Delft University of Technology

Algorithmic FX trading: a new backtesting approach for the venue selection

Publishable Version

Master in Applied Mathematics

Financial Engineering

Faculty of Electrical Engineering, Mathematics and Computer Science

Luca Ferretti (5671418)

September 22, 2023

Supervisor: Prof. A. Papapantoleon
Co-reader: Prof. F. Fang

MN Supervisor: E. Hazeveld,
Co-reader: H. Harutyunyan

Preface

This research project, conducted in collaboration between TU Delft and MN, a pension fund asset manager, focuses on the optimal venue selection in FX trading. The objective is to investigate how the venue selection affects trading performance and to improve MN trading execution algorithm, named ALGO. The research aims to propose a new approach for the venue selection problem by allocating weights to different venues instead of solely selecting the best one. It utilizes advanced statistical and machine learning techniques and develops a matching engine capable of reconstructing historical orderbooks for backtesting strategies. The outcomes of this thesis show clear ideas for improving the current venue selection model. The proposed models are anticipated to consistently outperform ALGO, leading to improved trade execution. The insights and methodologies developed in this research will contribute to further investigations in improving venue selection processes and optimizing execution strategies in the FX market. The thesis provides a comprehensive analysis of the problem, explores the mathematical framework, presents real-world data-driven approaches, and discusses the findings and conclusions, offering valuable insights and recommendations for future MN research.

Disclaimer: this version of the thesis does not contain any information deemed unpublishable by MN. The trading venues are renamed with AMS and RTM. Moreover, some details on ALGO and other minor changes have been removed and replaced with [...].

Acknowledgments

I wish to thank Prof. Antonis Papapantoleon, Prof. Fang Fang, Erwin Hazeveld and Hayk Harutyunyan for the great help they have given me during this research. I am also deeply thankful to Liakos Papapoulos, Tjerk Methorst, Koen Baak, Rick Lodder, Angelo Barisano, Felix Lokin and all the AXP team for providing me with all the resources and for teaching me a lot about how the world of asset managers works. The expertise and dedication of all of them played an important role during this thesis.

Additionally, I extend my appreciation to my family for their encouragement and understanding during this journey. Despite the distance, you have always been with me. I wish to thank India, for her unique way of understanding and supporting me in my life. I wish to thank Pietro, whose example I have always held in high regard in my choices. I want to thank all my friends in Italy who despite the distance are always there for me, and all my friends in Delft with whom I have shared beautiful moments along this journey.

Lastly, I acknowledge TU Delft for providing me the ambitious environment essential for my academic growth. At this university I was able to find new stimuli and opportunities that helped me to find again my passion for studying.

Luca Ferretti, September 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This research project is a collaborative effort between TU Delft and MN, a pension fund asset manager. It focuses on the investigation of the optimal venue selection in FX trading. Trading algorithms play a crucial role in financial markets, enabling efficient execution and risk management. MN, recognizing the significance of algorithmic trading, has developed their own proprietary algorithm named ALGO. The aim of this research is to explore the impact of venue selection on trading performance, leveraging MN's ALGO and applying advanced mathematical techniques to optimize execution outcomes in the dynamic FX market.

The objectives of this research are two. Firstly, it aims to propose a novel approach to tackle the venue selection problem in FX trading. Previous studies have not yielded significant results in this area, and therefore, this work seeks to explore new perspectives and methodologies to address the challenge. Secondly, the research aims to enhance the existing venue selection algorithm employed by ALGO. By leveraging advanced statistical and techniques and incorporating new insights gained from the research, the objective is to improve the accuracy and performance of ALGO's venue selection process. Ultimately, the goal is to develop an enhanced algorithm that can optimize the venue selection method and potentially outperform existing methods in the FX market.

The methodology changes the way to look into the venue selection method. It proposes a best allocation approach to the problem, trying to assign the best weight to each venue instead to just select the best one. Moreover, the used tools are new: this study has developed a matching engine capable of reading the data from the two venues and reconstructing the orderbook at each instance of time in the past. This makes it possible to backtest strategies using historical data to measure performance at both venues and to compare them. Using this approach, the research created datasets to analyze and to be used to train statistical and machine learning models.

The outcomes of this thesis are expected to yield several significant results. Firstly, the proposed approach is anticipated to demonstrate superior performance and practical applicability to ALGO. By shifting the focus towards quantifying venue superiority for optimal order allocation, the research aims to provide a more effective and suitable framework for ALGO's practical implementation. Secondly, the proposed models for the studied strategies are expected to consistently outperform ALGO, leading to a substantial improvement in the tested simulations. The performance enhancement of these models is expected to contribute to more efficient trade execution and better overall outcomes. Finally the insights and methodologies developed in this thesis will provide a basis for further investigations into improving venue selection processes and optimizing execution strategies in the FX market.

The first chapters of the thesis focus on a comprehensive analysis of the problem, providing a detailed exploration of the business setting and the current state of affairs that necessitate improvement. The subsequent chapters delve into the mathematical framework from a present perspective, elucidating the underlying mathematical principles and models employed in the research. The rationale for selecting a data-driven approach with backtesting strategies using real-world data and graphical analysis is thoroughly explained. The thesis then proceeds to introduce the results

obtained from the strategies and propose a new model with a proof of concept. Finally, the findings and conclusions derived from the current study are extensively discussed, providing valuable insights and recommendations for future research in the field.

Chapter 2

The Venue Selection Problem

The aim of this chapter is to introduce and explain in detail the problem addressed in this project. The chapter begins by explaining the importance of algorithms in trading. It then proceeds to provide an overview of the FX market, including its functioning, the concept of venues, and the various participants involved. The problem at hand is subsequently described, outlining the current venue selection process used by MN. Finally, the thesis presents different approaches to address this problem, selecting the best and offering potential solutions and enhancements.

2.1 Algorithmic trading

In recent years, algorithmic trading has become increasingly prevalent in financial markets. Algorithmic trading is the use of algorithms to make trading decisions, usually with a high degree of automation [1]. The spread of algorithmic trading has been driven by several factors, including advances in computing technology, improvements in data analysis and modeling techniques, and increased competition among financial firms. In addition, regulatory changes have made it easier for firms to adopt algorithmic trading strategies. Figure 2.1 represents a Goldman Sachs analysis of the spread of algorithmic trading by asset class from 2004 to 2017 and confirms its spread in all the markets.

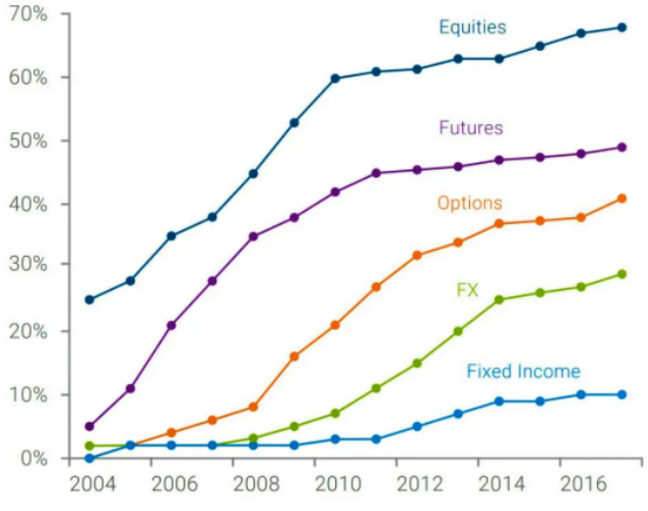


Figure 2.1: Market share of algorithmic trading by asset class [2].

There are two main advantages of algorithmic trading: first, the use of coded algorithms removes emotional biases that could compromise results; and second, algorithms can execute trades much faster than human reactions, providing a significant speed advantage in fast-paced markets. In fact, algorithmic trading can help traders make more informed decisions by processing vast

amounts of data in real-time, identifying patterns and correlations, and executing trades based on predefined criteria. This can lead to improved accuracy and consistency in trading decisions, as well as increased efficiency and cost-effectiveness. In addition, algorithmic trading can also help traders avoid common cognitive biases that can affect human decision-making. By relying on data-driven analysis and objective criteria, algorithmic trading can help traders make more rational and objective decisions. Moreover, algorithmic trading can also provide traders with the ability to test and refine their strategies using historical data and backtesting techniques. This can help traders identify weaknesses in their approach and optimize their strategies for better performance in real-world conditions. For these reasons combined with transparency of the transactions, trading data collection and cost reduction, MN decided to develop its own ruled based execution algorithm: ALGO. Before giving some background on ALGO and its current strategies, it is important to explain how trading works in practize.

2.2 FX trading

The foreign exchange (FX) market, also known as the currency market, is the largest and most liquid financial market in the world, with an estimated daily trading volume of over $6.6 trillion as of 2019 [3]. The FX market is where currencies are bought and sold by market participants (banks, corporations, governments and individual traders). The market operates 24 hours a day, 5 days a week, and spans multiple time zones, from Asia to Europe to the Americas.

One of the primary functions of the FX market is to facilitate international trade and investment. For example, a company in the United States that wants to import goods from Italy will need to exchange US dollars for EUR to pay for those goods. This exchange is typically facilitated through the FX market. Similarly, an investor who wants to buy a foreign stock will need to exchange their domestic currency for the foreign currency in order to make the purchase. In addition to facilitating international trade and investment, the FX market also plays an important role in managing currency risk. Companies that operate in multiple countries may be exposed to fluctuations in exchange rates, which can impact their revenues and profits.

FX market is an example over-the-counter (OTC) market: the trades are conducted directly by all the market participants, without an official market exchange. This fact makes the market decentralized, flexible and accessible, but on the other hand the absence of strict regulation makes it less transparent. Technological advancements have played a significant role in the evolution of the FX market. Electronic trading platforms have made it easier and faster for market participants to execute trades, and algorithmic trading has become increasingly popular.

Before explaining the concept of limit order and order book, it is important to define ticks and pips. In the foreign exchange (FX) market, "pip" and "tick" are both terms used to describe the small price movements in a currency pair. In the EUR/USD market, a pip is equal to 0.0001 while the tick is 0.00001 (0.1 pips). In summary, pips and ticks are both measurements of price movement in the FX market.

2.2.1 Limit orders and market orders

In the basic setup, a market has two types of orders: market orders and limit orders [4]. Market orders are executed immediately: by sending a market order, a trader wants to buy/sell a specific asset immediately, at the best price available. On the other hand, limit orders are not executed immediately. In fact, a limit order constitutes the fact of wanting to buy or sell an asset at a price other than the market price, usually less convenient for the counterpart. Limit orders are executed when matched with other orders and they are stored in a Limit Order Book (LOB, explained in the next subsection).

Overall, the choice between a market order and limit order depends on the investor's investment strategy, time horizon, and risk tolerance. Market orders are suited for investors who prioritize fast

execution, while limit orders provide greater control over execution prices at the cost of slower execution. Passive orders prioritize maximizing price advantage over speed of execution, while aggressive orders prioritize speed over price advantage. Market orders are considered highly aggressive, as they prioritize immediate execution at the best available prices in the market.

2.2.2 What is a Limit Order Book

The Limit Order Book (LOB) is a fundamental component of financial markets that displays all outstanding buy and sell orders for a specific financial instrument at various price levels. Traders can observe the bid (buy) and ask (sell) prices along with their corresponding quantities in the order book. When a trader wants to buy or sell, they can place a limit order specifying their desired price and quantity. If the price of a buy limit order is higher than or equal to the best available ask price, or the price of a sell limit order is lower than or equal to the best available bid price, a trade is executed. The order book matches buy and sell orders based on their respective prices, allowing traders to transact at the desired price levels. Figure 2.2 shows an example of limit order book.

Figure 2.2: A Limit Order Book records outstanding limit orders for a fixed time t. The green columns represent the volumes of the buy orders at each price, the red columns the volumes of the sell orders at each price [5].

One of the most important characteristics of the order book is that it constantly changes as market participants submit, modify, or cancel their limit orders. This dynamic nature reflects the evolving supply and demand in the market, enabling price discovery and efficient trading execution. By observing the order book, traders can assess market sentiment and liquidity, helping them navigate the market and execute their trades effectively. Having defined the LOB, it is now possible to define the trading venues.

2.2.3 What is a trading venue

According to the European Securities and Markets Authority (ESMA), a trading venue is defined as a multilateral system operated by an investment firm or a market operator which brings together multiple third-party buying and selling interests in financial instruments. In other words, the trading venue is a location where buy and sell orders for a financial-instrument are matched [6].

Trading venues are important for providing a centralized location where buyers and sellers can come together to execute trades in a fair and efficient manner. Some of the main properties of a trading venue include its trading rules, fees, transparency, and accessibility. Trading rules are the guidelines for how trades are executed on the platform, including order types, execution methods, and pre-trade and post-trade transparency. Fees are charged by the trading venue for access to the platform, and can include transaction fees, listing fees, and membership fees. Transparency is important for ensuring that market participants have access to relevant information about the market, such as the price and volume of trades. Finally, accessibility refers to the ability of different types of market participants to access the trading venue, which can impact the liquidity and efficiency of the market.

It is possible to distinguish two main groups of trading venues: Central Limit Order Book (CLOB) and Electronic Communication Network (ECN). CLOBs are trading systems that centralize and aggregate all buy and sell orders in a particular security or financial instrument. They maintain a central order book that displays all outstanding bids and offers, along with their corresponding quantities and prices. CLOBs typically provide a transparent and fair trading environment by allowing market participants to see the full depth of the market and place orders accordingly. When a buy order matches a sell order in terms of price and quantity, a trade is executed, and the transaction is recorded [7]. An important example of CLOB venue is AMS.

ECNs are electronic platforms that connect buyers and sellers directly, bypassing traditional intermediaries like market makers or specialists. They display a consolidated order book that includes the bids and offers from various participants, allowing for transparent and direct trading. Orders placed on ECNs can be executed immediately if there is a matching bid/offer available, or they may be added to the order book for future execution [8]. An important example of ECN venue is RTM. The main differences between ECN and CLOB are four:

- **Market Structure**: ECNs facilitate direct trading among participants, while CLOBs centralize and match orders from multiple participants.
- **Order Book Display**: ECNs display a consolidated order book with bids and offers from various participants, whereas CLOBs maintain a central order book. The key distinction is that ECNs consolidate the bids and offers into a single order book, whereas CLOBs maintain the central order book with all individual bids and offers. Both approaches provide transparency, but the display method differs.
- **Access to Participants**: ECNs offer access to a broader range of market participants, including institutions and retail traders, while CLOBs are typically accessed by specific market participants or members.
- **Transparency**: Both ECNs and CLOBs provide transparency, but ECNs often offer more visibility into individual participant orders. Having a consolidated order book it could be that some order are not visible to all the participants.

The explained differences could influence the market participant's choice to trade in one or another kind of venues. Overall, trading venues are essential for the functioning of financial markets, and the properties of a trading venue can have a significant impact on the efficiency and transparency of the market. If on one hand the price of currencies/assets in all the venues should be the same, the differences between venues (CLOB/ECN, market participants, different volumes etc.) could affect a trading strategy performance. In order to understand who are the market participants in the trading venues, the next section gives an overview about the two main categories of traders: market makers and market takers.

2.2.4 Market makers and market takers

To understand more, it is necessary to introduce some important concepts. First of all, let's define the interpreters acting in trading venues:

- A *(prime)-broker* is an intermediate party between the individual trader and the trading venue [6]. In general the individual traders are not able to place orders in the venues and they need brokers to indirectly trade.
- A *market maker* is an individual or firm that quotes both bid and ask prices for a specific security, providing liquidity to the market by ensuring that there are always buyers and sellers for that security. Market makers profit by buying securities at the bid price and selling them at the ask price, which creates a bid-ask spread. The bid-ask spread represents the market maker's profit margin and also serves as a measure of liquidity in the market.

  Market makers can also engage in principal trading, where they trade securities for their own accounts, rather than for clients. Principal trading can be a significant source of revenue for market makers, but it can also create conflicts of interest. To address this concern, some regulators require market makers to disclose their trading activities and to separate their market making and proprietary trading operations.

  Overall, market makers are important participants in financial markets, providing liquidity and depth to the markets they operate in. By actively quoting both bid and ask prices,

they ensure that buyers and sellers can always find a counterparty for their trades, which is essential for the efficient functioning of financial markets [9].

- A *liquidity pool* is a group of market participants who place buy and sell orders in an order book in order to provide liquidity. A trading venue may have several different liquidity pools into which a trader can place orders [6].
- A *market taker* is an agent who needs liquidity to ensure a reasonable price exists whenever they need to enter a trade or close an existing position. This allows them to execute trades at a reasonable price. They recognize that, in order to utilize market makers for trade execution, they may need to give up certain advantages, such as price advantages. Market takers tend to hold their positions for longer periods than market makers, which means they are generally less concerned about trading costs [10]. Figure 2.3 shows the main differences between market makers and takers.

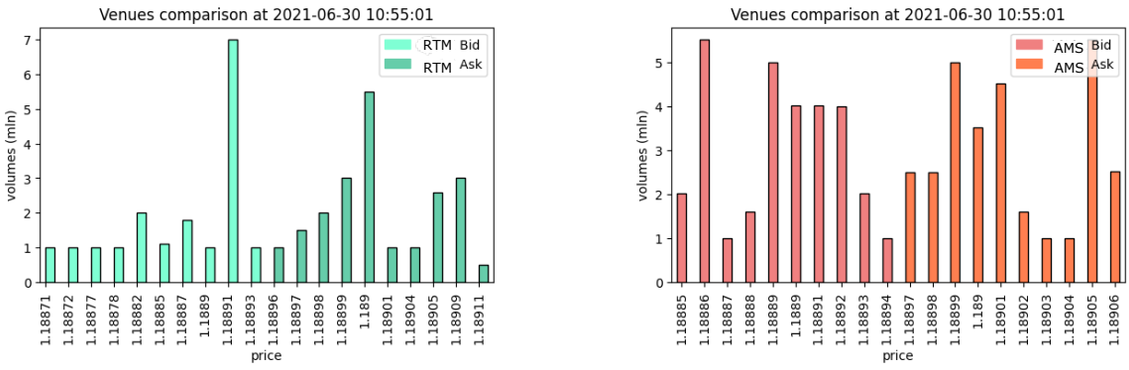
Figure 2.3: Market makers and market takers comparison [10].

To put this thesis into context, it is necessary to specify that MN is a market taker. FX trading is in fact performed to limit exposure to foreign currency risk and is not the main business of the firm. Given the size of the assets being traded, it is crucial for MN to trade them using the most optimised strategy possible. Even small improvements can equate to large potential portfolio efficiencies and/or cost savings. However, note that MN, using its own algorithms and trading large quantities, places substantial orders in the venues over time at a price calculated as fair. It could in fact be described as a *one-side market maker* that uses information from both venues to find a good buy/sell price for the currencies.

2.3 Problem description

The venue selection problem in trading refers to the challenge of choosing the optimal trading venue for executing a trade. This decision involves considering factors such as cost, speed, and quality of execution, as well as regulatory requirements. With the rise of electronic trading, the number of trading venues has increased, making the venue selection problem more complex [6].

In recent years, the venue selection problem has become increasingly important due to the growth of algorithmic trading and the increasing fragmentation of liquidity across multiple trading venues. With the growth of high-frequency trading, the speed of execution has become a key factor in the venue selection process, as traders seek to minimize latency and maximize the speed of execution [4]. A venue is distinguished from another venue by the difference in traded volumes, different spreads between bid and ask and other features. It is important to say that for the absence of arbitrage the venues have to have the same macro characteristics, but selecting and using deep features to study a venue it is possible to spot some differences between the venues. At exactly the same timestamp, venues could have different volumes and different prices (Figure 2.4) and these variables could determine the execution/not execution of an order in the next milliseconds.



(a) RTM - EUR/USD (b) AMS - EUR/USD

Figure 2.4: Comparison between the venues at 2021-06-30, 10:55:01.00

Figure 2.4 show the bid-ask prices for the euro currency in two different venues at the same time stamp (2021-06-30, 10:55:01.00). In the RTM venue (2.4a), a market participant who wants to buy 1 million euros immediately would have to pay 1.18896 per unit, while selling the euro position would yield 1.18893 per unit. In contrast, in the AMS venue(2.4b), the market taker would have to buy at a slightly higher price of 1.18897 per unit and sell at 1.18894 per unit. This example shows that there are differences between venues and the venue selection choice plays a crucial role to optimize the execution of the orders. It should also be noted that the distributions of the volumes present in each venue are completely different and they could play an important role in the venue choice with large volume orders.

In conclusion, fixing an order of buy/sell with a fixed quantity and a fixed price that has to be executed within a time interval, it is possible to define the venue selection problem like the choice of the best venue to achieve this aim (Figure 2.5).

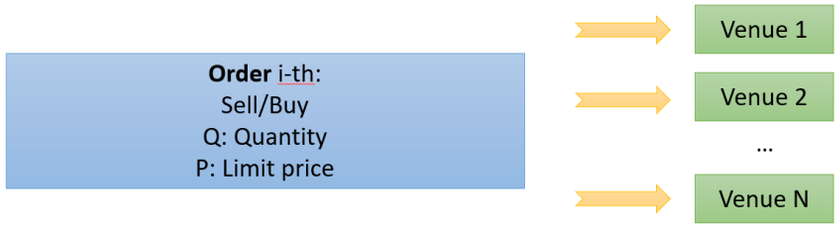


Figure 2.5: Definition of the Venue Selection problem: given a fixed order (blue), which is the best venue (green).

During this project the venue studied and analysed were only two: AMS and RTM. However, the approach used could fit with N venues. Defined the problem, it is now possible to give an overview of the business settings of this research and explain how the venue selection method works.

2.4 Business settings

MN is a Dutch non profit firm that provides pension administration services and pension investment management services with 140 billion euro of assets under management. This project is developed for the MN's Treasury department, specifically trying to improve the trading of foreign currency in order to hedge the currency risk exposure. The MN ALGO platform is an entire FX trading

execution algorithm developed internally by MN. It is the first example of a trading algorithm owned by a pension asset management company. The next subsections explain the main strategies and the venue selection method currently used.

2.4.1 ALGO: the strategies

Different market situations and different trader demands can lead to very different behaviour in the order execution. On one hand, there are some moments in which it pays to be passive while trying to minimise the cost of execution and, on the other hand, there are specific moments in which the trading strategy has to be aggressive putting more aggressive orders to buy/sell an asset in the shortest time period. In this second example the execution costs are usually larger. For these reasons, the ALGO execution algorithm provides two different strategies that can be executed in the two different situations:

- **Time Weighted Average Price (TWAP)**: Generically, TWAP is a family of execution algorithms that aims to execute a fixed order in a specific time interval [...].
- **Float**: The Float algorithm selects a relative position in the order book and remains at that position until a large order of friendly flow fills it when the market has sufficient liquidity. Unlike the TWAP algorithm, executing quickly is not crucial for Float algorithms, and the emphasis is on executing at a favorable spread [11].

This research backtests a strategy that is a copy of ALGO FLOAT strategy. The strategy has been selected because it is the most used strategy and was agreed with the MN quants. The price logic under the backtested strategy is be described in the next chapters.

2.4.2 ALGO: the venue selection method

ALGO is currently connected with two venues, in this research named AMS (headquartered and based in London) and RTM (headquartered in US, based in London) [...]. During the writing of this thesis a new venue selection logic has been implemented: this set of rules is referred to as parallel trading. Given a large volume to be traded, ALGO splits this amount in two quantities [...]. This latter quantities are divided in child orders and traded once per time using both venues [...]. Before the introduction of this logic, orders were sent to one venue per time. The goal of using both venues is to reduce the execution price and to buy/sell the currency quicker.

Figure 2.6 shows the ALGO logic explained: the total volume is splitted in two volumes (OR Operation) and than the consequent child orders are traded simultaneously once per time in both venues (AND Operation).

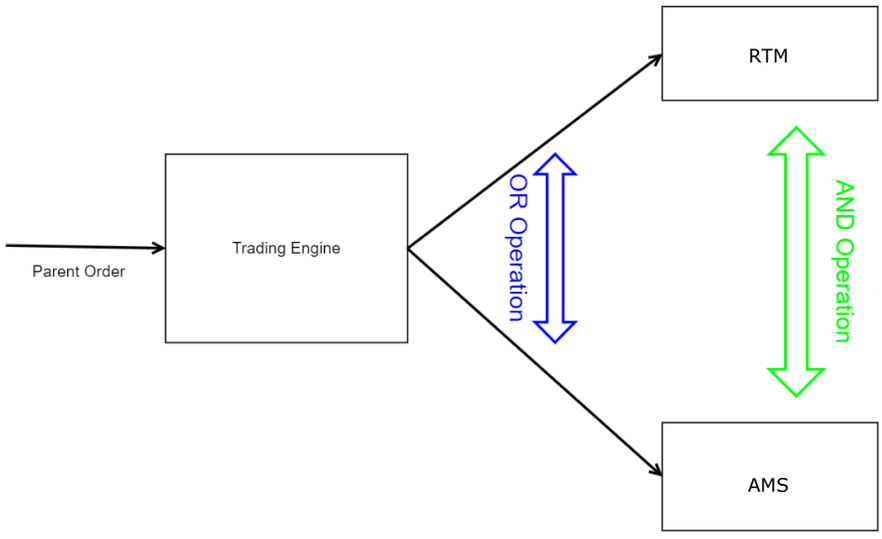


Figure 2.6: ALGO parallel trading scheme [12].

Once the first child order is executed, the second will be processed and executed and so on. Note that the first split is fixed *a priori* and is currently not dynamically calculated using venue market conditions. There is no reallocation of orders in case of partial execution of an order in one venue with the other. A calculation of fill probabilities at all venues could lead to an optimal allocation and possible dynamic reallocation of orders between venues [12].

2.5 Approach

The purpose of this part is to illustrate the logical flow of thought utilized for this thesis, expanding on past studies' methodologies, describing the new methodology provided, and offering techniques for putting it into reality. This topic is extremely rare in the literature and a new approach had to be thought of for the study to try to solve the venue selection.

2.5.1 Previous approach

In 2022, Smith [6] tried to model the problem with a pure probability estimation approach. This project was one of the master theses of the last year AXP. Using the data of all the trading activities of the venues (orders, executions, cancellations), the author tried to train and test some Machine Learning models (Random Forest based models) to forecast the probability of an order to be executed. The dataset used to train the model was the list of the order executed in both venues with some statistics of the order book computed when the order was received by the venues. Comparing the probabilities of executions of the each venue (output of the model), the best venue for the given order was selected within AMS and RTM (Figure 2.7).

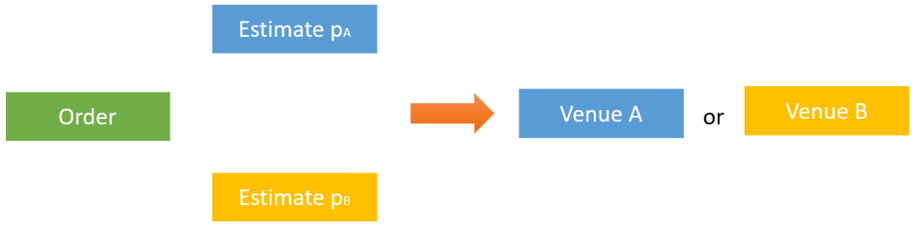


Figure 2.7: Approach used by Smith [6]: given an order, the trained models estimate the probability of execution in both venues and select the best one.

The project was the first venue selection project and this study follows in that vein. If, on the one hand, the data analysis part was read and used to understand the problem of venue selection, on the other hand the obtained Machine Learning models output were not significant. A method that always selects AMS outperformed the venue selection based on the random forest models. In fact, with the proposed choice only the 66.8% of the possible executions were achieved over the test set, while with the trivial AMS choice 74.8%. The results also revealed that the orders in the test set that were executed had high execution probability in both venues. This might imply that at the same timestamp the market conditions in both venues are extremely similar. It might also imply that for execution, it makes no difference whether an order is placed at one venue or another. If the market circumstances are favorable, the order will be fulfilled regardless of the venue.

The main problem of this approach is that the dataset used was a set of all the historical orders sent in both venues. As shown in Chapter 5, this kind of data contained a bias in the venue selection problem due to the larger liquidity in AMS with respect to RTM. In fact, being AMS the venue with more orders inserted, more orders are executed in this venue compared with RTM: this result, however, does not mean that one venue is better than the other, only that there are more market participants (the venue is more *liquid*). Because of this, the dataset used by Smith to train the random forest model, contained a bias: using a dataset with all the executed orders in both venues, the dataset was heavily unbalanced. Using this approach, AMS seems to be better in every situation and even using over/under-sampling techniques on the dataset. The obtained results were indeed not significant. From this previous research, it is clear that to deeply analyse the venue selection problem, the approach has to be completely changed.

2.5.2 Parallel trading approach

This research changes completely the approach used to try to solve the venue selection problem. In fact, one has to consider the fact that the logic of ALGO in the last period has been modified to allow parallel trading (see 2.4.2): both venues are used at the same time for the child orders chains. In order to exploit both the execution capabilities of the venues, the idea is to split an order in between the venues using different weights (Figure 2.8). The weights will depend on the probability of execution of one order: the higher the probability of a venue w.r.t. the other one, the higher the weight.

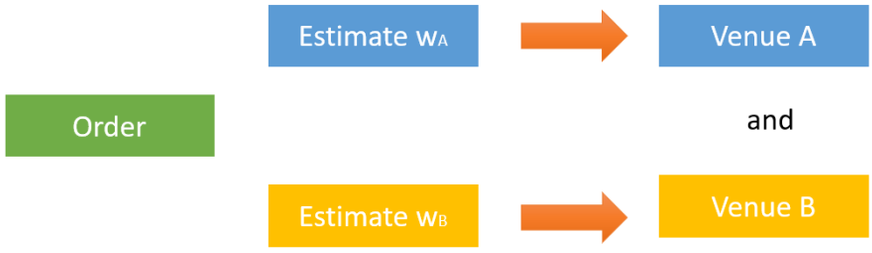


Figure 2.8: Parallel trading approach: given an order, the trained models estimate the probability of execution in both venues and assign weights to both.

At this moment, once the allocation is chosen, until the full execution of the orders ALGO cannot modify the allocation. A possible improvement could be a further modification that will optimize the trading procedure. This research tries to improve the ALGO trading procedure now: the focus is on improving the best initial order quantity allocation.

It is now possible to write down a mathematical formulation of the optimization problem described. Let's start with the constraints. Defining $w_A(t)$ the AMS weight at time t and $w_R(t)$ the RTM one, it is trivial to note that the weights follow the following relations for all t:

$$w_L(t) + w_C(t) = 1$$
$$w_L(t), w_C(t) \geq 0 \tag{2.1}$$

The first constraint is trivial and allows only one weight to be studied: this research will focus on the estimation of the optimal $\hat{w}_L$ (given it, the computation of $\hat{w}_C(t)$ is indeed trivial). The second constraint means that all the weights need to be positive: in all the strategies adopted, MN does not want to take any short position against the currency it wants to trade in one venue to trade in the other one.

Defined the constraints, it is possible now to define a vector x_t that represents all the features of both venues at time t. Furthermore, it is important to select a fixed trading strategy described by a set of rules Ω. Now, given w_A, x_t, Ω, let's define $f_\Omega(x_t, w_A)$ the function that describes the time of execution of the defined strategy. The optimization problem can be written as:

$$\hat{w}_L(t) = \arg\min_{w_A} f_\Omega(x_t, w_A) \tag{2.2}$$

This formula describes the goal of this research. Note that the output value of function f_Ω depends on the selected trading strategy Ω, the selected input statistics x_t and the selected weights w_A, w_R. The next chapters will describe the selected important statistics in x_t, the chosen strategy Ω and the chosen technique to estimate $\hat{w}_L$. The completely new construction of the descriptive features of the venues, the generation of new data, and the choice of models to train are not only completely different and innovative but also crucial factors that undeniably enhance the overall impact and validity of this study.

2.6 Starting point summary

This chapter has explained the starting point of this study. It is now important to summarize the highlights before continuing the research:

- ALGO, the MN algorithmic execution engine, is connected with two different venues: AMS and RTM.
- In general, given a limit order with a direction (buy/sell), a fixed price and a fixed quantity, the venue selection problem consists in the selection of the best venue in order to achieve the fastest execution for it (see Figure 2.5).

- ALGO computes and uses the same price in both venues: the venue selection is not about price, only speed of execution.
- ALGO is currently using a new parallel trading approach. Given a big order to be executed, it is splitted in two volumes [...].
- ALGO trades child orders in both venues [...].
- The previous research used the order executions data from both venues, trying to train some Machine Learning models to forecast the probabilities of execution. Despite this, the obtained results were not significant: a method that always selects AMS outperformed the venue selection based models. The approach, consistent with the old ALGO trading method, wanted to select only one venue: the one where the execution of the order had the higher probability (see Figure 2.7).
- This research proposes a new method based on the optimal order allocation selection for ALGO, compatible with the parallel trading approach developed during the time of this research. Equation 2.2 defines the problem in a mathematical point of view, while Figure 2.8 shows the approach.

Defined the starting point, it is important to mention that in the literature there is nothing directly applicable to this case. The venue selection problem is strongly linked to the industry, and its significance lies in the fact that there is a scarcity of useful publications addressing this specific issue. This implies that there is a lack of readily available knowledge and established guidelines for making informed decisions regarding venue selection in trading. Moreover, the logical approach of trying to estimate execution probabilities using execution data from both venues did not lead to any good results [6]. It is therefore necessary, based on the available data, to analyse and find out why the old approach did not work and to propose something new to try and estimate the values of w_A, w_R.

Chapter 3

Mathematical Framework of Order Book data

This section explains the general mathematical framework and the notation used in this research. It is divided in a first section focused on the mathematical description and statistics of the limit order book for a fixed time, a second section that considers the rolling time statistics and the last section that gives a general overview of the most important Limit Order Book (LOB) mathematical models. For all this section the traded pair is considered fixed (e.g. EUR/USD).

3.1 Properties of LOB for a fixed time

The goal of this first section is try to describe the LOB using all the information given in a fixed snapshot, taken in a fixed time t. Information is contained in statistics, with a rigorous mathematical definition that can be computed looking at the available data.

Definition 1 (Order) Given a specific direction d_x (buy or sell), a price p_x, a quantity q_x and a time of placement t_x a buy or sell order x is defined as:

$$x = (d_x, p_x, q_x, t_x)$$

In general, limit and non-market orders will be analysed in this thesis. When the term order is used, it will therefore be associated with limit order.

Definition 2 (Limit Order Book) A Limit Order Book (LOB) $\mathcal{L}(t)$ is the set of all active orders in a market at time t.

In the LOB, the orders are divided in *bid* (sell) and *ask* (buy). Usually large orders are divided into smaller orders executed little by little in order to limit the cost of execution. The total order is called *parent order* with a volume V, the small orders are the *child orders* with volumes $Q_1, \dots, Q_n$ such that $\sum_{i=1}^{n} Q_i = Q$ [11].

There are also numerous values that can help describe the status of an order book: the best bid, the best order and their values with respect to a value V.

Definition 3 (Best bid [11]) Given a Limit Order Book $\mathcal{L}(t)$, for a fixed time t, the best bid is defined as:

$$B(t) = \max_{\substack{x \in \mathcal{L}(t) | \\ d_x = buy}} p_x$$

Definition 4 (Best ask [11]) Given a Limit Order Book $\mathcal{L}(t)$, for a fixed time t, the best ask is defined as:

$$A(t) = \min_{\substack{x \in \mathcal{L}(t) | \\ d_x = sell}} p_x$$

Definition 5 (Best bid w.r.t. V [11]) Given a Limit Order Book $\mathcal{L}(t)$, for a fixed quantity V, the best bid with respect to V is defined as:

$$B_V(t) = \max\left\{ p_x : x \in \mathcal{L}(t) \text{ s.t } \sum_{\substack{y \in \mathcal{L}(t)| \\ d_y = buy, p_y \geq p_x}} q_y \geq V \right\}$$

Definition 6 (Best ask w.r.t. V [11]) Given a Limit Order Book $\mathcal{L}(t)$, for a fixed quantity V, the best order with respect to V is defined as:

$$A_V(t) = \min\left\{ p_x : x \in \mathcal{L}(t) \text{ s.t } \sum_{\substack{y \in \mathcal{L}(t)| \\ d_y = sell, p_y \geq p_x}} q_y \geq V \right\}$$

Note that if $V = 0$, than $B_0(t) = B(t)$ and $A_0(t) = A(t)$ (the best bid/ask is always the one at the top of the list). Other important variables are the spread, the mid-price and the pip.

Definition 7 (Spread [11]) Given a Limit Order Book $\mathcal{L}(t)$, the spread is defined as the difference between the best ask and the best bid at time t:

$$S(t) = A(t) - B(t)$$

Definition 8 (Mid Price [11]) Given a Limit Order Book $\mathcal{L}(t)$, the mid price (or simple mid), is the middle point between best ask and the best bid at time t:

$$M(t) = \frac{A(t) + B(t)}{2} = B(t) + \frac{1}{2}S(t)$$

The tick π of a LOB is defined as the minimum possible price movement. In the equities market usually a tick is equal to 0.01\$, for EURUSD in the FX market 0.00001\$. The spread of bid and ask is usually measured in pips: for the FX market a pip is defined as 10 pips (0.0001 for both EURUSD and EURGBP and 0.01 for EURJPY [11].

Definition 9 (n tick bid side volume [6]) Given a Limit Order Book $\mathcal{L}(t)$, the n ticks bid side volume is defined as the total volume on the bid side of the order book. That is:

$$V^B_{ntick}(t) = \sum_{\substack{x \in \mathcal{L}(t) | d_x = buy, \\ p_x \in P^B(t), P^B - \pi(t), \ldots, P^B - (n-1)\pi(t)}} q_x$$

Definition 10 (n tick ask side volume [6]) Given a Limit Order Book $\mathcal{L}(t)$, the n ticks ask side volume is defined as the total volume on the ask side of the order book. That is:

$$V^A_{ntick}(t) = \sum_{\substack{x \in \mathcal{L}(t) | d_x = sell, \\ p_x \in P^B(t), P^B + \pi(t), \ldots, P^B + (n-1)\pi(t)}} q_x$$

To better understand the definitions introduced, it's sufficient to observe an example of a FX Limit Order Book for a fixed time t. The pip value is 0.0001.

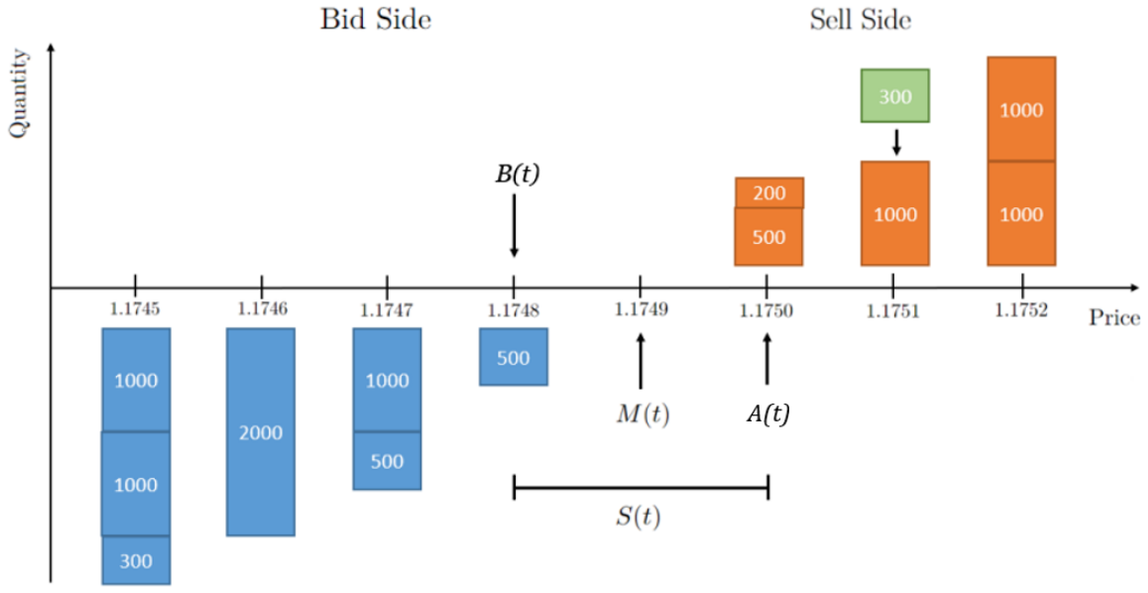


Figure 3.1: Example of the introduced metrics for a FX Limit Order Book. In the Figure the values of the best bid $B(t)$ and the best ask $A(t)$ are highlighted [6].

In the Example in Figure 3.1 the values of $B(t)$, $O(t)$, $M(t)$ and $S(t)$ are indicated. It is also possible to calculate the other values defined before:

$$B_{600}(t) = 1.1747$$

$$A_{600}(t) = 1.1750$$

$$V^B(t) = 6300$$

$$V^A(t) = 3700$$

$$V^B_{4ticks}(t) = 2000$$

$$V^A_{3ticks}(t) = 2000$$

All the statistics illustrated so far do not clearly show any imbalance between volumes. As will be explained in detail below, the LOB is a constantly changing system and it may be crucial to be able to measure any imbalance between bid and ask. It's further possible to define the bid-ask volume imbalance and the smart price [13].

Definition 11 (Bid-ask volume imbalance) Given a Limit Order Book $\mathcal{L}(t)$, fixing a number n of ticks, the bid-ask volume imbalance is the ratio between volumes included in n ticks from mid price:

$$I(t) = \frac{V^B_{nticks}(t)}{V^A_{nticks}(t)}$$

Definition 12 (Smart bid) Given a Limit Order Book $\mathcal{L}(t)$, considering the bid side and fixing a number N of levels, the smart bid is defined as the weighted sum of each level times the corresponding volume:

$$SB_l(t) = \sum_{i=1}^{l} \frac{P^B_i(t)}{V^B_i(t)}$$

where $V^A_i(t)$ is the total volumes of ask in correspondence with the prices $P^B_i(t)$.

Definition 13 (Smart ask) Given a Limit Order Book $\mathcal{L}(t)$, considering the bid side and fixing a number l of levels, the smart ask is defined as the weighted sum of each level times the corresponding volume:

$$SA_l(t) = \sum_{i=1}^{l} \frac{P_i^A(t)}{V_i^A(t)}$$

where $V_i^A(t)$ is the total volumes of ask in correspondence with the prices $P_i^A(t)$.

Definition 14 (Smart price) Given a Limit Order Book $\mathcal{L}(t)$, considering the values of smart bid $SB_l(t)$ and smart ask $SA_l(t)$, it is possible to define the smart price as:

$$SP_l(t) = \frac{SB_l(t) + SA_l(t)}{2}$$

Definition 15 (Price distance - buy case) Given a Limit Order Book $\mathcal{L}(t)$, a buy limit order posted at time t with a price $P(t)$, considering the values of best ask $A(t)$, it is possible to define the price distance for the buy order as as:

$$\Delta P_{buy}(t) = A(t) - P(t)$$

Definition 16 (Price distance - sell case) Given a Limit Order Book $\mathcal{L}(t)$, a sell limit order posted at time t with a price $P(t)$, considering the values of best bid $B(t)$, it is possible to define the price distance for the sell order as as:

$$\Delta P_{sell}(t) = P(t) - B(t)$$

3.2 Rolling features of LOB

The previous section discussed the properties of the LOB in a specific snapshot taken at time t: all the statistics introduced can give information only on the current state of the order book not considering the past. It is important to consider that rolling statistics play a crucial role in the financial markets description. Financial time series data represents a sequence of observations over time, and rolling statistics are used to track the changes in market prices, volumes, and other financial indicators. The aim of this section is define the most common rolling features used to describe the LOB: the rolling window T is fixed (it could be every timedelta e.g. 1sec or 1hour). I

Definition 17 (Log returns) Given a Limit Order Book $\mathcal{L}(t)$, considering the values of mid price $M(t)$ and $M(t-T)$, it is possible to define the logarithmic return at time t w.r.t. T as:

$$r_T(t) = \log(M(t)) - \log(M(t-T))$$

Definition 18 (Mid prices moving average) Given a Limit Order Book $\mathcal{L}(t)$, considering all the values of the mid prices from time t to time $t-T$ (e.g. $M(t)$ ... $M(t-T)$), it is possible to define the mid prices moving average as:

$$\mu_T(t) = \frac{1}{T} \sum_{i=T-t}^{t} M(i)$$

Definition 19 (Volatility) Given a Limit Order Book $\mathcal{L}(t)$, considering all the values of the returns from time t to time $t-T$ (e.g. $r_T(t)$ ... $r_T(t-T)$), it is possible to define the volatility at time t w.r.t. T as the standard deviation of these values:

$$\sigma_T(t) = \sqrt{\frac{1}{T-1} \sum_{i=t-T}^{t} (r_T(i) - \mu)^2}$$

with μ defined as the mean of the returns in the selected time interval.

Definition 20 (Order Flow) Given a Limit Order Book $\mathcal{L}(t)$, order flow is the quantity of buy and sell orders that are received in the LOB in the time interval $[t-T, t]$.

Chapter 4

Selected models

This chapter explains the theory of the used models in this research. The first part gives an overview about the time series theory and the correlation statistics. The second part introduces the Principal Components Analysis (PCA), used to preprocess data with high correlation. The models are than explained: linear regression model, XGBoost and Neural Networks. To give an introduction about the use of Machine Learning in Finance, some pillars are explained: robustness to overfitting, explainability, consistency and quickness.

4.1 Time series theory

This section wants to explain the main theoretical concepts behind the time series analysis used in this research. It starts defining a time series and than it explains the concept of stationarity used to study the correlation between the features of the data.

Definition 21 (Time series [14]) A time series is a doubly infinite sequence

$$(X_t)_{t\in\mathbf{Z}} = (\dots, X_{-2}, X_{-1}, X_0, X_1, X_2, \dots)$$

of random variables.

Definition 22 (Time series data [14]) Time series data consists of a finite collection of real numbers,

$$x_0, x_1, \dots, x_n.$$

In summary, a time series is the concept of a sequence of data points ordered by time, while time series data refers to concrete set of observations.

4.1.1 Stationarity

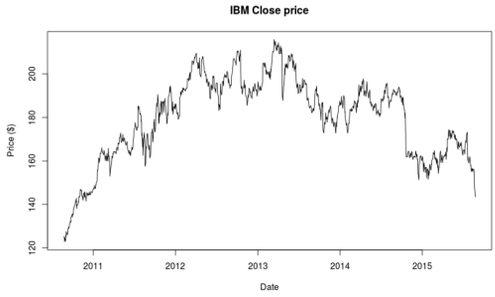
This subsection wants to define the concept of stationary time series. When a time series is stationary, patterns and relationships observed in the past hold in the future. This empowers analysts and researchers to create reliable predictive models that can anticipate future trends and outcomes. In order to define the concept of stationarity, it is important to understand what is the autocorrelation function $\gamma_x(\cdot,\cdot)$.

Definition 23 (Autocorrelation Function [15]) If $(X_t)_{t\in\mathbf{Z}}$ is a time series process such that $\mathrm{Var}\,(X_t) < \infty$ for each $t \in T$, then the autocovariance function $\gamma_x(\cdot,\cdot)$ of $\{X_t\}$ is defined by

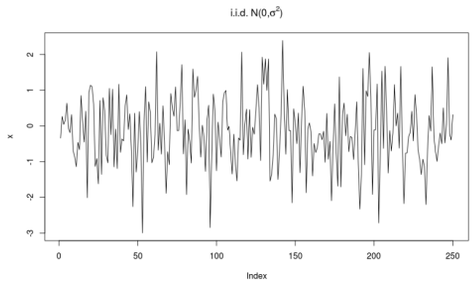
$$\gamma_X(r,s) = \mathrm{Cov}\,(X_r, X_s) = E\left[(X_r - E[X_r])\,(X_s - E[X_s])\right], \quad r,s \in T.$$

Definition 24 (Stationarity [15]) The time series $(X_t)_{t\in\mathbf{Z}}$ is said to be stationary if:
(i) $E|X_t|^2 < \infty$ for all $t \in \mathbb{Z}$,
(ii) $E[X_t] = m$ for all $t \in \mathbb{Z}$,
(iii) $\gamma_X(r, s) = \gamma_X(r + t, s + t)$ for all $r, s, t \in \mathbb{Z}$

In other words, stationarity in a time series refers to the property where statistical properties, such as mean and variance, remain constant over time, enabling reliable analysis and predictions. Figure 4.1 makes a comparison between a non stationary time series and a stationary one.



(a) IBM stock close prices [14].



(b) White noise $N(0, \sigma^2)$ [14].

Figure 4.1: Comparison between a non stationary time series in Figure 4.1a and a stationary time series in Figure 4.1b.

From the Figure 4.1a above, it is clear that the IBM close prices follow a trend which depends on time t, while the white noise generated data in Figure 4.1b do not. The first time series exhibits distinctive characteristics that are inherently tied to the ongoing trend in the stock market. This diverges from the case of a stationary time series like white noise, where this particular attribute does not hold true.stationary time series such as white noise.

ADF test

To check if a time series is stationary it is possible to use the Augmented Dickey-Fuller (ADF) test. The Augmented Dickey-Fuller (ADF) test [16] is a foundational tool in the world of time series analysis and econometrics. Its primary purpose is to figure out whether a given time series is showing something called a "unit root," which is basically a sign that the data isn't stationary. The ADF test assumes that data have a unit root, (non-stationary, null hypothesis) and wants to prove that it is stationary (alternative hypothesis). Performing the test and checking the p-value is possible to conclude if the time series is or is not stationary. All time series whose correlation was analysed in this study were checked with the ADF test.

ACF plot

The autocorrelation function (ACF) plot shows the correlation of a time series with its own lagged values. The horizontal axis of the plot represents the lag or the time interval between the observation and its corresponding lagged value. The vertical axis represents the correlation between the observation and its lagged value, with values ranging from -1 to 1. The correlation between the time series and its value k period ago is contained in the value corresponding with lag k. Figure 4.2 show an example of an ACF plot.

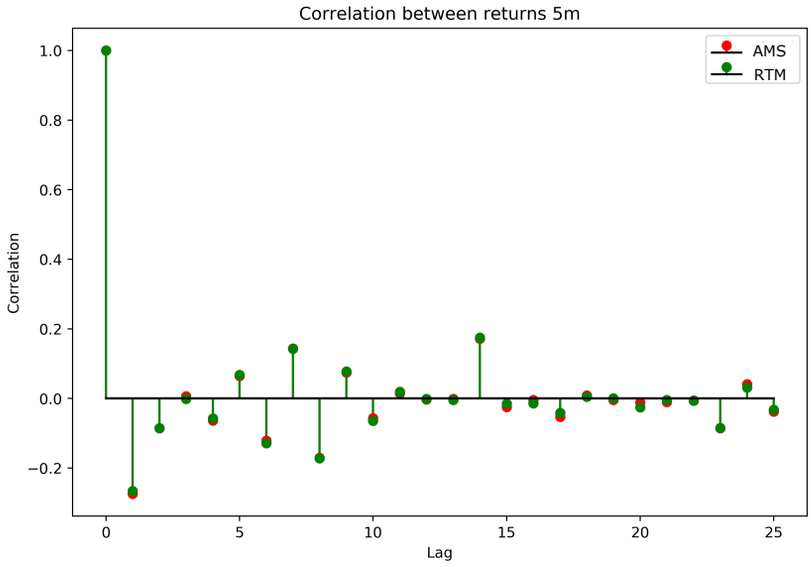


Figure 4.2: ACF rolling returns 5 min of the venues, 30/06/2021.

Let's analyse now the ACF plot in the Figure above. The first important thing to note is that the initial lag is around -0.25 for both venues. This negative correlation observed at the first lag indicates that when the current value of the time series decreases, there's a tendency for the previous time step to exhibit a slightly higher value. This pattern indicates an inverse relationship between these adjacent historical observations. Moreover, the lags following the first have a wave pattern, which proves a reverse correlation of the 5-minutes returns. After explaining and analysing an example of an ACF plot, e- it is possible to explain the preprocessing technique used in this study.

4.2 Principal Components Analysis

After the introductory section about time series analysis, it is possible to explain the preprocess technique used: the Principal Components Analysis (PCA) [17]. Let's assume to have matrix $E \in \mathbf{R}^{n \times p}$ in which every column is a random vector $X_1, X_2, \ldots, X_p$. The variance-covariance matrix of the random variables is defined as Σ. Let's now define, the matrix of the real observations $x_{i,j}$, is in the form:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,n} \\ x_{2,1} & \ddots & \ldots & \vdots \\ \vdots & \ldots & \ddots & \vdots \\ x_{p,1} & x_{p,2} & \ldots & x_{p,n} \end{bmatrix} \tag{4.1}$$

This matrix X represents the dataset. The variance-covariance matrix S derived from observations of random variables is an estimator of general variance-covariance Σ. It is now possible to create a new matrix (transformed dataset) V, defined as:

$$V = A^T X \tag{4.2}$$

with A the matrix of the orthonormal eigenvector of the variance-covariance S.

Explained variance in Principal Component Analysis refers to the proportion of the total variance in the data that is accounted for by each principal component. PCA is a dimensional reduction

technique that transforms a dataset into a new set of orthogonal variables called principal components. The explained variance of each principal component indicates how much information or variability in the original data is retained by that component.

When performing PCA, the principal components are ordered in terms of their explained variance, with the first component explaining the highest variance and subsequent components explaining progressively less variance. By examining the explained variance of each component, one can determine the relative importance or contribution of each component in representing the data.

The explained variance is typically expressed as a percentage and can be calculated by dividing the eigenvalue of each principal component by the sum of all eigenvalues. It helps in understanding the extent to which the principal components capture the variability of the original data. Higher explained variance indicates that the corresponding principal component retains more information from the original data, while lower explained variance suggests that the component may contain less relevant information.

Given the matrix V and the explained variances of the new features, it is indeed possible to select the most important columns (the ones with large explained variance), obtaining a new reduced dataset. The assumption is that the choice to use the new dataset does not reduce (too much) the information contained in it, but simplifies data management.

4.3 Machine Learning in Finance

Before starting to explain the project, the main features that a Machine Learning model for prediction in finance must have are explained [18].

- **Robustness to overfitting**: a model that performs well on the training set, but poorly with other data is completely useless. The market is indeed volatile and can undergo rapid regime changes that could cause problems for the model. It is therefore necessary to train the model using a large quantity of data, trying to avoid overfitting. There are models like XGBoost that guarantees less overfitting problems.

- **Explainability**: predictions must be explained by the input data. Having complex models that give unexplainable results could be a big problem in finance. On the one hand, it is difficult to entrust a portfolio/investment choices to a model that cannot be explained and, on the other hand, some companies are subject to regulation and have to explain their investment choices to the authorities and the models must be clear.

- **Consistency**: a model that is not accurate on a single prediction but is able to predict the sign of the returns (positive or negative) is preferable to a model that is accurate on a single prediction but not accurate in the long run.

- **Quickness**: Especially for Algorithmic Trading applications, the model has to be extremely efficient and quick to train/use. The models must act with a low latency, because even small lags could generate big losses.

Given these four pillars, it is now possible to explain the machine learning techniques used in this research. Note that the pillars were used in all the models selection done during this research.

4.4 Tree-based learners

To explain the eXtreme Gradient Boosting (XGBoost) model used in this research, it is important to understand the Machine Learning tree-based learners. The aim of this section is to explain the concepts between the tree-based learners used for regression [19]. The first object to understand is the decision tree.

4.4.1 Decision Trees

Let's assume to have a dataset of N points (x_i, y_i), $i = 1, \dots, N$ where $x_i \in \mathbf{R}^n$ and $y_i \in \mathbf{R}$. The goal is now to forecast the value of y_i given x_i, constructing the optimal partition of the feature space $\mathbf{R}^n$. In other words, given the maximum number of L subsets, to divide the space $\mathbf{R}^n$ in L subsets, each of which defined as an intersection of a certain number of half-spaces:

$$X_l = \{x \in \mathbf{R}^n : a_{lk}^T x \leq b_{lk}, k \in \{1, \dots, N_l\}\} \tag{4.3}$$

The subsets are built using two rules:

- mutually exclusiveness : for any l, k, the set $X_l \cap X_k$ is either empty or subset of a hyperplane. The probability that a point x_i ends up in the intersection should be zero.
- collectively exhaustiveness: it holds that

$$\bigcup_l X_l = \mathbf{R}^n \tag{4.4}$$

During the training process, the subsets X_l are trained and optimized using a loss function L. Defining $\hat{y}_i$ as the prediction of y_i given x_i, the most common loss function for the regression problems is the Mean Squared Error (MSE):

$$\text{MSE} = \sum_{i=0}^{N} (y_i - \hat{y}_i)^2 \tag{4.5}$$

The selection of the values a_{l_k}, b_{l_k} defines the tree. Given a value x_i, at every step, the value is compared with a_{l_k} or b_{l_k}, defining a specific path into the tree (Figure 4.3).

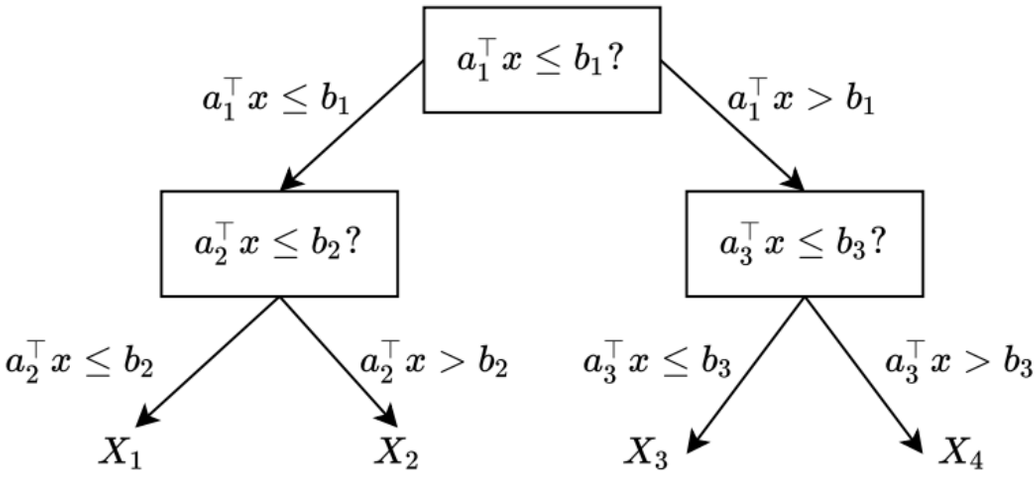


Figure 4.3: Hierarchical partitioning of the space into four subsets [19].

If the process might seem complicated to understand by looking at the formula above, Figure 4.4 shows the desired output in practice, comparing it with a polynomial fit in a regression problem:

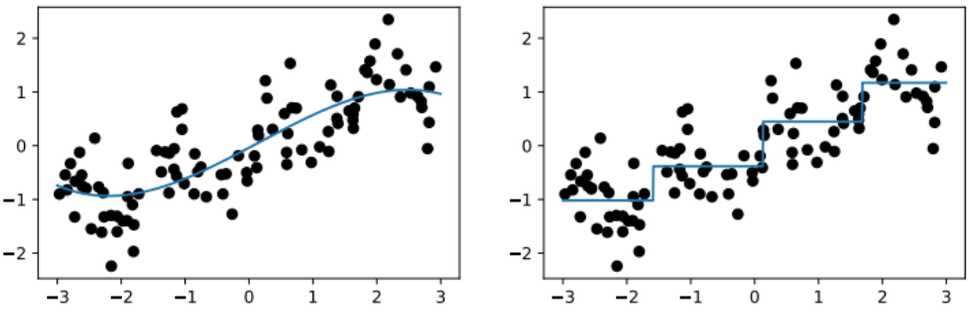


Figure 4.4: Comparison of a polynomial regression of degree 3 with a piecewise-constant regression through domain splitting [19].

One of the main hyperparameter of the decision tree model is the maximum depth d. Attaining a remarkably accurate classification or regression tree frequently entails constructing a tree with considerable depth d. On the other hand, this parameter could lead the model to overfitting problems. It's important to select d basing the choice on a test/validation set or using some hyperparameters tuning techniques [19]. To conclude, despite being machine learning models, decision trees are relatively simple to interpret and quick to fit [19]. To understand more about the optimization techniques for the training of decision trees see [20].

4.4.2 Random Forests

As previously mentioned, achieving a highly precise classification or regression tree often involves creating a tree with significant depth. However, this strategy comes with the drawback of potential overfitting. Furthermore, such a tree might exhibit a degree of arbitrariness. For instance, if two features yield similar outcomes upon splitting, the choice between them becomes less evident. This uncertainty extends to understanding the subsequent impact on further splits down the tree [19]. The solution to this problem for complex tasks was found in the random forest algorithm [21]. The Random Forest Algorithm combines the output of multiple (randomly created) Decision Trees to generate the final output. To figure this process, it's possible to observe Figure 4.5.

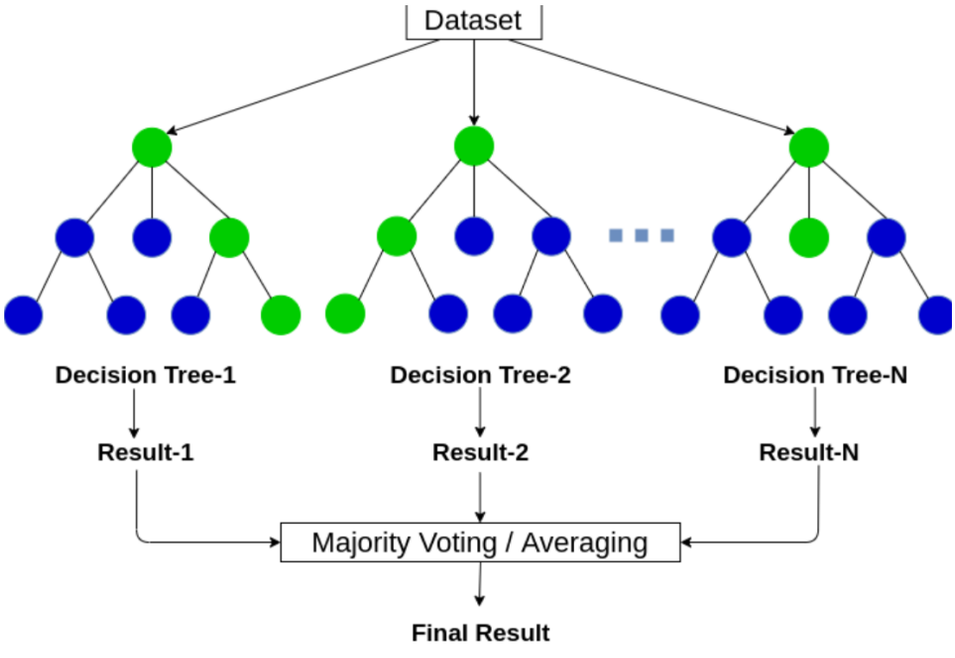


Figure 4.5: Random Forest [22].

The output of each tree are combined with an average (regression problem) and the output $\hat{y}$ is computed. To understand more about the optimization techniques for the training of random forests see [21].

4.5 XGBoost

Rather than relying on the aggregation or averaging of numerous randomly generated trees, an alternative approach can be considered: initialize the process by constructing a single tree. Subsequently, introduce an additional tree that specifically addresses instances for which the prior tree's classifications were erroneous. This method intentionally aims to rectify the shortcomings of the preceding tree rather than following a random approach [19].

To write this process in a mathematical point of view [19]: let's assume that from the training data the first model was built:

$$y = f_1(x) \tag{4.6}$$

Let's define the first model as $model_1 = f_1(\cdot)$. Given it, the idea is to train a second model that can improve the result of the first:

$$\text{model}_2(x) = \text{model}_1 + f_2(x) \tag{4.7}$$

The second model's loss function (using MSE) is:

$$L_2(x) = \arg\min_{f \in \mathbf{F}} \sum_{i=1}^{N} ((y_i - \text{model}_1(x)) - f(x_i))^2 \tag{4.8}$$

Repeating this step for N times, more and more models are created and trained to improve the past models. In general, at the m^{th} step, the $f_m(\cdot)$ model will be:

$$\text{model}_m(x) = \text{model}_{m-1} + f_m(x) \tag{4.9}$$

with the following loss function:

$$L_m(x) = \arg\min_{f \in \mathbf{F}} \sum_{i=1}^{N} ((y_i - \text{model}_{m-1}(x)) - f(x_i))^2 \tag{4.10}$$

This procedure is the base of one of the most common gradient boosting algorithm, the eXtreme Gradient Boosting (XGBoost) [23]. XGBoost, short for eXtreme Gradient Boosting, is a highly effective machine learning algorithm widely used for regression tasks across various domains, including applied mathematics. It has gained immense popularity due to its superior performance, flexibility, and ability to handle complex relationships in the data [1]. For these reason, combined to the fact that it is fast to train and use, it has been chosen as a model to be used in this research.

4.5.1 Hyperparameters

The training process in XGBoost involves optimizing the model's hyperparameters to achieve the best possible performance. Hyperparameters control various aspects of the model, including its complexity, regularization, and learning behavior. The most common hyperparameters in XGBoost for regression include [23]:

- `n estimators`: The number of trees (boosting rounds) in the ensemble.
- `max depth`: The maximum depth of each decision tree. Controls the complexity of the model.
- `learning rate`: The step size for each model update. Smaller values require more boosting rounds but can improve generalization.

In conclusion, optimizing the hyperparameters during the training process of XGBoost is crucial to achieve the best possible performance for regression tasks. These hyperparameters, including the number of estimators, maximum depth, and learning rate, play a significant role in controlling the model's complexity, regularization, and learning behavior, ultimately influencing its predictive accuracy and generalization capabilities.

4.6 Artificial Neural Networks

Artificial Neural Networks (ANNs) are a class of complex Machine Learning models inspired by the structure and function of biological neural networks in the human brain [24]. While ANNs have found success in various domains such as image recognition, natural language processing, and even some areas of finance, there are several reasons why they are not always the preferred choice for financial applications. The main problem is the conflict with three of the four pillars:

- Being complex structures, the training part could be difficult and the training part could be difficult and usually runs into overfitting problems.
- Being complex and deep structures, given the input, it is difficult to understand and explain the output.
- Being complex, both the training and the usage of the ANNs could be very slow. For example ALGO could not use a complex neural network for price prediction due to the need for low latency.

Despite these premises, this study used a proposed simple neural network structure. The structure will be explained in detail in the next chapters, but now it is important to understand the basics of the Neural Network's structure and the sigmoid layer.

4.6.1 General structure

The general structure of an artificial neural network (ANN) consists of three main components: input layer, hidden layers, and output layer. Neurons, or nodes, in each layer are interconnected through weighted connections, and the network's architecture can vary in terms of the number of layers and neurons within each layer. Data is fed into the input layer, and it propagates through the hidden layers to produce an output from the output layer (Figure 4.6).

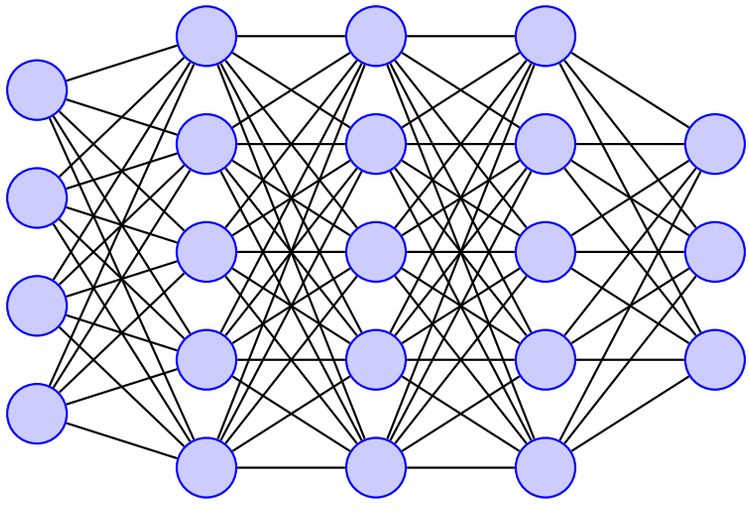


Figure 4.6: General structure of an Artificial Neural Network (realized with Tikz [25]).

Given the input values of each neuron, each one calculates its own activation function and propagates the value obtained to the next layer. Each connection is associated with a weight that is trained during the training process.

4.6.2 Sigmoid function

A sigmoid function can be described as a real function with well-defined boundaries, applicable to all real input values, and characterized by a positive derivative across its entire domain. Being a non linear function, the sigmoid demonstrates a significant level of smoothness [26]. The sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 - \exp^{-x}} \tag{4.11}$$

The output of $\sigma(x)$ is bounded between 0 and 1 (Figure 4.7):

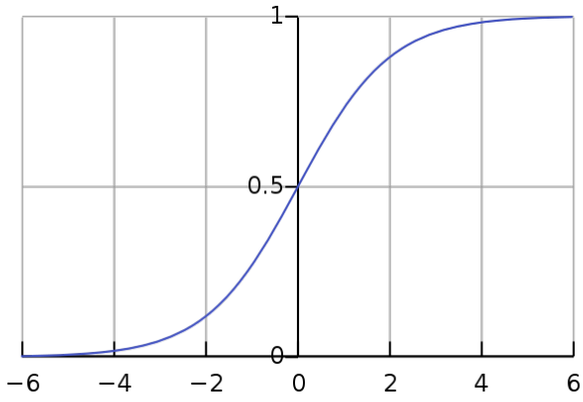


Figure 4.7: Sigmoid function [27].

The sigmoid function is one of the most common activation functions used in the Neural Networks. There are other functions such as ReLu or hyperbolic tangent [28], but in this research only the sigmoid is used. In fact, due to their complexity, the only Neural Network structure used is extremely simple. Again, it will be explained in the next chapters.

4.6.3 Training process

The training process in a Neural Network involves the optimization of model parameters to make the network learn from the provided data and improve its performance on a specific task [28]. This process typically consists of several steps and concepts:

- Data Preparation: the data pre-processing changes depending on the task and the case. This involves collecting, cleaning, and formatting your data into a suitable structure for training. Data is usually divided into three sets: the training set (used for training the model), the validation set (used to evaluate the model's performance during the training) and the test set (used to evaluate the model's performance at the end of the training).
- Model Architecture: selection the number of layers, the type of layers, the number of neurons in each layer, activation functions, and any other architectural choices.
- Inizialization: initialize the weights and biases of the network.
- Loss Function: selection of the loss function depending on the task.
- Forward Propagation: during each training iteration, input data is fed through the network's layers in a forward pass. Each layer performs a weighted sum of inputs, applies an activation function, and passes the result to the next layer. At the end the loss function is computed.

- Backpropagation: this is the process of calculating gradients of the loss with respect to the model's parameters. This is done by applying the chain rule of calculus in reverse order through the network layers. Gradients are used to determine how each parameter should be adjusted to reduce the loss.

The last two steps are repeated until a loss below a limit parameter is reached or an early stopping technique is activated to avoid overfitting. After the results have been calculated, the hyperparameters can be updated and modified, using numerous techniques. The most common technique used for many Machine Learning models is K-Fold Cross Validation [29].

4.7 K-Fold Cross Validation

Cross-validation is a data resampling method to perform hyperparameters tuning of a Machine Learning model [29]. It belongs to the family of Monte Carlo methods. The dataset is split into K subsets, or folds D_i with $i \in [1, K]$. In each iteration, one fold is set aside for validation, while the remaining $K - 1$ folds are used for training. This process is repeated K times, with each fold serving as the validation set once (Figure 4.8).

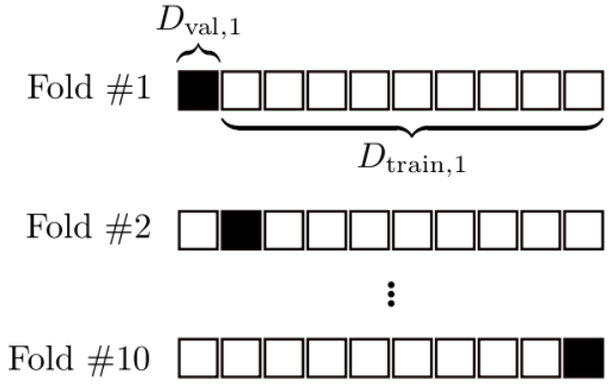


Figure 4.8: K-fold Cross Validation: at every step the black set is used as validation set and the white ones as training set [29].

During hyperparameter tuning, various combinations of hyperparameters are tested in these iterations. The goal is to find the set of hyperparameters that results in the best average performance across all validation folds.

K-fold cross-validation helps prevent overfitting [29]. In fact, the hyperparameters are trained used all the available sets multiple times, and each subset D_i is used both as training and validation set. This technique reduces the risk of the model memorizing specific patterns in the training data and provides a more reliable estimate of its real-world performance.

Chapter 5

First Data Analysis

The aim of this chapter is to analyse individual orders posted at venues. The first two sections give an overview of the available data and the importance of the venue selection in the FX market. The last section explains the analysis of orders using a 1-day dataset, comparing the size, side, elapsed time and volumes traded between venues. This first analysis is very important to try to grasp part of the complex dynamics of trading venues and to have concepts on which to base the subsequent study.

5.1 Dataset available

In order to effectively understand the problem and construct accurate models, it is crucial to understand the available datasets and the primary characteristics of the AMS and RTM data. Specifically, it is important to differentiate between two main types of data: the *tks* data and the *trd* data.

- The *tks* data is generated by new orders, cancellations, and modifications. All new orders received by the venue are stored in the *tks* data, identified by a unique id (*uid*), along with their corresponding time, side, quantity, and price. Cancellations and modifications of an order are also saved in this dataset, along with their respective timestamps.
- The *trd* data is generated by trades (market orders). In the LOB simulations, all orders are considered limit orders, including market orders which are read as limit orders. For example, if a firm buys 1m€ for the unit price of 1.18$, the order will be entered into the LOB as a limit order, regardless of whether it was originally a limit or market order. This is because market orders cannot differentiate between the types of orders executed, and they are therefore considered *very aggressive* limit orders that are executed immediately.

The available data are referred to all the orders insert in both venues in the selected days by all the market participants. In this research a set of heterogeneous days (different days of the week belonging to different months) is selected and studied. The choice is due to the very large size of the datasets available and the resulting data-processing time.

5.2 Overview: Complexity and importance of the venue selection (high frequency features analysis)

The aim of this subsection is to explain why the venue selection problem is important in trading showing the presence of information asymmetry between traders. High frequency data ($5ms$) will show the fact that the venue selection problem is on one hand complicated because the complexity of the orderbooks, and on the other hand important to improve the efficiency of the trading strategies. Note that taking into consideration that using high frequency features to spot opportunities into

the venues, the code must be efficient and fast. Strategies work best if competitor traders cannot compare the two venues: in this way MN could use the information asymmetry created to its advantage.

The selected time interval to study is 5 ms and the selected day is 30/06/2021 from 9:00:02 to 9:00:10. This eight seconds time interval is sufficient to show a general pattern present in all the datasets analysed that proves the information asymmetry. The features to study are the best bid and best ask in both venues: comparing this values every high frequency timestamp, it is possible to spot some trading opportunities. In Figure 5.1 it is possible to observe the changes in the best prices are happening in a time interval of just 6 seconds in both venues.

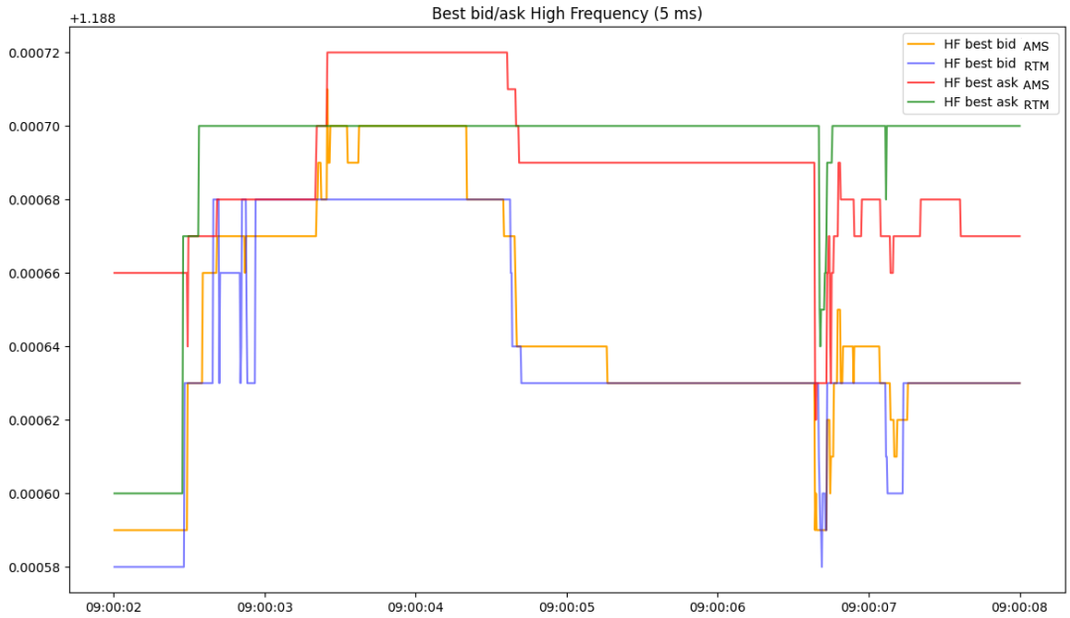


Figure 5.1: High frequency data of bid and ask in both venues from 9:00:02 to 9:00:08 of 30/06/2021.

Observing the best bid in one venue and the best ask in the other one is clear that some traders do not have access to both of them. For instance, around 9:00:03 the blue line and the red line have the same values. This fact means that in RTM (at least) one trader wanted to sell EUR for 1.18868 and in AMS (at least) one trader wanted to buy EUR for the same price. Not being in the same venue, the trade did not happen, but this fact could be a useful advantage for an informed trader. On the other way around, around 9:00:04 the opposite fact occurred: in RTM (at least) one trader wanted to buy EUR for 1.18870 and (at least) one trader wanted to sell EUR for the same price. Again, the trade did not happen for the same reason. These situations last a few fractions of a second, but using high frequency features could be spotted and used to find the most competitive price. Moreover, looking at the high frequency data of both venues it is possible to spot some arbitrage opportunities. In order to spot arbitrage opportunities, it is possible to study the inverted spread. Equations 5.1 and 5.2 contain the precise mathematical definition of the inverted spread.

$$\mathrm{IS}_1(t) = A^{\mathrm{AMS}}(t) - B^{\mathrm{RTM}}(t) \tag{5.1}$$

$$\mathrm{IS}_2(t) = A^{\mathrm{RTM}}(t) - B^{\mathrm{AMS}}(t) \tag{5.2}$$

When the inverted spread is negative there is an arbitrage opportunity: it is possible to buy and sell the asset at the same time using both venues having a gain without any risk. Let's assume, for instance, that at time $\hat{t}$ in AMS the best ask is 1.20 and in RTM the best bid is 1.21. The inverted spread $\mathrm{IS}_1(\hat{t})$ is indeed equal to $1.20 - 1.21 = 0.01 < 0$. To use the arbitrage, a market participant could buy the asset in AMS taking the best ask and sell it in RTM for the best bid. This is an arbitrage because without any risk exposure, the market participant has made a profit

of 0.01 per unit. Figure 5.2 shows the inverted spread values for 10 seconds. The choice of such a small interval is due to the fact that arbitrages last extremely short as informed participants tend to exploit them immediately.

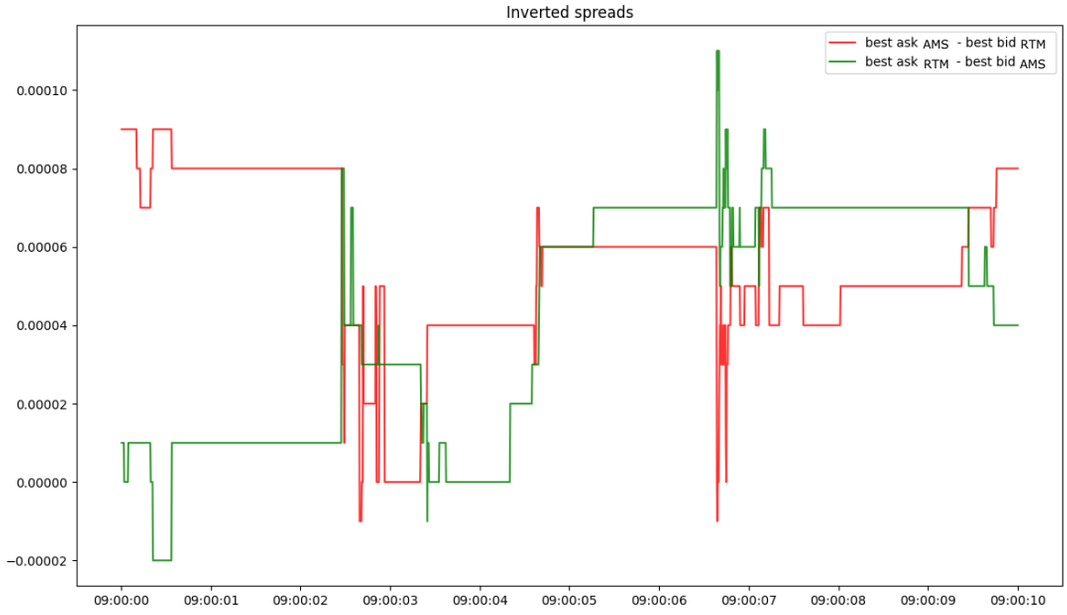


Figure 5.2: Inverted spreads from 9:00:00 to 9:00:10 of 30/06/2021.

The arbitrages in Figure 5.2 are extremely difficult to exploit (the negative peaks return immediately positive). It is important to specify that MN is not interested in exploiting market arbitrages. This is in fact not the goal of a pension fund, but of a high frequency trading firm. This research will simply find the most competitive prices to trade, not develop any arbitrage strategy. Furthermore, to exploit arbitrage between venues, one must have an infrastructure that guarantees low latency and speed of execution in the market. The presence of fast arbitrages is still important to prove the fact that using a efficient venue selection method, the execution's time and price can be improved by far, but the complexity of an environment in which every fraction of ms a lot of people are trading make the problem extremely complex and challenging.

5.3 Order analysis

To effectively compare venues, it is essential to select and analyze various statistics using different time intervals. In this subsection one specific dataset for each venue is analysed (2021-09-13, h. 9:00 - h. 17:00). The weekday has been selected randomly in the data available, and the time interval is selected depending on the fact that most trades take place at those times. It is important to notice that the chosen day does not present any special event that could influence the FX market (FED announcements, important political events etc.).

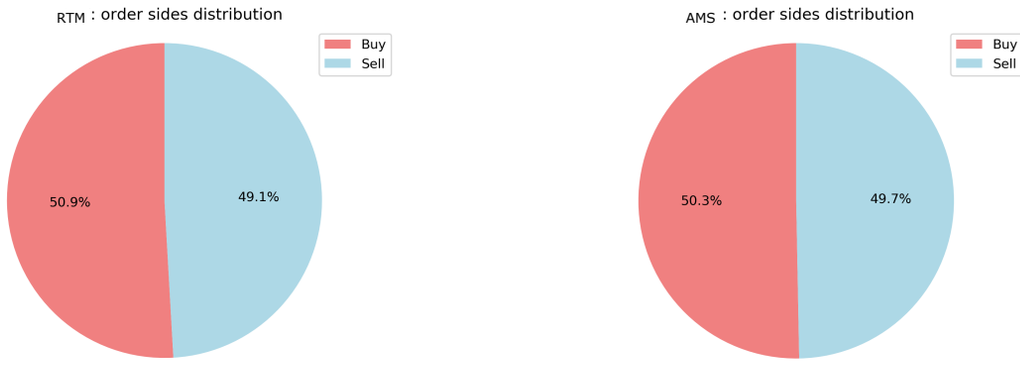
5.3.1 Venue liquidities

First of all it's possible to note that the number of orders in each venue is completely different: AMS has $\approx$ 2.5 million orders, while RTM only $\approx$ 500k. This fact is recurrent in all datasets analysed and it is the main difference between the two venues: AMS is more used than RTM (more investors are trading in AMS), consequently AMS could be more liquid. A *liquid* trading venue is a marketplace or exchange where there is a high volume of buyers and sellers, allowing for easy and quick execution of trades without significant price fluctuations. The difference between the number

of orders in the two venues plays a key role in the venue selection method. In order to compare the orders sent to the venues it is possible to analyse the size and side of each order.

5.3.2 Order sides

The first trivial question to ask is whether, for some reason, more buy orders might be sent in one venue and more sell orders in the other. Comparing the side of the orders in both venues it is possible to note that there is a perfect balance (Figure 5.3). In fact, RTM presents 50.9% of buy orders and AMS 50.3%: no important differences are noticed.



(a) RTM - EUR/USD (b) AMS - EUR/USD

Figure 5.3: Comparison between the sides of the orders in both venues (2021-09-13, h. 9:00 - h. 17:00)

If on one hand, the number of buy and sell orders is balanced in the venues, on the other hand this fact is not true using a rolling window of one minute. In Figure 5.4, it is possible to observe, the percentage of the new buy orders insert in the last minute: the results for both venues are different.

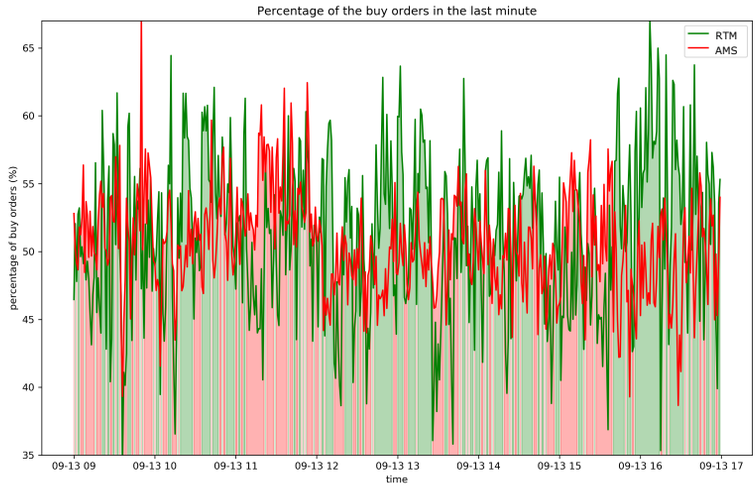


Figure 5.4: Percentage of the buy orders in the last minute in both venues (computed every minute). The area under the plot is green when the RTM percentage is greater than the AMS one and red in the other case (2021-09-13, h. 9:00 - h. 17:00).

Looking at the plot, it is possible to observe that for some time intervals of around 30 minutes, the percentage of the buy orders in one venue are continuously greater than in the other one. The mean of the difference (AMS - RTM) is $\mu = -0.71\%$, but the mean of the absolute difference is 5.74%: every minute, on average, the percentage of the new buy orders insert in the venue is 5.74% different w.r.t. the other venue's one. The variance value of the differences is $\sigma^2 = 49.59$: the values points are spread out over a wider range of values around the mean. This fact may suggest that the distribution over time of buy and sell orders at the two different venues may have differences.

In order to check the time correlation is possible to plot the ACF of the time series of the difference (Figure 5.5).

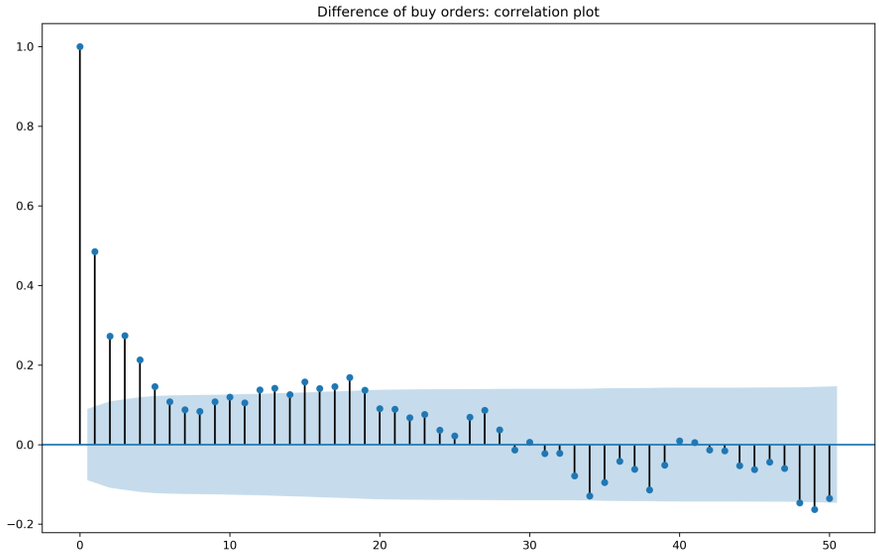
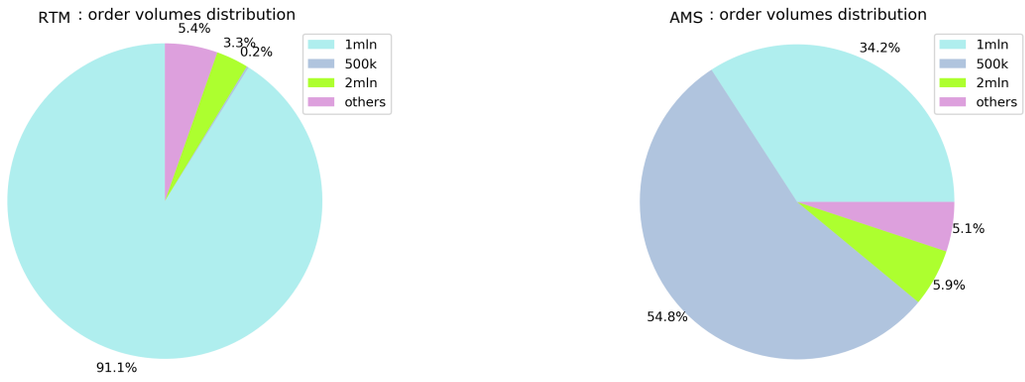


Figure 5.5: ACF plot of the difference of the percentage of buy orders in AMS and the same percentage in RTM (2021-09-13, h. 9:00 - h. 17:00). The shaded region in an ACF plot generated by Python represents the confidence interval for the correlation values, indicating the range within which correlations are likely to occur due to random fluctuations.

The ACF plot confirms that there is a strong temporal correlation: in fact, there are periods in which there is a positive difference value (lasting about 30 lags = 30 minutes) before the values become negative. This buy/sell study confirms that the quantity of buy/sell orders inserted in both venues is similar in a macro prospective (during an entire day), but it is statistically difference every minute. As Figure 5.4 and 5.5 show, there are periods in which more buy/sell orders are insert in one venues w.r.t. the other. This information could be useful as statistic for the venue selection.

5.3.3 Order sizes

Another meaningful difference is the volume of each order. In fact AMS presents approximately five times the orders in RTM, but they are generically smaller (Figure 5.6). While 91.0% of orders in RTM are 1 mln orders, AMS presents a more heterogeneous picture with 54.9% half million orders and only 34.2% of one million. In both venues more than 90% of the orders have these quantities.



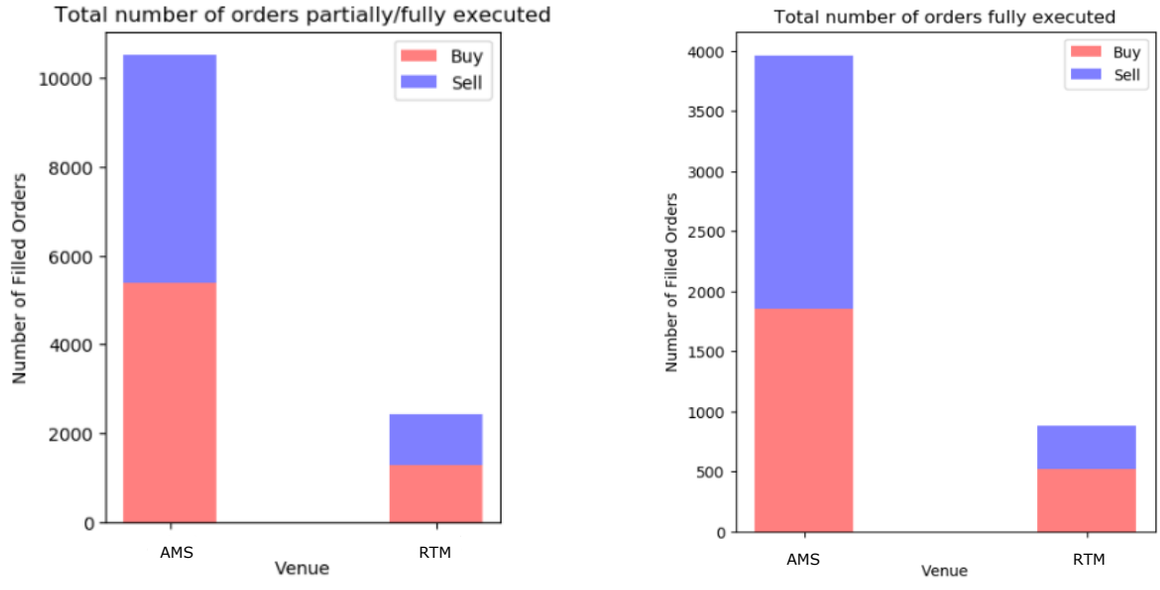
(a) RTM - EUR/USD (b) AMS - EUR/USD

Figure 5.6: Comparison between the volumes of the orders in both venues (2021-09-13, h. 9:00 - h. 17:00)

This information is very important in a case of splitting small orders between venues. It is indeed very important to consider the size of the orders on the venues in order to optimise executions. For example, using a 750k order in AMS may not be optimal: it would make more sense to choose between 500k or 1 million to ensure a more likely fit with the orders already present in the venue.

5.3.4 Traded volumes

In order to predict the fastest execution of an order between the venues, it is important to analyse the quantity of trades in the venues. As Figure 5.7 shows, from 9:00 to 17:00 AMS presents approximately five times the RTM trades. The fact occurs both with the data of executed and partially executed orders, and by looking only at the executed orders. Once again, the buy and sell proportions are respected in both venues.



(a) Total number of trades (fill + partial fill) (b) Total number of trades (complete fill)

Figure 5.7: Comparison between the total number of trades in both venues (2021-09-13, h. 9:00 - h. 17:00).

Even looking at the traded volumes the quantities in both venues have the same ratio (5x)

between AMS and RTM (Figure 5.8). The superiority of AMS over RTM can be attributed primarily to its substantial liquidity, which significantly expedites order execution. This observation underscores the crucial role of liquidity in optimizing trading performance. The abovementioned outcome has been noted also by Smith [6], who observed that opting for AMS consistently leads to better results [...].

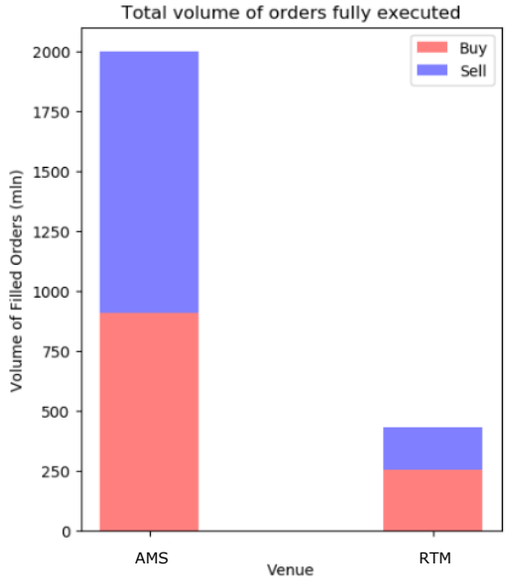
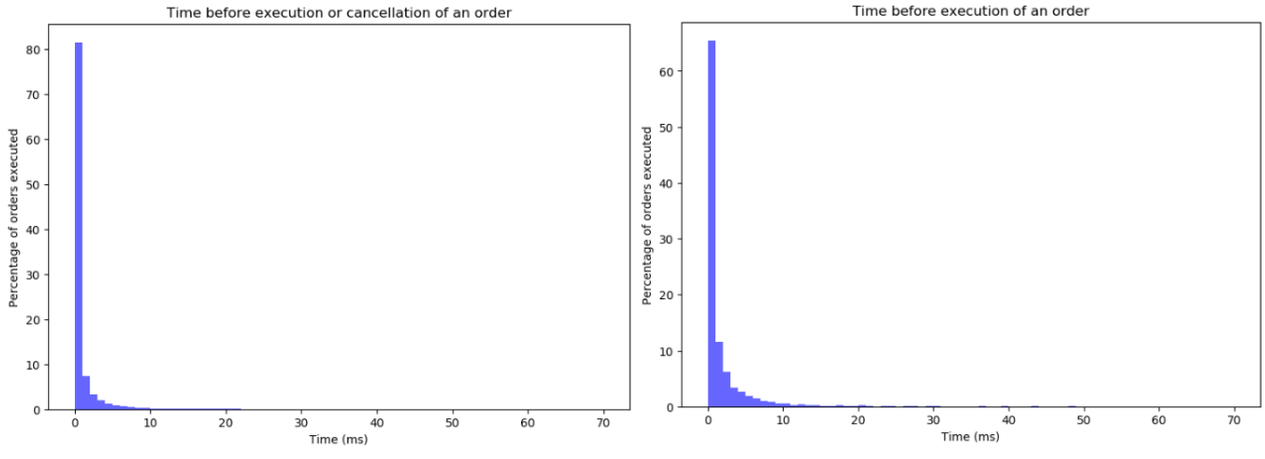


Figure 5.8: Traded volumes of fully executed orders in both venues (2021-09-13, h. 9:00 - h. 17:00)

One of the challenges of this research is therefore to detect when the RTM venue could offer faster executions than AMS, even if it has a way less liquidity.

5.3.5 Elapsed times

It is now possible to have an overview about the timing in the FX market. The amount of events that happen every fraction of a millisecond in the FX market makes it mandatory to use low-latency machines for trading. The first question to ask yourself is to read how long the orders stand at the venues on average: this quantity is defined the elapsed time of the orders. In order to study this statistic, it is possible to plot the lifetime of all orders in both venues (Figure 5.9).



(a) Elapsed time (cancellations and executions)

(b) Execution time

Figure 5.9: Life time of orders in both venues (EUR/USD, 2021-09-13, h. 9:00 - h. 17:00)

From the graph, one can immediately see that most orders are executed or cancelled after 1 ms. This is probably due to the fact that market makers have nanosecond technologies and are able to have extremely fast reaction times in cancelling an order that is no longer convenient or in executing an order placed at the venue. Furthermore, it can be deduced from these graphs that in the selected dataset 95.06% of the executed orders are executed within less than 10 ms, 97.39% within 20 ms and the 98.64% within 50 ms.

From the plot in Figure 5.9, one might argue that the distribution of the data follows a Poisson distribution. However, performing tests by fitting the distribution and conducting the chi-square test to verify if the distribution fits the data, the results contradict this hypothesis. The mean of the fitted Poisson distribution is significantly different from the data's mean and the p-value of the test is infinitesimal (much less than 0.05): there is no statistical evidence to support the claim that the data follows a Poisson distribution. In conclusion, the execution or cancellation times are similar in both venues, and there is no statistical evidence to suggest a difference in this regard between the venues.

In order to conclude the elapsed time analysis, it is possible to compute the mean and the standard deviation parameters of the distributions in each venue (Table 5.1).

Table 5.1: Mean μ and standard deviation σ of the elapsed time data for both venues (2021-09-13, h. 9:00 - h. 17:00).

	Execution or cancellation (ms)	**Execution (ms)**	**Cancellation(ms)**
AMS	$\mu = 2.059$ $\sigma = 32.300$	$\mu = 10.975$ $\sigma = 203.261$	$\mu = 2.035$ $\sigma = 30.605$
RTM	$\mu = 2.223$ $\sigma = 35.781$	$\mu = 11.372$ $\sigma = 64.907$	$\mu = 2.194$ $\sigma = 35.649$

If, on one hand, the means are not significantly different between the two venues, on the other hand a huge difference could be noticed comparing the standard deviation of the execution times. In fact, it is 64.907 for RTM and 203.261 for AMS. The reason why this happens is the presence of some extreme values in the AMS dataset which can increase a lot the variance. To avoid the extreme data, which could be due to errors in the data storage method, Table 5.2 shows the results considering just 99.9% of the data in each series.

Table 5.2: Mean μ and standard deviation σ of the elapsed time data for both venues considering data smaller than the 99.9% quantile (2021-09-13, h. 9:00 - h. 17:00).

	Execution or cancellation (ms)	**Execution (ms)**	**Cancellation(ms)**
AMS	$\mu = 1.426$ $\sigma = 9.772$	$\mu = 5.936$ $\sigma = 57.582$	$\mu = 1.419$ $\sigma = 9.703$
RTM	$\mu = 1.477$ $\sigma = 9.960$	$\mu = 11.372$ $\sigma = 64.907$	$\mu = 1.455$ $\sigma = 9.769$

Also in this case the first and the last columns do not show remarkable differences between the venues. On the other hand, the executed orders have a huge difference in the execution time. In fact, in 0.1% of removed data there were some extreme data for AMS that could shift a lot the statistics: the mean now is 5.963, 45.95% less than in Table 5.1. Moreover, the fact that RTM dataset is smaller (Figure 5.7) means that the statistics do not change even removing some extreme values.

5.4 Summary of the orders analysis

From this analysis it is possible to conclude that, excluding extreme cases, the orders in AMS are executed in less time with respect to RTM. To recap:

- The number of orders inserted in each venues is completely different: AMS has about five times the number of RTM orders. Considering also the volumes inserted in the venues in the analysed interval, it is possible to conclude that AMS is a more liquid venue than RTM.
- In general, there are no differences between buy and sell, but by analysing shorter intervals it is possible to spot differences. This could be due to noise, but could be useful when selecting venues.
- Just as the number of orders posted at venues is different, the number of trades is also different. The volumes traded show that in AMS on the selected day 4 times the volumes of RTM were traded.
- The sizes of the orders in the venue are different. In the selected day, 91.1% of the orders posted in RTM were 1 mln orders, while in AMS the majority of orders are split between 500k (54.8%) and 1 mln (34.2%).
- Orders are usually executed within less than 10 ms by 97.39% in both venues. This study will use 50ms (it is a common choice for trading algorithm). Despite this, the sample analysed shows that a shorter time might be optimal [...].

After this overview about orders, volumes, sides and timing, it is possible to analyze the some of the features defined in Chapter 3 with the aim to spot some useful information, trends or patterns that could be useful for the venue selection problem.

Chapter 6

Features comparison

This chapter continues the data analysis with the statistics introduced in Chapter 3, concluding it and explaining the problem with the current approach to venue selection. It is divided into three main sections: the first section analyzes the selected fixed time statistics while the second one the selected rolling statistics. The last section explains the limitations of the available data for the venue selection problem and proposes a backtesting data generation strategy.

6.1 Fixed time statistics

The fixed time statistics are metrics that depend only on the state of the order book at a specific time. The analysis of these statistics begins with the study of the mid price.

6.1.1 Mid price analysis

The first statistics studied is the mid price. The behaviour of this feature is expected to be extremely similar in both venues for the absence of high latency arbitrages. In fact, the mid price follows exactly the same trend in both the two venues. Figure 6.1 shows the mid prices computed every second (2021-09-07, h. 8:00 - h. 18:00).

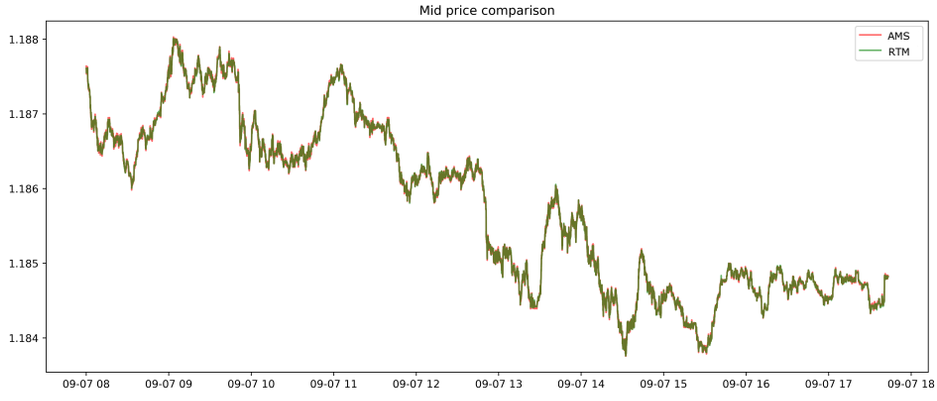


Figure 6.1: Comparison between the mid related features in both venues (2021-09-07, h. 8:00 - h. 18:00).

The pattern followed is exactly the same and no difference can be found: as also Smith [6] noted, the mid price is not a good statistic for the venue selection. In fact, the best bid and best ask follow the same patterns, but the spread could be useful for comparing venues.

6.1.2 Spread analysis

The next features to study is the spread (7) in both venues: spread analysis was conducted by analysing the intraday distributions. Five days were analysed from 8:00:00 to 18:59:59 checking the spread values every second. The mean of the spread values computed every second is reported in Table 6.1.

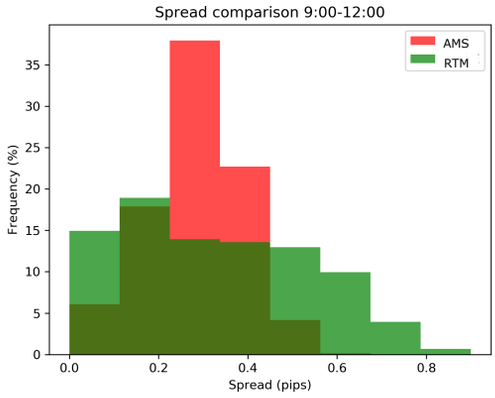
Table 6.1: Mean of the spread values in both venues (in pips). The values are obtained using ten different days and column $\Delta\bar{S}$ shows the difference.

Hour	$\bar{S}^{\mathbf{AMS}}$	$\bar{S}^{\mathbf{RTM}}$	$\boldsymbol{\Delta\bar{S}}$
8:00-9:00	0.33	0.33	0.00
9:00-10:00	0.29	0.36	-0.07
10:00-11:00	0.28	0.34	-0.06
11:00-12:00	0.31	0.34	-0.03
12:00-13:00	0.34	0.34	0.00
13:00-14:00	0.33	0.32	0.01
14:00-15:00	0.36	0.31	0.05
15:00-16:00	0.41	0.31	0.10
16:00-17:00	0.41	0.32	0.09
17:00-18:00	0.40	0.30	0.10
18:00-19:00	0.38	0.29	0.09

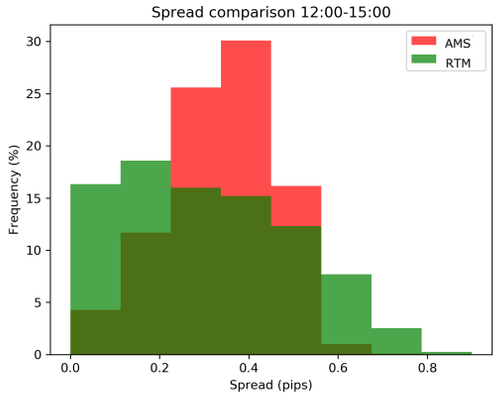
Taking these values, it is evident that during the day the spread values between the venues change and an intraday seasonality is present. Three daily periods can be identified within the day: the morning ($T = 0$, 8:00-12:00) in which on average the spread in AMS is slightly smaller than in RTM, the early afternoon ($T = 1$, 12:00-15:00) in which RTM has a slightly smaller spread than AMS and the late afternoon ($T = 2$, 15:00-19:00) in which delta values even reach -0.10. Note how these differences could play a very important role in venue selection: ALGO does not currently change its strategies depending on the time of day and this could be an important factor.

This intraday seasonality is due to the fact that in the morning the market is only open in Europe and the majority of traders are European. Then, in the early afternoon, the American market opens and consequently the number of American traders also increases. Towards the end of the day, the majority of traders are American due to time constraints. Being AMS based in London and RTM in Chicago (although they also have offices in Europe), it is possible to think that the geolocation of venues may have influenced the country from which traders trade in it. From the data one would think that in AMS there are more European traders placing volumes (in the morning the spread is smaller) instead in RTM more Americans (in the evening there seems to be a much smaller spread). This is just a hypothesis at the moment and will be confirmed by subsequent data.

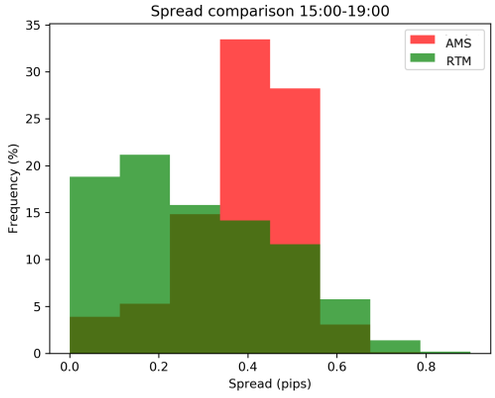
To better analyse the spread, one must also study its histograms to see its distribution and not just the average. Figure 6.2 shows the spread distributions during the three different daily periods $T = 0, 1, 2$.



(a) $T = 0$



(b) $T = 1$



(c) $T = 2$

Figure 6.2: Histogram of the spread distributions (in pips) during the different daily time periods. The shown data are obtained analysing the spread computed every second of five random days.

AMS has a completely different distribution with respect to RTM. The spread in AMS is in fact concentrated around values of 0.3 ($T = 0, 1$) and 0.4 ($T = 1, 2$) with an absence of tails. On the other hand, RTM has a flat distribution with heavy tails. The financial reason for this could again be related to the volume of the two venues: AMS has more volume and it might be more difficult for traders to change the spread much with very aggressive orders. On the other hand, RTM, having less volume, might be more prone to volatile spreads.

Moreover, the increments time series of the spread values for both venues are stationary (p-value of the ADFuller test equal to zero): it is possible to check the correlation between the time series. For $t = 0, 1$, the increments between the spread values in both venues have a correlation of 0.05, while in the afternoon 0.08. These small values may suggest that the spread (and thus the best bid and best ask) of venues may be skewed. It might be possible for a model to spot differences between venues based on these time series.

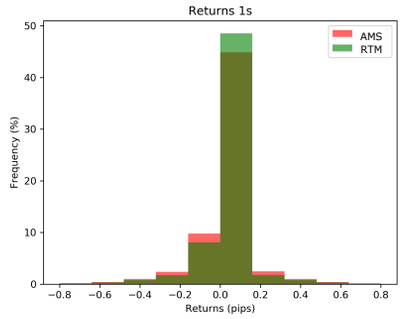
6.2 Rolling features analysis

After the selected fixed time order book statistics, it is now possible to analyze the rolling ones. The main statistics selected are the rolling returns and the rolling volatility.

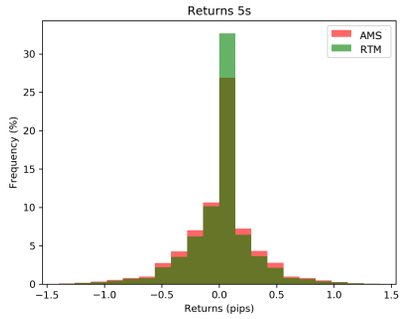
6.2.1 Returns analysis

The first rolling features analysed are the returns (not computed with the logarithm in this case, but just with the difference). Although, based on mid-price values, they should not be very useful

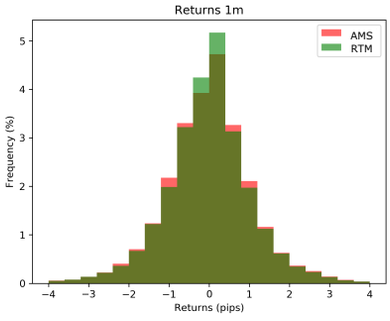
in venue selection, it is very useful to analyse them to get an idea of how the price moves in venues. The rolling returns features change substantially depending on the selected rolling window. Figure 6.3 shows the histograms of the returns with four different time intervals: 1s, 5s, 1m, 5m. The selected dataset is related to the 30/06/2021 from h. 9:00 to h. 19:00.



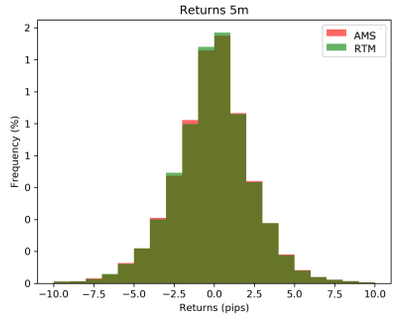
(a) rolling returns $1s$



(b) rolling returns $5s$



(c) rolling returns 1 min



(d) rolling returns 5 min

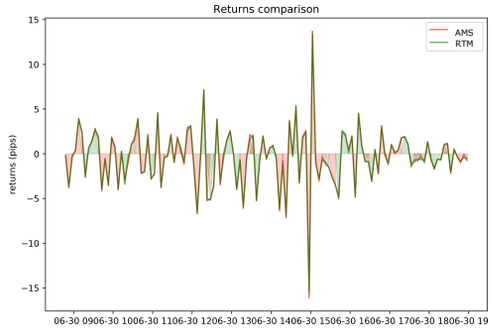
Figure 6.3: Returns comparison with different time intervals.

If, as expected, the differences between the histograms of the venues are not remarkable, the returns have difference wider shapes for every selected rolling time window. Figure 6.3a and Figure 6.3b show how the strong majority of the returns for these intervals are equal to zero. In the five days analysed (but in general in the EURUSD market), it happens often that the mid price remains stable within an interval of 1s or 5s. Figure 6.3c and Figure 6.3d show histograms centered in zero, but with infrequent null returns (around 3% for 1m and 5% for 5m). A Shapiro test was performed on the data, but for all the time series analysed, it has reported a null p-value: data are not Gaussian.

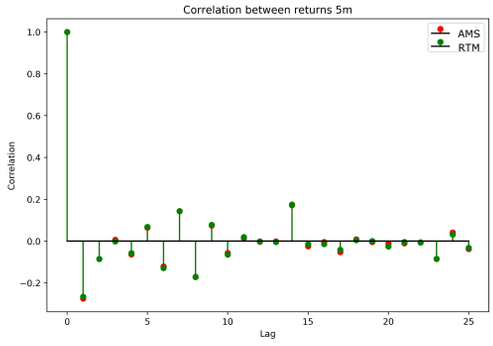
Time dependence

Being histograms, Figure 6.3 does not take into account the time correlation between the features. It is indeed possible to plot the time series of the selected returns trying to detect some trends and some differences between the venues. Again, one could expect no differences in the larger time intervals due to arbitrage absence principle. Analysing stationary time series, it is very important to look at the correlation between different lags with an ACF plot.

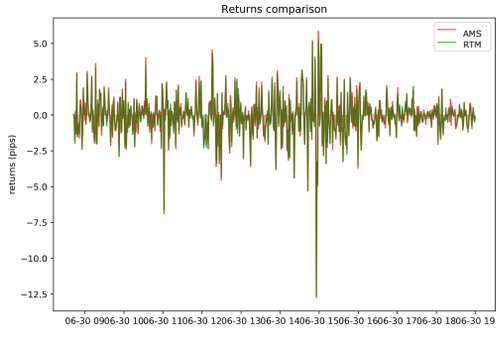
Let's start analysing the returns time series with the larger time windows and their ACF plots (5 min and 1 min, 30/06/2021 from 9:00 to 19:00) in Figure 6.4.



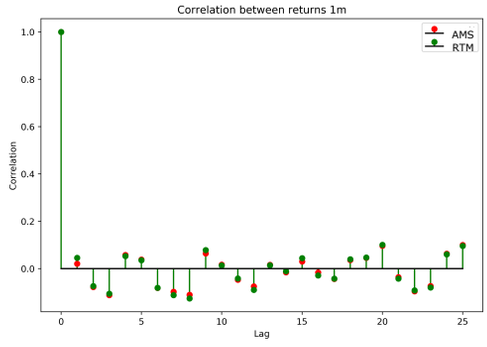
(a) Time series of 5 min rolling returns computed every 5 minutes.



(b) ACF rolling returns 5 min



(c) Time series of 1 min rolling returns computed every 1 minutes.

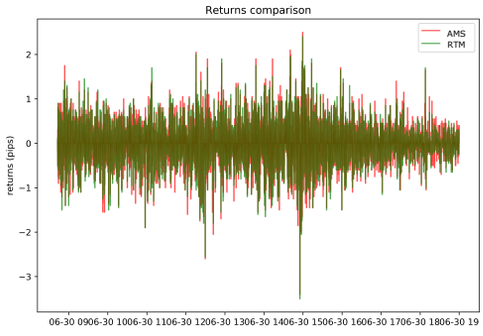


(d) ACF rolling returns 1 min

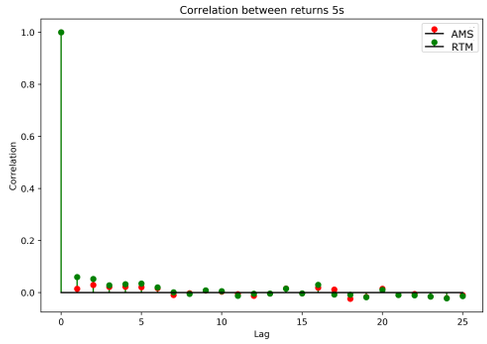
Figure 6.4: Time series and autocorrelation plot comparison between returns with relative large time intervals (5 minutes and 1 minute).

Once again, the venues show extremely similar trends and this is confirmed by the ACF plots. It is very interesting to point out the strong correlation present in the returns with relatively large time intervals: the trend is in fact not random, but seems to oscillate around zero with a trend. This fact is confirmed by ACF plots showing first negative and then positive correlations with past lags. While it is true that there does not seem to be any particular differences between the two venues, this fact is significant enough to understand the general market behaviour. For example, looking at Figure 6.9c one can deduce that it is likely that if I have had a positive return in the last five minutes, I will have a negative return in the next 5/10 minutes. The same phenomenon, although less pronounced and more random, is shown in Figure 6.9d.

When looking at plots with smaller intervals, the autocorrelation decreases more and more, to the point where it almost looks like noise. Looking at Figure 6.5, this phenomena is clearly observable. The continuous inverting behaviour of the returns is maintained, but it is much attenuated (in both venues), and even looking at the plot of the time series it is much more complicated to read. The peaks present are alternately red for AMS and green for RTM, without one venue showing a particularly different behaviour from the other.



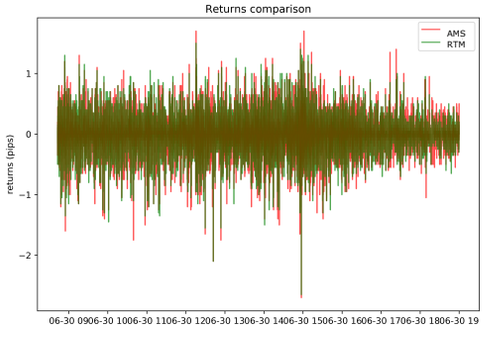
(a) Time series rolling returns 5s



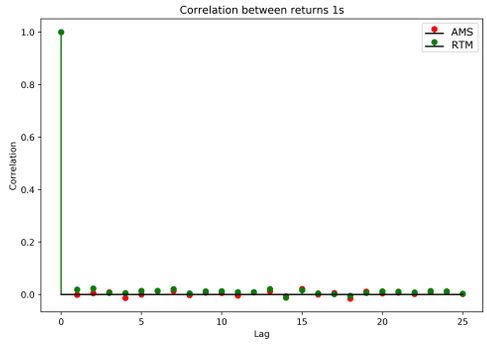
(b) ACF rolling returns 5s

Figure 6.5: Time series and autocorrelation plot comparison between returns with 5 seconds interval.

The analysis of these plots is concluded with the graphs of the 1-second returns. Again, the autocorrelation is minimal, although still present. There do not appear to be any particular differences between the two venues 6.6.



(a) Time series rolling returns 1s



(b) ACF rolling returns 1s

Figure 6.6: Time series and autocorrelation plot comparison between returns with 1 seconds interval.

It is evident from the graphs that the smaller the window of time, the more there is no correlation but just noise between the trend of returns. To confirm this idea and conclude the return analysis, the covariance matrix is shown in Table 6.2. Note that the only time series that show stationarity are the $1s$ and $5s$ returns. The other time series clearly show some patterns.

Table 6.2: Correlation between the different returns time series considered.

	r_{1s}^{AMS}	r_{1s}^{RTM}	r_{5s}^{AMS}	r_{5s}^{RTM}	r_{1m}^{AMS}	r_{1m}^{RTM}	r_{5m}^{AMS}	r_{5m}^{RTM}
r_{1s}^{AMS}	1	0.8	0.44	0.36	0.13	0.1	0.06	0.04
r_{1s}^{RTM}	0.8	1	0.43	0.46	0.14	0.13	0.06	0.06
r_{5s}^{AMS}	0.44	0.43	1	0.9	0.29	0.25	0.13	0.11
r_{5s}^{RTM}	0.36	0.46	0.9	1	0.32	0.3	0.14	0.13
r_{1m}^{AMS}	0.13	0.14	0.29	0.32	1	0.98	0.45	0.44
r_{1m}^{RTM}	0.1	0.13	0.25	0.3	0.98	1	0.46	0.45
r_{5m}^{AMS}	0.06	0.06	0.13	0.14	0.45	0.46	1	1
r_{5m}^{RTM}	0.04	0.06	0.11	0.13	0.44	0.45	1	1

The matrix shows an interesting fact: while once again there is a strong correlation between the variables with a large time window, the correlation is gradually attenuated as these windows become smaller. This fact could suggest that useful differences in venue selection could be contained in features with a very small time interval (in the order of magnitude of the ms). If on one hand, these features could be just random noise, on the other hand, due to the absence of arbitrage, they could play a key role in the venue selection problem. If something is happening in one venue, it will also happen in the other with a very small delay. Being able to detect these changes could lead to the choice of one venue over the other depending on one's buy or sell order.

6.2.2 Volatility

Financial volatility refers to the degree of variation and uncertainty in the prices or returns of financial assets, reflecting the level of risk and market fluctuations in the financial markets. Being one of the most important financial statistics it was considered important to analyse it. Like in the spread case, the first analysis is related to the hourly mean of our five days dataset. Volatility computed in this section is the standard deviation of the $50ms$ returns every 1 second. The first selected rolling time window is 5 minutes: the first study is related to large rolling time window. Table 6.3 shows the hourly means of the 5min volatility in both venues.

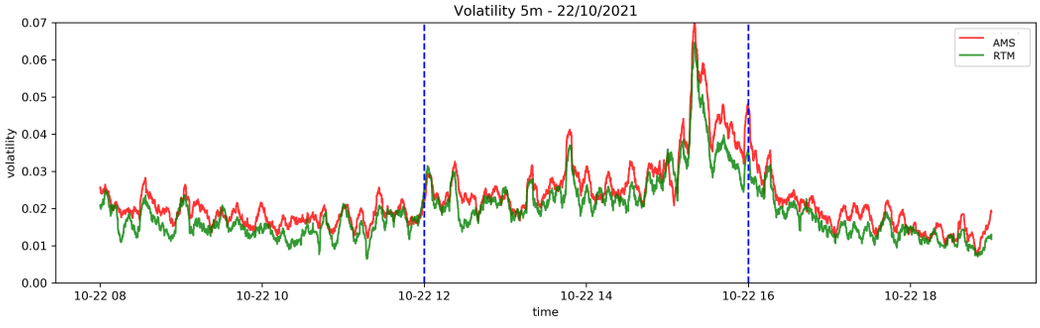
Table 6.3: Mean of the 5min volatility values in both venues (multiplied by 10^4).

Hour	$\bar{\sigma}^{\mathbf{AMS}}$	$\bar{\sigma}^{\mathbf{RTM}}$
8:00-9:00	0.023	0.020
9:00-10:00	0.022	0.020
10:00-11:00	0.021	0.018
11:00-12:00	0.020	0.018
12:00-13:00	0.030	0.026
13:00-14:00	0.029	0.025
14:00-15:00	0.030	0.026
15:00-16:00	0.031	0.027
16:00-17:00	0.020	0.018
17:00-18:00	0.016	0.013
18:00-19:00	0.014	0.012

As in the case of the spread, it is clear that it is possible to divide the hourly range into three groups. The morning hours, when the European market is open, are characterised by volatility around 0.02 in both venues. In the early afternoon, with the US market also open, the values increase a lot to above 0.03 in AMS. In this case the data also show high volatility from 15:00 to 15:59 (in the case of the spread this time interval was assigned to $T = 2$). In the late afternoon, approaching the close in Europe, the market on selected days appears more stable and less volatile.

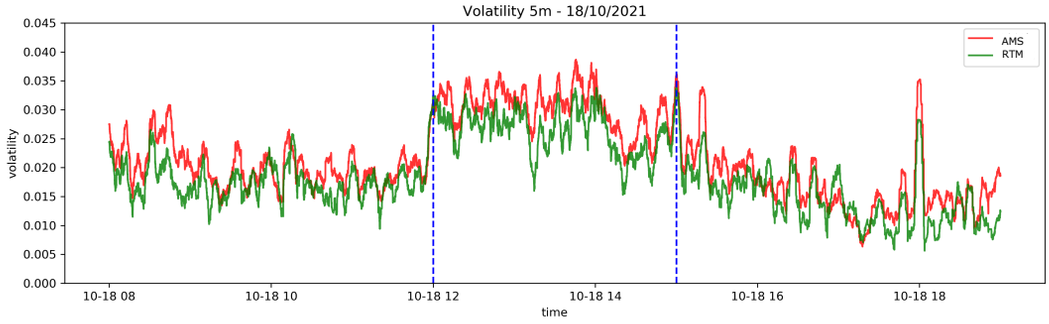
It is now possible to view two days of the five in the analysed dataset to study whether 15:00 belongs to the second or third group. 22/10/2021 shows great volatility between 15:00 and 16:00, with peaks as high as 0.07 (see Figure 6.7).

Figure 6.7: Time series of the 5 minutes volatility, 22/10/2021.



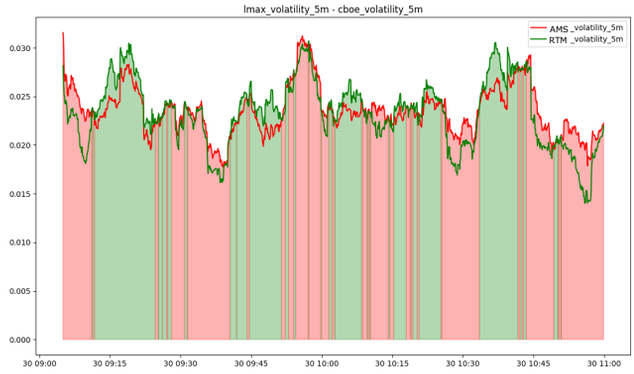
As the EURUSD market is continuously influenced by external factors from all over the world, it is absolutely logical that the price can have volatility spikes at any time (especially if both the European and American markets are open). An example of a low volatility day is 18/10 on which instead the division implemented for the spread could be perfectly functional. Volatility on this day remains below 0.04 throughout the day (Figure 6.8).

Figure 6.8: Time series and autocorrelation plot comparison between volatility with relative large time intervals (5 minutes and 1 minute).

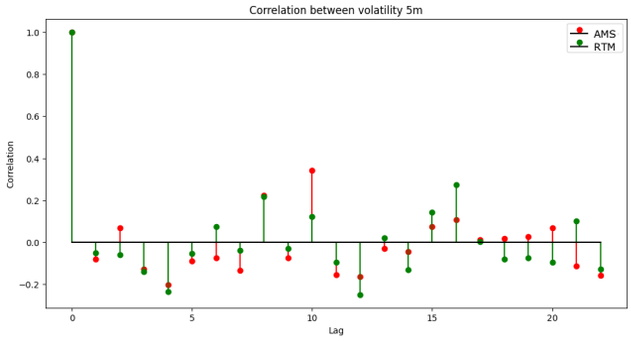


Regardless of the precise division, it is again important to note that days show intraday patterns with different volatility. Just as in the case of the spread, the fact that the stock markets are or are not open greatly influences the volatility in both venues.

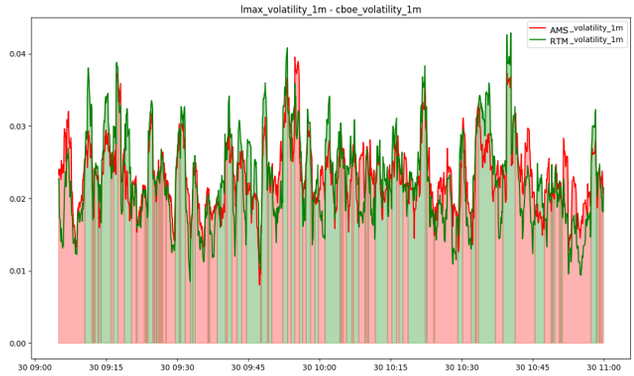
From the figures above, it can be seen that volatilities seem to follow similar, but not identical paths as in the case of mid-price or returns with large time intervals. The choice to use 50ms returns tries indeed to detect differences between the venues. As opposed to returns, volatility of the 50ms returns seems to be different, even in the larger time intervals. Looking at the 5 minutes and 1 minute time is possible to note different trends. Figure 6.9 shows data of 30/06/2021 from 9:00 to 11:00, but the result are generalisable.



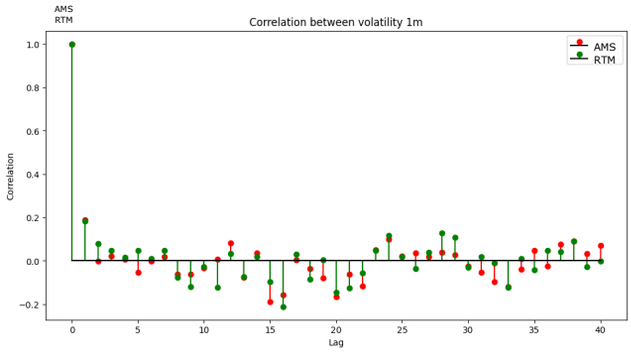
(a) Time series of 5 min rolling volatility computed every 5 minutes.



(b) ACF rolling volatility 5 min



(c) Time series of 1 min rolling volatility computed every 1 minutes.



(d) ACF rolling volatility 1 min

Figure 6.9: Time series and autocorrelation plot comparison between volatility with relative large time intervals (30/06/2021 9:00-11:00).

Although following the same mid price (as shown in the previous sections), the volatilities of the 50-ms returns are different. The ACF plots show that the correlations between the two different venues are also very different over time. There is clear statistical evidence that the two venues have different ways of following the mid price.

The hypothesis could now arise that at micro upper trends and/or lower trends a given venue has a different volatility from the other and that this phenomenon is also visible at relatively large time windows. In other words, it would be ideal to understand whether higher volatility values in AMS/RTM correspond to positive or negative price movements. Looking at Figure 6.10, it is impossible to identify a pattern in the trends: it is not true that at each upper/lower trend one venue is more volatile than another using this relatively large time interval (1 minute).

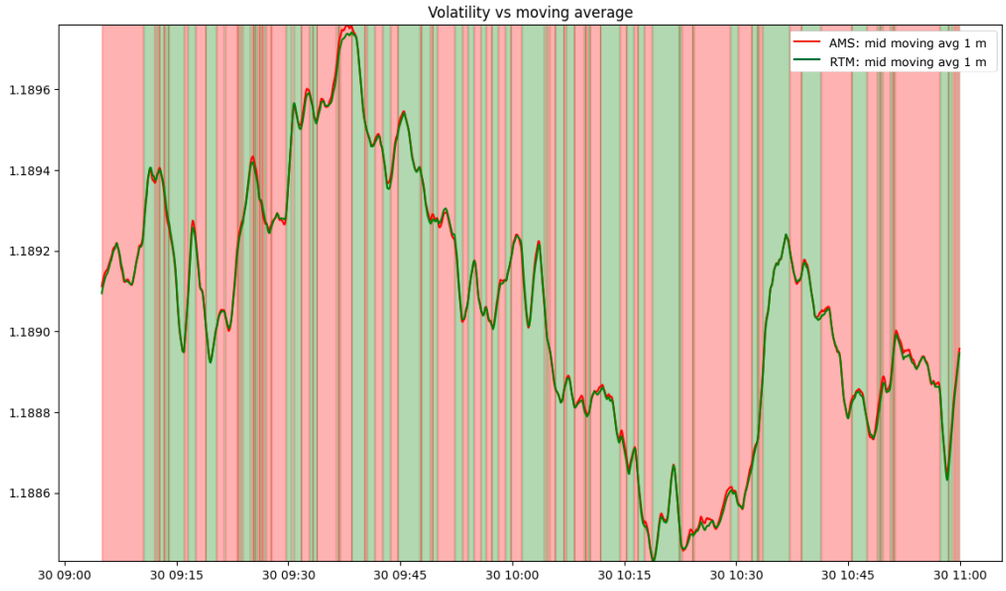


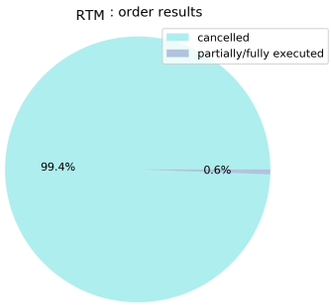
Figure 6.10: Comparison between the volatility values and the mid. The plotted lines are the 1minute timeseries in both venues and the background shows when the volatility is higher in RTM (green) or AMS (red).

From this volatility and returns analysis, it is clear that features based on macro price trends can detect macro trends present in both venues. Moving average and mid values are equal due to the absence of arbitrage, and thus the returns, when studied with time intervals in the order of magnitude of the second. The strong correlations between return data might be interesting for a macro trend prediction problem, but it could be too similar for a venue selection problem. On the other hand, features which use very small time interval could be not useful due to excessive noise. The primary objective of this study is to strike a balance between incorporating features that employ larger time intervals, capable of detecting macro trends that exhibit similarities across different venues, and features that use smaller time intervals, while ensuring that the generated signals are not too noisy.

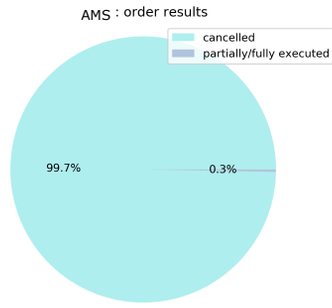
6.3 Comparison problems

This subsection explains the reasons behind the new approach proposed in this thesis to the problem of venue selection. It is possible to identify three main problems that data has for the venue comparison: the strong imbalance between executions and cancellations, the strong imbalance between the venues number of orders and the absence of equal orders in both venues. The aim of this subsection is try to explain these three problems and propose a solution.

Looking at the data, the majority of the orders is cancelled and not executed: for both venues the number of cancellations is extremely high (Figure 6.11): for RTM is 99.4%, for AMS 99.7%.



(a) RTM - EUR/USD



(b) AMS - EUR/USD

Figure 6.11: Comparison between the orders in both venues (2021-09-13, h. 9:00 - h. 17:00)

The portion of the executed orders is extremely small compared to the cancellations: the available dataset is strongly imbalanced. If one wants to calculate the probability of execution given an order by trying to train a model using historical orders as Smith did [6], problems may arise. In fact, this characteristic makes the model biased: in fact, it will tend to predict only non-execution of an order. To avoid this problem Smith, in her research, considered just the executed orders: all the cancellations are not considered during the training of the models. In doing so, Smith encountered a second major problem: as shown Figure 5.7, AMS executions are much more present with respect to RTM ones. In fact even in this case the dataset is imbalanced. In order to train the Random Forest model proposed, Smith proposes to oversample RTM data and undersample AMS one. The obtained result, as already mentioned, were not significant for the choice of venues.

However, the biggest problem for venue comparisons is the absence of comparable data. In a dynamic system like the limit order book, which changes every nanosecond, time plays a very important role. In all present data, there are no two identical orders placed in both venues at the same time. To compare venues in the most unbiased way possible, different data should be studied in which the orders sent to the venues are exactly the same. With this idea in mind, this study proposes an innovative technique for generating synthetic data for the venue selection using historical data.

Chapter 7

Synthetic data generation

In order to compare the venues without any kind of bias, a synthetic data generation approach has been used in this research. The aim of this subsection is to explain the main idea between the synthetic data generation, give an overview of the Python code that simulates the orderbook dynamics and propose different approaches used to generate new data.

7.1 Data generation: the basic idea

Synthetic data generation involves creating artificial data that mimics real-world data. In this case, the aim is to simulate orderbook dynamics to generate new data that can be used to compare venues without any bias. To achieve this goal the same synthetic order is sent to both venues with exactly the same characteristics and at the same time (Figure 7.1). The execution/not execution of the order is determined by the historical data of each venues and the amount of time preceding the execution is saved.

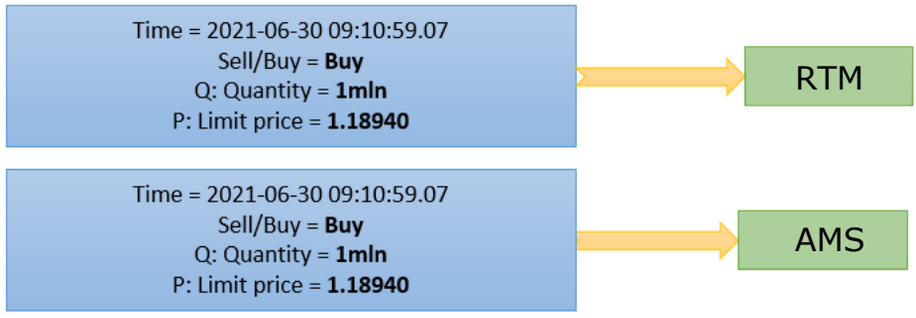


Figure 7.1: Synthetic data generation: two identical orders are sent to both venues at the same time.

In this way, the historical orders of each venues determine the execution of the mock order inserted, and the venues are judged without any kind of bias. In fact, the bias related to cancellations is avoided because the experiment never ends with a cancellation, and the bias related to the number of daily data from each venue is avoided because the experiments are run in parallel on AMS and RTM. Add to this the fact that the orders have exactly the same characteristics (direction, quantity and price) and the judgement on venues is totally impartial. The key point now is define the characteristic of the generic mock order insert in the LOB of both venues:

- **Timestamp and machine latency**: the timestamp has to be the same for both venues. At a generic time t_0 the order is sent to AMS and RTM and it will be read by the venues with a latency of 2 ms. The latency value is chosen by discussing with MN quants.

- **Quantity**: as shown in Figure 5.6, a good quantity could be 1 mln: this value and it is common in both venues (especially RTM) [...]. Another good point for this choice is that the quantity is not too large: the aim of the simulation is to be as realistic as possible and orders that are too large could alter the flow of the simulated LOB. In case of partial execution the order remains in the venues with the remaining quantity until all the quantity will be executed.

- **Direction**: The simulations are run with both buy and sell orders for the same time interval with two different simulations. In this way, two datasets are generated (buy and sell) and the data obtained are completely unbiased.

- **Price**: the choice of order price may vary depending on the type of experiment. It is obviously important that the price entered is the same in all venues. To benefit the venue with the most competitive best ask/best bid the proposal is to use min and max operation to give more important at the most competitive venue. In the subsection 7.3 the proposed strategies will be explained. Every strategy has a different price selection.

7.2 Data generation in practise: Python code explanation

In order to generate synthetic data, a public Python code with a complete LOB matching engine was found on GitHub [30]. The code has an MIT license which makes it public and available to everyone. It can reconstruct an orderbook and modify it step-by-step, every time a new order is added/cancelled/modified. The code is a matching engine because when there is a match between two orders the trade is automatically generated and the limit order book is dynamically updated. However, the original code required extensive modification to align it with the goals of this research. The main new parts coded were: the integration of the code with AWS to access venue data and the development of a simulation module to create statistics, track progress, and save experiment results.

To provide a general overview of the code base, we will introduce the two main classes: `orderbook` and `gateway`. The `orderbook` class represents a snapshot of the orderbook at a particular timestamp, and provides methods to access various pieces of information such as the mid-price, best ask, best bid, and volume distribution. On the other hand, the `gateway` class serves as the interface between the historical data and `orderbook`. This class loads all the historical trading data and processes data one-by-one modifying the orderbook class. Figure 7.2 illustrates this process for AMS (for RTM is exaclty the same with a different dataset). It's important to note that each venue has its own dedicated `gateway` and `orderbook` classes, which are specific to that venue.

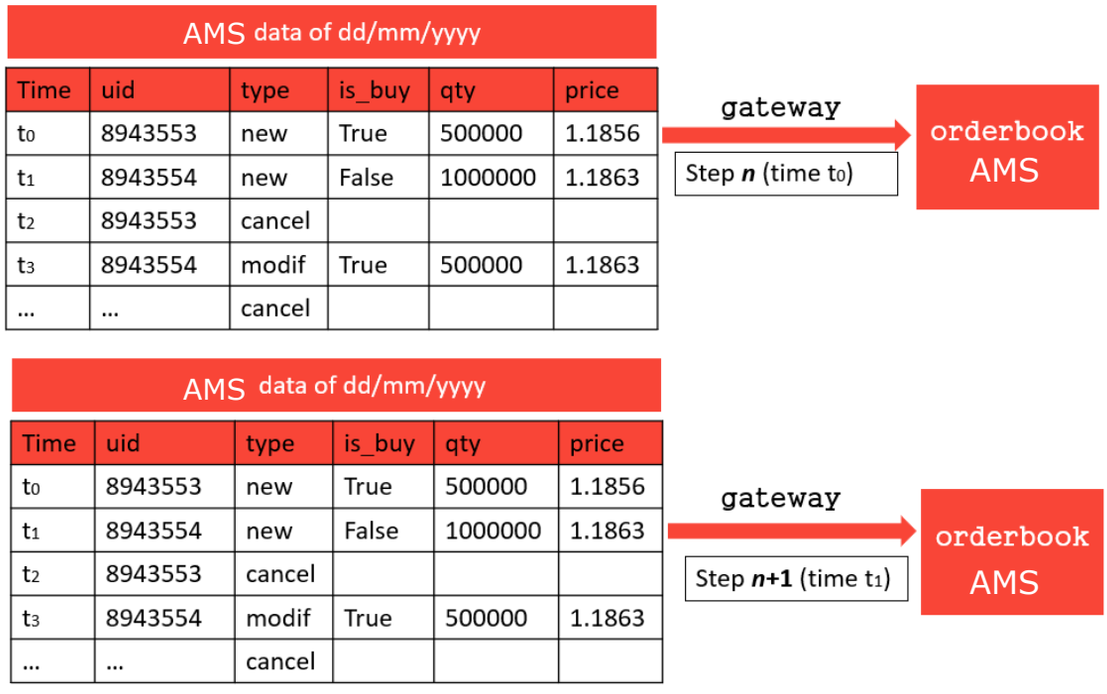


Figure 7.2: Explanation of the interaction between `gateway` and `orderbook` classes. The historical orders are processed one by one and at every timestamp it is possible to access the snapshot of the orderbook.

Considering that the average life time of each order is less than $15ms$ (Table 5.1), the choice to use one entire hour of data to load the historical orderbook before perform any strategy is accurate: it is possible to consider the constructed orderbook uagual to the historical orderbook. The next step is to understand how the code can add and process mock orders used to simulate a strategy. Note that the process explained in this section is repeated for both venues in the exactly same way. In order to clarify how a simulation can be run, it is possible to do an example. Let's assume that the orderbook is updated at time t_1 with a the historical orders. Furthermore, assume that at time $t_2 + \Delta t$ a mock order is to be entered for simulation:

- The order is read by the `gateway` class in the historical order. Note that the unique id (uid) of the order is defined negative in order to distinguish the mock orders from the historical ones. The mock order will be read by the `gateway` class and inserted into the `orderbook` class (Figure 7.3).

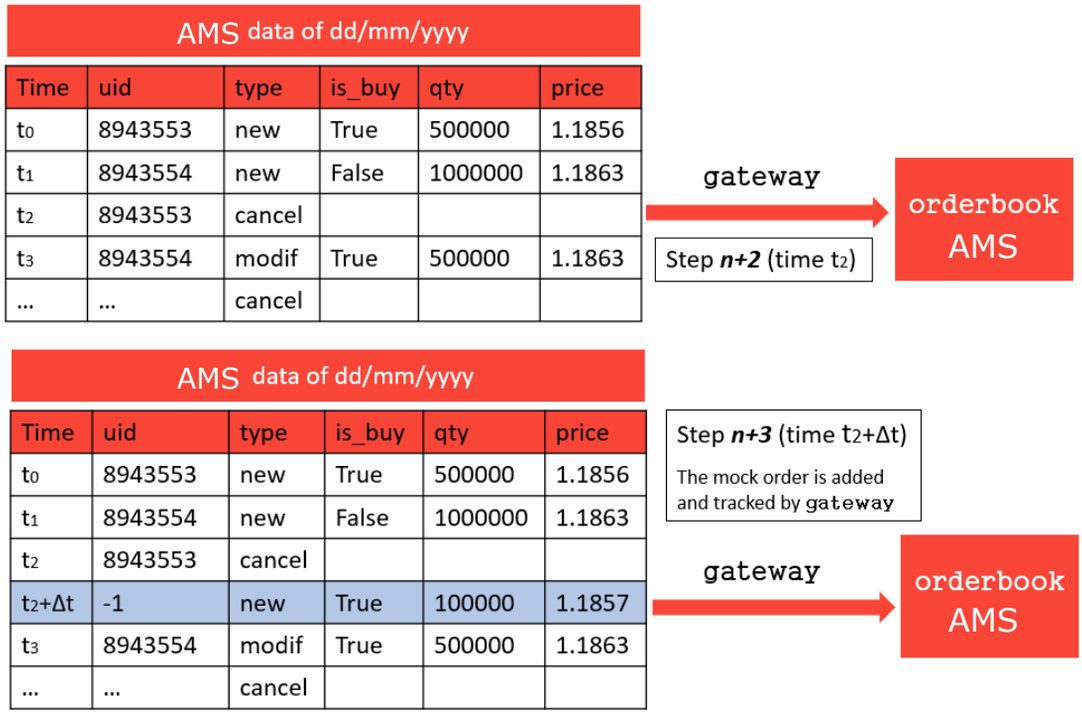


Figure 7.3: Explanation of how a single simulation begins. The mock order is inserted in the sequence of the orders with a negative uid. The `gateway` class processes it and adds it to the `orderbook` class.

- The historical order are processed one-by-one until the mock order is cancelled or it is executed. The choice of cancellation/not cancellation of the mock order depends on the chosen strategy. Let's assume that at time t_n the mock order is still in the venue (the strategy did not cancel the mock order) and a historical order generates a match (Figure 7.4). The simulation ends and all the information are saved in a csv file (order info, rolling statistics, snapshot statistics, time of execution in the venues).

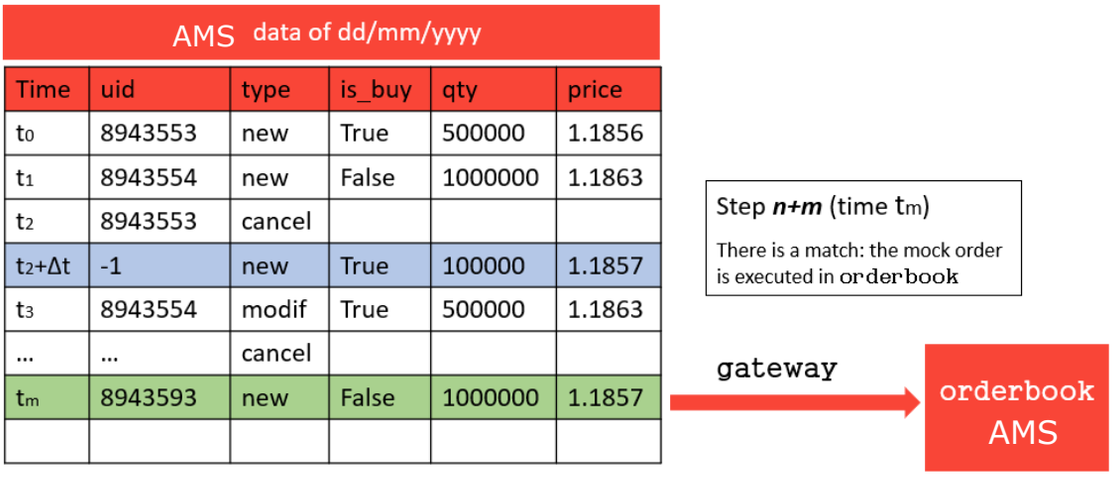


Figure 7.4: Explanation of how a single simulation ends. There is a match in the orderbook and the mock order is executed.

Note that the possible future cancellation of the historical matched order (order 8943593 in Figure 7.4) is processed by the `gateway` class but does not lead to any results. Furthermore,

it is important to highlight that the `orderbook` class provides an easy access to all the trades that involve the mock orders.

To conclude, performing the simulations and saving all the results, it is possible to build a dataset that can be used to train some statistical and Machine Learning models. In fact, the output of the simulations of both venues will be different because the historical orders and the statistics are different: the goal now is to train a model that can detect in which situation a venue could be better than the other one. Note how simulations do not claim to emulate reality exactly: by using only historical data it is not possible to claim total verisimilitude of the simulation. What a simulation aims to achieve is a clearer view of how venues work, comparing them and trying to spot the main differences. Although the simulations may not be entirely truthful, in fact, putting the focus on the sequence of historical orders in both venues could have very significant results.

7.3 Proposed strategies

In this subsection the strategies used in order to generate synthetic data. The strategy proposed depend on a ϕ parameter and they are all based on the ALGO floating strategy. The ϕ parameter defines the aggressiveness of each strategy: by backtesting different strategies with different aggressiveness, it is possible to compare locations in different situations, analysing different behaviours.

The proposed floating strategy (FLOAT) is a very similar version of the most used ALGO trading strategy. The proposed version is a simplified version of the real one discussed with the senior Quant of MN. The main reason why the real one is not shown is the fact that the parameters of the strategy are extremely confidential. Despite this, the proposed strategy is quite similar and easier to interpret and explain. It is a passive strategy that *floats* around selected price. In order to use the ALGO floating strategy one has to define a parameter ϕ, a limit time interval Δt and the mean of the mid prices in both venues at time t:

$$\bar{M}(t) = \frac{M^{\mathrm{RTM}}(t) + M^{\mathrm{AMS}}(t)}{2}$$

In the real ALGO Floating strategy, $\bar{M}(t)$ is calculated as a weighted average that assigns a weight to each venue based on the last update received by the venue. A recently arrived update is given a large weight, while an update that arrived earlier in time is given a smaller weight. The reason for this is that in reality it is not possible to access venue data at all times, but there is a latency in updates. In the proposed simulation, it is assumed that the venue updates occur at the same time and thus the weights are equivalent to each other. With this assumption, it is possible to calculate $\bar{M}(t)$ and establish the floating price:

In case of buy order: $P_{\mathrm{float}}(t, \phi) = \bar{M}(t) - \phi * 0.00001$

In case of sell order: $P_{\mathrm{float}}(t, \phi) = \bar{M}(t) + \phi * 0.00001$

The parameter ϕ or *level of passiveness* is fixed for every simulation. In the real ALGO version there is an analogous parameter and it is set manually by the MN trader at the start of execution of the parent order. With high values of ϕ the strategy favours the best price over execution speed, while with lower values the opposite occurs. Note that the price is rounded to the fifth decimal place to be an acceptable price in the FX market: this fact, that might seem insignificant, changes things a lot when the phi value is null approaching one mid price or the other.

Defined the initial price equal to the floating price, the mock order is sent to both venues. This logic is exactly the same that the ALGO one: the historical orders are processed by the code and after Δt milliseconds in each venue the order could be executed, partially executed or not executed. On the other hand, the floating price could change or it could be the same.

- If one order is fully executed the strategy ends. It is very important to note that the strategy ends even if in the other venues the order is not fully executed. In this case the new floating price is not considered.
- If there is not a full execution, the new floating price is computed. If the new price is equal to the previous one nothing happens and the strategy will wait other Δt milliseconds. If the new price is not equal, the old mock orders are cancelled and the remaining quantities in both orders are insert in corresponding venues with the same new floating prices. For instance, if in AMS the remaining quantity is 600k and in RTM it is still 1mln, the new orders have the same quantities (600k for AMS and 1mln for RTM) but with the updated floating price.

Again, the same strategy is applied to both venues and the venues with the fastest execution is preferred. Note that the strategy price is the same in both venues at every timestamp. When the strategy ends, the orderbook is re-initialized and a new floating strategy is immediately performed. Indeed, the ALGO float strategy has not a fixed duration, so there is not a fixed latency in which the strategy is performed.

7.4 Latency

This subsection will explain how the latency is managed in the simulations. With the hypothesis that all the updates from the venues occur at the same time (in reality this is not always true), there is still the reaction latency of the trading machine. In the codebase this value is defined and fixed at $2ms$ after the discussion with a MN quant. The strategies every Δt ms have to act with the venues. When a cancellation of an existing order and/or new order has to be sent, the message arrives whit τ ms of latency. To understand this important concept, it is possible to observe Figure 7.5 that describes this process.

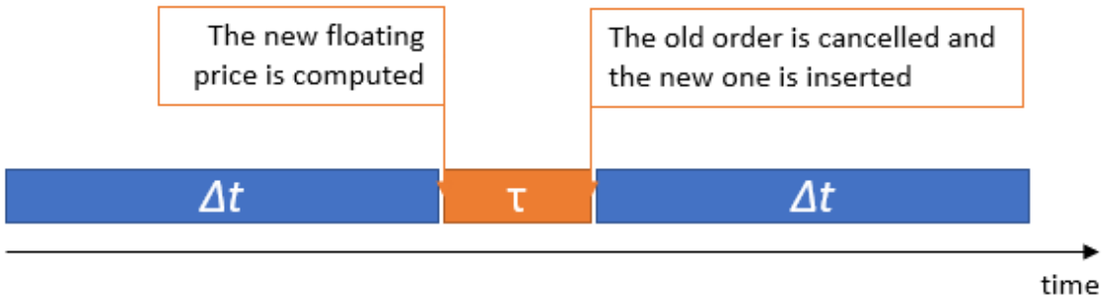


Figure 7.5: Description of the latency management in the codebase. After Δt ms the strategy computes the new price and sends updates delayed by τ ms.

Note that in the latency period a full/partial execution could occur: this case has to be considered. In reality, MN uses to modify the old orders putting a new price maintaining the remaining quantity (if there is any). The simulation emulate this fact, cancelling the old order and putting a new order with the price computed at time Δt with the remaining quantity at time $\Delta t + \tau$. If at time $\Delta t + \tau$ all the volume is already executed the simulation stops: it makes no sense to add an order with null volume.

7.5 Synthetic data generation summary

Synthetic data generation is considered the best approach for approaching the venue selection problem using an unbiased technique. By backtesting strategies in a consistent manner across both

venues, with identical parameters and timing, it becomes feasible to compare executions and find the best venue allocation.

The selected child orders size is 1 million because it is an extremely common size in both AMS and RTM. The latency is 2 ms (discussed with the MN quants) and the simulations are performed for both side buy and sell. For the price selection, a strategy similar to the ALGO FLOAT has been selected: it is the most used trading strategy by the MN traders. The only input parameter considered is $\phi \in [0, 4]$ and it describes the passiveness of the strategy. In this way, it is possible to analyse the different behaviours of the venues with different level of passiveness. Additionally, calculating a set of statistics before each simulation allows for assessing the correlation between execution outcomes and the initial statistical values.

Chapter 8

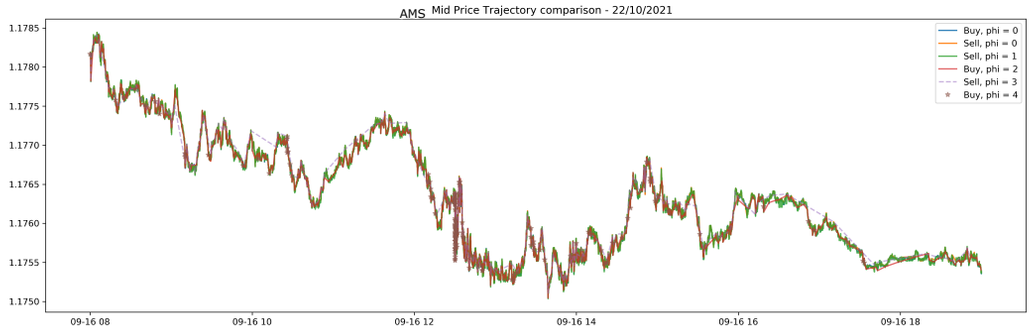
Synthetic Data Analysis: aggressive strategies

The aim of this section is to present the results of one day synthetic week data simulations for the aggressive strategies ($\phi = 0, \phi = 1$). A synthetic week is a set of one of every weekday from different weeks. Five weekdays from June to October 2021 were selected: Monday 18/10, Tuesday 7/09, Wednesday 30/06, Thursday 16/09 and Friday 22/10. All the days were studied from h. 8:00 to h. 19:00 (London time, CET+1). The decision to utilize a synthetic week instead of a regular week is motivated by an overfitting issue. The objective of the chapter is to identify general patterns in venues behavior. However, these patterns may be constrained to a particular period, and therefore the selection of a synthetic week is an attempt to limit this problem. Note that the simulations are performed in two different datasets. The first one is from 8:00 to 14:00, the second one form 14:00 to 19:00. In this way the orderbook is re-initialised at least one time with just historical data and the prices are compared.

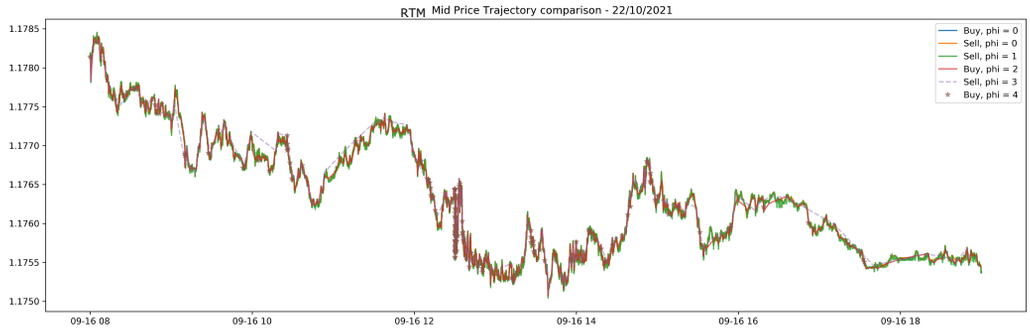
8.1 Strategy analysis rules

The first question that might arise from reading this chapter is why mock simulations are meaningful? The fact that all simulations are based only on historical data and not on real market participants can change a lot our mock scenario and modify the overall results: it is in fact possible that by continuously trying to trade aggressively on the same buy/sell side the price will be moved by some market maker to try to make a profit. If, however, the results were analysed not by considering a general overview but only by looking at individual one-million performances, then the significance of the simulations could increase. The single 1 mln buy/sell order, in fact, should in no way change the general price trend.

In order to compare the price trajectories of each orderbook, it is possible to plot the mid price in both venues for every day and for every simulation (Figure 8.1). In this way, it is possible to check that all the strategies (especially the most aggressive ones with $\phi = 0$) do not modify the mid price trajectory.



(a) AMS mid price trajectory



(b) RTM mid price trajectory

Figure 8.1: Checks of the mid price trajectories in both venues: the mid prices follow the same trajectory for all the strategies. The showed day is 22/10/2021, from 8:00 to 19:00 London time.

The plot does not show all the strategies to avoid confusion, but all the strategies have been tested and show all the same trajectory. The daily mid prices values are confirmed checking the values online. Given these results, it is possible to assume that all the simulations do not effect the trajectory of the mid prices.

Simulations are not to be interpreted as real trades, but the reading to be given is as follows: imagine the floating price as a line, and the simulation itself describes when this value is hit by the market with at least a million. The analysis of this chapter will be indeed focused on statistics computed on the single simulations, trying to find some common path or tendencies. Using the old approach described in 2.5.1 one should focus just on which venue was the fastest to execute the child order in each simulation in order to select the best venue. However, using the new parallel trading approach (see 2.5.2), the most important statistic studied is the difference between the volumes executed in each venue: in fact, the aim of this research is not select the best venue, but try to measure the performances in order to find the best order allocation. Given the i^{th} simulation, let's define the difference between the volume executed in both venues in that simulation:

$$\Delta V^i = V_{\mathrm{L}}^i - V_{\mathrm{C}}^i \tag{8.1}$$

where V_{L}^i is the volume executed in the i^{th} simulation in AMS and V_{C}^i in RTM. Given that the volumes are expressed in millions and each child order in every simulation has a quantity of 1 million, it is possible to write $V_{\mathrm{L}}^i, V_{\mathrm{C}}^i \in [0, 1]$. Consequently, $\Delta V^i \in [-1, 1]$ and some important cases be distinguished:

- If $\Delta V^i > 0$, in the i^{th} simulation, AMS achieved a complete execution, while RTM did not. The extreme case is $\Delta V^i = 1$, when RTM did not report any execution.
- If $\Delta V^i < 0$, in the i^{th} simulation, RTM achieved a complete execution, while AMS did not. The extreme case is $\Delta V^i = -1$, when AMS did not report any execution.

- If $\Delta V^i = 0$, the same $50ms$ time window a full execution (1 mln) occurred in AMS and in RTM. The i^{th} simulation has been *tied* by the venues.

Even if each day contains different characteristics, the aim of the next section is to highlight general patterns in the selected synthetic week. It has been analyzed the time distribution of the executions during a day, trying to find some meaningful common patterns. Moreover, in order to try to understand the behaviour of ΔV, correlations with selected statistics x_t will be analysed. After this, the goal is to try to find some models that, given x_t could forecast the best prediction as possible for ΔV. This models will be trained and tested using a split data in the synthetic week: the first four days will used as a training set, Friday 22/10 will be the test set. Each proposed model will be compared with two benchmarks: the prediction named ALGO which always predicts $\Delta\hat{V} = 0$ and the simple mean model that will be explained [...]. The selected reference metric is mean square error (MSE), being the most classic metric for regression problems.

Note that the results proposed in this section are extremely consistent: the models proposed can forecast ΔV values of a random day basing the forecast on just four random days in the past. The patterns analysed by the models are indeed generalisable and there is no risk of overfitting. Moreover, given the $\Delta\hat{V}$, it is possible to compute w_A using this formula:

$$w_A = \frac{1}{(2 - \Delta\hat{V})} \tag{8.2}$$

The equation does not aim to return the best value of w_A for a single two-million split: in fact, MN orders are usually much larger, which makes the choice more complex. The choice of quantity to allocate to each venue is in fact made at the beginning of the trading process, even if the quantities are large and the whole thing will take a long time. The proposed solution will be to split large orders into smaller orders and allocate in each venue multiple orders of 500k to try to speed up the process. The value of w_A represents the ratio that this research thinks is optimal.

8.2 Floating aggressive strategies with $\phi = 0$

Let's start the analysis with the aggressive strategies: in this section the floating strategy results with $\phi = 0$ are analysed. It is important to notice that this selection of ϕ makes the strategy the most aggressive of the used floating strategies. The buy and sell prices are both set to the average mid price of the venues. With this strategy, one can expect a large number of executions and a low average execution time. Running the analysis, a large number of executions is indeed verified: in a synthetic week, 608894 for the buy simulations and 621389 for the sell ones. It has been noted that in periods with very small spreads, the orders are market orders executed on. The interpretation of this result is that the line of the average of the mid prices (rounded the fifth decimal place) was hit more than 600k time in a synthetic week. 43168 for the sell ones.

8.2.1 ΔV results

With the most aggressive strategies, AMS outperforms RTM. On average for the buy simulations $\Delta V = 0.473$ and for the sell ones $\Delta V = 0.464$. Figure 8.2 shows the distribution of the ΔV: about 35% of the simulations are ended with $\Delta V = 1$.

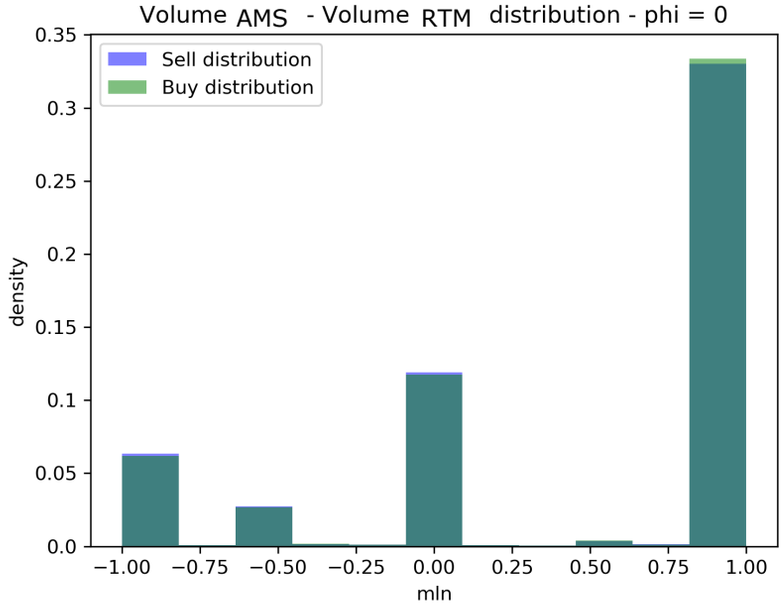
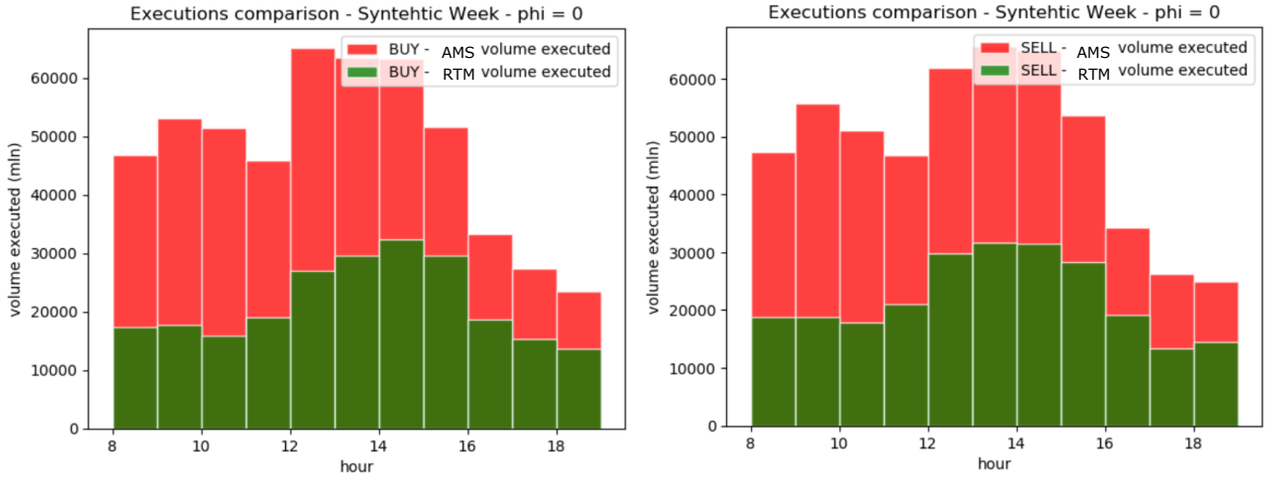


Figure 8.2: Difference between volumes traded in AMS and RTM for $\phi = 0$.

The $\phi = 0$ strategies could not be used by a real market participant for the entire duration of one day for the huge number of executions, but can help the analysis on the venue participants and their trades. The reason why AMS is preferable with respect to RTM is most likely the fact that more traders trade using the venue. Being the floating price with $\phi = 0$ a very aggressive price the most liquid venue is the best. Note that there is no remarkable difference between buy and sell ΔV distributions: Figure 8.2 shows that buy and sell bins have very similar heights overall. To further investigate the distribution, it is possible to plot the hourly distribution of the executions every day.

To conclude the first analysis is possible to analyze the distribution of the executions. For $\phi = 0$ AMS is always better than RTM for both buy and sell orders during all the hours (the differences between the executed volumes in the venues are always large). One important thing to notice is that during the time interval h. 12:00 - 15:00 the executed quantity increases by far (Figure 8.11).



(a) Buy orders

(b) Sell Orders

Figure 8.3: Execution hourly comparison for $\phi = 0$.

Once again, as in the case of volatility and spread, it is evident how the results of ΔV are influenced by the European/US markets.

Before proposing the models for forecasting ΔV, it is useful to summarise the observations in this first analysis. It is indeed possible to note that:

- **AMS outperforms RTM.** This fact is highlighted for both buy and sell orders ($\Delta V_{buy}, \Delta V_{sell} > 460k$). The only small exception is visible in the afternoon after h 16:00. There, RTM and AMS seem to have a comparable performance for $\phi = 1$. If with aggressive strategies the volume in front of your order in the queue should be the little in both venues, the AMS order flow plays a crucial role in the executions.

- **Different time intervals:** from h. 12:00 to 15:00 the executed volumes increase due to the fact that both EU and US traders are trading. Moreover in the late afternoon (after 15:00) RTM increases a lot its performance with respect to AMS: being a Chicago based venue, more people are trading in the US hours (AMS is indeed based in London). Defining the three period with the notation $T = 0$ for h. 8:00 - 12:00, $T = 1$ for h. 12:00 - 15:00 and $T = 2$ for h. 15:00 - 19:00, it is possible to compute the ΔV mean values in every interval (Table 8.1).

  Table 8.1: Mean Values of ΔV for $\phi = 0$ and $\phi = 1$.

T	ΔV_{buy} $(\phi = 0)$	$\Delta V_{sell}(\phi = 0)$
0	0.577	0.550
1	0.465	0.444
2	0.349	0.374

  Let's note that the hourly division has been limited to only three intervals because they are the most significant. Choosing a division into more subgroups could have led to overfit (the sample is only a synthetic week). The reduction of the efficiency of AMS during the day should be taken into consideration in order to obtain faster executions.

- **No remarkable buy/sell differences**: there is no important difference between buy and sell orders result for the aggressive strategies. Being the values of $\Delta\bar{V}$ extremely similar and checking the hourly plot, during the most used trading hours AMS has a constant better performance in both buy and sell sides. Table 8.1 shows that the mean values of ΔV seem to be similar. This fact is not taken into consideration in this first analysis, but it will be further analysed in the next sections. The fact that buy/sell orders could show different results in the execution, in fact, is clearly observed for the more passive strategies.

Having clarified these points, it is now possible to analyse the correlated features with the results in order to obtain as simple a model as possible for predicting ΔV.

8.2.2 Important statistics for $\phi = 0$

By analysing the other statistics for $\phi = 0$, it is clear that the real game changer in the venue selection for aggressive strategies is the venue's liquidity. With an aggressive strategy, that's the most important thing. If people are trading, your order will be executed faster because it is very convenient. In AMS the order flow is a way greater that in RTM and this fact does not depend on specific market conditions or statistics, but it depends only on the number of traders of the two venues.

The first and most important feature in venue selection in the aggressive case is the time of day, as clearly shown in Figure and in Table 8.1. Since the periods are so different, different models will be fit for each T value. Calculating the correlation between ΔV and all the other statistics, none

exceed the $-0.1/0.1$ correlation threshold except the difference between the best bid/ask in AMS and the best bid/ask in RTM. Let's define this quantities as:

$$\Delta B(t) = B^{\text{AMS}}(t) - B^{\text{RTM}}(t) \tag{8.3}$$

$$\Delta A(t) = A^{\text{AMS}}(t) - A^{\text{RTM}}(t) \tag{8.4}$$

Both time series are stationary and the ΔV value are stationary according to the Augmented Dickey–Fuller test: it is possible to study their correlation. ΔB has a remarkable Pearson correlation with ΔV_{sell} ($\rho = +0.32$), while ΔA has $\rho = -0.14$ for the buy case. These facts are logical: if, for instance, ΔB is large, the AMS best bid will be larger than the RTM's one: for a sell order in this situation the RTM execution will have an advantage as the price of the opposite side is lower, hence closer. For this reason, the value of the correlation with ΔV is negative. This reasoning also applies in the buy case with ΔA. Note that being aggressive orders, there isn't the correlation between the results of the strategies and the best price of the same order book side (buy order with the best bid or for the sell order with the best ask): being $\phi = 0$ a very aggressive strategy, the buy/sell orders are the first in line and the execution does not depend on where the best bid/ask is.

For the research's purposes, it is important to better analyse the relationship between the variables. Observing Figure 8.2, one can see how the output, continuous by definition, is similar to a categorical output. This fact makes the correlations non-linear and complicates the description. It is indeed a case where using linear models the residuals will not have a normal form but will have more peaks due to miss-classification errors. On the other hand, classification models will have to handle a large class imbalance problem (the class containing $\Delta V = 0$ is certainly much more represented) and their output will have to be converted back to a continuous value. Note in fact that for the Equation 8.2 and in general for the parallel trading approach, the one must not only select the best venue given the corresponding category, but also measure how better it can perform.

8.2.3 Proposed models for $\phi = 0$

The first proposed approach was a simple mean model. This model considers only the means of the data in the train set for every value of T. The model is extremely easy and has the only assumption that in the three periods there are Substantial differences between the values of ΔV. Performing the Kruskal-Wallis test on the three time series $\Delta V(T)$, with absolute certainty in every case (p-value = 0) i is possible to reject the null hypothesis and conclude that the data belong to three groups with clear differences. Here are the trivial formulas:

$$\hat{\Delta V}^{\text{buy}}(\phi = 0, T) = 0.577\, \mathbb{I}_{T=0} + 0.465\, \mathbb{I}_{T=1} + 0.349\, \mathbb{I}_{T=2} \tag{8.5}$$

$$\hat{\Delta V}^{\text{sell}}(\phi = 0, T) = 0.550\, \mathbb{I}_{T=0} + 0.444\, \mathbb{I}_{T=1} + 0.374\, \mathbb{I}_{T=2} \tag{8.6}$$

The goal now is try to use the other significant statistics for the models: ΔB for the sell case and ΔA for the buy case. To do that, the correlation of the factors have to be studied. The analyses obtained from this study are similar for both cases. Let's start observing Figure 8.4 that shows the relation between ΔB and ΔV^{sell} (case $T = 2$).

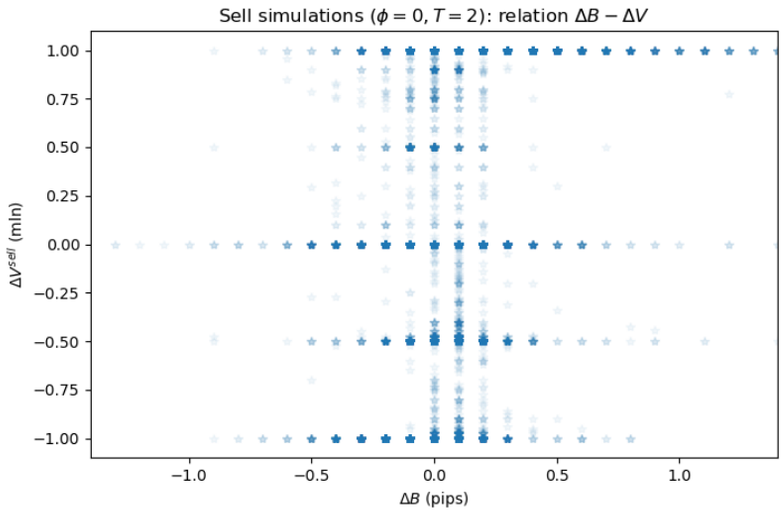


Figure 8.4: Relation for sell simulations ($\phi = 0, T = 2$) between ΔB and ΔV.

As anticipated, the output ΔV appears almost categorical. On the other hand, ΔB also appears categorical as the difference is always a multiple of 0.1 pip. The positive correlation noted in the coefficients is evidently non-linear and can be noticed by the tails at the top right and bottom left of the graph. As there are a large number of points in this graph, it is complex to understand what is happening around zero simply by looking at it.

The first attempt is to fit a simple linear regression model between the variables. The output is truncated in the interval $[-1, 1]$ (ΔV is in that interval). The main problem with this idea is the fact that the relationship between the variables is not linear: while the results may logically improve on those of ALGO or the mean, the model's residuals are neither normal nor heteroschedastic. The best way to understand these problems is visualize the model compared with the ones proposed before in Figure 8.5.

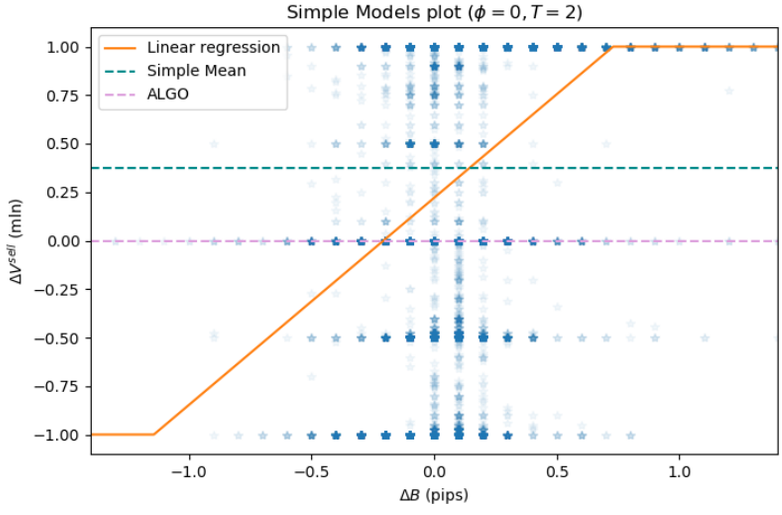


Figure 8.5: First models comparison: ALGO, simple mean and linear regression. Case sell, $\phi = 0$, $T = 2$.

it is clear that the model, although improving on the results of previous models, is unable to capture the relationship between the two variables. Moreover, note that the residuals obtained from this model are not normal as they will distribute themselves according to precise patterns

being categorical.

Another approach that could lead to strong overfitting is to exploit the fact that ΔB is also categorical. One solution would be to calculate the average value of ΔV for every possible value of ΔB. The model could certainly be influenced by the presence of outliers and the dataset studied, but it could certainly give clear indications on how the behaviour of the venues. To distinguish this proposed model to the simple mean model, it will be indicated as "mean per value" model. Figure 8.6 adds the model results in the previous plot.

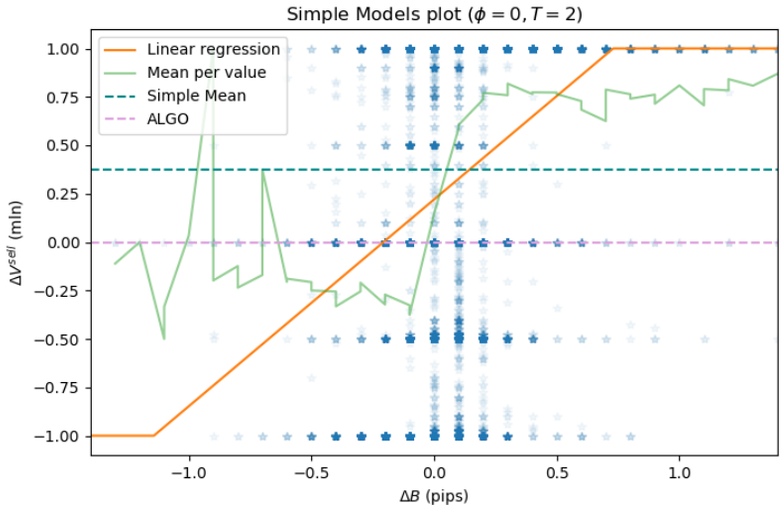


Figure 8.6: Second models comparison: ALGO, simple mean, linear regression and mean per value. Case sell,$\phi = 0$, $T = 2$.

The graph shows that the model may work well for the most common values of ΔB around zero, but is extremely oscillatory at the extremes (especially to the left). However, the fact that the central part clearly resembles a sigmoid can be exploited to fit an even better model that is robust to overfitting. Before going into the details of the proposed final model, it is worth mentioning that among the models attempted was a logistic regression model. The model, based on the sigmoid function, fictitious with python had numerous problems such as imbalance between classes (AMS outperforms RTM) and results that were extremely skewed by outliers. For this reason, the proposed function is not dictated by any coding language, but obtained with pen and paper after a series of logical reasonings.

In order to fit a shifted sigmoid, it is important to define the boundaries $[l_b, u_b]$ in which our in which the function is defined. All the $\Delta\hat{V}$ predicted by the model will be in this interval. The sigmoid function is now in the form:

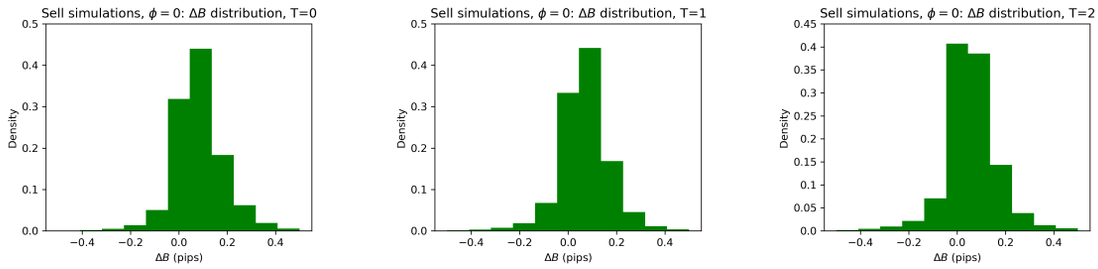
$$\sigma(x) = \frac{u_b - l_b}{1 + e^{-(\beta_0 + \beta_1 x)}} - l_b \tag{8.7}$$

Now, it is possible to estimate the values of β_0 and β_1. The way this parameters are fitted is simple: selecting the most common values of x (in the studied case ΔB and ΔA) and studying computing the values. For all the simulations, one of the two most common value of x is zero. Defining as $\bar{x}$ the non zero point in which the function is fitted, the equations of the parameters are:

$$\beta_0 = -\log\left(\frac{u_b - l_b}{\sigma(0) - u_b} - 1\right) \tag{8.8}$$

$$\beta_1 = -\frac{1}{\bar{x}}\left(\log\left(\frac{u_b - l_b}{\sigma(\bar{x}) - u_b} - 1\right) + \beta_0\right) \tag{8.9}$$

In order to fit all the models for every situation, we need to select the values of $l_b, u_b, \bar{x}$. Let's start with ΔB in the sell case: it is possible to plot the histograms in 8.7.



(a) $T = 0$ (b) $T = 1$ (c) $T = 2$

Figure 8.7: Distribution of ΔB sell simulations in every selected interval T.

Let's start considering the ΔB values in the sell simulations. Selecting $\bar{x} = 0.1$, $l_b = \bar{\Delta V}_{x \leq -0.2}$ (mean value of ΔV for $\Delta B \leq -0.2$) and $u_b = \bar{\Delta V}_{x \geq 0.2}$ (mean value of ΔV for $\Delta B \geq 0.2$), it is possible to fit a sigmoid function. Note that the boundaries are close to zero because the majority of the data is there. Moreover, if this strategy is used in production, it will be used for large split orders, and we don't want to allocate 90/100% of large orders to one venue. The proposed boundaries are $\approx [-0.25, 0.75]$: this choice allows obtaining values of $w_A \neq 1$. With this simple ideas, the results are extremely good, like Figure 8.8 shows.

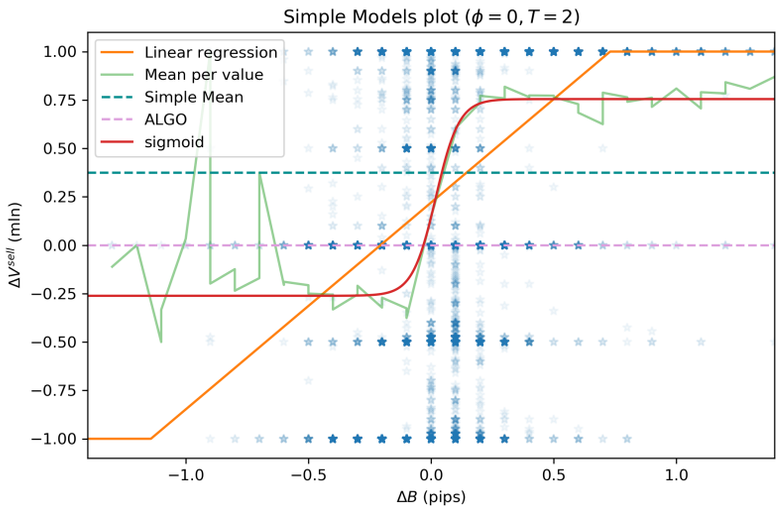


Figure 8.8: Last models comparison: ALGO, simple mean, linear regression, mean per value and sigmoid model. Case sell,$\phi = 0$, $T = 2$.

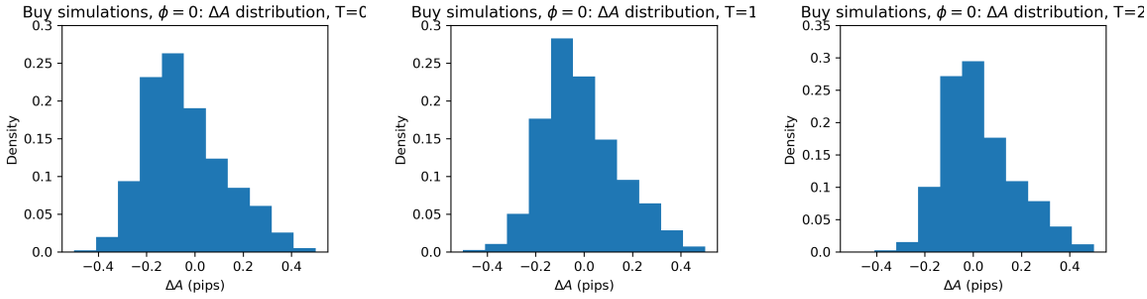
As shown in the figure, the sigmoid approach seems to fit the mean per value model perfectly, making it robust and logically consistent. To compare the results of every model except the mean per value which is not consistent, they were tested using 22/10 as a test set. The MSE is compared in Table 8.2.

Table 8.2: Mean squared error metrics comparison between ALGO approach, mean models, linear models and sigmoids. Sell case, $\phi = 0$.

T	ALGO	Mean models	Lin. models	σ models	Δ ALGO/σ
T=0	0.704	0.566	0.512	0.482	-31.5%
T=1	0.687	0.531	0.476	0.441	-35.8%
T=2	0.760	0.462	0.414	0.374	-50.8%

This Table proves what was anticipated. Among the simple models, the sigmoid-based model is the best model for understanding the behaviour of order execution in the sell case.

The same approach is applicable to the buy case. The only difference is that $x = -\Delta A$: in this case the correlation is positive and to fit the same sigmoid a negative correlation is needed. The selection of $\bar{x}, l_b, u_b$ again is data based (Figure 8.9).



(a) $T = 0$ (b) $T = 1$ (c) $T = 2$

Figure 8.9: Distribution of ΔA buy simulations in every selected interval T.

In the buy case, the tail are clearly heavier. This fact could be related to some fact of the markets outside FX. The selection of $\bar{x}$ is again 0.1 (the value $\Delta A = -0.1$ is the most represented non-zero value for all T). In order to give more important to the heavy tails, $l_b = \bar{\Delta V}(x \leq -0.3)$ and $u_b = \bar{\Delta V}(x \geq 0.3)$. The results show a strong improvement with respect to ALGO:

Table 8.3: Mean squared error metrics comparison between ALGO approach, mean models, linear models and sigmoids. Buy case, $\phi = 0$.

T	ALGO	Mean models	Lin. models	σ models	Δ ALGO/σ
T=0	0.770	0.444	0.441	0.440	-42.9%
T=1	0.736	0.457	0.454	0.456	-38.0%
T=2	0.687	0.552	0.549	0.543	-21.0%

Again the sigmoid models have the best MSE values for every value of T with respect to ALGO, but this time there is an exception for $T = 1$. In this situation, the linear model outperforms the sigmoid with -0.4% reduction in the MSE. In fact, it is important to point out that in this case the sigmoid model seems to be a little less performing than in the other case (it is comparable with the linear model). In fact, the model suffers from the very heavy tails of ΔA distribution and the performance is comparable to both linear models and mean models. Despite this, from the data it seems that the buy case is more difficult to predict even for the linear model with MSEs always similar to the mean models. In general, since the sigmoid model is extremely simple, easy to fit and explainable, it is the model selected as best.

Looking at the tables, it is clear that ALGO's current approach is not optimal for a very aggressive strategy: even simply averaging the executions in the venues of each hourly window improves it by a lot. A simple sigmoid models, trained not with logistic regression approach, but just considering the mean values of ΔV for $x = 0, \bar{x}$ and selecting to boundaries l_b, u_b has good result. The Logistic Regression, in fact, has to receive a categorical target value and suffers both the class imbalance and the heavy tails of the $\Delta B / \Delta A$ distributions.

In conclusion, for the more aggressive strategy analysed the ALGO approach would not be optimal at all: a lot of trading opportunities in AMS would be lost. [...]. It is crucial to divide the day into three periods using T and create three different propositions because the market clearly changes depending on the time of day. The very simple models proposed take into account the only feature that appears relevant. Given the models and Equation 8.2, w_A can be computed.

8.3 Floating aggressive strategies with $\phi = 1$

This section will analyse the output of the simulations with $\phi = 1$ strategies. The price of the orders is one tick below of the average of the mid prices in the buy case and one tick above in the sell case. Again, a large number of executions in the venues is expected, being a price still aggressive. In fact, with $\phi = 1$ the executions are 43157 for the buy simulations and 43168 for the sell ones.

8.3.1 ΔV results

It is possible to observe clear similarities with the $\phi = 0$ case. First of all, even with $\phi = 1$, AMS outperforms RTM. On average, for the buy orders $\Delta \bar{V}_{buy} = 0.271$ and $\Delta \bar{V}_{sell} = 0.285$ for sell orders. The percentage of $\Delta V = 1$ simulations is around 28%, like Figure 8.10 shows, but note that the value of ΔV is less than the corresponding value for $\phi = 0$: the strategy while being very aggressive is a little less so, but the AMS liquidity still makes the difference.

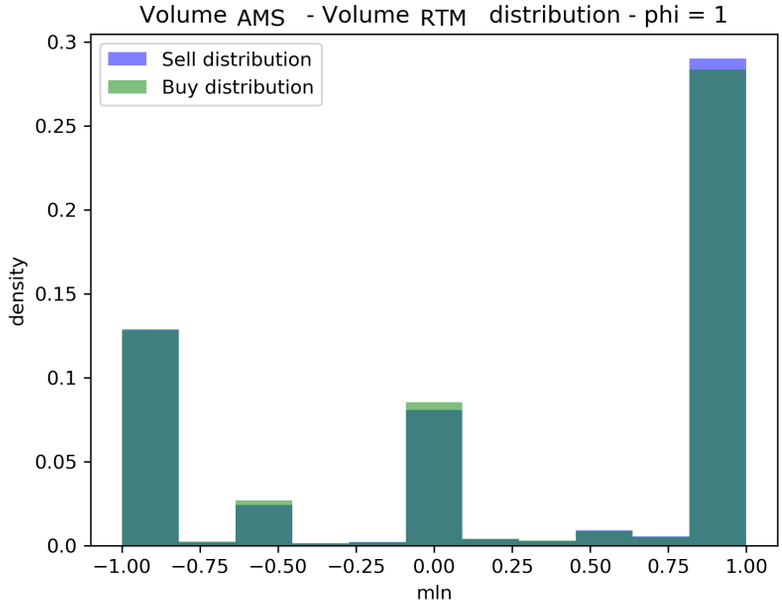
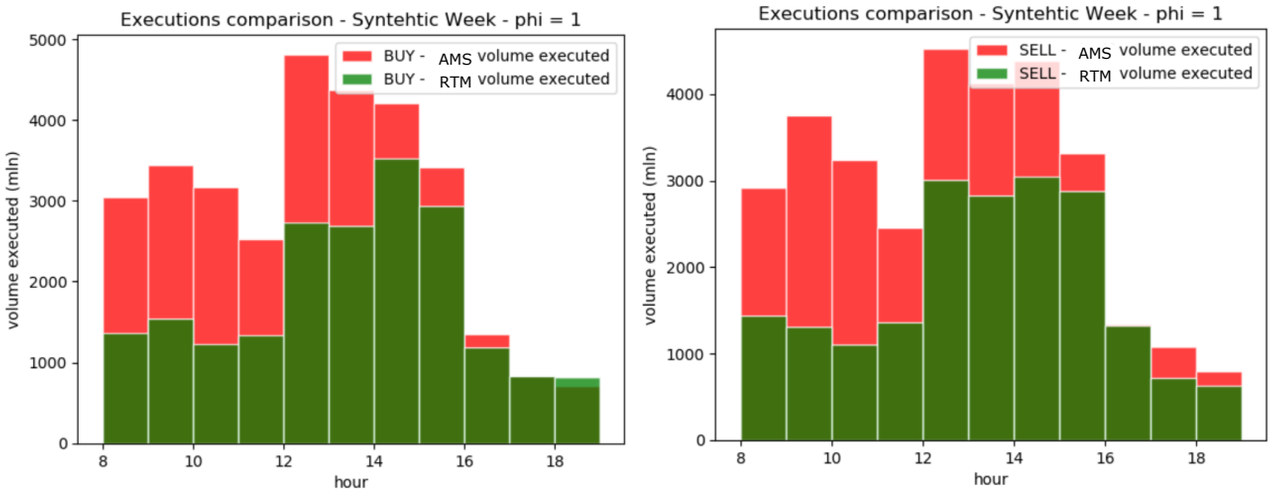


Figure 8.10: Difference between volumes traded in AMS and RTM for $\phi = 1$.

Figure 8.10 is clearly similar to Figure Figure 8.2 that represents the analogous $\phi = 0$ case. Moreover, considering the time, Figure 8.11 shows the hourly executed volumes.



(a) Buy orders.

(b) Sell orders.

Figure 8.11: Execution hourly comparison for $\phi = 1$.

Again, Figure 8.11 shows that it is possible to divide the daily time intervals in three different groups as in the previous case. The time selection is the same, assigning 15:00 to the last group as AMS decreases executions while RTM remains stable. To prove with numbers that the T division make sense also in this case, Table 8.4 reports the $\hat{\Delta V}(T)$.

Table 8.4: Mean Values of ΔV for $\phi = 1$.

T	$\bar{\Delta V}_{buy}(\phi = 1)$	$\bar{\Delta V}_{sell}(\phi = 1)$
0	0.442	0.467
1	0.243	0.231
2	0.055	0.100

If, as in the $\phi = 0$ case, there is no big sell and buy difference for $T = 0, 1$, this is not true for $T = 2$. In fact, looking at $\hat{\Delta V}$, AMS is still better in both cases, but if the sell case $\hat{\Delta V} = 0.1$, in the buy case it is *only* $\hat{\Delta V} = 0.055$, with a -45% percent reduction. This fact, which will be present in more passive strategies, may be due to factors outside the EURUSD market, but affecting it. After this first analysis on the results of the $\phi = 1$ simulations, it is possible to study the main statistics that ply a role in the ΔV results.

8.3.2 Important statistics for $\phi = 1$

Comparing this case to the previous one, the results are a little bit different: in fact, more statistics seem to be correlated to ΔV. Let's define the difference between the AMS volume in front and the RTM volume in front as δV_F (the choice of using δ a and not Δ is to distinguish this value from ΔV). Moreover, the differences between the moving average of best bid and best ask in the last 500 ms are correlated. This fact makes sense: a price movement in the last half-second could mean an opportunity for an aggrressive strategy. Note that for sure, it will not be an arbitrage (usually they are a way faster than this). Again, all the time series are stationary for the ADF test: the correlation is significant. Table 8.5 and Table 8.6 show the correlation matrix of most important features.

Table 8.5: Correlation matrix for the selected columns for $\phi = 1$ buy experiments

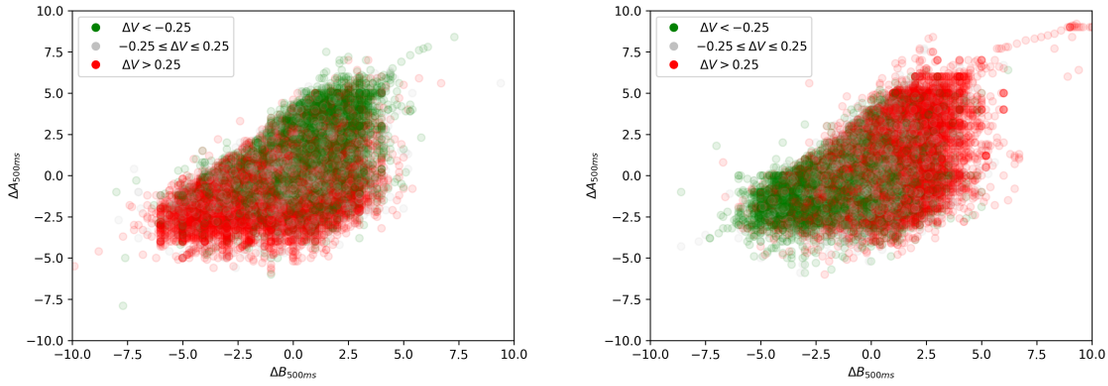
	ΔV	δV_F	ΔB	ΔA	ΔB_{500ms}	ΔA_{500ms}
ΔV	1	-0.29	-0.29	-0.39	-0.26	-0.4
δV_F	-0.29	1	0.66	0.48	0.53	0.39
ΔB	-0.29	0.66	1	0.6	0.79	0.49
ΔA	-0.39	0.48	0.6	1	0.46	0.78
ΔB_{500ms}	-0.26	0.53	0.79	0.46	1	0.54
ΔA_{500ms}	-0.4	0.39	0.49	0.78	0.54	1

Table 8.6: Correlation matrix for the selected columns for $\phi = 1$ sell experiments

	ΔV	δV_F	ΔB	ΔA	ΔB_{500ms}	ΔA_{500ms}
ΔV	1	-0.25	0.38	0.27	0.4	0.27
δV_F	-0.25	1	-0.38	-0.59	-0.35	-0.5
ΔB	0.38	-0.38	1	0.5	0.65	0.37
ΔA	0.27	-0.59	0.5	1	0.45	0.75
ΔB_{500ms}	0.4	-0.35	0.65	0.45	1	0.54
ΔA_{500ms}	0.27	-0.5	0.37	0.75	0.54	1

From the tables above, it is clear that all selected features are important for the prediction of the ΔV value, but that there is also correlation between them. Some information contained in the features is contained in the other features, like for ΔB and ΔB_{500ms} or for ΔA and ΔA_{500ms}. The propose solution, in order to have orthogonal independent features, is a simple PCA transformation. Before explain the proposed models, it is important to understand the correlation between the features with some plots.

Let's start with ΔB_{500ms} and ΔA_{500ms}: they are the first important rolling features included the analysis. For both sell and buy simulations the correlation between these statistics is high ($\rho = 0.54$): if the best bid has moved in the last 500ms, also the best ask may have moved in the same direction. This fact makes sense: the price of both sides of he LOB move in the same direction. The selected plot is a scatter-plot which uses the colors to highlight the correlations (Figure 8.12).



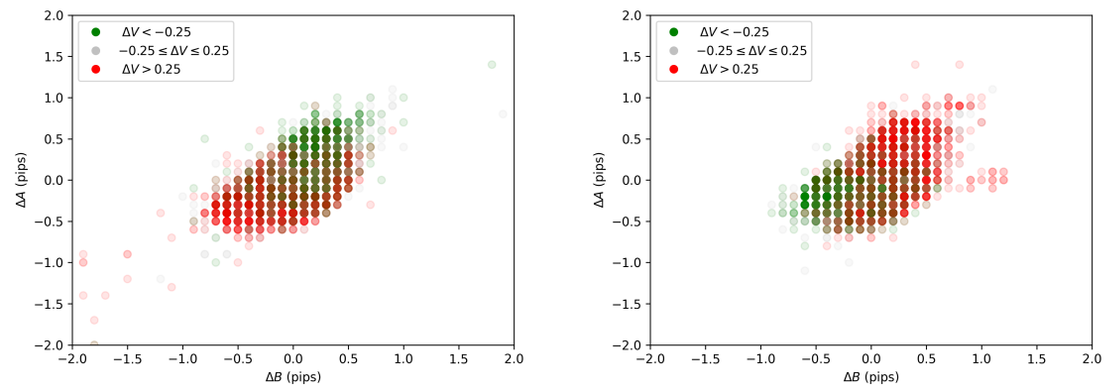
(a) Buy simulations.

(b) Sell simulations.

Figure 8.12: Correlation of ΔV with ΔB_{500ms} and ΔA_{500ms}. $\phi = 1$.

Figure 8.14a shows a negative correlation between ΔV for buy simulations and the selected features. With high values of them the balance goes to RTM (green), while the other way around is red (AMS). On the other hand, Figure 8.14a shows a positive correlation between the statistics and ΔV for the sell case. But what does this mean? Let's imagine a situation where both values of $\Delta B_{500ms} << 0, \Delta A_{500ms} << 0$. In this case, the scenarios are two: or a micro down trend is happening and AMS is faster than RTM to go down, or a micro upper trend is happening and AMS is slower than RTM to go up. It is logical that in a scenario of buy strategy, an order will be more attractive in a venue with the lowest prices. Conversely, in the event of a sale, the price will be more attractive in a market where prices are higher. Of course, the opposite reasoning is valid for $\Delta B_{500ms} >> 0, \Delta A_{500ms} >> 0$. In hybrid cases, the choice of venue is more complicated and must be based on other statistics. The correlation coefficients in Table 8.5 and Table 8.6 are confirmed both by the plots and the logic.

The same plots are proposed for ΔB and ΔA in Figure 8.13.



(a) Buy simulations.

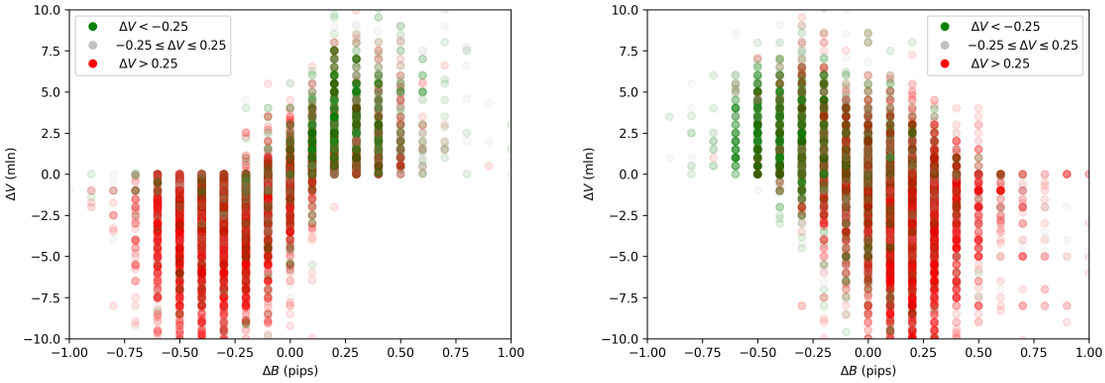
(b) Sell simulations.

Figure 8.13: Correlation of ΔV with ΔB and ΔA. $\phi = 1$.

Again, the correlation values in Table 8.5 and Table 8.6 are confirmed. However, a logical interpretation must be added: let's assume $\Delta B << 0, \Delta A << 0$. In this case, the best bid price and the best ask price of AMS is less than in RTM. From these assumptions, it is clear that an aggressive purchase price in AMS is more competitive than in RTM. RTM, in fact, will be the best venue to sell since prices are higher than in AMS.

The last relevant feature selected is δV_F, the difference between the volume in front of the order

in the venues. The correlation is logical: if my order has more value in front in AMS ($\delta V_F > 0$), will most likely have a faster execution in RTM ($\Delta V < 0$). The opposite is obviously true and these facts prove the negative correlation shown with ΔV in both buy and sell simulations. If this the value of ρ and the logical explanation were not enough, Figure 8.14 shows the relationship between ΔV_F, ΔB with ΔV. The choice to use ΔB was random: there is no particular reason.



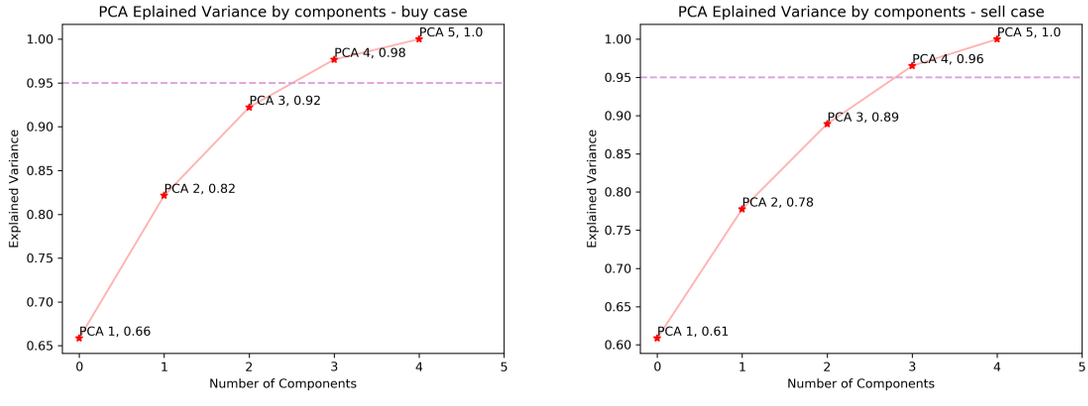
(a) Buy simulations.

(b) Sell simulations.

Figure 8.14: Correlation of ΔV with ΔB and δV_F. $\phi = 1$.

It is clear from the graph that at negative values of δV_F, AMS executions (red dots) are favoured. On the contrary, for positive values, RTM seems faster. After analysing all the relevant features from a logical, graphical and analytical point of view, it is possible to perform a Principal Component Analysis (PCA) reduction and start studying the models.

Given the five selected statistics in the train set, it is possible to normalize them using mean and standard deviation. Given that, performing PCA, it is possible to analyze the explained variance of the transformation. The explained variance by each component in Principal Component Analysis (PCA) represents the proportion of the total variance in the data that is captured by that component. It provides information about how much information each principal component carries. Figure 8.15 shows the explained variance of PCA components.



(a) Buy simulations.

(b) Sell simulations.

Figure 8.15: Explained variance of PCA components for $\phi = 1$.

In order to select a number of components in order to have an explained variance larger than 0.95, for both buy and sell case $N = 4$ is selected. Note that the test data are normalized using the mean and the standard deviation of the train set. Moreover, the PCA is trained on the train set and used on the test set without retraining.

8.3.3 Proposed models for $\phi = 1$

As in the $\phi = 0$ case, the first model proposed is the simple mean. As before, it consists on take the average of the training set on one particular time interval defined by T.

$$\Delta\hat{V}^{\text{buy}}(\phi = 1, T) = 0.442\,\mathbb{I}_{T=0} + 0.243\,\mathbb{I}_{T=1} + 0.055\,\mathbb{I}_{T=2} \tag{8.10}$$

$$\Delta\hat{V}^{\text{sell}}(\phi = 1, T) = 0.467\,\mathbb{I}_{T=0} + 0.231\,\mathbb{I}_{T=1} + 0.100\,\mathbb{I}_{T=2} \tag{8.11}$$

To keep it as simple as possible, the second proposed models are linear regression models, truncated into the interval $[-1, 1]$. This time, it will receive the 4 inputs x (outputs of the PCA model) and it will give the output:

$$\Delta\hat{V}(\phi = 1, T, x) = \min(\max(\beta_0 + \sum_{i=1}^{4} \beta_i x_i, -1), 1) \tag{8.12}$$

The first thing to notice this time is that, using more useful input features, the model outperforms the mean models in every case with a $\geq 16\%$ MSE reduction. On the other hand, the assumptions are still missed: being (again) the output ΔV pseudo-categorical (Figure 8.10) the residuals are neither Gaussian nor heteroschedastic as in the $\phi = 0$ case. Figure 8.16 shows the residuals plot and it is evident that there are distinct lines formed.

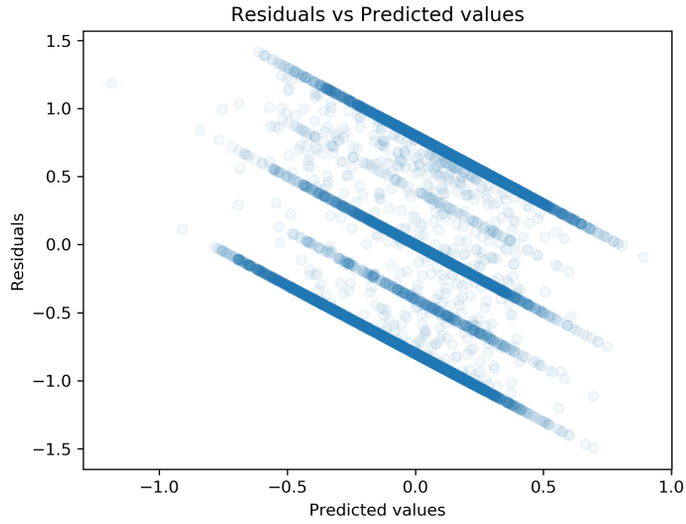


Figure 8.16: Residuals plot of the linear model in the sell simulations case, $T = 2$, $\phi = 1$.

Ideally, the residuals should be scattered randomly around the zero line without any discernible patterns. In this case, being the output pseudo categorical, the residuals have a lines pattern: for every value of the predictions, there are mainly three values of common residuals $-1, 0, 1$. This fact creates the lines: the model on one hand outperforms by far the ALGO model and the simple mean model, on the other hand it seems not adequate to capture the underlying relationship between the predictor variables and the response variable. The main problem here, it is that the relation is not linear and there is a large random component. The fact that so many traders are trading on both venues and that the market is indeed influenced by a huge amount of factors makes it very difficult to predict the best venue. To give an example, given the statistics the model might have chosen an biased allocation in AMS, but if a trader went to RTM with an opposite side market order it would make the choice in vain. The more people trade, the more unpredictable the venues. On the other hand, the linear regression model, so simple and trivial to fit and use, could be a simple and extremely understandable answer to the call for improving the current system.

Like, in the case before, a sigmoid based models are proposed. If before it was easy to fit with some simple underlying ideas to avoid outliers biases ($\sigma(\phi = 0) : \mathbb{R} \to \mathbb{R}$), now the problem is more complex ($\sigma(\phi = 1) : \mathbb{R}^4 \to \mathbb{R}$). The first attempt was a Logistic Regression defining three different classes, but again there were problems: the class weights should be trained and every combination buy/sell - T. Moreover, the assigned probabilities did not produce good results. For this reason a second model is proposed and analysed with a higher level of complexity. The model's structure is a simple neural network with just one input layer and one output layer. If usually the structure of the neural network (number of neurons, number of layers) is trained and tested to select the best performing one regardless of its complexity, in this case the structure is simple and fixed. This trivial selection is of course due to make everything simple and understandable. Figure 8.17 shows the simplicity of the selected structure.

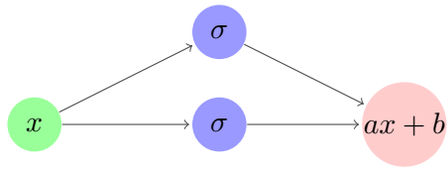


Figure 8.17: Simple structure of the proposed Neural Network: there are just one layer with two neurons with a sigmoid activation function and a linear output layer output layer (realized with Tikz [25]).

Let's observe and analyze the structure in Figure 8.17: the normalized input data ($x \in \mathbb{R}^5$) are processed by two neurons with a sigmoid function. These neurons should represent the probability of $\hat{\Delta V}$ to be equal to be positive or negative. Lastly, the final layer is a simple linear transformation of the result. In total, the neural network ha 15 trainable parameters: 12 for the first sigmoid layer and 3 for the linear output layer.

To further understand the neural network, the input statistics $x \in \mathbb{R}^5$ were multiplied with two different vectors (w_1, w_2) before entering the sigmoid neuron. Also two bias values are trained in this first layer. To summarize what the first layer does:

$$\sigma_i(x) = \frac{1}{1 + e^{-(w_i^T x + b_i)}} \quad \forall i = 1, 2 \tag{8.13}$$

Note that the total number of trainable parameters of the first layer is twelve: six for each sigmoid neuron, namely $w_i \in \mathbb{R}^5$ and $b_i \in \mathbb{R}$. The output of Equation 8.13 should represent the probability of the value ΔV value to be positive/negative given the input vector. Let's define these values as p_+ and p_-:

$$p_+ = \sigma_1(x) \in (0, 1)$$
$$p_- = \sigma_2(x) \in (0, 1)$$

Given the probabilities values, it is now needed to compute the output combining those with a linear transformation. The output layer computes the linear transformation:

$$\hat{\Delta V} = w_+ p_+ + w_- p_- + b \tag{8.14}$$

To be sure that the values of w_+ and w_- are logical and explainable they are initialised to 1 and -1 respectively. Furthermore, at the end of each train, it was checked that their values remained discordant.

Being a very simple neural network, the hyperaparmeters tuning was not long. After a grid search approach with a validation set, the selected hyperaparameters were: learning rate $\eta = 0.001$ (default value) and batch size $n = 128$. The training was performed with a max of 1000 epochs using a Early stopping techniques with a patience of 5 and restoring the best weigths obtained on the training set. The optimizer selected was Adam with the mean squared error as loss function.

To be coherent and find a general model, the structure and the hyperparameters were kept fixed for all the values of T and for both buy and sell simulations.

After the explanation of the models, it is now possible to show the models' results for buy and sell simulations on the same test set (Friday 22/10):

Table 8.7: Mean squared error metrics comparison between ALGO approach, mean models, linear models and simple neural networks. Buy case, $\phi = 1$.

T	ALGO	Mean models	Lin. models	NN models	Δ ALGO/LM	Δ ALGO/NN
T=0	0.822	0.644	0.580	0.538	-29.4%	-34.5%
T=1	0.777	0.676	0.583	0.572	-25.1%	-26.4%
T=2	0.726	0.725	0.638	0.615	-12.1%	-15.2%

Table 8.8: Mean squared error metrics comparison between ALGO approach, mean models, linear models and simple neural networks. Sell case, $\phi = 1$.

T	ALGO	Mean models	Lin. models	NN models	Δ ALGO/LM	Δ ALGO/NN
T=0	0.825	0.691	0.616	0.572	-25.3%	-30.7%
T=1	0.762	0.751	0.641	0.647	-15.9%	-15.1%
T=2	0.750	0.774	0.669	0.648	-10.8%	-13.6%

In both scenarios, linear models, and simple neural networks consistently exhibit superior performance in terms of MSE compared to the ALGO model. The percentage differences underscore the significantly larger improvement achieved by these alternative models when compared to the ALGO model. These findings strongly suggest the need for an improved execution algorithm strategy. Notably, the most pronounced disparity in performance is observed during the morning hours ($T = 0$). In this period, AMS significantly outperforms RTM, while the ALGO model's volume allocation leads to imprecise outcomes. It is worth mentioning that when comparing the mean models with the two proposed models, it becomes evident that in complex scenarios where selecting the optimal venue is challenging overall, the more intricate models outperform the simpler ones. This is particularly notable for $T = 1, 2$, where both venues yield comparable results and AMS's dominance is not evident.

Despite the simplified structure and low parameter count of the neural network, it consistently achieves results similar to linear regression, slightly better with the exception of $T = 1$ buy case. On the other hand, the simplicity of the linear regression model combined with the PCA makes the model easy and understandable. This research to date suggests the linear model, but with more data for training and new hyperparameters tuning the neural network might be the better choice.

8.4 Aggressive strategies summary

The first strategies analysed in this research were the aggressive ones, specifically $\phi = 0$ and $\phi = 1$. The cases have similar results:

- AMS outperforms RTM in both cases: with aggressive orders the liquidity and the fact that in AMS more people are trading play a crucial role.
- It is possible to divide each day in three different periods: $T = 0$ from 8:00 to 12:00, $T = 1$ from 12:00 to 15:00 and $T = 2$ from 15:00 to 19:00. Table 8.1 and Table 8.4 show clearly that the $\Delta\hat{V}$ values are different.

- In general, there are not remarkable buy/sell differences in the results. The only exception is noted for $T = 2$, $\phi = 1$ case. Here, the buy case shows a -45% reduction in the $\Delta\bar{V}$ value compared to the sell one.

- For $\phi = 0$ case, the important statistics are ΔA for the buy case and ΔB for the sell case. Since the order is very aggressive, the only statistic to check is the best price on the opposite side of the orderbook. It is indeed not important to look at one's own side as the order posted is most likely the first. The correlation is stronger for the sell case than for the buy one.

- For $\phi = 1$ case, the important statistics are the same for buy and sell case: ΔB, ΔA, ΔB_{500ms}, ΔA_{500ms} and δV_F. Since in this case the order is aggressive, but usually not the first, it is important to check the statistics of the same side as volume in front or best ask/best bid. The rolling features (ΔB_{500ms}, ΔA_{500ms}) explain some micro price movement that show an advantage of a venue.

- For $\phi = 0$ case, all the proposed models outperform ALGO. Mean models show very good results, but simple models such as linear regressions do as well. The best proposed model in this case is a shifted sigmoid that using just the value ΔB (for the sell case) and ΔA (for the buy case). Table 8.2 and Table 8.3 show the MSE of the models comparison. Having the sell case a stronger correlation between ΔB and ΔV the results are slightly better.

- For the $\phi = 1$ case, all the proposed models outperform ALGO. Table 8.8 and Table 8.7 show the MSE of the models comparison. While they continue to be better than ALGO, mean models perform much worse than linear models and simple neural networoks this time. The proposed neural network structure remains very simple and understandable and is the best model in terms of MSE. On the other hand, a simple regression model that uses the PCA data reaches very good results. Being very simple and explainable, the models chosen are linear ones.

It is now possible to begin the analysis of the more passive strategies, but first one thing must be noted. The more passive the strategy becomes, the longer the execution time. In fact, the number of completed simulations per hour decreases. This fact has two main consequences. First, it is necessary to generate datasets using more days in order to guarantee the reliability of the models and get better results. Second, it is increasingly complex to find working models. The duration of the simulations increases and the statistics analysed at the beginning may be totally unrelated to the final result of the simulation due to the longer and longer running time. It therefore begins to emerge that the fact that ALGO establishes the best execution a priori and does not dynamically change the allocations may not be the optimal choice [...].

Chapter 9

Synthetic Data Analysis: intermediate strategy

In this section an intermediate strategy ($\phi = 2$) is analysed. This strategy is classified as intermediate because it does not exhibit the behaviour of the most aggressive strategies, nor does it resemble the behavior of the most passive ones. The first section analyses the synthetic week and explain why to improve the analysis another week is needed.

9.1 Synthetic week limitation

The aim of this section is to explain why this choice to use another week of data was made and the selection criteria of the new days. In fact, in the selected synthetic week (the same used for aggressive strategies), in total 9975 simulations were performed for the buy side and 9945 for the sell ones. It is immediately noticeable that the number of simulations decreases due to the higher passivity. By generating less data, the results may be weaker and for this reason other simulations were performed.

The values of ΔV in the afternoon are much more balanced and one has that after 15:00 RTM outperforms AMS on average. Table 9.1 shows the $\Delta\bar{V}$ values of the simulations performed.

Table 9.1: Mean Values of $\Delta\bar{V}_{buy}$ and $\Delta\bar{V}_{sell}$ for $\phi = 2$, synthetic week.

T	$\Delta\bar{V}_{buy}$	$\Delta\bar{V}_{sell}$
0	0.324	0.338
1	0.113	0.165
2	-0.072	-0.003

If on one hand it is clear that RTM with more aggressive strategies becomes more competitive, on the other hand the buy/sell differences were spotted more and more after 12:00. The data shown during the simulated week suggests that it may be more favorable to buy in RTM and sell in AMS based on the observed values of ΔV. This phenomena could be related to the fact that one venue has the HQ in London and the other in Chicago. It is likely that the majority of the RTM market participants has to hedge the currencies with respect to the USD, and the majority of AMS market participants has to do the same with respect to the EUR (even though the official currency in London is the pound, many traders are EUR based). The way they hedge their portfolios with respect to the EURUSD market could be different.

Analysing this behaviour on a day-by-day basis, it was noted that four days show similar results to this one and are characterised by a lower mid-price trend throughout the day. The exception is the day of 18/10/2021, with a strong upper trend and a ΔV buy/sell distribution characterised by

a strong upper trend. In fact, 18/10/2021 shows a different behaviour: RTM outperforms AMS for the sell executions and AMS does the same for the buy one until h. 14:00. The reason could be related to the fact that this day is the only day that shows a clear upper trend for the mid price of EURUSD. If the price EURUSD is going up, EUR is taking value over the USD: in this scenario USD based market participants are buying EUR and EUR based market participants want to buy USD for a lower price. This hypothesis has to be check selecting other upper trend days. The opposite reasoning for the down trend present on other days could be the reason for the buy/sell difference. Note that the hedging procedures strongly depend on the values of the assets to which the firms are exposed.

A similar behaviour to that observed in the synthetic week was observed with real MN trading data in both venues and seems to be present only in certain periods. To really understand this fact, a deep analysis should be performed: the idea is that some important information might be contained in geographical indices such as the S&P500 and Euro Stoxx 50. Since this correlation is difficult to study and understand with the data available, this research will focus on the results and the modelling part.

9.2 Two weeks analysis

To increase the number of simulation in the training set, other five days were simulated and analysed. This time, days were selected that showed a clearer upper trend and the same week. The choice of adding a week to a synthetic week also has another reason: by using Thursday and Friday as a test set, it is possible to have more up-to-date data. The choice of trying to simplify the model's work is also due to the fact that the more passive the strategy, the more difficult the prediction. Browsing through the available second half-year data (some days had problems with missing data), the selected week is 21 to 25 June 2021. Although the week predates the synthetic week data, there is no logical link between them. Even if the days of the test set temporally precede those of the train set, the results are still valid. Given these two selected weeks, it is possible now to analyse the results of the simulations. In total, 20412 buy simulations and 20497 sell simulations were performed. Again, the first analysis is on the results of ΔV.

9.2.1 ΔV results

Like in the previous cases, even if the values are defined as continuous, they have a categorical shape. Figure 9.1 shows the ΔV distribution in the buy and sell simulations.

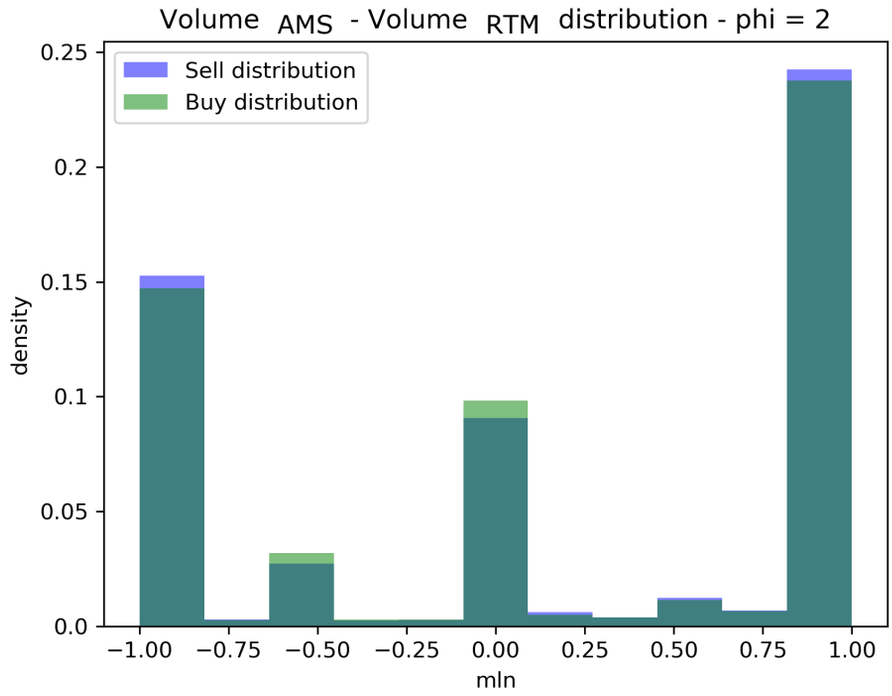


Figure 9.1: Difference between volumes traded in AMS and RTM for $\phi = 2$.

Again, the results show the typical the same intraday patterns with $T = 0, 1, 2$. Adding the historical week with the upper trend days, Table 9.2 shows that the buy/sell differences are mitigated during $T = 2$.

Table 9.2: Mean Values of $\Delta\bar{V}_{buy}$ and $\Delta\bar{V}_{sell}$ for $\phi = 2$, synthetic week and historical week.

T	$\Delta\bar{V}_{buy}$	$\Delta\bar{V}_{sell}$
0	0.300	0.267
1	0.119	0.153
2	0.002	-0.001

If with aggressive strategies AMS seems to be the venue that guarantees the fastest executions in every case, with more passive strategies this fact is not true. With two weeks of data there is a clear evidence that RTM, after 15:00, can achieve the same if not better results.

9.2.2 Important statistics for $\phi = 2$ data

Again, even if more than 100 statistics were computed, the highly correlated features are the same as in the $\phi = 1$ case: δV_F, ΔB, ΔA, ΔB_{500ms}, ΔA_{500ms}. The financial explanation is not repeated and all the plots were checked: they have extremely similar shapes to the ones shown in the previous chapter. It is possible indeed to observe the correlation matrices in Table 9.3 (buy case) and Table 9.4.

Table 9.3: Correlation matrix for the selected columns for $\phi = 2$ buy experiments

	ΔV	δV_F	ΔB	ΔA	ΔB_{500ms}	ΔA_{500ms}
ΔV	1	-0.23	-0.22	-0.23	-0.18	-0.21
δV_F	-0.23	1	0.57	0.47	0.39	0.32
ΔB	-0.22	0.57	1	0.8	0.79	0.70
ΔA	-0.23	0.47	0.8	1	0.69	0.79
ΔB_{500ms}	-0.18	0.39	0.79	0.69	1	0.82
ΔA_{500ms}	-0.21	0.32	0.69	0.79	0.82	1

Table 9.4: Correlation matrix for the selected columns for $\phi = 2$ sell experiments

	ΔV	δV_F	ΔB	ΔA	ΔB_{500ms}	ΔA_{500ms}
ΔV	1.00	-0.24	0.25	0.21	0.24	0.18
δV_F	-0.24	1.00	-0.44	-0.51	-0.31	-0.35
ΔB	0.25	-0.44	1.00	0.77	0.79	0.67
ΔA	0.21	-0.51	0.77	1.00	0.72	0.80
ΔB_{500ms}	0.24	-0.31	0.79	0.72	1.00	0.83
ΔA_{500ms}	0.18	-0.35	0.67	0.80	0.83	1.00

Again, there are highly correlated features: the linear models trained will use a PCA with $N = 4$ to decrease the number of features. There is indeed something to note: the time of execution of every strategy makes the allocation choice at the beginning more and more challenging, and the absence of highly correlated features is another factor that makes the choice more difficult. It is clear that being more passive the ΔV forecasting problem is getting more and more complicated.

9.2.3 Proposed models for $\phi = 2$ data

Again, as simple benchamarks, the mean models and the linear regression models are proposed. Being the problem more complicated, a lot of difference structures were trained and tested. Models as the sigmoid neural network presented before, a convolutional neural network [31], a TabNet [25] and a gradient boosting decision tree XGBoost [32]. All the models tested, with different hyperparameters and different features could not clearly outperform the linear regression models. Being XGBoost an easyly trainable decision tree algorithm, it was selected as the model to use for the forecasting problem.

Using the train set, a K-fold cross validation were performed to select the best hyperparameters for the ΔV forecasting problem. For both buy and sell cases and for all the time intervals $T = 0, 1, 2$ the best hyperparameters were the same: learning rate $\eta = 0.01$, number of estimator $n = 400$ and maximum depth $d = 3$. Given this values, is is possible to test the model on the test set:

Table 9.5: Mean squared error metrics comparison between ALGO approach, mean models, linear models and XGBoost models. Buy case, $\phi = 2$.

T	ALGO	Mean models	Lin. models	XGBoost	Δ ALGO/LM	Δ ALGO/XGB
T=0	0.769	0.689	0.636	0.647	-17.2 %	-15.9 %
T=1	0.734	0.716	0.581	0.577	-20.8 %	-21.4 %
T=2	0.675	0.677	0.648	0.643	-4.0 %	-4.8 %

Table 9.6: Mean squared error metrics comparison between ALGO approach, mean models, linear models and XGBoost models. Sell case, $\phi = 1$.

T	ALGO	Mean models	Lin. models	XGBoost	Δ ALGO/LM	Δ ALGO/XGB
T=0	0.774	0.678	0.598	0.592	-22.8 %	-23.5 %
T=1	0.771	0.735	0.622	0.607	-19.3 %	-21.3 %
T=2	0.707	0.707	0.694	0.690	-1.8 %	-2.4 %

Table 9.5 and Table 9.6 show that a complex algorithm as XGBoost and a simple linear model have the extremely similar results doing the same task. Moreover, considering that the simple linear model is not even a good model (the residual are neither gaussian nor heteroschedastic), it is clear that choosing the best allocation *a priori* and without being able to intervene dynamically will not achieve optimal results. This is also proven by the fact that the XGBoost model with only 5 features in Table 9.4 performs better than the model with more than 50 features. Moreover, even the Machine Learning models mentioned above failed to improve a linear model: there is every evidence that in order to get a working model we need to change the current approach to venue selection. In fact, given features at time t_0, ALGO for order allocation requires the model to predict that a price shift in the future will occur sooner in one venue than in the other. If this shift occurs in the immediate future, the model may be effective, but if the strategy is passive and the execution time is longer, this task becomes impossible.

9.3 Intermediate strategy summary

It is now possible to make a summary of the results of the intermediate strategy with $\phi = 2$:

- In the synthetic week a buy/sell difference was noted. This factor could be related to upper and lower trends during the day and the difference to the geographical location of the traders of the two venues. As this phenomenon is very complex to study and related to factors present not only in the FX market, this study will not consider it.
- As the execution time increased, a real week was runned to increase the size of the dataset.
- AMS outperforms RTM in the morning as noted for the other strategies. On the other hand RTM becomes more and more competitive in the early afternoon and for $T = 2$ the performances are perfectly balanced.
- The highly correlated features are again the same of $\phi = 1$ case. Even when using more features for complex Machine Learning models, the result does not improve.
- Numerous Machine Learning models with even high levels of complexity were trained. Neither structures such as convolutional neural networks nor Tabnets have had any noteworthy results. XGBoost is the only model that succeeds overall in improving the simple linear model, but once again the results are not remarkable.

- Given the poor results of such complicated models, it is evident that in order to find the optimal allocation, the approach to the problem must be changed. Given the statistics at time t_0 it is impossible to ask a model to find in which venue a price movement will occur sooner than in the other, considering the fact that the strategy is no longer aggressive, but starting to be more and more passive.

The problem that statistics at time t_0 are no longer useful in predicting the value of ΔV at the final time will certainly be encountered for subsequent strategies. In fact, the more passive the strategy, the longer its duration will be. With this idea, it is now possible to analyse the results of the most passive strategies.

Chapter 10

Synthetic Data Analysis: passive strategies

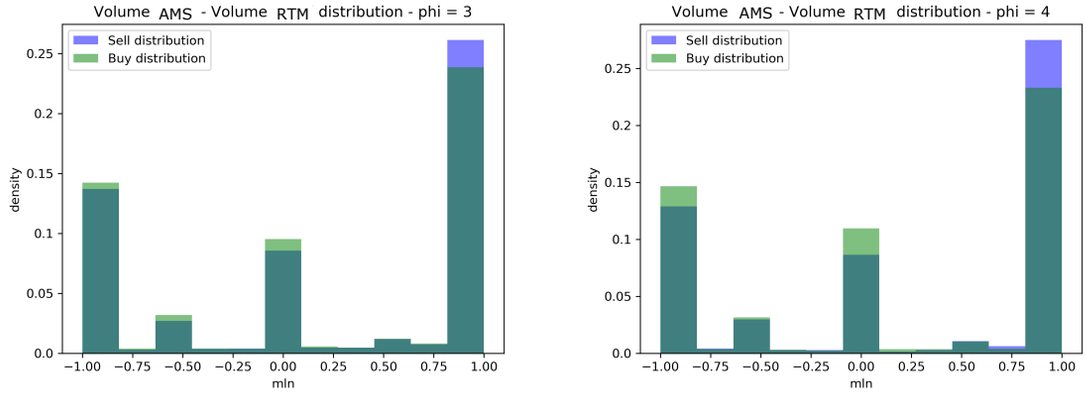
This chapter analyses the two passive strategies, obtained with $\phi = 3$ and $\phi = 4$. Being the strategies passive and indeed with not a lot of executions, the shown results take into consideration both the synthetic week and the historical week (same used for the $\phi = 2$ intermediate case).

10.1 Floating passive strategies with $\phi = 3$, $\phi = 4$

Being the results similar, this section analyses the strategies results obtained with $\phi = 3$ and $\phi = 4$. In total, during the two weeks backtested 4480 buy executions and 4331 sell executions were performed for $\phi = 3$. On the other hand, for $\phi = 4$, only 1266 buy executions and 1242 sell ones. It is evident that compared to the more than 20k executions obtained for $\phi = 2$ in the two weeks these strategies are extremely more passive. Let's start analysing the overall results of ΔV.

10.1.1 ΔV results

As in all the simulations shown before, ΔV has the typical categorical shape. Note that in this case, even the histogram shows some clear difference between buy and sell (Figure 10.1).



(a) $\phi = 3$.

(b) $\phi = 4$.

Figure 10.1: Difference between volumes traded in AMS and RTM for passive strategies.

From the figure above, it is possible to note that for more sell simulations obtained $\Delta V = 1$ with respect to the buy ones. To quantify this difference between the two sides of the orderbook Table 10.1 and Table 10.2 can be studied.

Table 10.1: Mean Values of $\Delta\bar{V}_{buy}$ and $\Delta\bar{V}_{sell}$ for $\phi = 3$.

T	$\Delta\bar{V}_{buy}$	$\Delta\bar{V}_{sell}$
0	0.357	0.328
1	0.126	0.236
2	-0.030	0.048

Table 10.2: Mean Values of $\Delta\bar{V}_{buy}$ and $\Delta\bar{V}_{sell}$ for $\phi = 4$..

T	$\Delta\bar{V}_{buy}$	$\Delta\bar{V}_{sell}$
0	0.348	0.321
1	0.067	0.305
2	-0.024	0.052

Table 10.1 and Table 10.2 for show that in the afternoon the more aggressive the strategy, the more the buy sell difference is accentuated in the analysed sample. For $\phi = 4$ and $T = 1$, $\Delta\bar{V}$ passes from 0.067 in the buy case to 0.305 in the sell case. Moreover, the hourly ΔV distribution confirms that the division $T = 0, 1, 2$ works well.

10.1.2 The limitation of the current approach for passive strategies

Even in this case the most important statistics to be considered are δV_F, ΔB and ΔA. Even if ΔB_{500ms} and ΔA_{500ms} show high correlation with ΔV, these values make worst the performances of the models. The logical reason is that the $500ms$ trend movement of the sides of the orderbook is not useful at all for a prediction of something that is going to happen more than one second later (e.g. the executions of a passive strategy's order). No one of the other features was useful to predict ΔV.

If for $\phi = 2$ it was amply proven that even with complex Machine Learning models it was not possible to predict the dynamics of the two venues, with even more passive strategies it is even more complex. As already explained, the duration of strategies before executions makes it impossible to predict *a priori* where the fastest execution will be. To remark again this concept, the same XGBoost model used for $\phi = 2$ was trained and selected. Starting with $\phi = 3$, Table 10.3 and Table 10.4 show the MSE obtained on the test set.

Table 10.3: Mean squared error metrics comparison between ALGO approach, mean models, linear models and XGBoost models. Buy case, $\phi = 3$.

T	ALGO	Mean models	Lin. models	XGBoost	Δ ALGO/LM	Δ ALGO/XGB
T=0	0.706	0.635	0.501	0.591	-29.1 %	-16.4 %
T=1	0.724	0.693	0.646	0.664	-10.7 %	-8.4 %
T=2	0.647	0.647	0.644	0.650	-0.5 %	0.4 %

Table 10.4: Mean squared error metrics comparison between ALGO approach, mean models, linear models and XGBoost models. Sell case, $\phi = 3$.

T	ALGO	Mean models	Lin. models	XGBoost	Δ ALGO/LM	Δ ALGO/XGB
T=0	0.777	0.627	0.548	0.639	-29.5 %	-17.8 %
T=1	0.759	0.653	0.597	0.597	-21.3 %	-21.3 %
T=2	0.705	0.705	0.697	0.663	-1.2 %	-6.1 %

The first fact to note is that the linear models outperform constantly the complex Machine Learning models. Moreover for $T = 2$ in the buy case even ALGO perform better. The fact that a tree based model cannot understand any relation between the features and the trivial ALGO

selection can outperform it is a clear signal that the *a priori* selection approach does not work at all for passive strategies.

It is essential to note that various intricate Machine Learning architectures have been explored in attempts to address the issue. In pursuit of solving the problem, exhaustive experimentation has been conducted, encompassing elaborate statistical methodologies, such as LSTM neural networks [33] with diverse architectures, and even the application of TabNet [25] in its unsupervised pre-trained version. Even these sophisticated models failed to outperform a rudimentary Linear Model serves as compelling evidence that predicting the outcome of a simulation *a priori* might indeed be an insurmountable challenge.

To conclude this analysis, Table 10.5 and Table 10.6 show the MSE obtained on the test set for the models in the $\phi = 4$. Note that only the Lienar models were trained in this case: there was a lack of data to train any Machine Learning model.

Table 10.5: Mean squared error metrics comparison between ALGO approach, mean models, linear models and XGBoost models. Buy case, $\phi = 4$.

T	ALGO	Mean models	Lin. models	Δ ALGO/LM
T=0	0.765	0.886	0.746	-5.1 %
T=1	0.670	0.657	0.663	-4.2 %
T=2	0.782	0.783	0.729	-10.3 %

Table 10.6: Mean squared error metrics comparison between ALGO approach, mean models, linear models and XGBoost models. Sell case, $\phi = 4$.

T	ALGO	Mean models	Lin. models	Δ ALGO/LM
T=0	0.852	0.646	0.527	-38.1 %
T=1	0.777	0.655	0.621	-20.1 %
T=2	0.711	0.702	0.691	-2.8 %

While the results of the linear models are extremely different for buy and sell, the fact that the errors have a high magnitude. Linear regression models generally perform better than Machine Learning models for passive strategies because they have predictions that on average work better, but they cannot capture complex dynamics. Because they are so simple and follow the logic of predicting what is generally the outcome (without using any kind of category), they are able to achieve extremely competitive results.

10.2 Summary of the results of passive strategies

To conclude the chapter, it is now possible to make a summary of the results of the passive strategies:

- If on one hand AMS performs a way better than RTM during morning hours ($T = 0$), on the other RTM is competitive in the late afternoon ($T = 2$). For $T = 1$ there is a huge difference between the values of $\hat{\Delta V}$, highlighted especially for $\phi = 4$ int Table 10.2. Again, the differences between the executions for buy and sell orders cannot be considered general, but they have to be further investigated with a specific research.
- No one of the selected statistics seem to be useful for the $\hat{\Delta V}$ estimation, with the exceptions of ΔA, ΔB and δV_F.

- The duration of the strategies and the complexity of the FX market makes the ΔV prediction *a priori* impossible. Machine Learning models do not work and the only alternative spotted is a simple regression model.

Having in mind all the results and the limitation of the runned simulations, it is now possible to propose a new approach for the venue selection problem. This approach will be explained in the next chapter and it will defined as the dynamical allocation approach.

Chapter 11

Dynamic allocation approach

All the previous chapters highlighted some limitations of the *a priori* allocation approach. In fact, deciding at the beginning the allocations of the volumes to both venues is not the best approach, especially for passive strategies. The duration of the strategies makes the selection extremely difficult because the future is largely unpredictable (even for complex machine learning models). The aim of this chapter is twofold: firstly, it explains how and why the current ALGO approach has been improved by this research and secondly, it proposes a possible concrete improvement for order execution.

11.1 How this research has improved ALGO

Before proposing a new approach to venue selection, it is important to show how simulation results and data analysis can be used for improving the current method. To recall it, the *a priori* selection consists of dividing a parent order of N mln into two child orders of $w_A N$ and $w_R N$ mln to be sent to AMS and RTM respectively. Note that usually N can have very high values (even 50/100 mln) and, even then, the split is only done once for all, at the beginning.

This study highlighted distinct areas of improvement in ALGO executions. The models chosen were regression models and not classification models for two main reasons:

- Management of class imbalance is complicated: classification models such as XGBoost or logistic regression were applied, but the error on test sets was always greater than the corresponding linear regression models. The models, in fact, suffered heavily from class imbalance and even when assigning class weights they always ended up classifying extreme classes ($\Delta V = 1$ or $\Delta V = -1$) or the middle class ($\Delta V = 0$). In practice, since what happens in the venues is extremely stochastic and the result of the actions of an enormous number of market participants, predicting the class is almost impossible. For this reason, regression models perform better: in fact, they often predict what works on average, avoiding extreme cases.

- ALGO application: for a practical application with $N \geq 2$, models predicting non-extreme values and more around the mean are preferable. If, for example, ALGO has to allocate $N = 20$ million to be executed *a priori* without having the possibility to modify the allocation, less extreme models are preferable, as they would send balanced quantities to both venues. Figure 11.1 show that the predictions obtained by the regression models (in the example for $\phi = 2$ and $T = 1$) are not extreme.

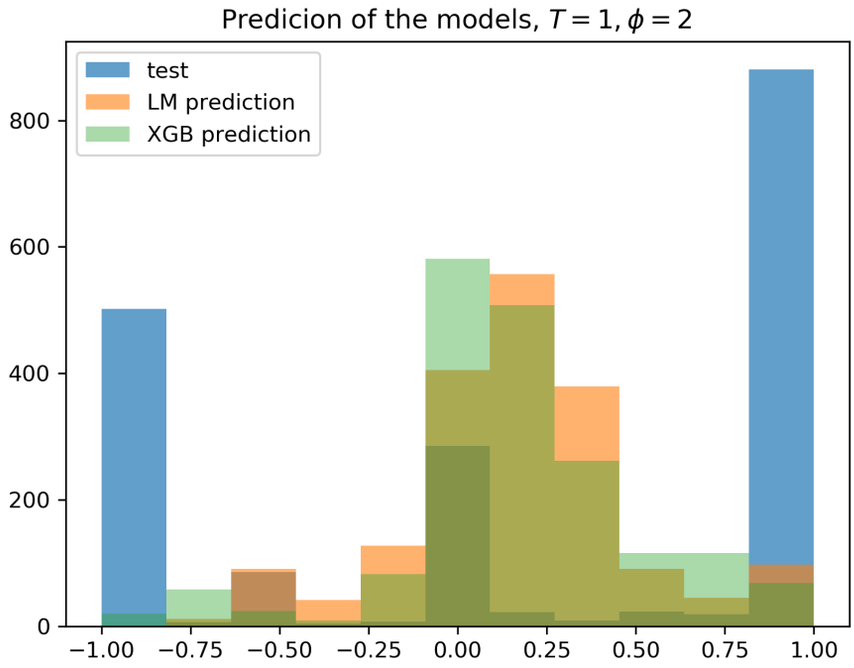


Figure 11.1: Histogram of the predictions of the test set for $T = 1$ and $\phi = 2$. Both the linear regression model and the XGBoost model predict values around the mean.

If the models predicted extreme values, parallel trading would lead to allocating a very large amount to one venue and leaving little or no volume to the other venue. If this choice could make sense for small quantities such as $N = 2$, it makes no sense if one deals with high N values: the allocation of large volumes to a single venue could lead to the loss of execution opportunities on the other. The choice to use $N = 2$ for the simulations is due to the fact that it is extremely simple and allows one to save the largest number of data possible. If, for instance, $N = 10$ had been selected, the data obtained would have been ten times less, making it necessary much more demanding data processing efforts.

Following these clarifications regarding the methodology, we can now outline the areas of possible improvement for the current venue selection algorithm:

- ALGO should consider the intraday seasonality of order executions clearly visible in the available data. Whatever the strategy adopted for venue allocation, the results associated with different periods were remarkably different. In general, from 8:00 to 12:00 (London hour, $T = 0$) AMS achieves better results than RTM, from 12:00 to 15:00 the volatility is higher because both the European and American markets are open and the number of executions increases. Lastly, in late afternoon (from 15:00 to 19:00 London hour), the number of executions decreases significantly and RTM appears more competitive. A first trivial step could use just the mean models, using the mean of the simulations backtested with the real ALGO strategies.

- ALGO should correlate the aggressiveness parameter ϕ to w_A. In fact, simulations clearly show that being AMS a more liquid venue, it performs better than RTM for aggressive orders in every intraday time interval: Very aggressive orders (similar to market orders) are executed much faster in AMS than in RTM.

- For very aggressive orders, the sigmoid model proposed for $\phi = 0$ could be a valuable option. Despite being very simple, it outperforms the linear model on the selected test set.

- ALGO should compute and consider values such as ΔB, ΔA and δV_F. Especially when dealing with small N, values such as those listed could be considered excellent predictors of the best venue allocation.
- While remaining simple and explicable, a linear model could manage venue allocation. This research has indeed shown that a simple linear models could outperform the current approach just considering the value of T and of the selected statistics. Being an extremely simple model, it could be trained often ensuring fast executions and explainability. In this case, it is recommended to use for training at least the week before the day you are trading.

After listing how the current method could be improved, it is time to explain its limitations and propose a new approach.

11.2 Limitation of the *a priori* approach

In addition to highlighting how parallel trading is an excellent strategy to achieve numerous executions using both venues, this research has highlighted some important limitations of the current venue selection method. As already mentioned, the limitations are highlighted by the venue selection for more passive orders. The need to allocate *a priori* the volumes and the impossibility to change the allocation makes the optimal venue selection problem impossible to solve. At time t_0 there is not enough information available to find the optimal allocation. Figure 11.2 shows the current venue allocation strategy.

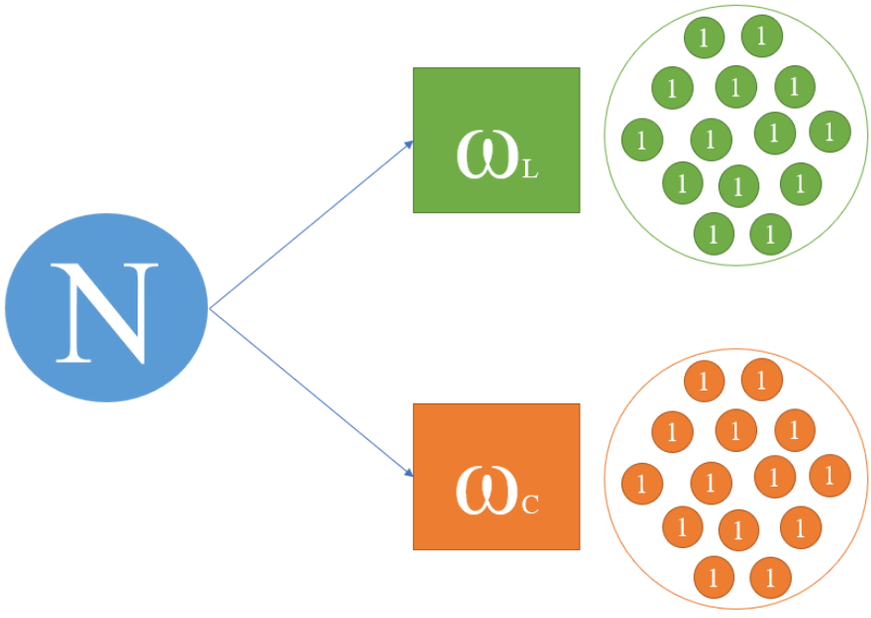


Figure 11.2: ALGO *a priori* venue selection algorithm.

From Figure 11.2 it is clear that the amount of millions to be traded (N) is split at time t_0 into two sets, one for each venues. The sets are successively divided in 1-mln child orders and executed one per time. The allocation decided *a priori* cannot be changed anymore, and this fact is a limitation for the speed of execution. In fact, when the subset allocated in one venue is fully executed, even just the remaining part is traded using just one venue. To prove how much this fact could slow down the execution process, a small experiment has been run.

Let's consider just one day of data, 25/06/2021 from h. 8:00 to h. 19:00. The goal now is to count the completed buy simulations with allocation *a priori* that day. In the next section this

result will be compared with the new proposed approach [...]. Assuming $N = 100$mln, let's assume that AMS and RTM is fifty-fifty. This values are used to make this example [...]. Let's define t_0 as the initial time of the simulation, t_1 as the time in which the fastest venue ends the executions and t_f as the ending time. Being the *a priori* approach, the first part is traded in parallel, but the last one in series. The volume R is indeed the amount of volume traded in series, while the value τ^{series} represents the amount of time in which the simulation has been performed in series. Table 11.1

Table 11.1: Parallel trading weak point: the end of every simulation is performed in series. This table shows the values of R and τ^{series} with one day of simulations (buy case, $\phi = 2$). Simulations with $T = 0$ are highlighted with a red background, those with $T = 1$ with a green background and those with $T = 2$ with a blue background.

t_0	t_1	t_f	Slow Venue	R (mln)	τ^{series}
08:00:00.00	08:51:35.10	09:11:10.80	AMS	10.02	00:19:35.70
09:11:10.80	09:49:24.25	10:14:55.60	RTM	17.62	00:25:31.35
10:14:55.60	11:08:21.45	11:54:26.75	RTM	26.19	00:46:05.30
12:00:00.00	12:31:58.41	12:34:27.56	RTM	17.99	00:02:29.15
12:34:27.56	12:44:01.51	12:53:50.96	AMS	6.92	00:09:49.45
12:53:50.96	13:01:05.31	13:30:41.06	AMS	37.99	00:29:35.75
13:30:41.06	14:00:02.51	14:06:52.46	RTM	10.73	00:06:49.95
14:06:52.46	14:34:10.91	14:49:31.66	RTM	25.17	00:15:20.75
15:00:00.00	15:06:45.50	15:14:22.05	RTM	15.83	00:07:36.55
15:14:22.05	15:53:00.41	16:00:54.85	RTM	8.47	00:07:54.44
16:00:54.85	16:57:57.75	18:01:56.70	AMS	14.28	01:03:58.95

Table above shows how each simulation has quantity R that does not use parallel trading. The volume to be traded in series on the observed day goes from a minimum of 6.92 to a maximum of almost 38 million. Considering that the goal of venue selection is the speed of executions, from this simple example it is evident that the current ALGO method is not optimal. Considering a random day of those of the sample, it can be seen that each simulation with $N = 100$ has a more or less long ending in parallel. This problem could be alleviated by using small values of N to reduce the delay or by changing the current approach to something more dynamic.

11.3 The proposed approach

The new proposed approach is defined as the dynamic allocation approach. Still using parallel trading on both venues, this approach intends to dynamically allocate orders in both venues. From the quantity N a sub-quantity M is subsequently taken (which could for example be equal to 2 million). This second quantity is dynamically allocated between the two venues with a split recalculated every Δt (selected time interval, [...]) based on the statistics of the two venues. Figure 11.3 tries to represent the proposed strategy.

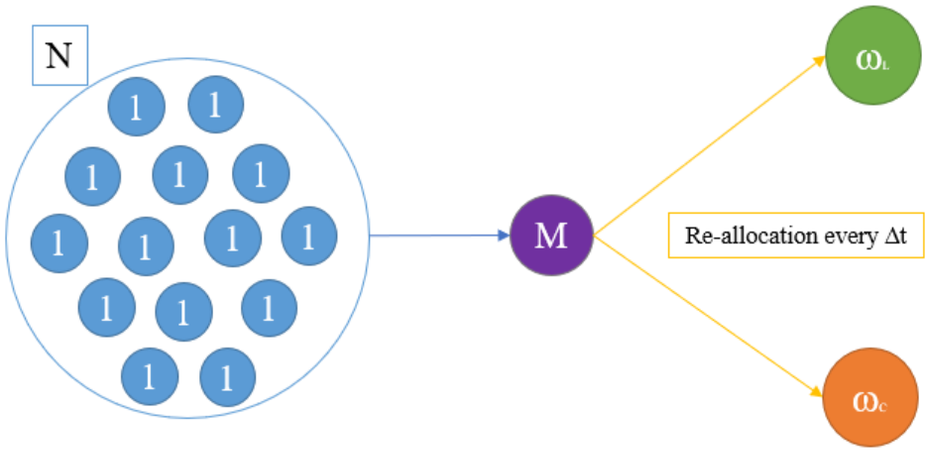


Figure 11.3: Proposed dynamical allocation approach.

Some observations are now in order:

- The value of M is the exposure risk parameter. As M increases, one is more exposed to buying or selling certain quantities on the market. If M is a high value in fact you are trying to buy/sell large quantities of currency. Large market moves could lead to large executions at prices which may subsequently be judged disadvantageous. A potential improvement could involve linking this parameter to the level of strategy aggressiveness minimizing the exposure with passive strategies and increasing it with aggressive strategies. Moreover, this parameter could be increased when the price is considered very fair by traders.

- In case of partial execution in one of the venues of an amount ϵ in an interval Δt, the amount to be split among the venues must be the same M. In other words, after each Δt the quantity to be allocated must always be M (unless the volumes to be executed are finished).

- The statistics to select the best venue must be saved and updated every Δt to have a constantly optimal and updated allocation. In this case, the values of w_A and w_R are indeed computed dynamically.

- The choice of w_A and w_R must take into account the common volumes of the orders present in the venues. Figure 5.6 clearly shows that probably a good choice would be to choose multiple splits of 500k.

- The value of Δt is the time interval parameter. [...]

- The price selection could be integrated in this logic. For example, imagine that from historical data two models were trained one more aggressive (model A) and one more passive (model B). Let's assume that the strategy is running at time t the models are estimating a two probabilities of executions in AMS p_A^{AMS} and p_B^{AMS}. If p_B^{AMS} is large, the model is predicting that there is likely to be an execution even at a less aggressive level. The quantity allocated in AMS could then be re-divided into two sub-groups with two different prices in order to try and exploit a more advantageous price.

This whole model could be very complex to develop and bring into production. If, however, points like the last one (the price selection) are complex, the simple concept of not having any *a priori* splitting of quantities between venues could be very simple and bring excellent improvements.

After explaining the main rules of the proposed new approach, a simple proof of concept will be tested to conclude this thesis. The results of this experiment will be useful for future researches.

11.4 Proof of concept

This section tries to show the potential of the proposed approach, explaining a simple proof of concept. The goal of this section is to show the potential of the idea, working in as simple a framework as possible. Being a simple case, just one day was considered: Friday 25/06/2021. Before the results analysis, it is important to explain the rule of algorithm.

11.4.1 Selected rules

Let's define the main parameters of the first trivial dynamical allocation model. As already mentioned, being a simple proof of concept, the selected rules are extremely trivial:

- $M = 2$: this choice is coherent with the current ALGO floating strategy.
- $\Delta t = 50ms$.
- Buy simulations.
- $\phi = 2$: being an intermediate case between an aggressive and a passive situation, it was considered optimal to use the intermediate strategy as a proof of concept.
- $w_A = w_R = 0.5$. There is not any model that computes the best venue allocation every Δt, but the exposure is constantly 1mln in AMS and 1mln in RTM.

the main objective of this approach is just to avoid the series trading part and use just the parallel one. The algorithm allocates dynamically 1 mln for AMS and 1 mln for RTM, without any forecasting model for ΔV. The results are indeed strong.

11.4.2 One day results

The benchmark for the results is Table 11.1. It's possible to observe that during the morning hours ($T = 0$) three $N = 100$mln simulations ended from h 8:00 to h. 11:54. In total, during the morning, these three simulations executed 300 mln. Using the selected trivial strategy, using the same day, the quantity increases to 400 mln. Table 11.2 shows the starting and the ending time of the four simulations performed in the morning.

Table 11.2: Proof of concept $N = 100$ mln executions ($T = 0$, buy case, $\phi = 2$).

t_0	t_f
08:00:00.00	09:05:14.50
09:05:14.50	09:51:46.90
09:51:46.90	10:38:32.60
10:38:32.60	11:48:29.30

During the morning, the simple strategies that avoid serial trading improve executions by 33% (from 300 mln to 400 mln). This excellent result is also replicated for $T = 1$, as Table 11.3 shows.

Table 11.3: Proof of concept $N = 100$ mln executions ($T = 1$, buy case, $\phi = 2$)..

t_0	t_f
12:00:00.00	12:32:39.86
12:32:39.86	12:41:16.21
12:41:16.21	13:01:02.71
13:01:02.71	13:09:54.51
13:09:54.51	13:46:41.26
13:46:41.26	14:18:02.26
14:18:02.26	14:51:45.51

Between 12:00 and 15:00 hours, a total of seven distinct $N = 100$ mln strategies concluded their execution. When we compare this outcome of 700 mln executed units with the data in Table 11.1, we observe an increase of 200 mln. This remarkable result, combined with that obtained by $T = 0$, suggests that avoiding serial trading could significantly speed up the order executions. To conclude, Table 11.4 shows that just three $N = 100$ mln simulations ended in the selected day.

Table 11.4: Proof of concept $N = 100$ mln executions ($T = 2$, buy case, $\phi = 2$).

t_0	t_f
15:00:00.00	15:10:09.30
15:10:09.30	15:47:08.15
15:47:08.15	16:47:17.20

The low volumes in both venues in the afternoon do not allow for a large number of performances and do not show a any significant improvement. In fact, both conclude three $N = 100$ simulations, the difference being that ALGO ends at 18:01 and the strategy at 16:47.

11.5 Summary of the new approach

The proposed dynamic allocation model could be a significant improvement to the venue selection problem. It is designed to remedy the *a priori* choice problem by proposing a dynamic allocation that continuously feeds both venue trading always in parallel. Using standard values of max exposure ($M = 2$ mln), aggressiveness ($\phi = 2$), total quantity ($N = 100$ mln) and fixing $w_A = w_R = 0.5$, one day was tested for buy simulations (25/06/2021).

The results of this simple experiment show a remarkable improvement: in total, from 11 simulations of ALGO, 14 simulations are performed and ended with the new approach. The improvement is larger for $T = 0$ (from 3 simulations to 4) and $T = 1$ (from 5 to 7), while for $T = 2$ the completed executions are the same number (3). This clear result obtained for just one day may suggest that a dynamic approach could significantly speed up the execution.

Chapter 12

Conclusions

In conclusion, the research question revolved around the FX trading Venue Selection problem and aimed to determine the optimal order allocation strategy that would enable simultaneous utilization of multiple venues to enhance execution speed. Since the published literature useful for this problem was not directly useful and the in-house study proved to be non-functional, the study had to find an alternative way. With the construction of a trading engine, the thesis backtested the strategies in both venues and compared them without any bias. The selected strategies were a simplified copy of the real ALGO Floating strategy, with a price selection based on the ϕ parameter that describes the level of passiveness. The results were analysed by distinguishing each strategy according to their aggressiveness. New models were proposed for each case, emphasising the importance of the main statistics used by the models. Finally, the limitations of the current ALGO approach were highlighted and a new dynamic approach was proposed that could be used in the future.

The first important conclusion is that the designed backtesting approach works. Simulation results obtained using data from 2021 are matched by real ALGO trading data from the last period. Although very computationally intensive, the possibility of reconstructing the orderbook at each time instant in the past and seeing results of the same strategies at the two venues makes it possible to compare and analyse their behaviour. The huge codebase built will also be made available to MN for future studies.

Moreover, this research highlighted a clear intraday seasonality. It divides the day in three main periods: $T = 0$ from 8:00 to 12:00 (London hour), $T = 1$ from 12:00 to 15:00 and $T = 2$ from 15:00 to 19:00. Checking the synthetic data created, AMS shows better results than RTM the first period, while RTM becomes more and more competitive in the afternoon, especially during $T = 2$. The reasons for this are attributed to the US market opening in the middle of the day, which brings large volumes to both venues. The data show that the algorithms for selecting venues should vary depending on the trading time: this information could be used in the next steps of ALGO development.

In general, it is clear from the synthetic data generated that for aggressive strategies AMS guarantees faster executions than RTM. This fact is shown by the results of the simulations obtained with $\phi = 0$ and $\phi = 1$. The liquidity and the fact that more people are trading make AMS better. This result was confirmed by the technical report based on real trading data of ALGO. For $\phi = 0$ simulations, the most just one statistic was detected as useful to forecast the result of the simulations: ΔB for the sell simulations, ΔA for the buy ones. The explanation is trivial: very aggressive orders that sometimes are market orders care just of the other side of the orderbook. On the other hand, for $\phi = 1$ strategies the important statistics were five: ΔB, ΔA, ΔB_{500ms}, ΔA_{500ms} and δV_F. In this case, the order is not always the first in line and the venue selection cares about both the best bid and the best ask (with their movement in the last half second) and the volume in front of the order. For both $\phi = 0$ and $\phi = 1$ the models proposed had good result trying to forecast the value of ΔV at the end of each simulation. For $\phi = 0$ a simple sigmoid model was proposed, for $\phi = 1$ a simple neural network. In this second case, because of the explicability

problems, a linear model could also be used.

Contrary to what has been observed for more aggressive strategies, the intermediate strategy ($\phi = 2$) uses a larger dataset: the synthetic week and a historical week. The highly correlated features remained the same as in the $\phi = 1$ case. An observation was made in the synthetic week indicating a buy/sell difference possibly linked to daily trends and geographical location of traders. Due to its complexity and broader market factors, this study won't delve into this phenomenon. Incorporating more features in complex machine learning models did not yield improved results. Again, AMS demonstrated morning superiority over RTM, consistent with other strategies. However, RTM showed increased competitiveness in the early afternoon, with balanced performance for $T = 2$. XGBoost is the only model that succeeds overall in improving the simple linear model, but the results are not noteworthy.

Similarly, in the intermediate cases involving passive strategies with $\phi = 3$ and $\phi = 4$, the extended dataset was employed. The outcomes underscore that *a priori* allocation for passive strategies is not conducive to optimal results. Evidently, as the price becomes more passive, the corresponding strategies endure for longer durations. For all the models used it was impossible to find a good prediction on the result of the strategy given the statistics at the beginning. Consequently, in such scenarios, ALGO's venue selection current approach should just rely on intraday seasonality, which was once again clear from the backtested data. It is also interesting to note that in more passive cases with longer strategies, more random factors come into play. For this reason, a choice such as fifty-fifty becomes more competitive when compared to models that are fully capable of predicting the output of simulations.

To conclude, the current ALGO venue selection method has many points for improvement. [...]. In fact, ALGO should increase the volumes associated with AMS in the case of more aggressive orders, as shown by the results of the strategies. Furthermore, the clear intraday seasonality in the trading data should be exploited by adjusting allocations according to the time of day: AMS has been shown to guarantee more executions in the morning than RTM, which becomes more competitive in the afternoon. If ALGO wanted to use other statistics, this study showed volume values at the front of the order δV_F, difference of best bids ΔB and asks ΔA are related with the results of the executions.

The main limitation highlighted found by this thesis in the *a priori* choice of allocations, however, is the fact that it is really complex to choose how to divide a large amount of mln at the beginning based on the statistics at that time (especially with more passive strategies). For this reason, this study proposes a new way to face the venue selection called the dynamic allocation approach. It would allow to act dynamically on the venues, changing the allocations every Δt. Furthermore, this method would eliminate the problem of trading in series that ALGO performs when a venue has executed all orders. As the proof of concept shows, in fact, this slows down executions.

Exploring avenues for further research reveals several promising directions that could enhance the understanding and applicability of the current study's findings. In general, parameters such as sleeping time Δt and elapsed time before an order is executed could be investigated (possibly with survival analysis strategies). Furthermore, if the idea of dynamic allocation were to be explored, parameters such as the M exposure or allocation models for w_A, w_R could be investigated.

References

[1] S. Jansen, *Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python*. Packt Publishing Ltd, 2020.

[2] L. Smigel, *79+ Amazing Algorithmic Trading Statistics (2023)*, 2022. [Online]. Available: https://analyzingalpha.com/algorithmic-trading-statistics, [Last accessed on 30.06.2022].

[3] Triennal Central Bank Survey, "Foreign exchange turnover in april 2019," Bank for International Settlements, Tech. Rep., 2019.

[4] Á. Cartea, S. Jaimungal, and J. Penalva, *Algorithmic and high-frequency trading*. Cambridge University Press, 2015.

[5] L. Xuan, "A general framework for modelling limit order book dynamics," Ph.D. dissertation, Imperial College London, 2022.

[6] R. C. Smith, "Algorithmic FX trading: Optimizing the venue selection mechanism," Dept. Economics and Finance, M.S. thesis, VU Amsterdam, MN, Amsterdam, 2022.

[7] A. Hayes, *Electronic Communication Network (ECN): Definition and examples*, 2022. [Online]. Available: https://www.investopedia.com/terms/e/ecn.asp, [Last accessed on 07.07.2022].

[8] *Central limit order book*, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Central_limit_order_book, [Last accessed on 26.02.2022].

[9] A. Bloomenthal, *Market maker definition: What it means and how they make money*, 2021. [Online]. Available: https://www.investopedia.com/terms/m/marketmaker.asp, [Last accessed on 31.08.2021].

[10] *Market Makers vs. Market Takers*, 2020. [Online]. Available: https://www.cmegroup.com/education/courses/trading-and-analysis/market-makers-vs-market-takers.html, [Last accessed on 31.08.2021].

[11] K. Baak, "Mathematical documentation ALGO project," MN ALGO internal wiki (unpublished), 2023.

[12] H. Harutyunyan, "Parallel trading," MN ALGO internal wiki (unpublished), 2022.

[13] M. Kearns and Y. Nevmyvaka, "Machine learning for market microstructure and high frequency trading," *High Frequency Trading: New Realities for Traders, Markets, and Regulators*, 2013.

[14] T. Nagler, "Introduction to time series," University Lecture - TU Delft, 2022.

[15] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer science & business media, 2009, pp. 11–12.

[16] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.

[17] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.

[18] M. F. Dixon, I. Halperin, and P. Bilokon, *Machine learning in finance*. Springer, 2020.

[19] P. K. Heinlein A, "Linear algebra and optimization for machine learning," University Lecture - TU Delft, 2023.

[20] S. Suthaharan and S. Suthaharan, "Decision tree learning," *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, pp. 237–269, 2016.

[21] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[22] A. Sharma, *Random Forest vs Decision Tree | Which is right for you?* 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/?social=google&next=https%3A%2F%2Fwww.analyticsvidhya.com%2Fblog%2F2020%2F05%2Fdecision-tree-vs-random-forest-algorithm%2F.

[23] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[24] X. Du, Y. Cai, S. Wang, and L. Zhang, "Overview of deep learning," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, 2016, pp. 159–164.

[25] S. Ö. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 6679–6687.

[26] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *International workshop on artificial neural networks*, Springer, 1995, pp. 195–201.

[27] Wikipedia, *Sigmoid function*, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Sigmoid_function.

[28] M. Matteucci, "Neural networks training and overfitting," University Lecture - Politecnico di Milano, 2021/2022. [Online]. Available: http://chrome.ws.dei.polimi.it/index.php?title=Artificial_Neural_Networks_and_Deep_Learning.

[29] D. Berrar *et al.*, *Cross-validation.* 2019.

[30] F. Merlos, *Python Matching Engine*, 2019. [Online]. Available: https://github.com/Surbeivol/PythonMatchingEngine.

[31] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[32] T. Chen, T. He, M. Benesty, *et al.*, "Xgboost: Extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.