is resting at that level. Traders may as a result see 34,700 as an area of potential "resistance" or "absorption": aggressive, impatient market buyers will be able to drive price beyond that level only if their aggregate market orders make up a value higher than the one of the resting limit orders. If that is not the case, market buyers will simply be "absorbed" as a result and price won't move past 34,700. Only further visits to that level with new buys hitting the ask will potentially result in price going higher. AI and ML strategies based on multi-level order book states tend to look at large bid or ask sizes as magnets attracting price, deciding on trade directionality based on whether the highest number of limit orders is resting above or below the current mid-price. See Tran et al., 2022 for further insights on the informativeness of order book data from a ML perspective.

Table 5 - 10 Order Book States at 20-level depth. Snapshot from the "book" data frame (specify time frame that is being represented in the table

received_time	origin_time	sequence_number	bid_0_price	bid_0_size	bid_1_price	bid_1_size	bid_2_price	bid_2_size	…	bid_19_price	bid_19_size	ask_0_price	ask_0_size	ask_1_price	ask_1_size	ask_2_price	ask_2_size	…	ask_19_price	ask_19_size	exchange	symbol
2023-07-01 00:00:00.052843520	2023-07-01 00:00:00.045000192	3.7675E+10	30471.99	2.82754	30471.95	0.98472	30471.83	0.98472	…	30469.52	0.00098	30472	12.25017	30472.04	0.00774	30472.09	0.00658	…	30476	0.13985	BINANCE	BTC-USDT
2023-07-01 00:00:00.157764352	2023-07-01 00:00:00.144999936	3.7675E+10	30471.99	2.85423	30471.83	0.98472	30471.79	0.10255	…	30469.49	0.04061	30472	12.2481	30472.04	0.00938	30472.09	0.00658	…	30476.01	0.08199	BINANCE	BTC-USDT
2023-07-01 00:00:00.350893568	2023-07-01 00:00:00.345999872	3.7675E+10	30471.99	1.27286	30471.74	0.3265	30471.61	0.00079	…	30469.49	0.04061	30472	11.99452	30472.04	0.00938	30472.09	0.00658	…	30475.53	0.00632	BINANCE	BTC-USDT
2023-07-01 00:00:00.451638016	2023-07-01 00:00:00.446000128	3.7675E+10	30471.99	1.28557	30471.74	0.3265	30471.61	0.00079	…	30469.49	0.04061	30472	11.85067	30472.04	0.00938	30472.09	0.00658	…	30476	0.13985	BINANCE	BTC-USDT
2023-07-01 00:00:00.555460864	2023-07-01 00:00:00.546000128	3.7675E+10	30471.99	1.2846	30471.74	0.3265	30471.61	0.00079	…	30469.52	0.00098	30472	9.31884	30472.04	0.00938	30472.09	0.00658	…	30476.46	0.00079	BINANCE	BTC-USDT
2023-07-01 00:00:01.051793920	2023-07-01 00:00:01.046000128	3.7675E+10	30471.99	2.28605	30471.92	0.32951	30471.79	0.98472	…	30469.53	0.32822	30472	9.94344	30472.04	0.00938	30472.09	0.00658	…	30475.49	0.00079	BINANCE	BTC-USDT
2023-07-01 00:00:01.250915328	2023-07-01 00:00:01.246000128	3.7675E+10	30471.99	2.26549	30471.92	0.32951	30471.79	0.98472	…	30469.54	0.24028	30472	9.95118	30472.04	0.00164	30472.09	0.00658	…	30475.04	0.03	BINANCE	BTC-USDT
2023-07-01 00:00:01.350973952	2023-07-01 00:00:01.345999872	3.7675E+10	30471.99	2.26649	30471.92	0.32951	30471.79	0.98472	…	30469.54	0.24028	30472	9.95118	30472.04	0.00164	30472.09	0.00658	…	30475.04	0.03	BINANCE	BTC-USDT
2023-07-01 00:00:01.750977536	2023-07-01 00:00:01.746000128	3.7675E+10	30471.99	2.2691	30471.79	0.98472	30471.74	0.3265	…	30469.53	0.32822	30472	9.68866	30472.04	0.00164	30472.09	0.00658	…	30475.3	0.1	BINANCE	BTC-USDT
2023-07-01 00:00:01.950739456	2023-07-01 00:00:01.946000128	3.7675E+10	30471.99	3.16192	30471.79	0.98472	30471.74	0.3265	…	30469.54	0.24028	30472	9.68866	30472.04	0.00164	30472.09	0.00658	…	30475.04	0.03	BINANCE	BTC-USDT

Data source: Crypto Lake API, retrieved with Python

3.2.2. Order Book Analysis: Measuring Imbalance the "Traditional Way"

In line with previous work, imbalances will be computed as OFIs (Order Flow Imbalances) and TFIs (Trade Flow Imbalances).

The Crypto Lake API data frame presented in Table 5, after appropriate manipulation, can be used to measure OFIs, while a different data frame from the same provider can be used to measure TFIs.

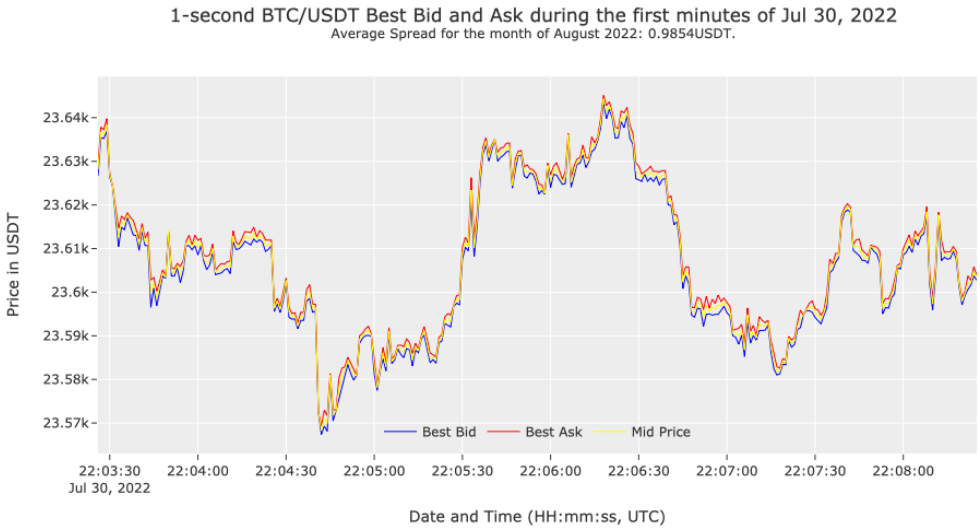
First, the "book" data frame is useful to visualize current best bid and ask prices and the mid-price can be computed as the simple average between the two:

$$mid\_price = \frac{ask\_0\_price + bid\_0\_price}{2}$$

Variables are named as they are in the code created to download, manipulate and visualize this data, available in the Appendix.

Figure 16 shows a sample of bid, ask and mid-price data visualized from the "book" dataframe.

Figure 16 - Best Bid, Best Ask and resulting Mid Price for BTC/USDT, sample



Data Source: Crypto Lake API, original computation and visualization via Python

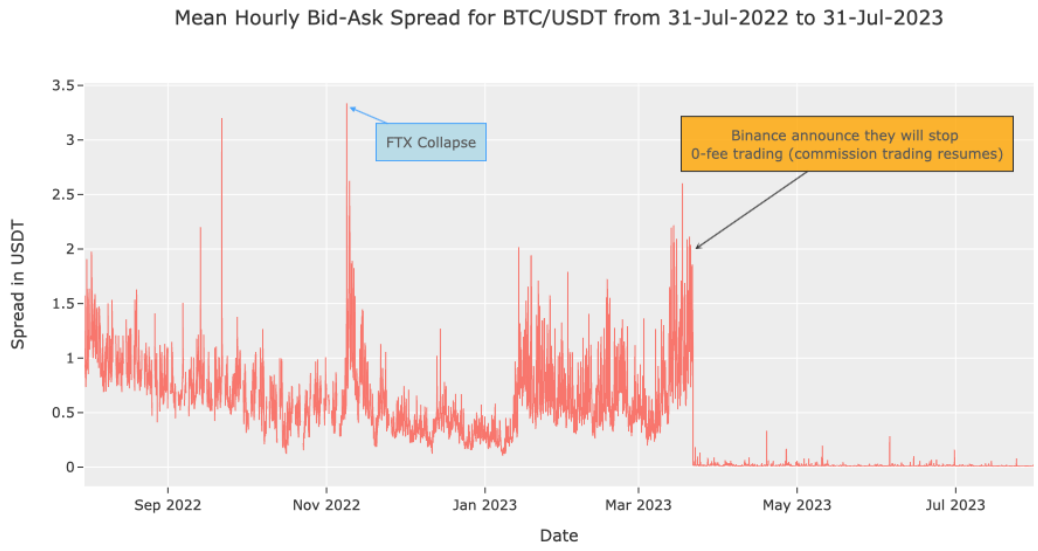
The current spread, visualized in Figure 17, is given by the difference between best ask and best bid. The change in fee structure is evident from March 2023 onwards, and the link

between volatility, liquidity and spread is well represented by the spike in correspondence of the FTX collapse.

OFIs will be computed following the already presented Equation (1), from Cont et al. (2014) for e:

$$e_n = I_{\{P_n^B \geq P_{n-1}^B\}} q_n^B - I_{\{P_n^B \leq P_{n-1}^B\}} q_{n-1}^B - I_{\{P_n^A \leq P_{n-1}^A\}} q_n^A + I_{\{P_n^A \geq P_{n-1}^A\}} q_{n-1}^A$$

Figure 17 - Average hourly spread on BTC/USDT trading on Binance. The end of 0-fee trading is easily recognizable



Source: Crypto Lake API, original computation and visualization via Python

Computed es will then be summed together over different periods of time (i.e., upsampled) to form OFIs. Figure 18 shows the average hourly e for the year in consideration. A change in the value of e is evident after commission-free trading is removed. The sharp increase in volatility is most likely due to the thinner liquidity caused by users leaving the exchange as a result of the new fee structure, effectively causing each order book event to have a higher impact on the order book state compared to the period when commission-free trading attracted enough traders to have more resistant and resilient order book states.

Actual OFIs, resampled at an hourly frequency, showcase the exact same features at a different scale, as Figure 18 shows.

TFIs will be simply computed as the algebraic sum over different time periods of executed trades. Buy trades quantities will have positive values (e.g., 0.12BTC) while sell trades quantities will have negative values (e.g., -0.3BTC). In the end, controlling for the

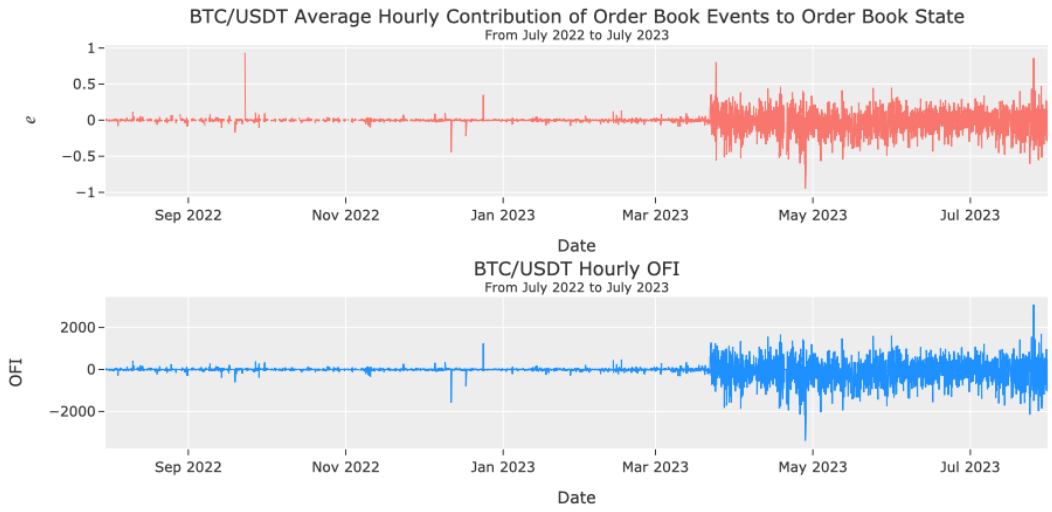
structural break in March 2023, shocks e, OFIs and TFIs appear to behave as stationary series and can be subject to standard statistical modelling. Hence, before moving to AI and ML modelling, to check if Binance data backs previous results of OFIs and TFIs having explanatory power for contemporaneous price changes, a linear regression is applied to different resamplings of the data frame in Table 5, of which columns OFI, TFI and mid_price_change are the main interest.

Resamplings include periods of 1 second, 10 seconds, 1 minute, 10 minutes and 1 hour, while mid-price changes are computed as follows:

$$mid\_price\_change_t = \frac{mid\_price_t - mid\_price_{t-1}}{0.01}$$

Where 0.01 is the tick size of the BTC/USDT quotes downloaded (i.e., BTC price is expressed in USD with a 2-decimals precision).

Figure 18 - Average value of e, resampled at an hourly frequency and Hourly OFI



Source: Crypto Lake API, original computation and visualization via Python

Because of the exogenous and arbitrary structural break caused by the shift in fees, the regression will separately analyze the periods before and after the change, with the aim of gauging possible resulting differences. OFI data is sufficient for a comparison of the periods going from July 31st, 2022 and July 31st, 2023. TFI data is unfortunately insufficient to measure changes between the two periods, as it starts from March 15th. Nevertheless, results

on OFI will be helpful in inferring the degree of TFI influence on mid-price changes during the no-fee period. R^2, coefficient values and p-values for the period after March 2023 are presented in Table 6.

Coefficients on OFI and TFI are to be read as the number of ticks by which the average expected mid-price increases following a one-unit increase in OFI or TFI. As highlighted above, the tick size in this case is 0.01USDT. Being OFI and TFI expressed in BTC, one exemplifying interpretation would be that a 1BTC increase in net order flow over a time frame of 10 seconds results in a contemporaneous 0.20USDT increase in the price of BTC (that is 20 USDT cents, roughly 20 dollar cents). With BTC trading around 30,000USDT at the time of writing, increasing net order flow by 30,000USDT would result in a ~0.0007% price increase.

A similar interpretation goes for TFIs, switching "net order flow" with "net trade flow". Mind that net order flow changes in this model may come with the same probability from order submission, cancellation, or execution. These three events intuitively happen with different relative frequencies, however this model assumes for simplicity that those frequencies are the same.

Table 6 - Results from multiple Linear Regressions of mid-price changes on OFI and TFI resamplings

MID PRICE *After* March 15, 2023

	OFI				**TFI**			
	α	β_{OFI}	p-value β_{OFI}	R^2	α	β_{TFI}	p-value β_{TFI}	R^2
1sec	0.6775	17.9619	0.0000	14.14%	0.3598	39.3421	0.0000	23.45%
10sec	7.2019	20.3229	0.0000	26.19%	4.2491	49.7505	0.0000	33.22%
1min	39.5303	18.6759	0.0000	30.69%	25.7922	50.2353	0.0000	37.96%
10min	281.3032	13.5135	0.0000	21.55%	224.1506	44.4374	0.0000	48.55%
1H	1049.6956	8.7653	0.0000	11.15%	1066.6631	36.1475	0.0000	51.78%

	OFI & TFI					
	α	β_{OFI}	β_{TFI}	p β_{OFI}	p β_{TFI}	R^2
1sec	0.8153	12.8937	33.7395	0.0000	0.0000	30.26%
10sec	8.6595	14.4258	39.6030	0.0000	0.0000	45.03%
1min	49.1858	13.1909	39.5288	0.0000	0.0000	51.54%
10min	383.6885	8.4488	39.1965	0.0000	0.0000	56.30%
1H	1604.2967	4.5088	34.0852	0.0000	0.0000	54.56%

The first relevant finding analyzing data from after the fee structure change is that using TFI as a regressor results in systematically better linear fit compared to using OFI. This is in line with Silantyev's 2019 findings where TFIs had better explanatory power than OFIs in

modelling contemporaneous mid-price changes, and goes against the original 2014 work on stocks by Cont et al. However, one important observation is that R-squareds are lower across the board compared to the two studies. This could mean that either the relation is moving from being somewhat linear to being more nonlinear, or (more likely) that over the 12 years that separate today from Cont et al.'s work (and over the 4 years from Silantyev's), the potential edges given by this imbalance-price change relation may have been exploited by traders and partially reflected/incorporated by prices.

Since the R-squared on the model using TFI as a sole regressor appears to be increasing with the time frame, longer periods have been considered and the final result is that the R-squared increases up to the 7-hour time frame (55.98% R^2). It then starts deteriorating when considering 8-hour periods and longer.

The best single-regressor linear model thus appears to be the one considering 7-hour TFIs. However, the model considering the impact of both OFIs and TFIs has systematically better linear fit, and in particular the common impact of 10-minute OFIs and TFIs explains 56.30% of the 10-minute change in mid-price. The congruence with Silantyev's findings is probably due to the current fee structure: maker fees (to post limit orders) at execution are on average lower than taker fees[30], which renders market (taker) orders more expensive and thus potentially more informative as they might better represent trader intention compared to limit orders. This might have intuitively been the case during the no-fee period as well, with the main difference being that taker orders were more expensive as a result of the bid-ask spread instead of the higher taker fees. One crucial difference must be considered, though: although across the two periods market takers still paid more than makers to get in and out of trades, makers went from gaining the bid-ask spread (i.e., reducing the total cost of their trades by the bid-ask spread amount) to having to pay to enter the market (albeit still less than takers). This drastic change has potentially had two major effects on trading activities on Binance:

- First, a liquidity drought (showcased in Figure 14), as many makers might not have any incentives to post limit orders anymore.
- Second, an increase in the explanatory power provided by OFIs compared to the no-fee period. Extending Silantyev's concept, if posting limit orders suddenly becomes more expensive, then it would be reasonable to expect an increase in these orders' informativeness, as traders posting limit orders mainly because of the foreseen bid-

[30] This is a normal situation for almost all fee-based exchanges, and configures a rather successful method for ensuring an appropriate level of liquidity without the need for designated market makers.

ask spread gain will inevitably leave the market. This could then result in adverse selection against a subset of noise traders.

To verify the validity of the second point, a comparison of explanatory power of OFIs before the fee structure change and after is in order, with results in Table 7.

The 0-fee period sees a dramatic decline in explanatory power for OFIs, effectively being able to explain at most 5% of each mid-price change. These results make intuitive sense and confirm what highlighted above: in presence of a significant and volatile bid-ask spread (as pictured in Figure 17), limit orders are far less likely to be filled compared to market orders. An impatient market buyer would be immediately filled some cents to a couple of USDT above the current mid-price, on average. A limit buyer would be forced to post an order some cents to a couple of USDT below the current mid-price, without great confidence of being filled. If an informed buyer would then want to enter the market at short notice, this difference will likely result in the choice of a market order. On average, then, one can expect that during the 0-fee period OFIs had less explanatory power compared to the current environment. As far as TFIs go, data is insufficient to run a similar comparison. However, it would be reasonable to expect quite mixed results: differences in explanatory power of TFIs would be a function of how much more (or less) expensive taker orders were during the no-fee period and after. Since the bid-ask spread during the no-fee period was significantly volatile, informativeness of taker orders was potentially higher in periods when the bid-ask spread was high enough to render taker costs higher than they currently are, and lower in periods when the bid-ask spread was low enough to render taker costs lower than they currently are.

Table 7 - Linear Regression of mid-price changes on OFI, comparison Before vs After the fee structure change

	Mid-Price Regressed on OFI							
	***Before* March 15, 2023**				***After* March 15, 2023**			
	α	β_{OFI}	p β_{OFI}	R^2	α	β_{OFI}	p β_{OFI}	R^2
1sec	0.0221	19.7767	0.0000	1.02%	0.6775	17.9619	0.0000	14.14%
10sec	0.3461	34.3766	0.0000	2.10%	7.2019	20.3229	0.0000	26.19%
1min	2.5058	42.7023	0.0000	3.02%	39.5303	18.6759	0.0000	30.69%
10min	27.8345	48.4102	0.0000	4.04%	281.3032	13.5135	0.0000	21.55%
1H	172.6350	49.8971	0.0004	5.03%	1049.6956	8.7653	0.0000	11.15%

3.2.3. Testing for Structural Breaks

To verify that splitting the time series around March 15th is a correct approach to assess the relationship between OFIs and mid-price changes, a Chow test is in order (Chow, 1960). Chow's diagnostic procedure tests for coefficient equality between two separate regression models. Intuitively, if a dataset does not incorporate any structural break, coefficients returned by different regressions on various splits of that dataset should not be significantly different from one another. If, instead, coefficients resulting from regressions run on different portion of the same dataset were to be significantly different, that would be a rather telling indicator that data dynamics are not constant along the entire set of datapoints analyzed.

Chow's test is particularly popular in time series analysis and fits well in the case currently considered. The dataset containing mid-price changes and OFIs is split into two parts, specifically on March 15th, to check if the OFI coefficient is significantly different between the two periods. In case a significant difference was to be found, Chow' test would not irrefutably link that difference to the change in fee structure adopted by Binance, rather, it would simply state that the coefficients are significantly different between the two periods. The statistician running the test would then simply have a reasonable belief that the cause of such difference in parameters ought to be linked to Binance's policy instead of other events that might have occurred around the same time, without being as salient.

The original OFI regression presented above can be analytically represented as follows:

$$y_t = \alpha + \beta OFI_t + \epsilon_t$$

The original dataset is split on t_{split} = March 15th, 2023 and two different regressions are run on each of the two resulting datasets:

$$y_{t|t<t_{split}} = \alpha_1 + \beta_1 OFI_{t|t<t_{split}} + \epsilon_{t|t<t_{split}}$$

$$y_{t|t>t_{split}} = \alpha_2 + \beta_2 OFI_{t|t>t_{split}} + \epsilon_{t|t>t_{split}}$$

And the null hypothesis $H_0: \alpha_1 = \alpha_2,\ \beta_1 = \beta_2$ is tested, assuming i.i.d. normal errors, by computing the following F-statistic:

$$\frac{(S_C - (S_1 + S_2))/k}{(S_1 + S_2)/(N_1 + N_2 - 2k)} \sim F_{k,\, N_1+N_2-2k}$$

Where S_C is the sum of squared residuals (RSS) from the full original regression, S_1 and S_2 are respectively the RSS from the first and second split regressions, N_1 and N_2 are the number of observations in each split dataset and k is the number of parameters in each regression (in this case, $k = 2$).

Running the test in Python returns an F-statistic of 1977.61 with a largely sub-1% p-value, which strongly suggest rejecting the null hypothesis. This confirms the preliminary observation made while inspecting the OFI time series chart. As a result, splitting the dataset and running separate regressions is not only informative, as it quantifies the difference in OFI informativeness before and after the change in fees, it is also one of the few correct ways of dealing with Binance data from the analyzed period. As a matter of fact, Chow's test results confidently say that analyzing this dataset by running one single linear regression would be an incorrect approach and would provide distorted coefficients. The approach used above is thus appropriate.

Turning back to results after March 15th, a somewhat powerful linear relationship is certainly an interesting finding. However, analyzing contemporaneous relationships is not advantageous if development of operational implementations is the objective. To make this relationship actionable, one should be able to predict it over time. Unfortunately, the explanatory power of OFIs and TFIs when considering one-step-ahead price changes is basically nil, as far as the dataset used in this analysis is concerned. However, if OFIs and TFIs are autocorrelated in time as suggested by earlier claims of "order flow memory", one could build an autoregressive model, quantify one-step ahead forecasts of OFIs and TFIs and measure the correlation of those forecast with one-step-ahead price changes. If that came out to be significant, those forecasts might successfully be implemented in a trading strategy; food for thought.

This analysis casted some light on how cryptocurrency markets can be disruptive and dynamic, with an ever-evolving microstructure. From arbitrary, structure-altering decisions (such as Binance's shift in fee policy) to scandals, manipulations and collapses resulting from the current lack of regulation, studies may find contrasting results even if not too far apart in time, with mixed outcomes being quite frequent.

One important point to reiterate is that this analysis has only focused on one exchange because of the interesting structural break it organized. In general, aggregating market data

from as many exchanges as possible is a highly suggested practice if generality of results is the main objective. Generality that, by construction, is extremely difficult to achieve when studying such a decentralized and heterogeneous landscape as cryptocurrency markets.

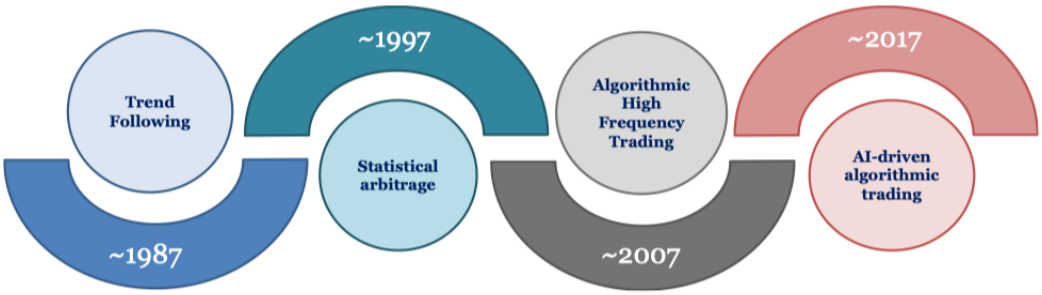
Assuming linear relationships in financial markets is a very inviting idea, however, complex systems will be more likely to showcase non-linear relationships. One modern way to capture complex, non-linear patterns is to use AI and ML solutions. The next Chapter will explore the capabilities of AI and ML algorithms in order flow analysis, presenting one simple, original, implementation of ML in analyzing the same dataset this section has tried to model.

Chapter IV Machine Learning Implementations For Order Flow Analysis In Cryptocurrency Markets

The disruption and fast-paced evolution characterizing cryptocurrency markets have contributed to the birth of novel techniques and tools used by analysts and traders, with Artificial Intelligence (AI) and Machine Learning (ML) models seeing particular success. This chapter will explore these innovative technologies: their strengths, their caveats, and opportunities for future developments. An original implementation of a simple ML model will be presented, as a mean of comparison with the traditional OLS modelling showcased in Chapter III.

4.1. Machine Learning: a Brief Introduction

Figure 19 - Historical developments in trading and the advent of AI



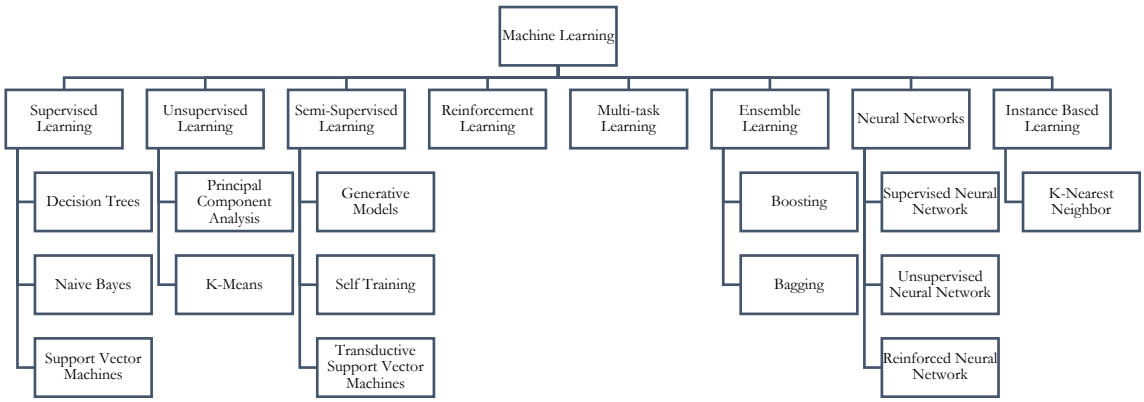
Source: OECD, 2021

"Machine Learning" (ML) identifies a subset of Artificial Intelligence (AI) that involves the use of algorithms and aims at improving computer system performance in carrying out specific tasks, through learning from data. Performance is usually evaluated in the form of predictive precision, robustness and scalability, although measuring ML performance often involves more complex considerations. ML allows for optimization, minimizing the errors arising when predicting complex patterns in the data. The main advantage of ML consists in the tweakable degree of autonomy with which an algorithm can learn how to correctly conduct a task, resulting in a trained model that can operate with reasonable levels of human supervision and maintenance, sometimes reaching full autonomy. An additional, not negligible advantage of ML is the capability of modelling complex nonlinear relationships without major human intervention, thus making pattern recognition tasks less cumbersome for data scientists and statisticians. Applications of ML span over RSa vast array of fields: from data mining to image processing and recognition, to predictive analytics and complex

statistical problems. As shown in Figure 20, this technology has extremely flexible application possibilities as a result of the large number of algorithms available today.

For an ML model to be properly trained, vast amounts of data are needed: in fact, ML and the concept of "Big Data" are strongly related and intertwined, as the latter enables the existence of the former. One burden related to ML modeling, particularly when dealing with fields where this technology has never been applied before, consists in the time consuming data collection, preparation and manipulation process that is a crucial step before feeding the model. Inadequate data granularity, unordered datapoints and totally unstructured datasets pose the risk of creating a poor model, according to the secular "garbage in, garbage out" principle. Advancements in the field, however, are making it possible for algorithms to find patters even in highly unstructured datasets, e.g. via unsupervised or semi-supervised neural networks.

Figure 20 – Types of Machine Learning algorithms



Source: Mahesh, 2020, adapted

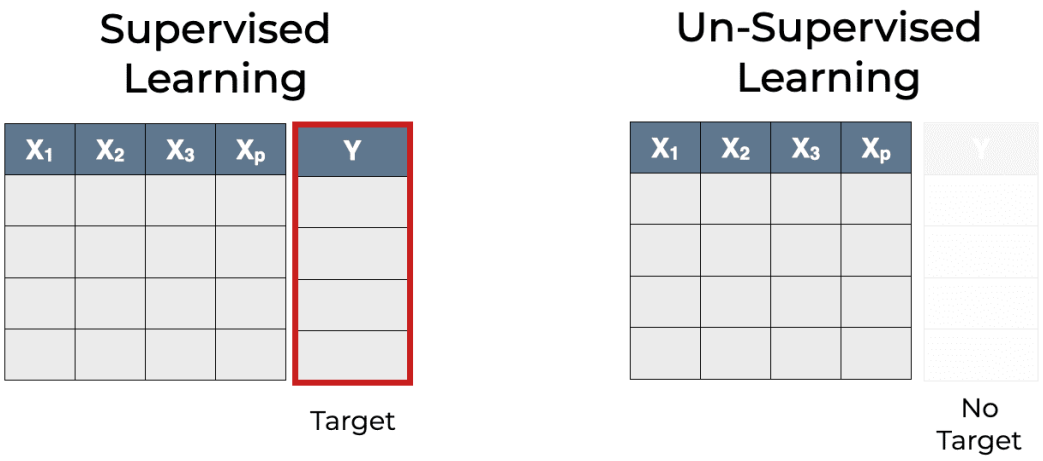
4.1.1. Supervised vs Unsupervised Learning

Supervised and unsupervised learning are two key paradigms in ML, each with their own set of methodologies and goals. Supervised learning algorithms learn from labeled datasets, including the input data and the corresponding output values. By studying the fundamental patterns and correlations between inputs and outputs, algorithms are trained to generate predictions or classifications based on new datapoints. Unsupervised learning, on the other hand, operates without labeled data and seeks to find hidden structures, patterns, or clusters,

rather than predicting values. Algorithms investigate the intrinsic properties of the data and organize it into meaningful clusters, often revealing previously unnoticed insights.

Intuitively, these sets of algorithms are named after the degree of human supervision needed for their correct training: supervised learning algorithms require supervision in the form of providing an output column, while unsupervised algorithms autonomously find hidden structures in the data, without the need for human-provided "true" outputs. As already outlined above, supervised and unsupervised algorithms do not provide the same type of output: supervised algorithms usually provide numerical predictions or categorical classifications based on already pre-defined possible classification sets, while unsupervised algorithms organize data in as many clusters as it is appropriate, based on the data structure at hand. Figure 21 presents an example of the different datasets used in supervised and unsupervised learning problems.

Figure 21 - Supervised vs. Unsupervised learning datasets



Source: Ebner, 2021

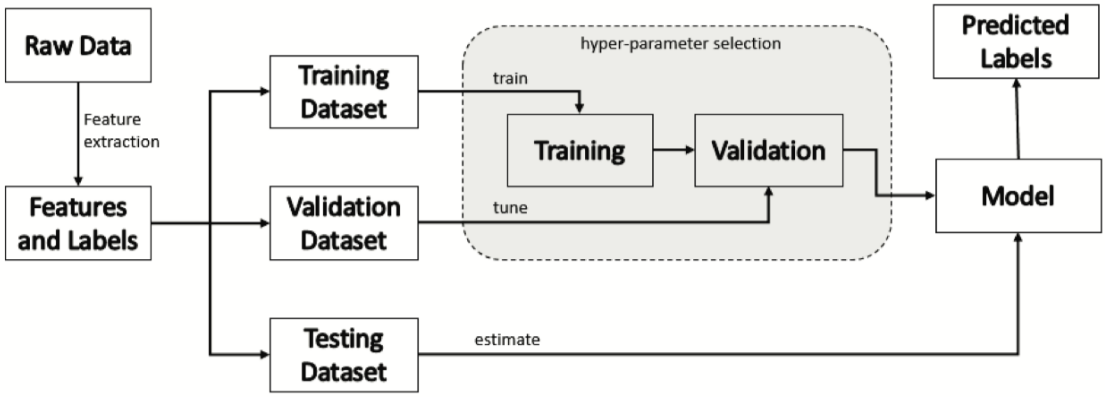
Figure 22 shows a typical workflow used in supervised learning modelling, while Figure 23 shows a typical unsupervised learning pipeline.

Features from supervised and unsupervised algorithms can be merged together in instances of partially labeled data, forming so-called "semi-supervised" learning algorithms.

As far as cryptocurrency order flow analysis goes, thanks to the high granularity of order book data available, ML models have been flourishing and a vast array of implementations have been presented in the literature. For predictive implementations, supervised learning is adopted without caveats of data availability, while unsupervised learning is usually adopted

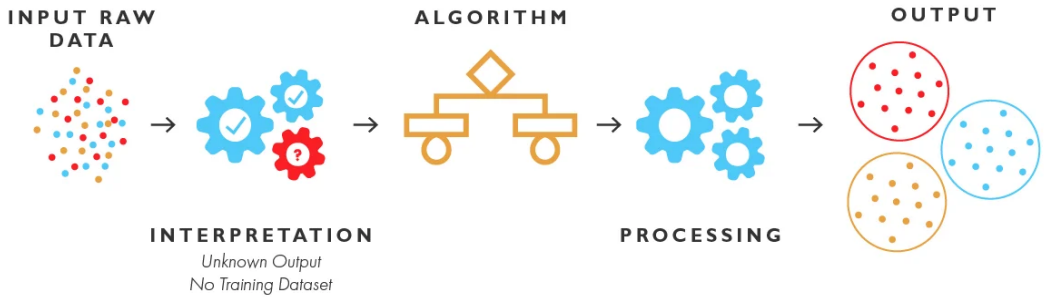
where the aim is not price prediction, rather finding hidden patterns and dynamics in order book data. Section 4.2 highlights some relevant work presented in the field.

Figure 22 - Supervised learning workflow



Source: Basavaraju et al., 2019

Figure 23 - Unsupervised learning pipeline



Source: datixinc.com

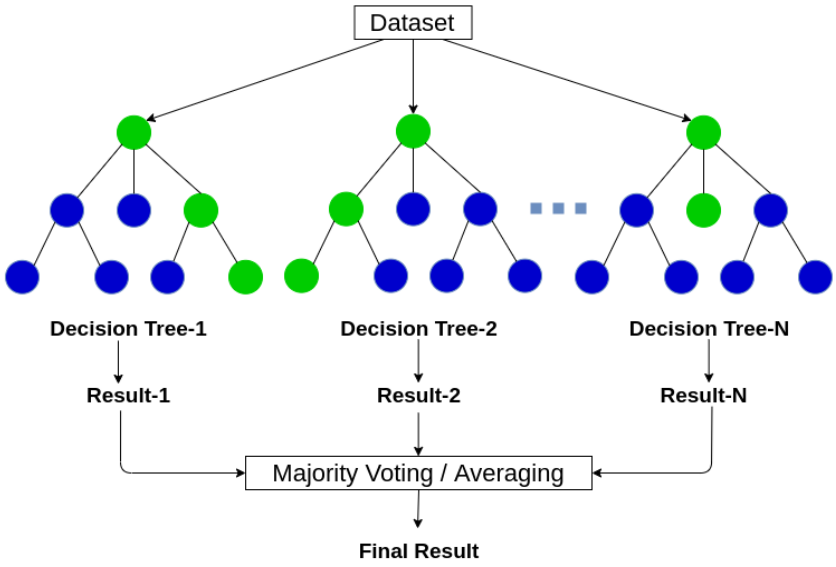
4.1.2. Improving Accuracy, Preventing Overfitting: Ensemble Methods

With ML models being extremely adaptable and able to scout hidden patterns in data, one of the most feared issues by supervised ML developers and analysts is so-called "overfitting". Overfitting consists in a model internalizing too much of the noise present in the data used for training, and as a result performing well in the training phase while performing poorly in the testing phase and in production. Put differently, an overfit model does not generalize well from observed data to out-of-sample data (Ying, 2019). Among other things, ensemble methods propose to reduce overfitting by combining multiple

algorithms to improve accuracy and robustness, also reducing predictive errors. Moreover, Dong et al. (Dong et al., 2020) point out that traditional ML methods may showcase inadequate performance when dealing with particularly complex, imbalanced data with high dimensionality. This makes a strong case for ensemble methods, which aim at using the combined power of multiple algorithms to improve accuracy and robustness, reducing predictive errors. Although ensemble learning has mostly been explored in the field of classification (Krawczyk et al., 2017) its power can be extremely useful in the realm of regression and prediction problems and, as a part of it, order flow analysis.

Ensemble methods consist in training multiple ML algorithms (such as decision trees) on different data substets, getting predictions from each algorithm (also called "weak learner") and merging those predictions to form one single output. In classification problems, the most common merging method is the so-called majority vote. Classification results are pooled together and the class resulting from the majority of the algorithms is chosen as the final output, as Figure 24 shows. Regression ensemble methods work in a similar way, with the only difference that the final output is usually a weighted average or a median of the values predicted by the weak learners (Krawczyk et al., 2017).

Figure 24 – Ensemble learning framework



Source: engineersgarage.com

Ensemble methods can be divided into three main classifications: bagging, boosting and stacking, presented below.

Bagging

Bootstrap aggregating, abbreviated as "bagging", is mostly used in classification and regression problems. This ML method increases model accuracy and reduces overfitting by using multiple "weak" decision trees and merging their single predictions.

The bagging process can be divided into bootstrapping and aggregation.

Bootstrapping is a sampling strategy that uses the replacement approach to obtain samples from the whole population set. Sampling a dataset with replacement contributes to the randomization of the selection process. The resulting samples are run through weak learners to obtain multiple results.

Aggregation is used to combine these results and randomize the output. Predictions will be inaccurate without aggregation since every outcome will be different. The aggregation process can be based on probability bootstrapping procedures or include all outcomes from weak learners, i.e. through a simple arithmetic mean.

Bagging is useful because it combines weak base learners to generate a single, more stable strong learner. It also reduces variance, which reduces model overfitting. It has, however, the disadvantage of being rather hardware intensive. Powerful machines would then be needed to apply bagging methods to high-dimensionality and high-granularity data.

Boosting

Boosting is based on the sequential fitting of a dataset by a series of weak learners. Starting from an initial learner, each subsequent learner will fit the same dataset with the inclusion of the prediction error produced by the preceding learner. Based on the error function of choice applied to all models (mean squared error, root mean squared error, etc.) this ensemble method effectively "boosts" performance by feeding the prediction error of one model into the computations of a following one, thus improving predictive power.

The two most popular boosting implementations use decision trees as weak learners and are called AdaBoost (adaptive boosting) and XGBoost (extreme gradient boosting).

AdaBoost gives different weights to the output of each weak learner. Specifically, it assigns larger weights to previously misclassified results, so that subsequent learners can shift focus from the correct outputs towards the wrong ones and adapt their predictions accordingly, reducing classification errors.

XGBoost is probably the best known ensemble method and has a wide array of applications thanks to its flexibility. Similarly to AdaBoost, XGBoost gives more weight to the most challenging datapoints. The main feature setting this method apart is the way it

evaluates the chosen loss function: it includes a gradient descent optimization procedure where model parameters are iteratively adjusted to minimize the gradient of the loss function. Moreover, larger weights are given to weak learners showcasing better performance on the training data. This might be a potential source of overfitting, so as a countermeasure XGBoost includes a regularization parameter to prevent excessive complexity.

XGBoost is generally preferred because of its regularization capabilities and the possibility to be run via parallel computing (i.e., running multiple instances of the same algorithm on multiple CPU cores to make computations faster), both aspects where AdaBoost is lacking (Bentéjac et al., 2021). This ensemble method will be used in Section 4.3, presenting a machine learning twist to the analysis carried out in Chapter III.

Stacking

The ensemble methods presented above use a rather simple approach in the results combination phase (majority voting, weighted averages, simple averages).

Stacking takes results combination one step further and consists in training an ML model on how to best combine the outputs produced by weak learners[31]. When high levels of predictive accuracy are needed, data scientists may turn to stacking methods. However, these techniques can prove computationally intensive and can pose data leakage concerns. As a result, careful cross-validation steps are needed to ensure unbiasedness in these methods.

4.2. Literature Review on Machine Learning Models for Order Flow Analysis

AI and ML tools have been used for some time in cryptocurrency time-series price analysis. Various implementations such as long-short–term memory and generalized neural networks (Lahmiri and Bekiros, 2021), neuro-fuzzy controllers (Atsalakis et al., 2019), ensemble algorithms and artificial neural networks (Mallqui and Fernandes, 2019), decision trees (Huang et al., 2019), support vector machines (Mallqui and Fernandes, 2019; Aggarwal et al., 2020) and deep feed-forward neural networks (Lahmiri & Bekiros, 2021) have been presented in academic research.

The world of crypto order flow analysis through AI and ML models is in its evolution phase, with deep, reinforcement learning and neural networks techniques being the most popular applications. Jha et al. (2020) achieved 71% walk-forward accuracy on a 2-second time horizon for bitcoin's spot price predictions based on LOB data from the Coinbase

[31] Some stacking methods rely on simpler models such as linear regression.

exchange. They advocate for the use of deep learning techniques based on cryptocurrency LOBs containing more complex phenomena compared to traditional markets, as a result of higher and less stable latencies. They found improvements in model performance when training spanned longer periods and noticed stickiness in mid-price predictability for periods up to one minute. They also suggest that this predictability may come from the model likely capturing the persistence of order book imbalances on the 5 best bid and ask levels. The authors also point to cryptocurrency order books showcasing a "memory" of at least 10 seconds on Coinbase. Wray, Meades and Cliff (Wray et al., 2020) demonstrate that an appropriately designed deep learning neural network (DLNN) can autonomously learn to act as a high-performing algorithmic trading system merely by passive observation of an existing trader's transactions. The author's model trains by receiving a trader's order as input and matching it to the state of the level-2 LOB at the time of the order. The model will then adapt to LOB states, submitting orders based on the trader's behavior during similar LOB states observed during the training phase. Kolm, Turiel and Westray (Kolm et al., 2021), applied artificial neural networks to NASDAQ order books and showed an interesting finding that has space in crypto markets, too: training models on order book states, which they describe as a complex non-stationary multivariate process, yields poorer performance compared to training with order flow, which in turn represents stationary quantities derived from the LOB. Finally, Schnaubelt (2022) used deep reinforcement learning to analyze over 3.5 million order book states from crypto exchanges Bitfinex, Coinbase and Kraken, successfully training a model that could learn optimal limit order placement strategies and proving the case for using deep reinforcement learning techniques in the realm of execution optimization, which could reveal strategic for large investors and asset managers.

4.3. Machine Learning in Practice: Order Flow Analysis on BTC/USDT

The same data used and manipulated in Chapter III will now be used to test the potential of ML techniques. Because of computational power concerns, only data from after March 15th, 2023 will be used. Since OLS regression revealed that 10-minute OFIs and TFIs combined had the highest R^2 when explaining mid-price changes, only this sampling rate will be used for this practical example. The following sections will go through the process of choosing and training an ML model for a regression problem, comparing results with the traditional OLS approach. Differently from Chapter III, data will be split in a training and a testing set with the popular 70/30 partition, so that model performance will be compared and evaluated mainly on the testing set. This analysis and the related charts are all original

work. The Python code for data collection, manipulation, training, testing and charting can be found in the Appendix.

4.3.1. Model and Feature Selection

Model selection should be mainly driven by the type of problem to be solved and the data structure at hand. In this case, data has already been cleaned to be used in Chapter III and is orderly arranged in tabular form, with one column dedicated to the output variable (the mid-price change). This first observation suggests turning to supervised learning techniques and leave unsupervised ones aside. Mind, unsupervised techniques can still be applied to the data at hand, however their application is quite beyond the main objective of this analysis (which consists in fitting the data to find a relationship between OFIs, TFIs and mid-price changes, and training a model that will be able to estimate mid-price changes based on OFIs and TFIs). For example, unsupervised techniques might be successfully used to find hidden patterns in order book data as presented in Table 5.

Moreover, with mid-price changes being a numerical value, classification techniques are out of scope, too. Classification might rather be applied if one simply wanted to predict the sign of the mid-price change as a function of TFIs and OFIs.

Thus, since the problem at hand can be tackled by a supervised algorithm, and the output variable is numerical and the objective is to model it via regression, one viable ML approach is to use the XGBoost algorithm. As already outlined in Section 4.1.2, XGBoost is an ensemble method that leverages the combined power of multiple weak decision trees, incorporating prediction errors in each subsequent iteration. This model is promising as it will potentially increase predictive accuracy, robustness and will be more adaptable in front of new potential changes in underlying data dynamics.

As highlighted above, the features this model will be using are 10-minute OFIs and TFIs. Three additional features will be added (and will be included in the OLS model, too, for a fair comparison): minute of observation (from 0 to 59), hour of day (from 0 to 11) and day of week (from 1 to 7). This is to mainly see whether the ML model will give weight to them in the training process, as it would indirectly hint to a possible confirmation of the findings by Caporale and Plastun (2019) and Aharon and Qadan (2018) outlined in Chapter II, concerning weekly temporal patterns in bitcoin prices.

4.3.2. Training and Avoiding Overfitting

One key aspect to be wary of when training a ML model is overfitting. In fact, simply fitting the default version of XGBoost to a dataset will almost certainly result in overfitting, if a sufficient number of estimators/learners are used. One of the easiest and most popular ways to check for overfitting is to split the dataset at hand into a training and a testing set, subsequently measuring and comparing prediction error metrics between training and testing data. If prediction errors appear to be significantly lower in training data compared to testing data, the model is most likely suffering from overfitting issues. To reduce the risk of overfitting, XGBoost includes a vast array of parameters that can be tweaked with the aim of improving generalization of results. The process of tweaking parameters in a ML model is referred to as "hyperparameter tuning". The following code snippet (also available in the Appendix) showcases a way in which XGBoost hyperparameters can be automatically tuned without the need for human trial and error:

[23]:
```
from skopt import BayesSearchCV
from skopt.space import Integer, Real
from sklearn.pipeline import Pipeline

np.int = int

pipe = Pipeline(steps = [('clf', xgb.XGBRegressor(random_state = 1))])

search_space = {
    'clf__max_depth': Integer(2,8),
    'clf__learning_rate': Real(0.001, 1.0, prior='log-uniform'),
    'clf__subsample': Real(0.5, 1.0),
    'clf__colsample_bytree': Real(0.5, 1.0),
    'clf__colsample_bylevel': Real(0.5, 1.0),
    'clf__colsample_bynode': Real(0.5, 1.0),
    'clf__reg_alpha':Real(0.0, 10.0),
    'clf__reg_lambda': Real(0.0, 10.0),
    'clf__gamma': Real(0.0, 10.0)
}

opt = BayesSearchCV(pipe, search_space, cv = 10, n_iter = 50, scoring =␣
 ↪'neg_mean_absolute_error', random_state = 1)

opt.fit(X_train, y_train)
```

Specifically, hyperparameter are tuned in this example leveraging Bayesian optimization and cross-validation, using the mean absolute error as a metric to judge model performance. All keys in the search_space dictionary are hyperparameters, while the relative values delineate the perimeter inside which those hyperparameters will be tweaked by the Bayesian optimization algorithm. The most important ones are the maximum depth of each tree, the learning rate of the algorithm (how much of the previous learner's error is fed forward into the next learner), alpha and lambda (regularization hyperparameters). The best resulting model has the following hyperparameter values:

```
[32]: opt.best_params_

[32]: OrderedDict([('clf__colsample_bylevel', 1.0),
                   ('clf__colsample_bynode', 1.0),
                   ('clf__colsample_bytree', 1.0),
                   ('clf__gamma', 10.0),
                   ('clf__learning_rate', 0.11217197417799707),
                   ('clf__max_depth', 2),
                   ('clf__reg_alpha', 10.0),
                   ('clf__reg_lambda', 2.6977042269121503),
                   ('clf__subsample', 1.0)])
```

Hyperparameter tuning proved to be effective in this example: the mean absolute error (MAE) for the standard, non-tweaked XGBoost model fit to the analyzed dataset is equal to 362.15 for the training set and 574.29 for the testing set. This clearly denotes overfitting, as the model significantly struggles to generalize from training to testing data. After tuning, the MAE is equal to 511.35 for the training set and 532.34 for the testing set. MAEs are hence closer, representing reduced overfitting. An increased MAE for the training set is not necessarily bad, if the lower initial MAE resulted from unnecessary noise internalization by the model.

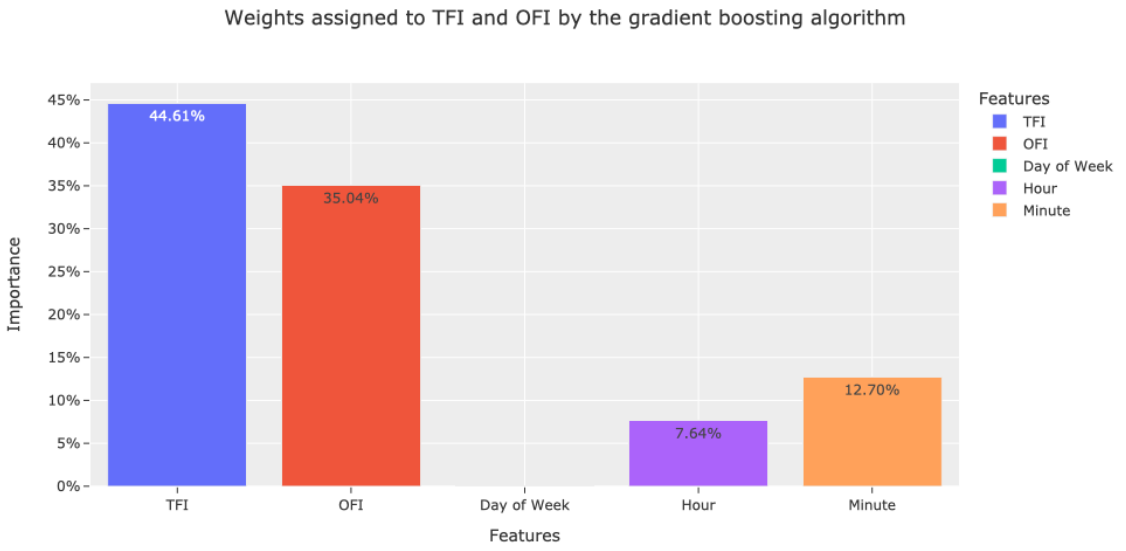
MAE is used as the error metric of choice in this example, as it is rather popular when measuring ML models performance. MAE is computed as follows:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n} \quad (2)$$

Where y_i is the i-th prediction and x_i is the corresponding observed value. Taking the absolute difference of the two yields the so-called absolute error $|e_i|$, which divided by the number of observations returns the MAE.

XGBoost suffers of one of the most frequent caveats in ML: its processing appears to happen in an opaque "black box" that is not readily analyzable by a human data scientist. However, the model produces valuable insights that can help one understand how its decision and computation process comes to realization. One of the most important insights is feature importance: the model, after being trained, returns an array of weights with one weight for each feature it has been trained on. In this specific case, the model has been trained on five features in total, and Figure 25 shows the importance or "weight" assigned by XGBoost to each feature during the training process.

Figure 25 - Feature importance assigned by XGBoost

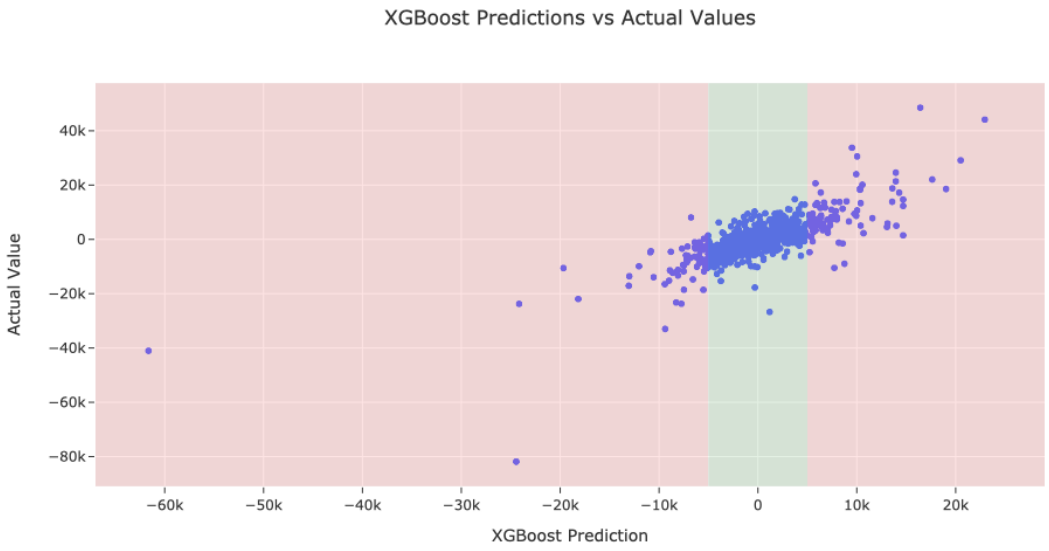


Original visualization via Python

As expected, XGBoost gives more weight to TFIs than OFIs. This is in line with OLS findings from Chapter III. Interestingly, the minute and hour features are given non-negligible importance, while the day of week is not considered at all. These weights are the result of a 100-tree model optimized over 50 iterations with 10 cross-validation steps. Intuitively, changing any of these three variables will result in a more or less significant difference in hyperparameter values and finally produce different weights. This specific setup appears to be overall satisfactory, as it showcases reduced overfitting and, as Figure 26 shows, out-of-sample predictions are accurate for price changes ranging from -5,000 to +5,000. Higher variance can be noticed when predicting outliers: this can be a possible area of improvement if further modelling was to be carried out.

XGBoost predictions are linearly correlated at 61.45% with observed mid-price changes when the observed mid-price changes are between -5,000 USDT and +5,000 USDT. This is a rather satisfactory result when considering that the edge provided by OFIs and TFIs has most likely been somewhat eroded since the first studies discovering their predictive power.

Figure 26 - XGBoost predictions vs actual values. The green shaded area showcases the highest correlation between the two



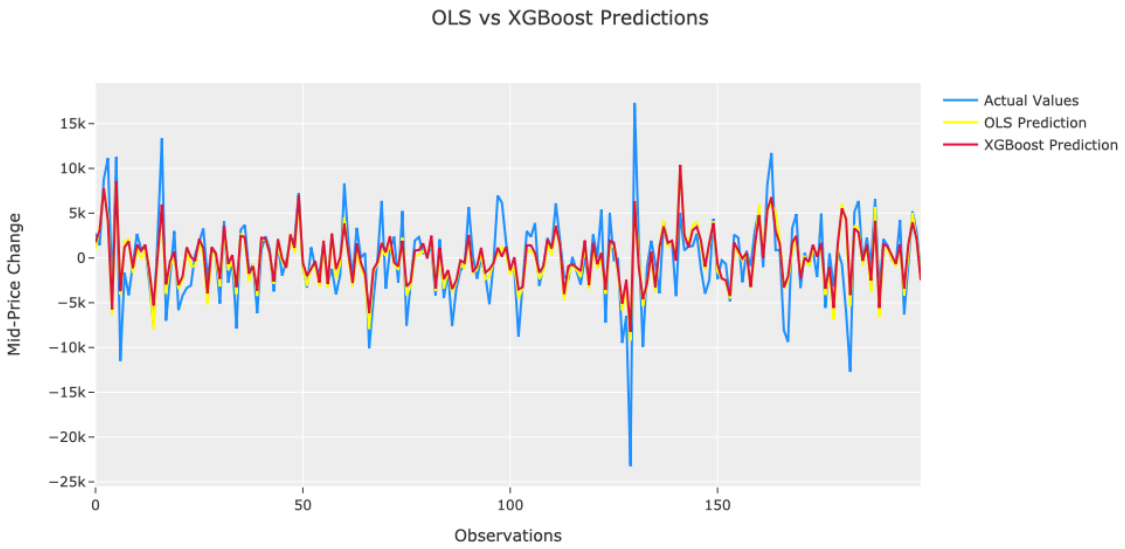
Original visualization via Python

4.3.3. Results: OLS vs ML

XGBoost brings an improvement over the traditional OLS method presented in Chapter III. First, testing-set MAE is 532.34 with XGBoost versus 595.69 with OLS. Second, XGBoost has the potential of being more flexible as it is very computationally efficient in front of higher dimensionality, while maintaining robustness. Hyperparameter tuning allows for a great degree of model personalization. Finally, regression is only one of the possible applications of XGBoost: ranking and classifications are two additional scopes where this innovative ensemble method can prove extremely useful.

Figure 27 shows a plot of part of the testing set, together with OLS and XGBoost predictions. Overall, as the difference in MAE might have suggested, there are no significant differences between the two sets of model predictions, apart from an arguably slightly better performance of XGBoost during more volatile periods. This is mainly due to the fact that, as outlined above, the relationship between OFIs, TFIs and mid-price changes is linear and does not showcase evident signs of non-linearity. XGBoost might have had sensibly better performance compared to OLS if the underlying relationship was non-linear.

Figure 27 - XGBoost vs OLS predictions



Original visualization via Python

Overall, this XGBoost experiment proves the viability and potential of ML models in modelling order flow dynamics. Further developments are to be expected in this ever-evolving landscape, producing new opportunities while, on the other hands, posing some risks and caveats linked to the very nature of these new techniques. The following section touches on some of the concerns analysts and traders might want to be wary of when implementing ML models.

4.4. Risks and Concerns of ML Models in Algorithmic Trading and Analysis

With finance being a low signal-to-noise ratio environment (De Prado, 2018), the use of ML techniques must be accompanied by an appropriate degree of caution, particularly if these models are to be implemented in highly volatile markets such as cryptocurrency. The process involved in creating an actual, fully fledged ML-enabled trading strategy is extremely complex: data sourcing, collection, processing, manipulation, hardware infrastructure, software development, feature engineering, model tuning, testing, validation, cross-validation, backtesting, production and maintenance are tasks calling for the work of an entire team of professionals, rather than one single analyst or trader. These steps add complexity and pose possible sources of operational risk when moving from one process to the other, potentially leading to inadequate model production and, in the end, wasted time and money. One crucial aspect when developing ML strategies is having a proper team and

implementing a meta-strategy made up of independently evaluated sub-tasks (De Prado, 2018).

ML modelling should also not have as a sole objective to create compelling backtests by iteratively fitting models to a dataset up until positive returns appear. This could easily result in confirmation bias and data leakage, as this process might push developers to create overfit models with the sole purpose of maximizing backtest performance. The resulting backtest would then be a "survivor" among all the possibly implementable strategies over the analyzed time horizon, by construction (thus also including survivorship bias in the mix). What should rather be analyzed is the importance assigned by the ML models to selected features over time. Intuitively, this importance might change as a result of historical events or other market dynamics. The changes in feature importance should be closely monitored to ensure consistency in predictive capabilities.

The wider application of ML techniques in a trading context might also result in a changed trading activity cycle: human traders are used to analyzing price dynamics based on time. However, the advent of computer-based techniques, which do not necessarily follow time intervals as close as they follow actual trading activity (such as amount of volume exchanged, number of increased or decreased ticks, etc.), is shifting importance from time to other variables. This might catch traditional traders off-guard when price will stop behaving in a neat time-organized fashion (De Prado, 2018). The same rule of caution applies to academic research and, in a way, applies to the examples presented in Chapters III and IV, too.

Traders and analysts should also avoid putting too much faith in ML models capabilities: supervised learning models, for instance, still tend to require stationary datasets to provide the best estimates. One should resist the urge to pass unprocessed, uninspected data into supervised learning algorithms, as the cost of this practice would readily materialize via the garbage-in, garbage-out paradigm.

Many ML funds suffer from at least one of the outlined caveats, resulting in frequent failures to deliver the promised performance (De Prado, 2018).

The black-box paradigm however still remains the most widespread caveat for virtually all ML implementations. Many users of ML models will neglect the algorithm's inner workings and limit their work to fitting datasets with a specific model. This poses interpretability concerns further down the line and might result in a very bad look for an ML-enabled company if it finds it virtually impossible to explain to a client the underlying process backing a specific decision.

4.5. Possible Implications of ML Use on Market Efficiency

The OECD has pointed out some potential effects of widespread ML usage on financial markets in their 2021 Artificial Intelligence, Machine Learning and Big Data in Finance report (OECD, 2021). Specifically, the potential use of similar models by a vast pool of traders might boost contagion and unexpected correlations, particularly in periods of market stress. Herding behavior and the paradigm of a "one-way market" might result in unstable structure and price dynamics, increasing volatility. At the same time, ML implementations might erode existing edges, reducing arbitrage margins and reducing bid-ask spreads, thus benefitting retail participants. As a result, proprietary firms might have stronger incentives in opaquely disclosing information to retain an advantage, worsening the already lacking explicability and interpretability of many ML outputs. This lack of transparency might be intentionally used as an informational shield by spoofers and other market manipulators, as they might shift responsibility towards autonomous algorithms if inquired about questionable market practices. This adds to the potential caveats on accountability arising from autonomous computer systems. Moreover, since ML models are widely used by market makers and liquidity providers, nowadays market stability and efficiency go hand in hand with cybersecurity: malicious disruptions of service targeting major liquidity providers may cause wild market instability through illiquidity, effectively hindering market solidity.

Overall, ML implementations could both boost and hinder market efficiency: time will ultimately determine the final verdict, as a function of how market participants will decide to adopt these novel technologies.

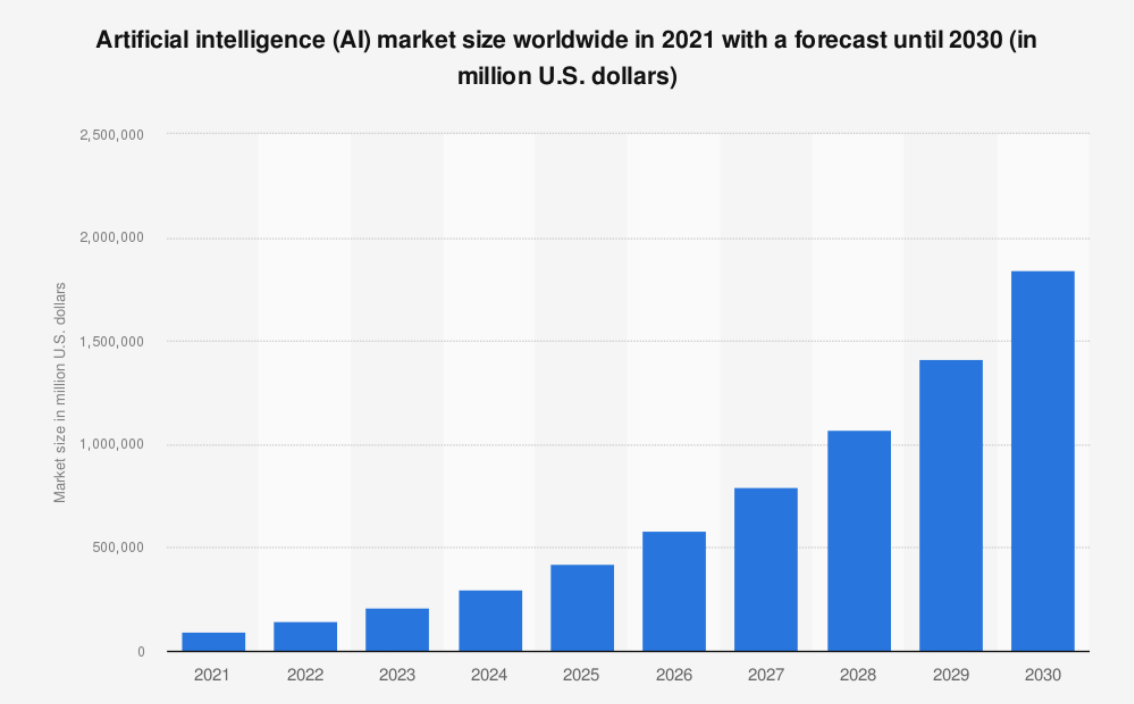
4.6. Future Trends for AI and ML in Cryptocurrency Markets

Artificial Intelligence and Machine Learning technologies are experiencing exponential growth in a wide set of application fields and are projected to reach a $1.8 trillion market size in 2030 (Statista, 2023b; Senn-Kalb and Mehta, 2023 see Figure 28). These innovations are set to be part of much of our lives and will become an integral component to the widest array of activities.

Joining two ever-evolving markets such as AI/ML and cryptocurrency makes for great growth potential and a great basis for future advancements that could spill over to adjacent

fields. The following subsections touch on some potential trends and drivers of future developments.

Figure 28 - Global artificial intelligence market size 2021-2030



Source: Next Move Strategy Consulting, © Statista (Statista, 2023b)

4.6.1. Advanced Trading Algorithms

One of the most prominent applications of ML in cryptocurrency markets revolves around trading systems: at this stage, as previously highlighted, the black-box paradigm hinders output interpretability and debugging of divergent algorithms. Moreover, current algorithms rely on extremely large datasets for training purposes, which can prove computationally intensive and prevent broader "retail" adoption. Future algorithm advancements might contribute to the democratization of ML use in cryptocurrency trading and, for the most part, in trading as a whole. It would be reasonable to expect, though, that based on the inherent digital and democratizing nature of cryptocurrency, these advancements will touch the world of crypto and DeFi first. One potential driver of such developments may be represented by transfer learning algorithms, which focus on results generalization and knowledge transfer across domains (Zhuang et al., 2020; Davchev et al., 2020). Transfer learning makes it possible to re-use knowledge from previous applications (called source domains) for slightly different problems (called target domains), thus

decreasing the need for vast amounts of training data. Transfer learning can be a valuable leverage in text classification, which is the underpinning of sentiment analysis, another valuable driver in the cryptocurrency space.

4.6.2. Sentiment Analysis

One of the greatest debates in the cryptocurrency landscape revolves around the intrinsic value of crypto assets. Many argue that this intrinsic value is basically nil, making a case for expectations, opinions and trust being the true determinants of crypto assets value. If that is the case, one way to interpret crypto price movements would be to explore investor's sentiment. However, with millions of market participants scattered all around the world, gauging real-time sentiment and translating it into actionable insights may prove challenging. ML applications to sentiment analysis mainly consist in classification of text corpuses retrieved from news articles, blogs, forums and social media posts. Phrases are evaluated based on the sentiment they appear to express (joy, disgust, trust, doubt and many others) and the entire corpus is given an aggregate classification. If many text corpuses convey positive sentiment at the same time, one could expect positive short-term returns on the crypto asset object of those texts. In fact, this strategy has vastly been applied in academia with rather positive outcomes (Colianni et al., 2015; Szabó, 2017; Valencia et al., 2019; Inamdar et al., 2019; Wołk, 2020). Future trends consisting of more efficient algorithms and more accurate classifiers can increase the informativeness of sentiment analysis and provide a better overall picture of participants intentions.

4.6.3. Explainable AI

Many AI and ML algorithms, such as Deep Neural Networks, do not have the capability of compellingly explaining their underlying inference process and results (Xu et al., 2019). Explicability, interpretability and transparency are of essential importance to end users, as model decisions might not always be intuitive or even rational at first hand. This brought to the birth of the broader explainable AI (XAI) field. Unfortunately, one of the most prominent findings at this stage is that model explicability is inversely correlated with model accuracy (Xu et al., 2019), with decision trees being the most explainable but least accurate models, and Deep Neural Networks being the least explainable but most accurate ones. AI researchers have lately been devoting some efforts in breaking the most complex black-box algorithms to ensure adequate disclosure to end users. Transparency design and "post-hoc" (or "ex-post") explanation are the two main courses of action as of now, with transparency

design being focused on actually explaining the inner workings of algorithms and post-hoc explanation focusing on output interpretation, justification and rationalization in an ex-post fashion.

In 2017, the Defense Advanced Research Projects Agency (DARPA) launched a $50 million project on explainable AI, with the aim of producing "glass-box" models that could be reasonably explainable to a technologically aware human being. This field is expanding and has the potential of shedding light on the complex functioning of AI algorithms, potentially resulting in increased market participation and adoption. Moreover, explainable algorithms could become a mandatory requirement for professional and institutional traders, to limit opportunistic adoption of opaque models to carry out damaging market activities such as manipulation.

4.6.4. Decentralized Finance

Classification algorithms have the potential to vastly improve security in the world of DeFi. Specifically, risk assessment and management processes can be rendered more efficient through ML-enabled borrower credit scoring and fraud detection.

Smart contracts can become more complex over time to include specific covenants and offer a truly tailor-made contract creation experience, with implementation, monitoring and breach detection having the potential of being improved by new AI systems.

4.6.5. Blockchain Interoperability

One of the greatest caveats of having multiple different blockchains for different use cases involves interoperability: in fact, blockchains have defined boundaries and implementing inter-blockchain processes can prove challenging and cumbersome. The simple task of swapping ethereums for bitcoins would involve a 2-step process if advancements in cryptocurrency conversion didn't come as far as they have. One should have converted ethereums for fiat currency and again fiat currency for bitcoins, incurring in market risk during the process and intuitively spending more in commissions. Novel ML and AI algorithms and their advancements can be leveraged by interoperability protocols to provide a more seamless inter-blockchain experience for end users, potentially reaching the stage in which most blockchains can interface with each other and complex multi-blockchain projects will really be able to take off.

This final Chapter delved into the world of Artificial Intelligence, with an exclusive focus on currently available Machine Learning methods and their potential in the cryptocurrency space. A simple practical Machine Learning application was proposed for Binance data on BTC/USDT, highlighting the possible improvements brought on top of more traditional OLS models. Finally, some notes of caution were in order in a context where complex models can be abused or not implemented correctly. Future trends are extremely promising and confirm the intuition that AI and ML are here to stay and gradually improve how humans interface with financial markets. With the multi-faceted nature of cryptocurrency, these models are set to improve end user experience and friendliness, playing a crucial role in crypto adoption and backing the expansion of this innovative digital landscape.

Conclusions

This thesis has explored the world of market microstructure and its application to cryptocurrency markets. The broader concept of market microstructure was introduced first, suggesting an exhaustive taxonomy of market participants. The focus then shifted towards types of orders and their impact on price discovery, emphasizing the importance of liquidity and order flow in market efficiency.

Moving on to cryptocurrency market microstructure, this thesis discussed the unique characteristics of cryptocurrencies and how they translate into decentralized transactions. It also proposed an original order flow analysis application to the Bitcoin/Tether US pair traded on Binance, testing the predictive power of order flow imbalances on mid-price changes and how it can change as a result of exogenous changes such as fee structure modifications. Results from the traditional OLS approach confirmed the explanatory power of both trade flow imbalances and order flow imbalances with respect to contemporaneous mid-price changes, verifying previous literature, although noticing a relative decrease in R-squareds that might suggest price is internalizing this relationship. The fee structure change operated by Binance made for a great case study, strongly confirming that exogenous changes can have significant impacts on market microstructure. Specifically, this thesis found order flow imbalances to be more informative during a fixed-percentage fee scheme compared to a no-fee scheme based on volatile bid-ask spreads. Overall, 10-minute OFIs and TFIs combined have been found to have the best predictive power with respect to contemporaneous mid-price changes. A machine learning extreme gradient boosting algorithm has been fitted to the same data as an exercise, finding improvements in MAE metrics and noticing that the model tends to give more feature importance to TFIs rather than OFIs during training. Out-of-sample predictions from the OLS and the XGBoost models have been presented as overall comparable in accuracy, mainly as a result of the defined linear relationship characterizing OFI, TFI and mid-price changes interactions. The conclusion was that the XGBoost model would have probably performed better than the OLS one in case of non-linear relationships in the dataset.

In the final chapter, this thesis touched on the use of artificial intelligence and machine learning for order flow analysis in cryptocurrency markets. The differences between supervised and unsupervised learning were introduced and the advantages of using ensemble methods to improve accuracy and prevent overfitting were highlighted. Various improvements offered by the XGBoost model were brought to the table (such as

regularization capabilities and the support for parallel computing), while emphasizing the need for continuous research and briefly presenting some state-of-the-art research on the field.

A note of caution on how these complex systems should be adopted and handled was highlighted, with the aim of increasing awareness on the darker sides and caveats of ML modelling. A subsequent exploration of the impact of AI and ML models on market efficiency was proposed, concluding that these novel techniques may be "good servants of a bad master", and stressing the fact that market-destabilizing practices may well be the end result if these models were to be used inadequately or maliciously.

This thesis concluded with a discussion on potential future trends of AI and ML applications in the world of cryptocurrency, highlighting that advanced trading algorithms, enhanced sentiment analysis, advancements in explainable AI, improved DeFi security and expansions of blockchain interoperability protocols will likely be some of the most salient results of continuous developments in the field.

In conclusion, financial markets offer an extremely fertile ground for future research and disruptive innovation. Gazing into the horizon, the boundaries of finance, technology and AI appear to be blurring away, unveiling a world of extraordinary opportunities that is only waiting to be explored.

Bibliography

Aggarwal, D., Chandrasekaran, S., & Annamalai, B. (2020). A complete empirical ensemble mode decomposition and support vector machine-based approach to predict Bitcoin prices. *Journal of Behavioral and Experimental Finance*, *27*.

Aharon, D. Y., & Qadan, M. (2018). What drives the demand for information in the commodity market? *Resources Policy*, *59*, 532–543.

Amihud, Y., & Mendelson, H. (1987). Trading mechanisms and stock returns: An empirical investigation. *The Journal of Finance*, *42*(3), 533–553.

Atsalakis, G. S., Atsalaki, I. G., Pasiouras, F., & Zopounidis, C. (2019). Bitcoin price forecasting with neuro-fuzzy techniques. *European Journal of Operational Research*, *276*(2), 770–780.

Banks, E. (2010). *Dark pools: The structure and future of off-exchange trading and liquidity*. Palgrave Macmillan.

Baruch, S. (2005). Who benefits from an open limit-order book? *The Journal of Business*, *78*(4), 1267–1306.

Basavaraju, A., Du, J., Zhou, F., & Ji, J. (2019). A machine learning approach to road surface anomaly assessment using smartphone sensors. *IEEE Sensors Journal*, *20*(5), 2635–2647.

Baur, D. G., Hong, K., & Lee, A. D. (2018). Bitcoin: Medium of exchange or speculative assets? *Journal of International Financial Markets, Institutions and Money*, *54*, 177–189.

Bentéjac, C., Csörgő, A., & Martinez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*, 1937–1967.

Biais, B., Declerck, F., & Moinas, S. (2016). *Who supplies liquidity, how and when?*

Bouchaud, J.-P. (2009). Price impact. *ArXiv Preprint ArXiv:0903.2428*.

Bouchaud, J.-P., Bonart, J., Donier, J., & Gould, M. (2018). *Trades, quotes and prices: financial markets under the microscope*. Cambridge University Press.

Boulton, T. J., & Braga-Alves, M. V. (2020). Short selling and dark pool volume. *Managerial Finance*, *46*(10), 1263–1282.

Brogaard, J., Hendershott, T., & Riordan, R. (2014). High-frequency trading and price discovery. *The Review of Financial Studies*, *27*(8), 2267–2306.

Brogaard, J., Hendershott, T., & Riordan, R. (2019). Price discovery without trading: Evidence from limit orders. *The Journal of Finance*, *74*(4), 1621–1658.

Brown, P., Thomson, N., & Walsh, D. (1999). Characteristics of the order flow through an electronic open limit order book. *Journal of International Financial Markets, Institutions and Money*, *9*(4), 335–357.

Caporale, G. M., & Plastun, A. (2019). The day of the week effect in the cryptocurrency market. *Finance Research Letters*, *31*.

Chaboud, A., Hjalmarsson, E., & Zikes, F. (2021). The evolution of price discovery in an electronic market. *Journal of Banking & Finance*, *130*, 106171.

Chaum, D. (1983). Blind signatures for untraceable payments. *Advances in Cryptology: Proceedings of Crypto 82*, 199–203.

Chordia, T., Roll, R., & Subrahmanyam, A. (2008). Liquidity and market efficiency. *Journal of Financial Economics*, *87*(2), 249–268.

Chow, G. C. (1960). Tests of equality between sets of coefficients in two linear regressions. *Econometrica: Journal of the Econometric Society*, 591–605.

Cohen, G. (2022). Algorithmic trading and financial forecasting using advanced artificial intelligence methodologies. *Mathematics*, *10*(18), 3302.

Colianni, S., Rosales, S., & Signorotti, M. (2015). Algorithmic trading of cryptocurrency based on Twitter sentiment analysis. *CS229 Project*, *1*(5), 1–4.

Cont, R., Kukanov, A., & Stoikov, S. (2014). The price impact of order book events. *Journal of Financial Econometrics*, *12*(1), 47–88.

Davchev, J., Mishev, K., Vodenska, I., Chitkushev, L., & Trajanov, D. (2020). Bitcoin price prediction using transfer learning on financial micro-blogs. *The 16th Annual International Conference on Computer Science and Education in Computer Science*.

De Prado, M. L. (2018). The 10 reasons most machine learning funds fail. *The Journal of Portfolio Management*, *44*(6), 120–133.

Debelle, G., Whitelaw, J., Vikstedt, H., Fréchard, I., Perez, S., Zajonz, R., Lai, K., Marras, M. L., Takeuchi, A., Jung, H., Sordo Janeiro, A., Ng, K., Maag, T., O'Connor, J., Nordstrom, A., & Ho, C. (2011). *High-frequency trading in the foreign exchange market*.

Demirgüç-Kunt, A., Klapper, L., Singer, D., & Ansar, S. (2022). *The Global Findex Database 2021: Financial inclusion, digital payments, and resilience in the age of COVID-19*. World Bank Publications.

Domowitz, I., & Steil, B. (1999). Automation, trading costs, and the structure of the securities trading industry. *Brookings-Wharton Papers on Financial Services*, *2*, 33–92.

Domowitz, I., & Wang, J. (1994). Auctions as algorithms: Computerized trade execution and price discovery. *Journal of Economic Dynamics and Control*, *18*(1), 29–60.

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*, 241–258.

Ebner, J. (2021). *Supervised vs Unsupervised Learning, Explained*. Sharp Sight. https://www.sharpsightlabs.com/blog/supervised-vs-unsupervised-learning/

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

Garg, H. (2023). *Consensus Mechanisms in Blockchain*. Shiksha Online. https://www.shiksha.com/online-courses/articles/consensus-mechanisms-in-blockchain/

Garman, M. B. (1976). Market microstructure. *Journal of Financial Economics*, *3*(3), 257–275.

Glosten, L. R., & Milgrom, P. R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, *14*(1), 71–100.

Goldstein, M. A., & Kavajecz, K. A. (2000). Eighths, sixteenths, and market depth: changes in tick size and liquidity provision on the NYSE. *Journal of Financial Economics*, *56*(1), 125–149.

Härdle, W. K., Harvey, C. R., & Reule, R. C. G. (2020). Understanding cryptocurrencies. In *Journal of Financial Econometrics* (Vol. 18, Issue 2, pp. 181–208). Oxford University Press.

Harris, L. (2003). *Trading and exchanges: Market microstructure for practitioners*. OUP USA.

Hayes, A. (2022). *Block Trading Facility (BTF): What it is, How it Works, Example*. Investopedia. https://www.investopedia.com/terms/b/block-trading-facility.asp

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

Inamdar, A., Bhagtani, A., Bhatt, S., & Shetty, P. M. (2019). Predicting cryptocurrency value using sentiment analysis. *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 932–934.

Jaisson, T. (2015). Market impact as anticipation of the order flow imbalance. *Quantitative Finance*, *15*(7), 1123–1135.

Jeon, Y., Samarbakhsh, L., & Hewitt, K. (2021). Fragmentation in the Bitcoin market: Evidence from multiple coexisting order books. *Finance Research Letters*, *39*, 101654.

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

Johnson, J. (2019). Daily Trading Patterns in the Bitcoin/Euro Market. *Euro Market (November 14, 2019)*.

Johnson, T. C. (2008). Volume, liquidity, and liquidity risk. *Journal of Financial Economics*, *87*(2), 388–417.

Kolm, P. N., Turiel, J., & Westray, N. (2021). Deep order flow imbalance: Extracting alpha at multiple horizons from the limit order book. *Available at SSRN 3900141*.

Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion*, *37*, 132–156.

Kurihara, Y., & Fukushima, A. (2017). The market efficiency of Bitcoin: a weekly anomaly perspective. *Journal of Applied Finance and Banking*, *7*(3), 57.

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

Lahmiri, S., & Bekiros, S. (2021). Deep learning forecasting in cryptocurrency high-frequency trading. *Cognitive Computation*, *13*, 485–487.

Lee, Y.-T., Liu, Y.-J., Roll, R., & Subrahmanyam, A. (2004). Order imbalances and market efficiency: Evidence from the Taiwan Stock Exchange. *Journal of Financial and Quantitative Analysis*, *39*(2), 327–341.

Madhavan, A. (2000). Market microstructure: A survey. *Journal of Financial Markets*, *3*(3), 205–258.

Mahesh, B. (2020). Machine learning algorithms - a review. *International Journal of Science and Research (IJSR).*, *9*(1), 381–386.

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

Mendelson, H. (1982). Market behavior in a clearing house. *Econometrica: Journal of the Econometric Society*, 1505–1524.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*.

Narain, A., & Moretti, M. (2022). Regulating Crypto: The right rules could provide a safe space for innovation. September, 2022. *Finance & Development, IMF*.

OECD. (2020). The tokenisation of assets and potential implications for financial markets. *OECD Blockchain Policy Series*.

OECD. (2021). *Artificial Intelligence, Machine Learning and Big Data in Finance: Opportunities, Challenges, and Implications for Policy Makers*.

Reiff, N. (2022). *What Was the First Cryptocurrency?* Investopedia. https://www.investopedia.com/tech/were-there-cryptocurrencies-bitcoin/

Schnaubelt, M. (2022). Deep reinforcement learning for the optimal placement of cryptocurrency limit orders. *European Journal of Operational Research*, *296*(3), 993–1006.

Senn-Kalb, L., & Mehta, D. (2023). *Artificial Intelligence: in-depth market analysis*.

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

Statista. (2023a). *Crypto Pulse Check*. https://www.statista.com/study/133052/statista-crypto-pulse-check/

Statista. (2023b). *Machine Learning*. https://www.statista.com/study/111190/machine-learning/

Statista. (2023c). *Stock exchanges*. https://www.statista.com/study/25684/global-stock-exchanges-statista-dossier/

Subrahmanyam, A. (2009). The implications of liquidity and order flows for neoclassical finance. *Pacific-Basin Finance Journal*, *17*(5), 527–532.

Szabó, D. A. (2017). *Impact of social media on cryptocurrency trading with deep learning*.

Takemiya, M. (2023). ALT: Aggregate Liquidity Technology. *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 1–6.

Toth, B., Palit, I., Lillo, F., & Farmer, J. D. (2015). Why is equity order flow so persistent? *Journal of Economic Dynamics and Control*, *51*, 218–239.

Tran, D. T., Kanniainen, J., & Iosifidis, A. (2022). How informative is the order book beyond the best levels? Machine learning perspective. *ArXiv Preprint ArXiv:2203.07922*.

Tredinnick, L. (2019). Cryptocurrencies and the blockchain. *Business Information Review*, *36*(1), 39–44.

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

Wołk, K. (2020). Advanced social media sentiment analysis for short-term cryptocurrency price prediction. *Expert Systems*, *37*(2), e12493.

Wray, A., Meades, M., & Cliff, D. (2020). Automated creation of a high-performing algorithmic trader via deep learning on level-2 limit order book data. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1067–1074.

Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., & Zhu, J. (2019). Explainable AI: A brief survey on history, research areas, approaches and challenges. *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*, 563–574.

Xu, K., Gould, M. D., & Howison, S. D. (2018). Multi-level order-flow imbalance in a limit order book. *Market Microstructure and Liquidity*, *3*(4).

Yan, B., & Zivot, E. (2010). A structural analysis of price discovery measures. *Journal of Financial Markets*, *13*(1), 1–19.

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, *1168*, 22022.

Zhu, H. (2014). Do dark pools harm price discovery? *The Review of Financial Studies*, *27*(3), 747–789.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.

Sitography

1inch Protocol. URL: https://app.1inch.io/ [accessed on 09/23/2023]

Analytics Vidhya. Introduction to XGBoost Algorithm in Machine Learning. URL: https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/ [accessed on 09/23/2023]

Basel Committee on Banking Supervision. Prudential Treatment of Cryptoasset Exposures. URL: https://www.bis.org/bcbs/publ/d545.pdf [accessed on 09/23/2023]

Binance Exchange. URL: https://www.binance.com/ [accessed on 09/23/2023]

Blockspot. List of Cryptocurrency Exchanges. URL: https://blockspot.io/exchange/ [accessed on 09/23/2023]

Buy Bitcoin Worldwide. Bitcoin Halving Clock. URL: https://buybitcoinworldwide.com/halving/ [accessed on 09/23/2023]

DARPA. Explainable Artificial Intelligence (XAI). URL: https://www.darpa.mil/program/explainable-artificial-intelligence [accessed on 09/23/2023]

Datix. What is Machine Learning? URL: https://datixinc.com/blog/what-is-machine-learning/ [accessed on 09/23/2023]

Depository Trust & Clearing Corporation. About DTCC. URL: https://www.dtcc.com/about/businesses-and-subsidiaries/dtc [accessed on 09/23/2023]

dYdX Exchange. URL: https://trade.dydx.exchange/ [accessed on 09/23/2023]

EDX Markets. About EDX. URL: https://edxmarkets.com/about/ [accessed on 09/23/2023]

Financial Stability Board. FSB finalises global regulatory framework for crypto-asset activities. URL: https://www.fsb.org/2023/07/fsb-finalises-global-regulatory-framework-for-crypto-asset-activities/ [accessed on 09/23/2023]

Futures Industry Association. ETD Tracker. URL: https://www.fia.org/fia/etd-tracker [accessed on 09/23/2023]

Gemini Staking. URL: https://www.gemini.com/en-us/staking/ [accessed on 09/23/2023]

Genesis Global Trading. URL: https://genesistrading.com/ [accessed on 09/23/2023]

INX Exchange. URL: https://www.inx.co/ [accessed on 09/23/2023]

Kaiko Blog. URL: https://blog.kaiko.com [accessed on 09/23/2023]

Kaiko. URL: https://www.kaiko.com [accessed on 09/23/2023]

Numpy Documentation. URL: https://numpy.org/doc/1.26/ [accessed on 09/23/2023]

Openocean Protocol. URL: https://app.openocean.finance/ [accessed on 09/23/2023]

Pandas Documentation. URL: https://pandas.pydata.org/docs/ [accessed on 09/23/2023]

Paxful Exchange. URL: https://paxful.com/ [accessed on 09/23/2023]

Plotly Documentation. URL: https://plotly.com/python/ [accessed on 09/23/2023]

Plotly. API Reference for Python. URL: https://plotly.com/python-api-reference/ [accessed on 09/23/2023]

Polars Documentation. URL: https://pola-rs.github.io/polars-book/ [accessed on 09/23/2023]

Pyarrow Documentation. URL: https://arrow.apache.org/docs/python/index.html [accessed on 09/23/2023]

Python Documentation. URL: https://docs.python.org/3/ [accessed on 09/23/2023]

Renegade Dark Pool. URL: https://renegade.fi/ [accessed on 09/23/2023]

Scikitlearn User Guide. URL: https://scikit-learn.org/stable/user_guide.html [accessed on 09/23/2023]

Skopt User Guide. URL: https://scikit-optimize.github.io/stable/user_guide.html [accessed on 09/23/2023]

Statsmodels Documentation. URL: https://www.statsmodels.org/stable/index.html [accessed on 09/23/2023]

Tradingview Chart. URL: https://www.tradingview.com/chart/ [accessed on 09/23/2023]

Uniswap Protocol. URL: https://app.uniswap.org/ [accessed on 09/23/2023]

XGBoost Documentation. URL: https://xgboost.readthedocs.io/en/stable/ [accessed on 09/23/2023]

Appendix

What follows is a Jupyter Notebook converted into PDF format and attached to this thesis. This Notebook was created by the author in its entirety, leveraging Python packages documentation and online examples. Only the most salient outputs are included. When a piece of code outputs a chart, the number of the Figure showing said chart in this Thesis is provided.

Jupyter Notebook Containing the Python Code Used in This Thesis

September 24, 2023

1 Import Modules

Import modules. For the lakeapi module to work properly, use a Python version < 3.11, this script works on 3.9.6.

[ ]:
```
import lakeapi # To download order book states and trades data via the Crypto␣
↪Lake API
import datetime # To manipulate datetime objects
import time # To access and convert time
import pandas as pd # To manage tabular datasets
import math # To perform computations
import numpy as np # To perform computations and manipulate arrays
from binance.client import Client # To establish a connection to the Binance API
import myKeys # Custom module containing personal Binance API public/secret keys
import plotly.express as px # For charting
import plotly.graph_objects as go # For charting
import plotly.figure_factory as ff # For charting
import requests # To perform HTTPS requests (mainly downloading data from␣
↪Coingecko)
from pyarrow.parquet import ParquetFile # To manage the efficient Parquet file␣
↪format by Oracle
import pyarrow as pa # For a quicker experience with large datasets
from sklearn.metrics import r2_score # To compute the R-squared metric
from chow_test import chow_test # To perform the Chow Test
import polars as pl # To manipulate high-dimensional datasets while managing␣
↪memory efficiently
from tqdm import tqdm # To add a progress bar to lenghty for loops
import statsmodels.api as sm # To run OLS regression
import xgboost as xgb # To train and deploy XGBoost models (Machine Learning)
from sklearn.metrics import mean_absolute_error # To compute the MAE metric
from skopt import BayesSearchCV # For hyperparameter tuning applied to XGBoost␣
↪models
from skopt.space import Integer, Real # For hyperparameter tuning applied to␣
↪XGBoost models
from sklearn.pipeline import Pipeline # For hyperparameter tuning applied to␣
↪XGBoost models
```

1

2 Data Collection

2.1 Download Data from LakeAPI

Show all crypto pairs available for download

[ ]:
```
lakeapi.available_symbols(table='book')
```

2.1.1 Download Order Book States and Store Them in a Parquet File

Divide downloads in batches to avoid overloading RAM. The result will be 12 files that will need to be merged.

[ ]:
```
# Create a list with 12 date ranges to split the download process into 12␣
↪batches
dates = [
    [datetime.datetime(2022,7,31), datetime.datetime(2022,8,31)],
    [datetime.datetime(2022,9,1), datetime.datetime(2022,9,30)],
    [datetime.datetime(2022,10,1), datetime.datetime(2022,10,31)],
    [datetime.datetime(2022,11,1), datetime.datetime(2022,11,30)],
    [datetime.datetime(2022,12,1), datetime.datetime(2022,12,31)],
    [datetime.datetime(2023,1,1), datetime.datetime(2023,1,31)],
    [datetime.datetime(2023,2,1), datetime.datetime(2023,2,28)],
    [datetime.datetime(2023,3,1), datetime.datetime(2023,3,31)],
    [datetime.datetime(2023,4,1), datetime.datetime(2023,4,30)],
    [datetime.datetime(2023,5,1), datetime.datetime(2023,5,31)],
    [datetime.datetime(2023,6,1), datetime.datetime(2023,6,30)],
    [datetime.datetime(2023,7,1), datetime.datetime(2023,7,31)]
]
t = 1
# Iterate over the 12 date ranges
for datepair in dates:
    print(f'This iteration is considering the period between {datepair[0]} and␣
↪{datepair[1]}')
    books = lakeapi.load_data( # This is the actual function downloading the␣
↪data
        table='book',
        start = datepair[0],
        end = datepair[1],
        symbols = ['BTC-USDT'],
        exchanges = 'BINANCE'
    )
    # Save to an efficient parquet file
    books.to_parquet(f'BTCUSDT_Book_part{t}.parquet',␣
↪use_deprecated_int96_timestamps = True)
    t += 1
    del books
```

Merge the files.

[ ]:
```
i = 1
books = pd.DataFrame()
for i in range(1,16):
    book = pd.read_parquet(f'BTCUSDT_Book_part{i}.parquet')
    books = pd.concat([books, book], axis = 0)

books = books.drop_duplicates()
books = books.reset_index()
books.head()
```

[ ]:
```
books.to_parquet('BTCUSDT_Book_Year.parquet', use_deprecated_int96_timestamps =␣
↪True)
```

2.1.2 Download Trades and Store Them in a Parquet File

Same approach used for order book states, however limiting the download to march 15 - july 31 2023 for computational reasons.

[ ]:
```
dates = [
    [datetime.datetime(2023,3,15), datetime.datetime(2023,3,31)],
    [datetime.datetime(2023,4,1), datetime.datetime(2023,4,30)],
    [datetime.datetime(2023,5,1), datetime.datetime(2023,5,31)],
    [datetime.datetime(2023,6,1), datetime.datetime(2023,6,30)],
    [datetime.datetime(2023,7,1), datetime.datetime(2023,7,31)]
]
i = 1
for datepair in dates:
    print(f'This iteration is considering the period between {datepair[0]} and␣
↪{datepair[1]}')
    trades = lakeapi.load_data(
        table='trades', # here the table is 'trades', before it was 'book'
        start = datepair[0],
        end = datepair[1],
        symbols = ['BTC-USDT'],
        exchanges = 'BINANCE'
    )
    tradeFinal.to_parquet(f'BTCUSDT_Trades_part{i}.parquet',␣
↪use_deprecated_int96_timestamps = True)
    i += 1
    del trades
    del tradeFinal
```

Merge the files, keeping only the important columns.

[ ]:
```
tradeFinal = trades.reset_index(drop = True)[['received_time', 'side',␣
↪'quantity', 'price']]
```

```
tradeFinal.to_parquet('BTCUSDT_Trades.parquet', use_deprecated_int96_timestamps␣
 ↪= True)
```

2.2 In case downloading/writing to file fails, before deleting cached files try below (with appropriate modifications eg folder path)

[ ]:
```
import os
import json

# Directory containing folders
folder_path = '/Volumes/SSDEsterno/.lake_cache/joblib/lakeapi/_read_parquet/
 ↪_read_parquet_nocache'

# List to store data
data_list = []

# Loop through folders
for folder_name in [x for x in os.listdir(folder_path) if 'File' in x]:
    folder_dir = os.path.join(folder_path, folder_name).strip('.DS_Store')

    # Load metadata from metadata.json
    metadata_file = os.path.join(folder_dir, 'metadata.json')
    with open(metadata_file, 'r') as f:
        metadata = json.load(f)

    # Load data from output.pkl
    output_file = os.path.join(folder_dir, 'output.pkl')
    output_data = pd.read_pickle(output_file)

    # Append data to list
    data_list.append({'FolderName': folder_name, 'Metadata': metadata,␣
 ↪'OutputData': output_data})

# Create DataFrame
df = pd.DataFrame(data_list)

# Print the resulting DataFrame
print(df)
```

Extract the aws paths from which data is fetched:

[ ]:
```
paths = []
for i in df.Metadata:
    paths.append(i['input_args']['path'])
len(paths)
```

Fetch Data

```
[ ]: import awswrangler as wr

dfWs = pd.DataFrame()
i = 1
for path in paths:
    print(f'Appending dataframe n. {i} out of {len(paths)}... ({round((i/
 ↪len(paths))*100,0)}%)')
    dfWs = pd.concat([dfWs, wr.s3.read_parquet(path[1:-2])], axis = 0)
    i += 1

dfWs.head()
```

```
[ ]: dfWs['timestamp'] = pd.to_datetime(dfWs['timestamp'], unit = 'ns')
dfWs['receipt_timestamp'] = pd.to_datetime(dfWs['receipt_timestamp'], unit =␣
 ↪'ns')
dfWs.rename(columns={'timestamp': 'origin_time', 'receipt_timestamp':␣
 ↪'received_time', 'amount': 'quantity'}, inplace=True)
dfWs.set_index('received_time').reset_index(inplace=True)
```

```
[ ]: tradeFinal = dfWs[['received_time', 'side', 'quantity', 'price']]
tradeFinal.head()
```

```
[ ]: tradeFinal.to_parquet('BTCUSDT_Trades.parquet', use_deprecated_int96_timestamps␣
 ↪= True)
```

3 Data Manipulation

3.1 Computing Nanosecond "Shocks" to the Order Book e

The nanosecond-granularity file will be opened with Polars. It will then be resampled into seconds and transferred to Pandas. Only Polars can handle the original 140 million + rows.

```
[ ]: plDf = pl.read_parquet('BTCUSDT_Book_Year.parquet')
```

Compute nanosecond imbalances.

```
[ ]: e = pl.Series([np.nan], dtype = 'float64', index=lakeDf['received_time'])
length = len(e)
for n in tqdm(range(1, length), total=length, desc='Processing'):
    e[n] =\
    (plDf['bid_0_price'][n] >= plDf['bid_0_price'][n-1])*plDf['bid_0_size'][n]␣
 ↪-\
    (plDf['bid_0_price'][n] <=␣
 ↪plDf['bid_0_price'][n-1])*plDf['bid_0_size'][n-1] -\
    (plDf['ask_0_price'][n] <= plDf['ask_0_price'][n-1])*plDf['ask_0_size'][n]␣
 ↪+\
    (plDf['ask_0_price'][n] >= plDf['ask_0_price'][n-1])*plDf['ask_0_size'][n-1]
```

```
es = pl.DataFrame(e, columns = ['e'])
plDf = plDf.hstack(es)
```

3.2 Resampling Order Book Data

Resample the database at 1-second frequency (Polaris sees resampling as a subtype of groupby, differently from pandas). Shocks must be summed when resampling to have a coherent value. Keeping only the last available shock will result in precious information loss. This procedures returns, among other things, 1-second Order Flow Imbalances (OFIs).

[ ]:
```
plDf = plDf[['received_time', 'bid_0_price', 'bid_0_size', 'ask_0_price',␣
 ↪'ask_0_size', 'e']].set_sorted('received_time')
plDf = plDf.groupby_dynamic('received_time', every = '1s').agg(pl.
 ↪col('bid_0_price').last(),
                                                                pl.
 ↪col('bid_0_size').last,
                                                                pl.
 ↪col('ask_0_price').last(),
                                                                pl.
 ↪col('ask_0_size').last(),
                                                                pl.col('e').sum)
```

Convert to a pandas dataframe.

[ ]:
```
lakeDf = plDf.to_pandas()
# delete the Polars dataframe to free up RAM
del plDf
# remove missing values
lakeDf = lakeDf.dropna().reset_index(drop=True)
```

Check the range covered by the dataframe.

[ ]:
```
print(f'This dataframes starts on {min(lakeDf.received_time)} and ends on␣
 ↪{max(lakeDf.received_time)}')
```

Save the file.

[ ]:
```
lakeDf.to_parquet('BTCUSD_Full_With_Shocks_Clean.parquet',␣
 ↪use_deprecated_int96_timestamps = True)
```

3.3 Computing Trade Flow Imbalances from Trades Data

Read the parquet file (this time directly with pandas). Specify pyarrow as engine for quicker read times.

[ ]:
```
trades = pd.read_parquet('BTCUSDT_Trades.parquet', engine = 'pyarrow')
```

Assign the correct sign to traded quantities based on the sign column provided by the Crypto Lake API.

```
[ ]: trades['signed_quantity'] = np.where(trades['side'] == 'sell',␣
      ↪-trades['quantity'], trades['quantity'])
     trades.drop('quantity', inplace =True, axis = 1)
     trades = trades.rename(columns = {'signed_quantity':'quantity'})
```

Compute 1-second TFIs by simply resampling the dataframe and summing over the quantity column.

```
[ ]: TFI = trades[['received_time', 'quantity']].resample('1S', on = 'received_time')
     TFI.dropna(inplace = True)
     TFI.reset_index(inplace = True, drop = True)
```

4 Plotting and Charting

4.1 Plotting Figure 11

Getting access to Binance

```
[ ]: api_key = myKeys.key
     api_secret = myKeys.secret

     client = Client(api_key, api_secret)
```

Collecting data from Coingecko (BTC/USD, USDT/USD)

```
[ ]: df = pd.DataFrame()
     # CoinGecko API endpoint for historical price data
     base = ['bitcoin', 'tether']
     for curr in base:
         url = "https://api.coingecko.com/api/v3/coins/"+curr+"/market_chart/range"
         vs_currency = "usd"
         from_timestamp = int(datetime.datetime(2020, 1, 1).timestamp())  # Replace␣
      ↪with your desired start date
         to_timestamp = int(datetime.datetime.now().timestamp())

         # Parameters for the API request
         params = {
             "vs_currency": vs_currency,
             "from": from_timestamp,
             "to": to_timestamp,
         }

         # Fetch historical data from CoinGecko API
         response = requests.get(url, params=params)
         data = response.json()
```

```
    df[['Timestamp'+curr+'USD','Close'+curr+'USD']] = pd.
 ↪DataFrame(data['prices'], columns= ['Timestamp', curr+'USD'])
    df['Timestamp'+curr+'USD'] = pd.to_datetime(df['Timestamp'+curr+'USD'],␣
 ↪unit = 'ms')
```

Getting data from Binance (BTC/USDT)

[ ]:
```
start = int(time.mktime(datetime.datetime(2020,1,1).timetuple()))*1000
end = int(time.mktime(datetime.datetime.now().timetuple()))*1000
lines_BTC = client.get_historical_klines("BTCUSDT", start_str = start, end_str␣
 ↪= end, interval='1d', limit = 1000)
```

[ ]:
```
linesDf_BTC = pd.DataFrame(lines_BTC, columns=["OpenTime", "Open", "High",␣
 ↪"Low", "CloseBTC", "Volume", "CloseTimeBTC", "QuoteVolume", "Trades",␣
 ↪"TakerBaseVolume", "TakerQuoteVolume", "void"])
linesDf_BTC = linesDf_BTC.astype('float', errors='ignore')
# Convert timestamp (in milliseconds) to datetime. Timezone is UTC.
linesDf_BTC['OpenTime'] = pd.to_datetime(linesDf_BTC.OpenTime, unit='ms')
linesDf_BTC['CloseTimeBTC'] = pd.to_datetime(linesDf_BTC.CloseTimeBTC,␣
 ↪unit='ms')

df['TimestampBTCUSDT'] = linesDf_BTC['CloseTimeBTC']
df['CloseBTCUSDT'] = linesDf_BTC['CloseBTC']
df = df.astype('float', errors='ignore')
plotDf = df.drop(['TimestamptetherUSD', 'TimestampBTCUSDT'], axis=1)
```

Plotting

[ ]:
```
fig = make_subplots(rows = 2, cols= 1, subplot_titles=("BTC/USDT vs BTC/USD",␣
 ↪"USDT/USD"))
trace1 = go.Scatter(x = plotDf['TimestampbitcoinUSD'], y =␣
 ↪plotDf['CloseBTCUSDT'], marker=dict(color='red'), name='BTC/USDT')
trace2 = go.Scatter(x = plotDf['TimestampbitcoinUSD'], y =␣
 ↪plotDf['ClosebitcoinUSD'], marker=dict(color='lightblue'), name='BTC/USD')
dat = [trace1, trace2]
fig.add_traces(data=dat)
fig.add_trace(go.Scatter(x = plotDf['TimestampbitcoinUSD'], y =␣
 ↪plotDf['ClosetetherUSD'], marker=dict(color='blue'), name='USDT/USD'),␣
 ↪row=2, col=1)

# Create the layout with custom dimensions
fig.update_layout(
    width=1600,   # Customize the width (in pixels)
    height=800,   # Customize the height (in pixels)
    template = 'ggplot2',
    legend = dict(
        orientation = 'h',
```

```
        yanchor = 'middle',
        y = 0.5,
        xanchor = 'center',
        x = 0.5
    )
)
fig.update_xaxes(title_text='Date', row=1, col=1)
fig.update_yaxes(title_text='Price', row=1, col=1)

fig.update_xaxes(title_text='Date', row=2, col=1)
fig.update_yaxes(title_text='Price', row=2, col=1)

fig.show()
```

Import the previously saved "lakeDf".

[ ]:
```
lakeDf = pd.read_parquet('BTCUSD_Full_With_Shocks_Clean.parquet', engine =␣
↪'pyarrow')
```

4.2 Computing Mid-Price and Bid-Ask Spread

Import the previously saved "lakeDf".

[ ]:
```
lakeDf = pd.read_parquet('BTCUSD_Full_With_Shocks_Clean.parquet', engine =␣
↪'pyarrow')
```

[ ]:
```
lakeDf['mid_price'] = round((lakeDf['ask_0_price'] + lakeDf['bid_0_price'])/2,2)
lakeDf['spread'] = lakeDf['ask_0_price'] - lakeDf['bid_0_price']
lakeDf = lakeDf.dropna().reset_index(drop=True)
lakeDf.head()
```

4.3 Plotting Figure 16

[ ]:
```
smallLakeDf = lakeDf[:300]
meanSpreadAug22 = lakeDf[lakeDf['received_time'].between('2022-07-30',␣
↪'2022-08-31')]['spread'].mean()
newnames = {'bid_0_price':'Best Bid', 'ask_0_price': 'Best Ask', 'mid_price':
↪'Mid Price'}

px.line(smallLakeDf, x = 'received_time', y = ['bid_0_price', 'ask_0_price',␣
↪'mid_price'],
       template='ggplot2', color_discrete_sequence=['blue','red', 'yellow'],
       labels = {'value': 'Price in USDT', 'received_time': 'Date and Time (HH:
↪mm:ss, UTC)'},
       title = f'1-second BTC/USDT Best Bid and Ask during the first minutes␣
↪of Jul 30, 2022<br><sup>Average Spread for the month of August 2022:␣
↪{round(meanSpreadAug22,4)}USDT.'
```

```
        ).update_layout(legend = dict(
            orientation = 'h',
            yanchor = 'top', y = 0.1,
            xanchor = 'center', x = 0.5,
            bgcolor = 'rgba(0,0,0,0)',
            title = '')
        ).update_traces(line_width=1).for_each_trace(lambda t: t.update(name =␣
↪newnames[t.name]))
```

4.4 Plotting Figure 17

[ ]:
```
meanHourlySpread = lakeDf.set_index('received_time')['spread'].resample('1h').
↪mean().reset_index()
```

[ ]:
```
px.line(meanHourlySpread, x = 'received_time', y = 'spread',
        template='ggplot2',
        labels = {'spread': 'Spread in USDT', 'received_time': 'Date'},
        title = 'Mean Hourly Bid-Ask Spread for BTC/USDT from 31-Jul-2022 to␣
↪31-Jul-2023',
        ).update_layout(legend = dict(
            orientation = 'h',
            yanchor = 'top', y = 0.1,
            xanchor = 'center', x = 0.5,
            bgcolor = 'rgba(0,0,0,0)',
            title = '')
        ).add_annotation(
            x='2022-11-09 20:00',
            y=3.3,  # Adjust the y-coordinate based on your data,
            ax = 70,
            ay = 30,
            borderpad = 8,
            text='FTX Collapse',
            showarrow=True,
            arrowhead=2,
            arrowcolor='dodgerblue',
            bgcolor='lightblue',
            bordercolor = 'dodgerblue',
            opacity=0.8,
            font=dict(size=12)
        ).add_annotation(
            x='2023-03-22 20:00',
            y=2,  # Adjust the y-coordinate based on your data
            ax = 130,
            ay = -90,
            borderpad = 8,
```

```
            text='Binance announce they will stop <br>0-fee trading (commission␣
↪trading resumes)<sup>*</sup>',
            showarrow=True,
            arrowhead=3,
            arrowcolor='black',
            bgcolor='orange',
            bordercolor = 'black',
            opacity=0.8,
            font=dict(size=12)
        ).add_annotation(
            x='2023-07-01 00:00',
            y=0,  # Adjust the y-coordinate based on your data
            xshift = -140,
            yshift = -50,
            borderpad = 8,
            text='<sup>*</sup>BTC/USDT will keep trading at 0-fees, however␣
↪adding fees to <br>other pairs made users flee the exchange, resulting in␣
↪trading volume drainage.',
            showarrow=False,
            arrowhead=3,
            arrowcolor='black',
            bgcolor='rgba(0,0,0,0)',
            align = 'left',
            opacity=0.8,
            font=dict(size=10)
        ).update_traces(line_width=1)
```

From around march 20 2023 spread goes to near 0. This is the result of an evolution in Binance's campaign on zero maker and taker fees on BTC/USDT trading. When fees were first removed (around july 8 2022), spreads for the BTC-USDT trading pair soared because market makers could no longer count volume towards Binance's VIP trading fee program. To compensate, they had to increase spreads which was essentially a way of transferring fees to price takers. Spreads instantly plummeted once fees were re-added, and are currently at .004bps. Commission free trading consisted in bringing maker and taker fees to 0. Cost for takers was then represented by the spread. With commission trading, costs are represented by maker and taker fees, rather than spreads. Maker and taker fees are computed based on user status (https://www.binance.com/en/fee/trading). The stop of commission free trading caused a sharp decline in trading volume on Binance so now Binance has resumed commission free trading in specific pairs such as BTCUSDT. However, people had already left the exchange and now a combination of binance wanting to favor a recoup in trading volume and other currencies actually making money from the resumed fee scheme sees the spread on the most liquid pair on the exchange still low. This is a great opportunity for checking the explanatory power of OFIs and TFIs with and without significant bid-ask spreads. Now, with 0 fees and low spread BTCUSDT could suffer less informativeness not only of limit orders but also of market orders. Remember Silantyev (2019) argued that what makes market orders more informative than limit orders is their cost of execution.

Sources:

https://www.binance.com/en/support/announcement/binance-launches-zero-fee-bitcoin-trading-10435147c55d4a40b64fcbf43cb46329

https://blog.kaiko.com/binance-volume-plummets-after-end-of-zero-fee-trading-e122c384cb9e

4.5 Plotting Figure 18

[ ]:
```
meanHourlyShock = lakeDf.set_index('received_time')['e'].resample('1h').mean().
 ↪reset_index()
hourlyOFI = lakeDf.set_index('received_time')['e'].resample('1h').sum().
 ↪reset_index()
```

[ ]:
```
fig = make_subplots(rows = 2, cols= 1, subplot_titles=(["BTC/USDT Average␣
 ↪Hourly Contribution of Order Book Events to Order Book State<br><sup>From␣
 ↪July 2022 to July 2023</sup>",'BTC/USDT Hourly OFI<br><sup>From July 2022 to␣
 ↪July 2023</sup>']))

fig.add_trace(go.Scatter(x = meanHourlyShock['received_time'], y =␣
 ↪meanHourlyShock['e'], name='e'), row=1, col=1)
fig.add_trace(go.Scatter(x = hourlyOFI['received_time'], y = hourlyOFI['e'],␣
 ↪marker=dict(color='dodgerblue'), name='OFI'), row=2, col=1)

# Create the layout with custom dimensions
fig.update_layout(
    template = 'ggplot2',
    showlegend = False,
).update_traces(line_width = 1)
fig.update_xaxes(title_text='Date', row=1, col=1)
fig.update_yaxes(title_text=r'$e$', row=1, col=1)

fig.update_xaxes(title_text='Date', row=2, col=1)
fig.update_yaxes(title_text='OFI', row=2, col=1)

fig.show()
```

We see that order book events have higher impact after the reintroduction of fees. This could be given either by the thinner liquidity resulting from volume leaving the exchange.

5 Statistical Analysis

5.1 Computations for Tables 6 and 7.

Preparing the data for the analysis.

[2]:
```
book = pd.read_parquet('BTCUSD_Full_With_Shocks_Clean.parquet', engine =␣
 ↪'pyarrow')
trades = pd.read_parquet('BTCUSDT_Trades.parquet', engine = 'pyarrow')
```

[ ]:
```
book['mid_price_onestepback'] = book['mid_price'].shift()
book.dropna(inplace = True)
book['mid_price_change'] = (book['mid_price']-book['mid_price_onestepback'])/0.
↪01
book.dropna(inplace = True)
book.reset_index(inplace = True, drop = True)
book.head()
```

[ ]:
```
trades = trades.merge(book[['received_time', 'mid_price_change']], on =␣
↪'received_time')
trades.rename(columns = {'mid_price_change':'TFI_mid_price_change'}, inplace =␣
↪True)
trades.head()
```

Run multiple OLS regressions, capture p-values, coefficients and R-squareds.

[5]:
```
for i in ['OFI', 'TFI', ['OFI','TFI']]:
    for j in ['1s', '10s', '60s', '10min', 'H']:
        OFI = book[['received_time', 'mid_price_change', 'e']].
↪set_index('received_time').resample(j).sum().dropna().reset_index()
        OFI = OFI.rename(columns = {'e': 'OFI'})
        TFI = trades[['quantity', 'received_time', 'TFI_mid_price_change']].
↪set_index('received_time').resample(j).sum().dropna().reset_index()
        TFI = TFI.rename(columns = {'quantity': 'TFI'})
        OFITFI = OFI[['received_time', 'mid_price_change', 'OFI']].
↪merge(TFI[['received_time', 'TFI']], on = 'received_time')
        OFITFI['tomorrows_mid_price_change'] = OFITFI['mid_price_change'].
↪shift(-1)
        OFITFI = OFITFI.loc[OFITFI['TFI'] != 0]
        OFITFI = OFITFI.dropna().reset_index(drop=True)
        dfAfterMarch = OFITFI.loc[OFITFI['received_time'] > '2023-03-15']
        X = sm.add_constant(dfAfterMarch[i])
        mod = sm.OLS(dfAfterMarch['mid_price_change'], X)
        fitted_model = mod.fit(cov_type = 'HC3')
        print(f'{i} - {j}: R^2 = {fitted_model.rsquared}')
        match i:
            case 'OFI':
                print(f'{i} - {j}: Parametri = cost. {fitted_model.params[0]}␣
↪({fitted_model.pvalues[0]}), OFI {fitted_model.params[1]} ({fitted_model.
↪pvalues[1]})')
            case 'TFI':
                print(f'{i} - {j}: Parametri = cost. {fitted_model.params[0]}␣
↪({fitted_model.pvalues[0]}), TFI {fitted_model.params[1]} ({fitted_model.
↪pvalues[1]})')
            case ['OFI', 'TFI']:
```

```
            print(f'{i} - {j}: Parametri = cost. {fitted_model.params[0]}␣
↪({fitted_model.pvalues[0]}), OFI {fitted_model.params[1]} ({fitted_model.
↪pvalues[1]}), TFI {fitted_model.params[2]} ({fitted_model.pvalues[2]})')
      if i == 'OFI':
          OFI = book[['received_time', 'mid_price_change', 'e']].
↪set_index('received_time').resample(j).sum().reset_index().dropna()
          OFI = OFI.rename(columns = {'e': 'OFI'})
          dfBeforeMarch = OFI.loc[OFI['received_time'] <= '2023-03-15']
          X = sm.add_constant(dfBeforeMarch[i])
          mod = sm.OLS(dfBeforeMarch['mid_price_change'], X)
          fitted_model = mod.fit(cov_type = 'HC3')
          print(f'|| OFI PREMARZO || {i} - {j}: R^2 = {fitted_model.
↪rsquared}')
          print(f'|| OFI PREMARZO || {i} - {j}: Parametri = cost.␣
↪{fitted_model.params[0]} ({fitted_model.pvalues[0]}), OFI {fitted_model.
↪params[1]} ({fitted_model.pvalues[1]})')
```

```
OFI - 1s: R^2 = 0.14140220848266283
OFI - 1s: Parametri = cost. 0.6774803668571057 (7.20052712604536e-09), OFI
17.96191174541342 (0.0)
|| OFI PREMARZO || OFI - 1s: R^2 = 0.01021159203205102
|| OFI PREMARZO || OFI - 1s: Parametri = cost. 0.02212317794913104
(0.5760368583904538), OFI 19.776697074093793 (8.09720562502785e-10)
OFI - 10s: R^2 = 0.2618800725763827
OFI - 10s: Parametri = cost. 7.201884387388341 (1.0860728881229892e-10), OFI
20.32290239624941 (0.0)
|| OFI PREMARZO || OFI - 10s: R^2 = 0.02099115508110383
|| OFI PREMARZO || OFI - 10s: Parametri = cost. 0.3460857633143225
(0.397700929538896), OFI 34.37664902277029 (2.2291589854760117e-06)
OFI - 60s: R^2 = 0.3068674124487143
OFI - 60s: Parametri = cost. 39.530302445437286 (3.730341381788538e-09), OFI
18.675933162365332 (0.0)
|| OFI PREMARZO || OFI - 60s: R^2 = 0.030226882446430903
|| OFI PREMARZO || OFI - 60s: Parametri = cost. 2.5057760221382765
(0.3160024053349405), OFI 42.702325215699176 (5.0421010062743455e-06)
OFI - 10min: R^2 = 0.2155443541701042
OFI - 10min: Parametri = cost. 281.3032415868338 (5.9782092277080865e-05), OFI
13.51347690993642 (5.039195025321387e-259)
|| OFI PREMARZO || OFI - 10min: R^2 = 0.04039363760544845
|| OFI PREMARZO || OFI - 10min: Parametri = cost. 27.834460762927833
(0.2616790833031467), OFI 48.410179588514346 (8.283408613851283e-06)
OFI - H: R^2 = 0.11151914672569152
OFI - H: Parametri = cost. 1049.6955996474726 (0.01718390289693645), OFI
8.765334474169595 (3.049381893071303e-20)
|| OFI PREMARZO || OFI - H: R^2 = 0.05027773365420429
|| OFI PREMARZO || OFI - H: Parametri = cost. 172.6349698376903
(0.23929624269025918), OFI 49.8971164232005 (0.00043056552603164185)
```

```
TFI - 1s: R^2 = 0.23451624581952824
TFI - 1s: Parametri = cost. 0.35984258311864825 (0.0011493907251324382), TFI
39.34211510597495 (0.0)
TFI - 10s: R^2 = 0.3322133030563864
TFI - 10s: Parametri = cost. 4.249108969784905 (6.308125739188464e-05), TFI
49.75048868039831 (0.0)
TFI - 60s: R^2 = 0.37957490682717543
TFI - 60s: Parametri = cost. 25.7921814273969 (5.546741535465709e-05), TFI
50.23525795198911 (4.004139628388473e-271)
TFI - 10min: R^2 = 0.48550984035012024
TFI - 10min: Parametri = cost. 224.15064455712832 (9.181574570617056e-05), TFI
44.43735919140279 (8.774678594511879e-86)
TFI - H: R^2 = 0.5177932800847826
TFI - H: Parametri = cost. 1066.6631412021102 (0.0014338892695675604), TFI
36.14751889792962 (6.523476495106878e-41)
['OFI', 'TFI'] - 1s: R^2 = 0.3026228613044004
['OFI', 'TFI'] - 1s: Parametri = cost. 0.8152604829738843
(1.1510444513456617e-14), OFI 12.893671137750841 (0.0), TFI 33.73953193929013
(0.0)
['OFI', 'TFI'] - 10s: R^2 = 0.4503424392579427
['OFI', 'TFI'] - 10s: Parametri = cost. 8.659500993098144
(3.1605525753493742e-19), OFI 14.425789028719038 (0.0), TFI 39.602966553712264
(0.0)
['OFI', 'TFI'] - 60s: R^2 = 0.5154194958811058
['OFI', 'TFI'] - 60s: Parametri = cost. 49.18575272153887 (9.405963225335e-19),
OFI 13.19090517886142 (0.0), TFI 39.52880241825095 (1.2532646632904664e-142)
['OFI', 'TFI'] - 10min: R^2 = 0.5630118619993687
['OFI', 'TFI'] - 10min: Parametri = cost. 383.6884641838664
(1.4497385573454747e-13), OFI 8.448833753329673 (1.1571154722036147e-93), TFI
39.19654559805259 (9.262087760982371e-59)
['OFI', 'TFI'] - H: R^2 = 0.5456157742954209
['OFI', 'TFI'] - H: Parametri = cost. 1604.296675697175 (6.211441986612492e-07),
OFI 4.508827497201775 (6.166572411472432e-08), TFI 34.08515789255368
(1.3920957951522411e-34)
```

Check at what frequency the R-squared for TFIs starts deteriorating.

[69]:
```
for j in ['1s', '10s', '60s', '10min', 'H', '2H', '5H',␣
↪'6H','7H','8H','9H','10H', '24H', '48H', '72H']:
    TFI = trades[['quantity', 'received_time', 'TFI_mid_price_change']].
↪set_index('received_time').resample(j).sum().dropna().reset_index()
    TFI = TFI.rename(columns = {'quantity': 'TFI'})
    TFI = TFI.loc[TFI['TFI'] != 0]
    TFI = TFI.dropna().reset_index(drop=True)
    dfAfterMarch = TFI.loc[TFI['received_time'] > '2023-03-15']
    X = sm.add_constant(dfAfterMarch['TFI'])
    mod = sm.OLS(dfAfterMarch['TFI_mid_price_change'], X)
    fitted_model = mod.fit(cov_type = 'HC3')
```

```
    print(f'TFI - {j}: R^2 = {fitted_model.rsquared}')
    print(f'TFI - {j}: Parametri = cost. {fitted_model.params[0]}␣
↪({fitted_model.pvalues[0]}), TFI {fitted_model.params[1]} ({fitted_model.
↪pvalues[1]})')
```

```
TFI - 1s: R^2 = 0.2345162458169937
TFI - 1s: Parametri = cost. 0.3598423817531081 (0.0011493930274200633), TFI
39.34211510551858 (0.0)
TFI - 10s: R^2 = 0.3321451400803441
TFI - 10s: Parametri = cost. 4.246492885612526 (6.322147313695188e-05), TFI
49.719115117186 (0.0)
TFI - 60s: R^2 = 0.37944672161226856
TFI - 60s: Parametri = cost. 25.765330842590455 (5.606931611715446e-05), TFI
50.20210756979513 (7.340544584006939e-271)
TFI - 10min: R^2 = 0.4857151506237821
TFI - 10min: Parametri = cost. 225.051796664854 (8.489762088428948e-05), TFI
44.42565345609536 (8.972414736136679e-86)
TFI - H: R^2 = 0.5182030041322121
TFI - H: Parametri = cost. 1068.8874087679635 (0.0013909472216402457), TFI
36.200135420678606 (6.620920808555084e-41)
TFI - 2H: R^2 = 0.5254195184974495
TFI - 2H: Parametri = cost. 2002.4944412391085 (0.0018836757598941264), TFI
34.510588474446166 (7.22868756179862e-38)
TFI - 5H: R^2 = 0.5554305598806185
TFI - 5H: Parametri = cost. 3624.100456654454 (0.015745666805510334), TFI
28.55507821171643 (1.34043394272804e-17)
TFI - 6H: R^2 = 0.5231099537051651
TFI - 6H: Parametri = cost. 5166.019965831808 (0.00832407903617682), TFI
30.01914729454667 (2.414557913985617e-16)
TFI - 7H: R^2 = 0.5598547069960933
TFI - 7H: Parametri = cost. 5306.900020171191 (0.004442286673186316), TFI
30.7145052415254 (1.4172773179348922e-30)
TFI - 8H: R^2 = 0.4977711122459547
TFI - 8H: Parametri = cost. 6490.04198294729 (0.019997889023068195), TFI
28.72789921374428 (2.1805140061957475e-17)
TFI - 9H: R^2 = 0.46601701963681086
TFI - 9H: Parametri = cost. 5799.774113622301 (0.029778356396866393), TFI
26.793881193685596 (6.5012963419866476e-15)
TFI - 10H: R^2 = 0.479909620854193
TFI - 10H: Parametri = cost. 6233.651461756111 (0.033375661370248215), TFI
27.726606456431274 (6.324124844604824e-12)
TFI - 24H: R^2 = 0.5041018574210574
TFI - 24H: Parametri = cost. 19567.55730906192 (0.07397451096184608), TFI
26.275104148268934 (7.004295601550297e-06)
TFI - 48H: R^2 = 0.4804663349464474
TFI - 48H: Parametri = cost. 23079.67482626052 (0.10036656235075762), TFI
23.88872507660867 (1.9721153154492106e-05)
```

```
TFI - 72H: R^2 = 0.4778911059834118
TFI - 72H: Parametri = cost. 28497.88642729067 (0.23793054031692307), TFI
26.957530284951233 (0.008770165736130096)
```

5.2 Chow Test

Read the parquet file.

[ ]:
```
df = pd.read_parquet('BTCUSD_Full_With_Shocks_Clean.parquet', engine =␣
↪'pyarrow')
```

Compute mid-price change and resample at 1-min frequency.

[ ]:
```
df['mid_price_stepback'] = df['mid_price'].shift()
df['mid_price_change'] = (df['mid_price']-df['mid_price_stepback'])/0.01
del df['mid_price_stepback']
df.dropna(inplace = True)
df.reset_index(inplace = True, drop = True)
df = df[['received_time', 'e', 'mid_price_change']].resample('1min', on =␣
↪'received_time').sum()
df.dropna(inplace = True)
df.reset_index(inplace = True)
```

Find the index value for March 15th 2023 (where the dataset should be split).

[ ]:
```
march15 = df.loc[df['received_time'] == '2023-03-15'].index.values[0]
```

Run the Chow Test knowing where to split the dataset.

[9]:
```
chow_test(y_series=df['mid_price_change'], X_series=df['e'],
        last_index=int(march15-1),
        first_index=int(march15),
        significance=.05)
```

```
Reject the null hypothesis of equality of regression coefficients in the two
periods.
Chow Statistic: 1977.611959882749, P_value: 1.1102230246251565e-16
```

[9]:
```
(1977.611959882749, 1.1102230246251565e-16)
```

The output suggests there is indeed a structural break between the two datasets resulting from the split on March 15th 2023.

6 Machine Learning Application: XGBoost

6.1 Preparing Data and Setting a Benchmark (OLS)

[ ]:
```
df = pd.read_parquet('BTCUSD_Full_With_Shocks_Clean.parquet', engine =␣
 ↪'pyarrow')
trades = pd.read_parquet('BTCUSDT_Trades.parquet', engine = 'pyarrow')
```

Merge the two dataframe on the important columns, to have OFIs and TFIs in the same dataframe.

[ ]:
```
trades = trades.merge(df[['mid_price_change', 'e', 'received_time']], on =␣
 ↪'received_time')
```

Resample at 10-minute frequency, since the best outcome in linear regression came from this frequency.

[ ]:
```
trades10min = trades[['received_time', 'quantity', 'e']].
 ↪set_index('received_time').resample('10min').sum().reset_index()
trades10min = trades10min.merge(trades[['received_time','mid_price_change']].
 ↪set_index('received_time').resample('10min').sum().reset_index(), on =␣
 ↪'received_time')
trades10min['future_price_change'] = trades10min['mid_price_change'].shift(-1)
trades10min.dropna(inplace = True)
trades10min.reset_index(inplace = True, drop = True)
```

Add 3 variables related to time, to check if they have a role in the relationship between mid-price change and OFIs/TFIs and if XGBoost takes them into consideration.

[21]:
```
trades10min['dayofweek'] = trades10min['received_time'].dt.dayofweek
trades10min['hour'] = trades10min['received_time'].dt.hour
trades10min['minute'] = trades10min['received_time'].dt.minute
```

Create a training and a testing set to validate the model.

[22]:
```
X_train = trades10min[['quantity', 'e', 'dayofweek', 'hour', 'minute']][:
 ↪round(len(trades10min)*0.7)]
y_train = trades10min['mid_price_change'][:round(len(trades10min)*0.7)]
X_test = trades10min[['quantity', 'e', 'dayofweek', 'hour',␣
 ↪'minute']][round(len(trades10min)*0.7):]
y_test = trades10min['mid_price_change'][round(len(trades10min)*0.7):]
```

Run a linear regression on the newly created training set, to have an appropriate benchmark for XGBoost.

[23]:
```
X = sm.add_constant(X_train)
mod = sm.OLS(y_train, X)
fitted_model = mod.fit(cov_type = 'HC3')
print(f'TFI - 10min: R^2 = {fitted_model.rsquared}')
```

```
print(f'TFI - 10min: Parameters = const. {fitted_model.params[0]}␣
↪({fitted_model.pvalues[0]}), TFI {fitted_model.params[1]} ({fitted_model.
↪pvalues[1]})')
```

```
TFI - 10min: R^2 = 0.5429632641293771
TFI - 10min: Parametri = cost. 171.80965785063756 (0.00010929827505235165), TFI
36.93147374124314 (3.014803089210358e-32)
```

Predict testing values based on the OLS model (this will be used later in the plot for Figure 27)

[24]:
```
X = sm.add_constant(X_test)
pred_ols = fitted_model.predict(X)
```

6.2 Hyperparameter Tuning

Adjust the main tweakable hyperparameters with a Bayesian Search algorithm, based on the mean absolute error metric.

[25]:
```
np.int = int

pipe = Pipeline(steps = [('clf', xgb.XGBRegressor(random_state = 1))])

search_space = {
    'clf__max_depth': Integer(2,8),
    'clf__learning_rate': Real(0.001, 1.0, prior='log-uniform'),
    'clf__subsample': Real(0.5, 1.0),
    'clf__colsample_bytree': Real(0.5, 1.0),
    'clf__colsample_bylevel': Real(0.5, 1.0),
    'clf__colsample_bynode': Real(0.5, 1.0),
    'clf__reg_alpha':Real(0.0, 10.0),
    'clf__reg_lambda': Real(0.0, 10.0),
    'clf__gamma': Real(0.0, 10.0)
}

opt = BayesSearchCV(pipe, search_space, cv = 10, n_iter = 50, scoring =␣
↪'neg_mean_absolute_error', random_state = 1)

opt.fit(X_train, y_train)
```

[25]:
```
BayesSearchCV(cv=10,
              estimator=Pipeline(steps=[('clf',
                                         XGBRegressor(base_score=None,
                                                      booster=None,
                                                      callbacks=None,
                                                      colsample_bylevel=None,
                                                      colsample_bynode=None,
                                                      colsample_bytree=None,
early_stopping_rounds=None,
```

```
                                                    enable_categorical=False,
                                                    eval_metric=None,
                                                    feature_types=None,
                                                    gamma=None, gpu_id=None,
                                                    grow_policy=None,
                                                    importance_type=None,
interaction_constraints=…
                            'clf__learning_rate': Real(low=0.001, high=1.0,
prior='log-uniform', transform='normalize'),
                            'clf__max_depth': Integer(low=2, high=8,
prior='uniform', transform='normalize'),
                            'clf__reg_alpha': Real(low=0.0, high=10.0,
prior='uniform', transform='normalize'),
                            'clf__reg_lambda': Real(low=0.0, high=10.0,
prior='uniform', transform='normalize'),
                            'clf__subsample': Real(low=0.5, high=1.0,
prior='uniform', transform='normalize')})
```

Return the hyperparameters for the best model.

[26]:
```
opt.best_params_
```

[26]:
```
OrderedDict([('clf__colsample_bylevel', 1.0),
             ('clf__colsample_bynode', 1.0),
             ('clf__colsample_bytree', 1.0),
             ('clf__gamma', 10.0),
             ('clf__learning_rate', 0.11217197417799707),
             ('clf__max_depth', 2),
             ('clf__reg_alpha', 10.0),
             ('clf__reg_lambda', 2.6977042269121503),
             ('clf__subsample', 1.0)])
```

Compare XGB's and OLS's MAE in the testing set.

[27]:
```
pred = opt.best_estimator_.predict(X_test)
mae_train = mean_absolute_error(y_train, opt.best_estimator_.predict(X_train))
mae_test = mean_absolute_error(y_test, pred)
mae_test_ols = mean_absolute_error(y_test, pred_ols)
print(f'XGB MAE (train): {mae_train}\nXGB MAE (test): {mae_test}\nOLS MAE␣
 ↪(test): {mae_test_ols}')
print(f'XGB\'s MAE is {round((mae_test_ols/mae_test - 1)*100)}% smaller than␣
 ↪OLS\'s MAE.')
```

```
XGB MAE (train): 511.3501955656913
XGB MAE (test): 532.3349311240983
OLS MAE (test): 595.6931368354922
XGB's MAE is 12% smaller than OLS's MAE.
```

Check for the usefulness of hyperparameter tuning: launch a completely fresh model without any tuning and see how it performs from a MAE and overfitting perspective.

[28]:
```
nohptxgb = xgb.XGBRegressor(n_estimators = 50)
nohptxgb.fit(X_train, y_train)
nohptpred = nohptxgb.predict(X_test)
nohpt_mae_train = mean_absolute_error(y_train, nohptxgb.predict(X_train))
nohpt_mae_test = mean_absolute_error(y_test, nohptpred)
print(f'XGB MAE (train): {nohpt_mae_train}\nXGB MAE (test): {nohpt_mae_test}')
```

```
XGB MAE (train): 362.14935061505315
XGB MAE (test): 574.2943293853928
```

Hyperparameter tuning effectively results in reduced overfitting and lower MAE in the testing set. Overfitting is evident in this case of not-tuned model since the MAE is significantly lower in the training set compared to the testing set.

6.3 Plotting Figure 25

[ ]:
```
fidf = pd.DataFrame(opt.best_estimator_.steps[0][1].feature_importances_,␣
 ↪columns=['Importance'])
fidf['FeatureName'] = ['TFI', 'OFI', 'Day of Week', 'Hour', 'Minute']

px.bar(fidf, x = 'FeatureName', y = 'Importance', text = fidf['Importance'].
 ↪apply(lambda x: f'{x*100:.2f}%'), color = 'FeatureName', title = 'Weights␣
 ↪assigned to TFI and OFI by the gradient boosting algorithm',␣
 ↪labels={'FeatureName': 'Features', 'value': 'Importance'}).
 ↪update_yaxes(tickformat=".0%").update_layout(template = 'ggplot2')
```

6.4 Plotting Figure 26

[ ]:
```
fig = px.scatter(x = pred, y = y_test,
                 title = 'XGBoost Predictions vs Actual Values',
                 labels = {'x':'XGBoost Prediction', 'y':'Actual Value'}).
 ↪update_layout(template = 'ggplot2')

fig.add_vrect(
    x0=-67000,
    x1=-5000,
    fillcolor="red",
    opacity=0.1,
    line_width=0,
)
fig.add_vrect(
    x0=-5000,
    x1=5000,
    fillcolor="green",
    opacity=0.1,
```

```
    line_width=0,
)
fig.add_vrect(
    x0=5000,
    x1=29000,
    fillcolor="red",
    opacity=0.1,
    line_width=0,
)
```

6.5 Plotting Figure 27

[ ]:
```
fig = px.line(title = 'OLS vs XGBoost Predictions')
fig.add_trace(go.Scatter(y = y_test.values[800:1000], marker =␣
↪dict(color='dodgerblue'), mode = 'lines', name = 'Actual Values')).
↪add_trace(go.Scatter(y = pred_ols[800:1000], mode = 'lines', marker =␣
↪dict(color='yellow'), name = 'OLS Prediction')).add_trace(go.Scatter(y =␣
↪pred[800:1000], mode = 'lines', marker = dict(color='crimson'), name =␣
↪'XGBoost Prediction')).update_layout(template = 'ggplot2', xaxis_title =␣
↪'Observations', yaxis_title = 'Mid-Price Change')
```