A Novel Optimal Execution Strategy

Using data-driven methods and stochastic modeling, with application in the FX Spot Market

by

Luca Fornaro

to obtain the degree of Master of Science in Applied Mathematics
at the Delft University of Technology,

Student number:	5754267		
Project duration:	January 8, 2024 – August 27, 2024		
Thesis committee:	Dr. F. Yu,	TU Delft,	daily supervisor
	Prof. dr. ir. G. Jongbloed,	TU Delft,	responsible professor
	E. Hazeveld	MN,	external supervisor

Abstract

This thesis presents a novel approach to optimize execution strategies in the Foreign Exchange Spot Market, focusing on the application of data-driven methodologies and stochastic modeling. It begins by proposing a new measure to evaluate the limit order book volume imbalance, which considers multiple price levels and their weighted impact on market predictions. Then the research employs a combination of convolutional neural networks and long short-term memory models to analyze the limit order book data and its imbalances, i.e. the previously introduced limit order book volume imbalance and the order flow imbalance, which takes into consideration their time evolution. This methodology allows for the exploration of price movements and the identification of optimal execution strategies by predicting future mid-price movements, which are embedded in the drift term of the stochastic model. The findings indicate that the developed models outperform traditional strategies by adapting more effectively to market dynamics. Further, this work develops a backtesting environment that simulates market conditions to empirically validate the effectiveness of the proposed strategies against historical data. The results demonstrate a superior performance in terms of profitability and cost-efficiency compared to traditional models that do not utilize the drift term from predictive analytics in their stochastic model of the mid-price. Future work may extend these methodologies to other financial markets and explore prediction models that include the influence of multiple assets.

Keywords: FX Spot Market, Algorithmic Trading, Stochastic Modeling, Deep Learning, Optimal Execution Strategy.

Acknowledgements

I would like to extend my heartfelt thanks to everyone who has significantly contributed to my academic journey and the completion of this thesis. My time at TU Delft has been profoundly enriching, providing both immense learning and substantial personal growth. I would also like to thanks all those who have been an important part of my life over these last two years, a period that has taught me immensely and changed my life.

First and foremost, I would like to express my gratitude to my supervisors, Dr. Fenghui Yu and Erwin Hazeveld. Your guidance was extremely important and constructive; you helped me navigate both the mathematical aspects of my thesis and its business and practical applications. Thanks to you, I have learned more than I could have imagined, and this thesis could not have reached such a level of excellence without your support. I also thank Prof. Geurt Jongbloed for being part of the thesis committee. Additionally, I would like to extend my gratitude to MN for not only providing all the necessary resources for this thesis, including data, but also for the support throughout this project. I am also thankful to all my colleagues for your assistance and for the opportunity to collaborate on this project.

I deeply thank my family - Cristina, Vincenzo, Paolo, Daniele, and Camilla - who are the pillars of my life. Thank you for your unwavering support in every decision I make, and for always being there, even from a distance, during my journey. Your support and love have allowed me to live this period with serenity and satisfaction, knowing that I have a solid family that is always ready to assist me, listen to me and help me, even from afar.

I also want to express my appreciation to all my friends in Italy who have always been present and have continued to be a part of my life despite being apart: Luca, Marco, Antonio, Anna, Daniela, Lisa, Sara, Ilaria, Chiara, Davide, Martina, Elisa, Colosso, Niccoló, Riccardo, Carlo, Pietro, Valentina, Valeria, Aurora, Simone, and Camilla. Your enduring friendship have been a constant source of motivation and inspiration.

Lastly, I must express enormous gratitude to my friends in Delft. You have been my fellow travellers on this adventure, which we somehow faced together. You have been my family in the Netherlands, making me feel at home even though I was far

from Italy. Thanks Alessandro, Francesca, Gaia, Pietro, Laura, Stefano, Alessio, Sofia, Barbara, Carlo, Joris, Juliette, Giacomo, Mine, Riccardo, Isabella, Marco, Filippo, Gianluca, Juan, Sebastian, Gerasimos, Tobias, Francesco, Fernando, Albert, Gideon, Panos, Giorgia, Tomás, Matthias, Albin, Songyan, Luca, Andrea, Tommaso, Can, Vittorio, Vrinda, Clementine, Emma, Amanda. Thank you for all the moments we have shared, in one way or another. Your friendship and support have been essential in helping me reach where I am today and in making these two years an unforgettable and enriching experience.

Delft, 27 August 2024
Luca Fornaro

Contents

List of Figures

List of Tables

1 Introduction

Algorithmic trading represents a significant advancement in the financial markets, utilizing complex algorithms to execute trades at high speeds and with great efficiency. This method uses advanced mathematical models and vast computational resources, enabling traders to execute large volumes of orders at very fast speeds. Such capabilities are very important in modern finance, as they help to minimize costs, maximize efficiency, and improve the overall timing and precision of trades.

Due to its versatility, algorithmic trading can be applied across various markets, including the Foreign Exchange (FX) market. The FX market is the global marketplace for trading national currencies against one another and is the world's largest financial market, with daily turnovers of approximately $7.5 trillion [3]. It operates 24 hours a day across various international time zones, facilitating continuous trading activities. Unlike other financial markets, FX has no centralized physical location; trading is conducted mainly electronically over-the-counter (OTC), which means that all transactions occur via computer networks among traders globally. Trades in this market are settled "on the spot".

Many companies engage in FX trading not only for speculative investment but also for essential risk management and operational purposes. For example, MN, the company collaborating in the research for this thesis, is a leading Dutch asset manager for pension funds which manages assets worth 147 billion euros ([23]) and invests internationally, necessitating currency conversions into euros, U.S. dollars, among others. These trades typically occur in the FX spot market and are often facilitated by banks which charge a fee. To reduce costs and gain more control over trades, MN has developed ALGO, an automated system that executes trades algorithmically and autonomously on the FX spot market, enhancing its competitive edge and prestige.

The purpose of this thesis is to find an optimal execution strategy by predicting the movement of the mid-price of an asset based on imbalances in the limit order book. The theoretical strategy will be tested on real-world data, the FX Spot Market data, on the exchange euro-dollar (EUR/USD). This strategy will ultimately be used by MN to optimize the execution of ALGO's trades. This objective and its underlying methodology will be explained in more detail in the subsequent sections. This chapter is structured

in the following way: first, Sections 1.1, 1.2 and 1.3 will introduce fundamental concepts necessary for understanding the topic of the thesis and its context. Then, in Section 1.4 the purpose of this thesis will be clearly defined, followed by Section 1.5 that analyzes various approaches in the literature. Finally, the major contributions of this thesis will be presented in Section 1.6, followed by an overview of the thesis structure in Section 1.7.

1.1 Algorithmic Execution and Algorithmic Trading

With the development of increasingly advanced technologies, the financial sector is witnessing a growing prevalence of algorithmic trading. This form of trading utilizes computational algorithms and mathematical models to execute orders. These algorithms operate at high speeds, making decisions based on various factors such as volatility and market conditions. They enable rapid and quantitative decision-making without relying on human capabilities and judgments. Nowadays, these algorithms are employed by various market participants, including individuals, companies, and institutions. The Dutch Authority for Financial Markets (AFM) distinguishes between two key terms: algorithmic trading and algorithmic execution [16].

Algorithmic trading refers to the use of computer algorithms to automate the entire trading process. These algorithms analyze vast amounts of market data to identify trading opportunities, make decisions, and execute trades based on predefined rules and parameters. The primary goal of algorithmic trading is to enhance trading efficiency, improve profitability, and manage risk by removing human emotion and bias from trading decisions. This form of trading encompasses a wide range of strategies, including trend following, statistical arbitrage, market making, and momentum trading, each designed to exploit specific market conditions and inefficiencies.

Algorithmic execution, on the other hand, focuses on the optimal execution of large orders. It is particularly important for institutional investors and large traders who need to buy or sell substantial quantities of assets without causing significant price disruptions. Execution algorithms are designed to break down large orders, the parent orders, into smaller, strategically timed trades, called child orders, that can be executed over a specified period or based on certain market conditions. The primary objective of algorithmic execution is to achieve the best possible execution price while minimizing slippage and market impact. Common execution algorithms include VWAP (Volume Weighted Average Price), TWAP (Time Weighted Average Price)[1], and implementation

[1]The Time-Weighted Average Price (TWAP) strategy is a trading algorithm used to execute large orders over a specified time period. It aims to minimize market impact by breaking down large orders

shortfall strategies.

The focus of our research will be on algorithmic execution of child orders, since we are not interested in exploiting arbitrage opportunities or engaging in high-frequency trading. This is also the objective of MN.

1.2 The Limit Order Book

We mentioned that we will use the limit order book. This is central to our thesis, and in this section we are going to explain what it is and how it works. The Limit Order Book (LOB) is a system used by trading venues for organizing buy and sell orders of a particular asset from participants in a market. The LOB displays orders with prices and quantities: orders to buy are called bids and orders to sell are called asks. This system plays a crucial role in the functioning of electronic financial markets, providing a transparent mechanism for price discovery. Within it, there are two distinct types of orders that an agent can choose to adopt: limit orders and market orders.

Definition 1.2.1 (Limit Order)**.** Limit orders (LOs) are instructions to buy or sell a specific quantity of an asset at a specified price or better. They are added to the LOB and executed when the market price reaches the order price, offering traders control over the price at which they trade. Limit orders, or resting orders, are GTC (Good 'Til Canceled), meaning that are a type of order that remains active until either the order is canceled or filled.

Definition 1.2.2 (Market Order)**.** Market orders (MOs) are instructions to buy or sell an asset immediately at the best available current price. They do not specify a price and are executed against the best available limit orders on the opposite side of the book. Market orders are IOC orders (Immediate Or Cancel orders), meaning that they will execute all or part of the order immediately, and then any part unfilled ill be cancelled.

The primary difference between limit and market orders lies in the execution and price. Limit orders control the price but may not execute immediately, while market orders execute immediately but do not control the price, potentially leading to significant price slippage in fast-moving markets. It is important to note that, in general, market orders are considered aggressive orders because they are executed immediately and consume liquidity. In contrast, limit orders are typically more passive and provide liquidity to the market. When choosing to use a limit order, there is no certainty that it will be executed, and if it is, the timing of the execution is unknown. Therefore,

into several smaller ones, each priced near the TWAP.

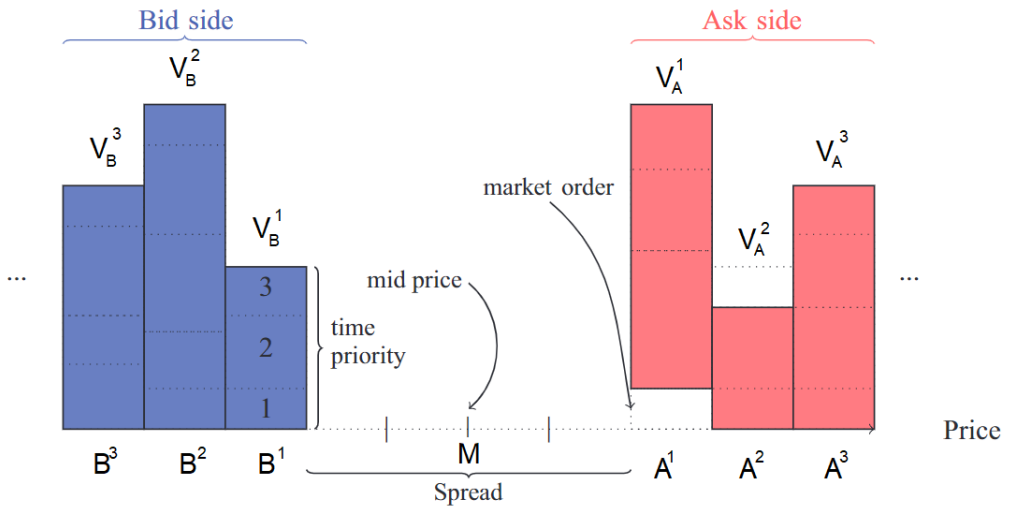


Figure 1.1: Limit Order Book representation. The price levels are denoted as B^i and A^i and their corresponding volume is represented by V_B^i and V_A^i. [from [21], p.369]

using a limit order requires a careful trade-off between the execution time and the best possible price that can be obtained.

For instance, if we consider a sell limit order, the more aggressive it will be, i.e. the closer it is to the best bid, the higher the probability that it will be executed. However, this also implies that it will be executed at a less favorable price. On the other hand, if the limit order is more passive, positioned deeper within the limit order book, it may yield higher profits if executed, but the probability of execution decreases. Indeed, market orders are always matched with limit orders at the best possible price. This consideration is very important for our thesis.

As we can understand, the limit order book is the collection of all outstanding limit orders at a given moment. All the buy limit orders are collected on the bid side, while the sell limit order are collected on the ask side. However, this limit order book is in constant change: it changes whenever a new order is added, or an existing order is canceled or modified. These changes can occur within just a few milliseconds. A representation of the limit order book is provided in Figure 1.1.

In Figure 1.1 B^1 and A^1 denote the Best Bid and Best Ask, defined as:

Definition 1.2.3 (Best Bid)**.** The Best Bid B^1 in a limit order book is the highest price for which a buy limit order is submitted.

Definition 1.2.4 (Best Ask)**.** The Best Ask A^1 in a limit order book is the lowest price for which a sell limit order is submitted

In the limit order book, there will be different prices that ask limit orders (and respectively buy limit orders) can have. These are called *levels*, and the first level corresponds to the lowest (and respectively, highest) price, which is the one closest to the mid-price (so, the Best Ask and Best Bid). Each price further away is one more level.

Definition 1.2.5 (Ask and Bid Price at level l)**.** The Ask (or Bid) price at level l, denoted by A^l (or B^l) is the price of the ask (or bid) limit orders at level l.

The difference between the Best Ask and the Best Bid is called Spread S:

Definition 1.2.6 (Spread)**.** The spread S in a limit order book is the difference between the Best Ask and the Best Bid ($S = A^1 - B^1$). It is an indicator of the liquidity and volatility of the asset; a narrower spread often indicates high liquidity and low volatility.

Moreover, the average of the Best Ask and Best Bid is called Mid-Price:

Definition 1.2.7 (Mid-price)**.** The mid-price M is calculated as the average of the Best Bid and the Best Ask ($M = \frac{A^1+B^1}{2}$). It represents a theoretical price of the asset if one were to average out the highest price someone is willing to pay and the lowest price someone is willing to sell.

Several methods exist for computing this quantity, such as incorporating multiple levels of the LOB or calculating a weighted average of these levels. We selected this particular method due to its simplicity and prevalent usage in the literature [9].

The smallest increment that a price can have is called tick size. In the case of FX Spot Market for the EUR/USD currency rate, this corresponds to 0.00001, i.e. 10^{-5}.

1.3 Trading Venue

Most of the currency exchanges (or in general, trades) take place in specific systems known as trading venues. The European Securities and Markets Authority (ESMA) describes a trading venue as a system where multiple third-party interests in buying and selling financial instruments converge [15]. This system is managed by either an investment firm or a market operator. Essentially, a trading venue is a place where purchase and sale orders for a financial instrument find their match.

Trading venues play a crucial role as they offer a centralized place where trades can be executed in a (hopefully) fair and efficient way, bringing together buyers and sellers. Key characteristics of a trading venue include its trading regulations, cost structure, transparency, and accessibility. Trading regulations outline the procedures for executing trades on the platform, encompassing order types, execution methods,

and transparency before and after the trade. The trading venue imposes fees for using the platform, which may consist of transaction fees, listing fees, and membership fees. Transparency is a significant aspect, ensuring that market participants have access to pertinent market information, such as trade prices and volumes. Lastly, accessibility pertains to the ease with which various types of market participants can access the trading venue, influencing the market's liquidity and efficiency.

Thanks to MN we have been provided with the limit order book data from two different venues (that we will denote as *Venue1* and *Venue2*) of the FX Spot Market. This data will be used to conduct our research.

1.4 Problem Description and Main Concepts

Now that we have introduced all the fundamental concepts necessary to understand the research question of this thesis, we will now explain it in more detail. The aim of this thesis is to find an optimal execution strategy. This will be used by MN, which has developed, as previously mentioned, ALGO, an automatic execution algorithm for the FX Spot market. The strategy will be found primarily by predicting how the exchange price evolves based on the imbalance information of the limit order book, and based on that, optimizing the execution. In particular, we will also focus on the aggressiveness of the order. To understand the idea of this thesis more intuitively, we present an example: if, for instance, we want to post a sell limit order, and we know based on the forecasts that the mid-price will rise, then we want our order to be more passive, i.e., with a higher price, in order to wait for the mid-price to rise and come in our favor. If, on the other hand, based on the forecasts we expect the mid-price to fall, then we want our limit order to be more aggressive (and therefore submitted with a lower price), in order to execute the order as soon as possible and avoid the mid-price moving away from us, causing a loss of money.

We will focus on EUR/USD as an example, as the methodology is completely generalizable. Moreover, given that we have data from two different venues, we will also investigate whether these venues have a reciprocal predictive effect. In other words, we will study if it's possible to use the limit order book data from one venue to make more accurate predictions about the mid-price movement of the other.

An important factor to consider is the time required for the strategy to identify the optimal execution strategy. Given that we are operating within a financial market, it is important that the strategy is capable of quickly determining the optimal price. This rapid response is essential to prevent market changes that could make previously information obsolete, thus potentially diminishing the strategy's effectiveness.

1.5 Literature Review

Our research problem is divided into two distinct phases. The first phase involves studying predictive models capable of forecasting the movement of the mid-price, specifically attempting to forecast whether, in a future interval, the mid-price will increase or decrease, and by what margin. The second phase addresses the problem of finding an optimal execution strategy.

1.5.1 Predictive Models

Many models have been proposed to predict future movements of the mid-price in order to quantify returns. Many of these methods consider data from the limit order book, and in particular, attempt to use the imbalances from these limit order books to make predictions.

Imbalances in the LOB can be measured in various ways. [6] measure it as the difference in volumes between ask and bid, focusing only on the first level. This is then stochastically modeled and incorporated into the overall model of mid-price movement dynamics. Other research, like [11], [5], [7] or [20], focuses not only on the momentary difference in volumes but also on its temporal evolution, thereby considering the Order Flow Imbalance (OFI). In our research, we will use both modalities and compare them. In addition, we will also include the use of multiple levels, as also done by [11], where multiple levels of the limit order book are taken into account and aggregated together through the use of Principal Component Analysis (PCA) (PCA is a statistical technique that reduces the dimensionality of data by identifying the principal components that capture the most variance in the data (Appendix A)), or as [31], where the OFI for multiple levels is also defined here.

Once we have defined how to measure LOB imbalances, many researches have tried to use them to predict mid-price movement, i.e. the impact that the OFI or volume imbalances have on the mid-price. A significant research is [11], which examines the predictive capabilities of OFIs in forecasting future movements of the mid-price in equity markets. Their innovative approach integrates OFIs from multiple levels of the LOB into a single composite variable using PCA, termed the integrated OFI. This methodology provides a more accurate understanding of mid-price dynamics than models relying solely on top-level data, as proved by [11]. The authors study mid-price and cross-impact models, which assess how the order flow in one asset influence mid-prices in other assets within the same market. Structurally, their model resembles a regression framework where the dependent variable, the mid-price movement, is modeled as a function of both own-asset and cross-asset OFIs. This approach allows them to evaluate the contemporaneous and predictive impacts of order flows, comparing models where only the OFIs of one asset are considered, with models where multiple assets are in-

cluded. We will draw inspiration from their models, but instead of applying them across different assets, we will apply them to the order flow imbalances of limit order books from different venues within the same market (the FX Spot market). Additionally, we will extend their models by incorporating a new measure of volume imbalance.

In a very similar way, [5] also attempts to utilize order flow in their predictions. As [11], it incorporates OFI in the prediction of mid-price movement through a regression, and it studies the cross effect given by the influence of other assets. However, it does not aggregate multiple levels of the LOB together, unlike [11]. Both these researches explore the concept of cross-impact models within their framework. These models examine how order flows in one asset could potentially affect mid-prices in another, a hypothesis they test by including cross-impact terms in their regression model. The findings of [5] suggest that while the OFI significantly enhances model accuracy for predicting mid-price movements, the inclusion of cross-impact terms does not contribute substantially more beyond what the OFI offers. This indicates that their regression model, by effectively utilizing order flow data, can capture essential market dynamics without necessarily relying on the more complex cross-impact parameters. On the other side, [11] concludes that including the cross impact effect instead improves prediction capabilities.

Another example where regression models are used to predict mid-price movements using OFI, is [31], where it also proves the importance of including multiple levels of the limit order book for predictions. [31] utilizes the OFI from multiple levels and integrate all these into the prediction model, rather than using only the first level. Unlike [11], where OFIs from multiple levels are aggregated using PCA, in [31] they are all used independently in the regression model as parameters. The authors apply, as [11], both ordinary least squares (OLS) and Ridge regression techniques to empirically fit a linear relationship between MLOFI and contemporaneous changes in mid-price. Their findings demonstrate that incorporating order flow data from deeper in the LOB significantly improves the out-of-sample goodness-of-fit of their models.

All these approaches propose more "statistical" methods for making predictions. Another approach that has proven to be very efficient in the literature is the use of deep learning, which leverages complex neural network architectures capable of capturing non-linear relationships and patterns in data that are often not apparent or are too intricate for traditional models. This ability makes deep learning particularly effective in predicting the mid-price of financial assets, where mid-price movements can be influenced by multiple, interrelated factors operating across different time scales. [20] studies the application of deep learning techniques to predict high-frequency stock returns using data from the LOB. They specifically employ a variety of deep learning architectures, including Long Short-Term Memory (LSTM) networks, to analyze order flow imbalances and their predictive power on stock mid-prices at multiple time horizons. The authors demonstrate that deep learning models, particularly those trained on

more granular and refined inputs such as order flow data rather than direct order book states, significantly outperform traditional statistical and machine learning models. By employing Convolutional Neural Network (CNN) networks, they are able to aggregate data together and find a measure of imbalance, and by using LSTM networks, they capture the temporal dependencies inherent in order flow data, which proves crucial for accurately forecasting future mid-price movements. The paper not only tests these models against traditional benchmarks but also investigates the predictive power of models over several time horizons.

Another interesting research utilizing LSTM is that of [32]. The authors propose a sophisticated method utilizing Deep Neural Networks (DNNs) for predicting stock mid-prices. The paper elaborates on how deep learning, specifically through the use of LSTM networks, can be leveraged to improve the accuracy of predicting stock mid-price movements. Unlike [20], [32] does not aggregate data using CNNs; however, both use LSTM networks. LSTMs are capable of capturing temporal dependencies, which are crucial for analyzing financial markets. The authors compare their DNN-based model with traditional prediction models and demonstrate superior performance in terms of prediction accuracy and reliability. In our research, we will also employ DNN models, using both CNNs and LSTM networks together. Additionally, we will explore extending these models to incorporate data from multiple LOBs simultaneously.

1.5.2 Optimal Execution

Over the past few decades, there has been extensive research on how to optimally execute trading strategies, with the goal of optimizing order execution for greater profit. In order to solve this problem, a multitude of factors are considered, such as historical stock prices and various market conditions. Traditionally, researchers employ stochastic models for the dynamics of stock prices. These models are subsequently calibrated using available data. Within this framework, once the model has been established, achieving an analytical solution to the optimization problem is highly challenging due to the complexities of the models employed. In the few instances where such a solution is feasible, stochastic optimal control techniques are utilized. Otherwise, numerical computational methods are adopted to to find the optimal execution strategy. The pioneers of this topic were [2], and [4], who laid the foundation for many subsequent studies. They assumed the price movements follow a geometric Brownian motion. Since then, research on this topic has increased. Research on optimal execution has been conducted using market orders ([6], [7]), limit orders ([27] , [9], [12]), and combinations of both ([8], [18]). We will focus in our research only on the placement of limit orders, based on the setup described in [8].

Some studies, such as [8] or [18], do not incorporate signals from the LOB into their models for additional insights on mid-price movements. [8] presents an optimal

execution strategy that integrates both limit and market orders to enhance trading efficiency. Using a stochastic model to describe the dynamics of mid-price movements, and assuming that the market orders arrive following an independent Poisson process, the authors develop a strategy for executing large orders while considering constraints on order volume and depth. In our thesis, we will take their stochastic market modelling setup and adapt it to the use of limit orders only. Similar to [8], [18] also includes the possibility of utilizing both limit and market orders. However, it focuses on modeling the spread as a continuous-time Markov chain, and it is on this model that the decision to use limit or market orders is based. The optimal control problem is resolved using dynamic programming methods, as in our case.

Other research includes information from the LOB, such as Order Flow or Volume Imbalance, within the stochastic model. This addition provides significant insights that prove crucial for developing execution strategies: [6] employ stochastic models to describe the dynamics of the limit order book, incorporating the volume imbalance of the LOB into their considerations. This is not calculated deterministically, but is modelled with stochastic models, using jump processes. This complexity precludes an analytical solution to the problem. However, our model enables an analytical approach by integrating signals, including volume imbalances, from the LOB. In a similar way, [7] develops a strategy to find the optimal speed for market orders, incorporating the order flow of market orders within the stochastic model. Unlike our research and [6], they do not look at the volume imbalances of the LOB, but only at the frequency of other market orders, studied as order flow and modelled stochastically.

An alternative approach to addressing the optimal execution problem involves the application of Reinforcement Learning (RL) or Deep Reinforcement Learning (DRL) methodologies. These techniques enable an agent to learn an optimal behavior through iterative trial-and-error interactions within a dynamic environment, under the guidance of a reward feedback system. Stochastic models often cannot fully represent the complex dynamics of the limit order book, leading to necessary assumptions for solving optimal execution problems. RL methods can better approximate these complexities but face challenges like high computational costs and the need for detailed hyperparameter tuning. One of the most popular RL algorithm is Q-learning [30], that learns the quality of actions, indicating how beneficial they are for an agent to perform under specific circumstances. It iteratively updates these quality values, known as Q-values, using the Bellman equation until it finds an optimal policy, which guides the agent to choose the action that maximizes the expected reward for each state. [25] adapted the Q-learning technique to better address the optimal execution problem. They achieved this by employing a DNN to estimate the Q-function, which is integral to Q-learning. The developed model is a fully connected Neural Network trained using Double Deep Q-Learning that takes into consideration the current state of the limit order book. This technique is applied to nine different stocks and the results show that it outperforms

TWAP on seven out of nine stocks. Also [10] employ in their research RL to address the optimal execution problem, with a particular focus on Double Deep Q-Network learning and a novel variant of the reinforced deep Markov model. These models are utilized to devise an optimal trading strategy for a FX triplet. A similar approach was used by [14], who leverage RL to identify a strategy for pairs trading, a strategy that matches a long position with a short position in two highly correlated stocks, anticipating that any divergence in their price movements will eventually converge.

1.6 Main Contributions

This thesis, a collaborative research effort between TU Delft and MN, makes a substantial contribution to the scientific literature on these topics: firstly, it proposes a new way of measuring the volume imbalance of the limit order book by incorporating weighted deeper levels of the limit order book, not just the first. Additionally, using the concept of order flow imbalance, it extends existing methods to predict the evolution of an asset's price: in particular, it proposes an innovative new method based on deep learning models that evaluates whether considering multiple assets (or venues) in our predictions can enhance the accuracy of our forecasts. Moreover, the most significant contribution of this thesis is to incorporate these predictions into a stochastic model where the forecast is used as the drift of the price dynamics, and with techniques of stochastic optimal control, it defines an optimization problem whose solution (found analytically, and therefore can be computed more rapidly) provides a new and innovative execution strategy. All this will then be trained and simulated (on a market simulator created by the authors as backtesting environment) on real and historical data from two venues of the FX Spot market, focusing on the Euro-Dollar exchange. We focus only on this exchange because the methodology is the same for all currency exchanges. Moreover, we consider two venues because these are the data we have available; however, all methods developed and researched in this thesis are extendable to different venues (or even different assets). The results are promising and show that our novel execution strategy allows orders to be executed with a higher profit compared to the strategy that does not include the predictions as a drift term. This approach makes this research highly versatile, and an excellent starting point for new future studies. All data analysis, model implementations, simulations, and other computational tasks within this thesis were conducted using Python.

1.7 Structure

The structure of this thesis is as follows. In this chapter (Chapter 1) we introduced the central theme of the thesis, discussing key concepts necessary for understanding the topic and reviewing the principal scientific research available in the literature. In Chapter 2, we will continue with the description of the dataset and we will give some visualizations in order to understand how it is structured. Then, in Chapter 3, we define what constitutes a volume imbalance in the limit order book and explore methods for measuring it. We also present empirical evidence demonstrating the utility of these imbalances in predicting future returns. Subsequently, the research continues with Chapter 4, which introduces another commonly used method for measuring LOB imbalances (the Order Flow Imbalance) and discusses methods for predicting the future return of the exchange price. These method are the forward-looking price impact model and the Order-Flow-CNN-LSTM model. Following that, Chapter 5 introduces the stochastic model to describe the dynamics of the mid-price and the market., and later it defines and solves the stochastic optimization problem in order to derive the optimal execution strategy. Finally, Chapter 6 provides empirical simulations that apply these execution strategies to real data, comparing their performance against benchmark strategies. The thesis concludes with Chapter 7, where we summarize the findings and discuss their implications and potential for future research.

2 Preliminary Data Analysis

In this chapter, we will conduct a comprehensive analysis of the available data. We will describe the dataset's structure and characteristics, and illustrate its features through visualizations. Specific days or time intervals will be selected randomly from our dataset to serve as examples in our visualizations.

2.1 The Dataset

Let us start by discussing the available dataset and its structure. As already mentioned, the dataset we are considering has several features: first of all, each row indicates an update of the LOB. Each time a limit order is posted, executed, modified, or cancelled, the LOB will update, and therefore a new row of the dataset will be created with the new information (including also the time when this update occurs). The columns we are considering are shown in Table 2.1, along with an explanation of what they represent and what type of objects they are in Python. The name of the columns will also be the notation that we will use in our research.

It is important to note that our dataset offers five levels, namely $i \in \{1, ..., 5\}$, and thus we will have five different levels in the ask side, and five different levels in the bid side. This will also be the reason why in the course of our research we will consider only

Column	Description	Data Type
T	Time of update of the LOB	Datetime
A^i	Ask price of level i	Float
V_A^i	Volume of A^i	Int
B^i	Bid price of level i	Float
V_B^i	Volume of B^i	int
S	Spread	Float
M	Mid-Price	Float

Table 2.1: Description and data type of the columns of the dataset available

the first five levels of the LOB, as they are the only ones we have available. However, all the theoretical modeling we will carry out is generalizable to a different number of levels. Moreover, we have access to two different LOBs, since we have data from two different venues. Given the significant disparity in the values of these features, standardization becomes a crucial step to ensure comparability and to train models. We employ the *min-max normalization* technique for this purpose, which is mathematically represented as:

$$\mathbf{X}_{stand} = \frac{\mathbf{X} - \min(\mathbf{X})}{\max(\mathbf{X}) - \min(\mathbf{X})}. \tag{2.1}$$

Finally, the outliers have been removed using the Z-score method [24]. The Z-score method is a statistical technique used to identify outliers within a dataset by measuring the distance of each data point from the mean in terms of standard deviations. Mathematically, the Z-score of a data point x_i is calculated using the formula:

$$Z_i = \frac{x_i - \mu}{\sigma} \tag{2.2}$$

where x_i is the value of the data point, μ is the mean of the dataset, and σ is the standard deviation of the dataset. The Z-score indicates how many standard deviations a data point is from the mean. Data points with a Z-score beyond a specified threshold (for our research, we used the score ± 12) are considered outliers.

2.1.1 Visualizations

Our analysis continues with an examination between the euro-dollar exchange rates at two distinct venues, Venue1 and Venue2. Our focus will be on the progression of mid-prices at both venues, which will be represented through comparative plots (differences in mid-prices, logarithmic returns and their differences).

In the course of this thesis, we will use log returns due to their advantageous properties, such as ease of summation, and because they are commonly used in the literature to measure returns [11].

Definition 2.1.1 (Log Return)**.** Let $\{M_t\}_{t\in\mathcal{Z}}$ be the time series of the mid-price of an asset. The *log return* R_t at time t is defined as

$$R_t = \log \frac{M_{t+1}}{M_t}.$$

Clearly, the definition of log return can be tailored based on the future interval under consideration: for instance, potential values of t could be represented by each possible update of the LOB, or the state of the LOB at every predefined time interval, such as 50ms.

Our exploration starts with an examination of mid-price movements. Specifically, Figure 2.1 shows the fluctuations in the mid-price at the Venue1 for the euro-dollar exchange (EUR/USD) over an interval of 20 minutes of the day September 14th, 2023. This visualization serves as a representative example of the dynamics we are studying. As we can see, the time series moves quite randomly, although there are trends (for a while it is up, and then it goes down)

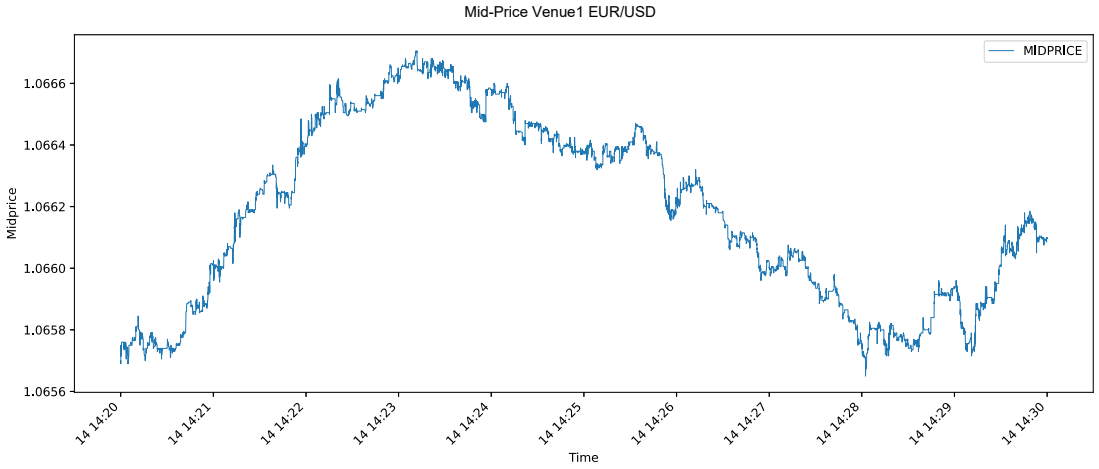


Figure 2.1: Mid-price movements, Venue1, EUR/USD, for a time interval of 20 minutes (September 17th, 2023).

Let us now compare the mid-price of both venues simultaneously: Figure 2.2 presents the movements of mid-prices for both Venue1 and Venue2 for an interval of one entire day (January 17th, 2024). The movements appear to be nearly identical and follow the same trend, going to indicate that the price differences between the different venues are very small, not perceptible through plots. The observed similarity between venues suggests that data from one could potentially be used to predict future returns in the other, indicating that certain information may be anticipated earlier in one venue than another, and that is also what we will try to investigate in this thesis. This insight could be advantageous for enhancing predictive models and refining execution strategies.

Our analysis continues with an examination of the differences in the mid-prices from the order books of both venues. This comparison is shown in Figure 2.3.

Let us now look at the log-returns and their difference, to see how they behave. Figure 2.4 shows the difference in log returns between Venue1 and Venue2 for a time interval of 20 minutes (again, from September 14th, 2023), taken as an example.

As we can see, in both plots the values vary around zero, symmetrically and fairly

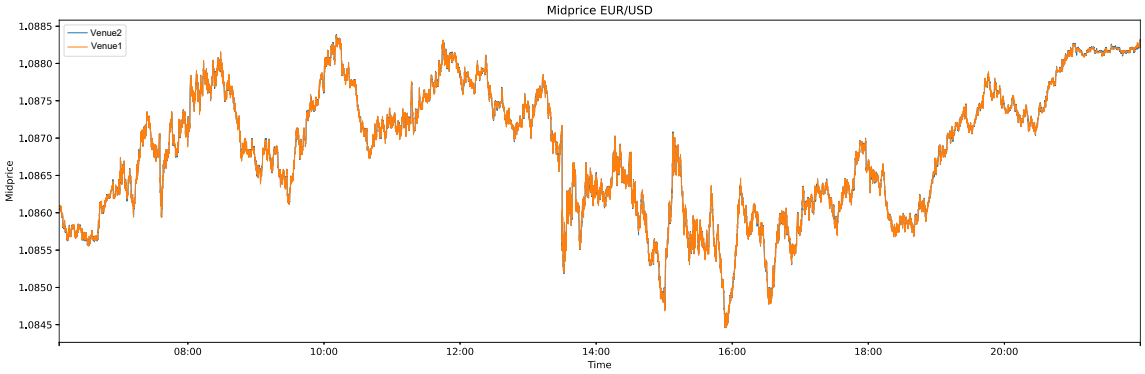


Figure 2.2: Mid-price movements Venue1 and Venue2, EUR/USD, for a time interval of January 17th, 2024.

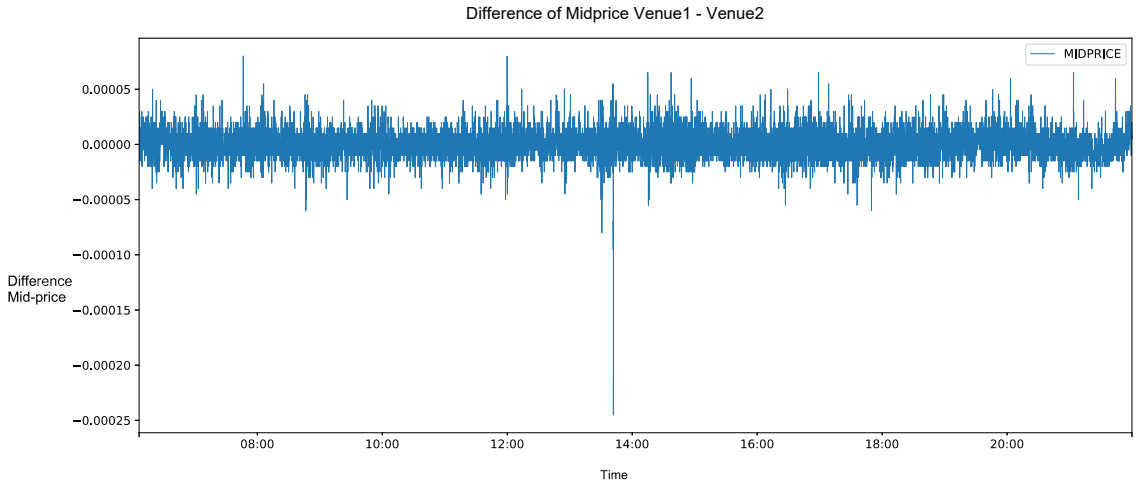


Figure 2.3: Difference of mid-prices Venue1 - Venue2, EUR/USD, for a time interval of one day (January 17th, 2024).

consistently. This means that there does not appear to be one venue that is always ahead of the other (in which case, we would have had that perhaps the values were always positive or negative).

Finally, we will now show the different values that the column Spread have in a dataset that include a randomly constructed week. This week is created by selecting five days from different weeks and combining them to form a generic week without specific trends. This approach ensures generality of the results and avoids the computational burden of a larger dataset. The days taken into considerations are: January 17th, 2024;

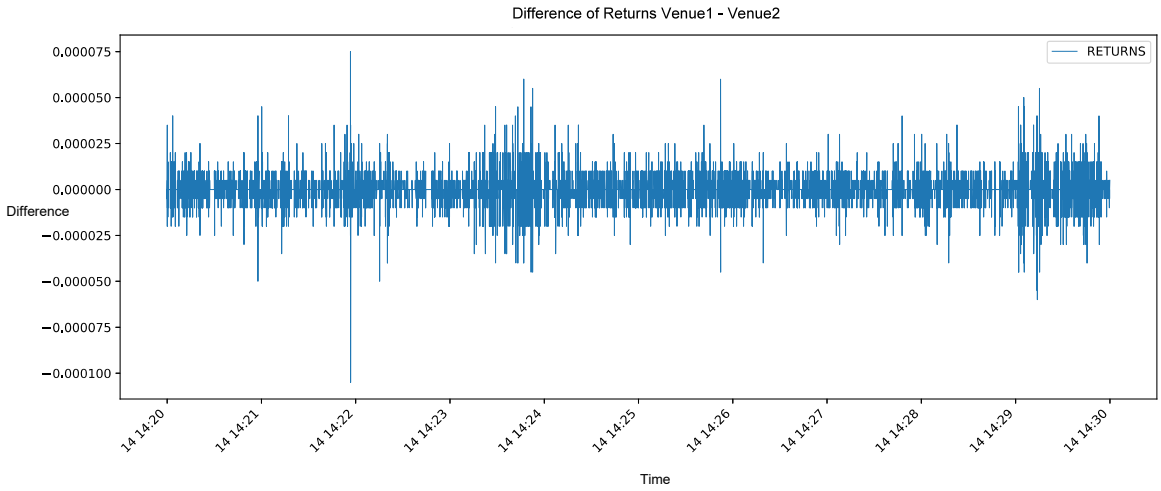


Figure 2.4: Difference of log returns Venue1 - Venue2, EUR/USD, for a time interval of 20 minutes (September 17th, 2023).

February 19th, 2024; March 5th, 2024; April 26th, 2024; May 16th, 2024. This is shown in Figure 2.5 (for Venue1) and Figure 2.6 (for Venue2). As we can see, in Venue1 the

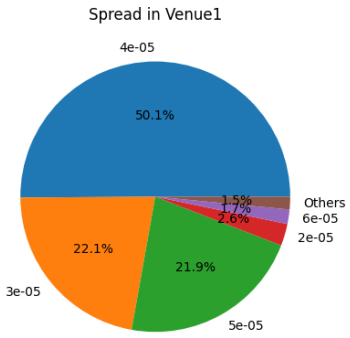


Figure 2.5: Values of Spread for Venue1

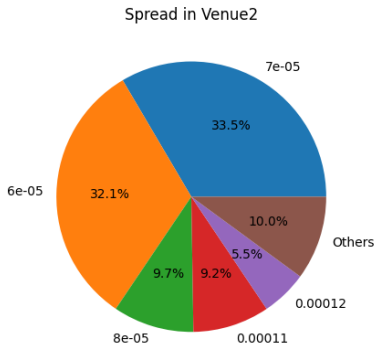


Figure 2.6: Values of Spread for Venue2

majority of the spread values are 3, 4, or 5 ticks for Venue1, while for Venue2 they are 6 or 7 ticks. This is also due to the fact that Venue1 has more updates and seems to have more liquidity. In fact, if we look at the number of updates of the two venues for this randomly selected week, we have that Venue1 has 6223678 rows, while Venue2 has 1389827 rows, so, Venue1 has around 4.5 times the number of rows of Venue2.

3 Limit Orderbook Volume Imbalances

The Limit Order Book is an indispensable resource in the research of trading strategies, given its capacity to provide a real-time representation of market participants' intent to transact a specific asset at various prices. This feature of the LOB makes it a rich source of information for predicting imminent price movements.

One of the most studied metrics within the LOB to capture market sentiment is the limit order book volume imbalance, which is a measure of difference between the quantity of buy and sell orders at a specific price level. This metric serves as an indicator of market pressures related to supply and demand, offering valuable insights into potential future price trajectories. A significant imbalance, characterized by a preponderance of buy orders over sell orders or vice versa, may signal an impending price shift. For example, a surge in buy orders could be indicative of robust demand, potentially driving the price up. In contrast, a predominance of sell orders could imply an oversupply, which might trigger a price decline. Furthermore, volume imbalance can be a potent tool for identifying trading opportunities. Traders who can adeptly interpret these imbalances may be positioned to forecast short-term price fluctuations and execute trades that align with these predictions, potentially yielding profitable outcomes.

In this chapter, we will discuss these topics in greater detail, including the introduction of a new method for measuring volume imbalance that incorporates multiple price levels in a weighted manner. We will then empirically demonstrate, using the real data available to us, how this measure can serve as a reliable indicator of future mid-price movements, thereby validating the foundational intuition of this research. We will also analyze the predictive ability in relation to the future temporal interval under consideration.

3.1 Measure for Volume Imbalance

In the literature, such as in [5] and [6], the formula in Equation 3.1 is proposed to calculate the Order Book Volume Imbalance at time t (ovi_t), which is based only on the volumes of the best bid and the best ask, without considering the other levels of the limit order book.

$$ovi_t = \frac{V^1_{A,t} - V^1_{B,t}}{V^1_{A,t} + V^1_{B,t}}, \tag{3.1}$$

where $V^1_{A,t}$ and $V^1_{B,t}$ are the total volumes at time t of LOs posted at the best ask and the best bid, respectively. Clearly $ovi_t \in [-1, 1]$, and when ovi_t is close to 1, there is more demand to sell and therefore we will expect the price to fall, while when it is close to -1 we expect the price to go up.

However, having access to information beyond the first level of the Limit Order Book, which includes the best ask and best bid, can be extremely useful. This multi-level data provides a more comprehensive view of market dynamics, making it an interesting factor to consider when creating a measure of volume imbalance. The first level of the LOB, which directly indicates the prices and volumes that would be executed by market orders, naturally holds significant importance. However, incorporating information from multiple levels can provide a more accurate understanding of market sentiment.

In light of this, this thesis introduces a novel measure of volume imbalance that considers the first l levels of the LOB. This measure will be referred as $ovi^\alpha_t(l)$, indicating the volume imbalance at time t, and it is defined in the following definition.

Definition 3.1.1 (Multi-level Order Book Volume Imbalance of order l)**.** Given a limit order book with l levels considered, let $V^i_{A,t}$ be the total volume at the i-th level on the ask side at time t, and let $V^i_{B,t}$ be the total volume at the i-th level on the bid side at time t. Let S_t be the spread at time t, let $\delta^i_{A,t}$ and $\delta^i_{B,t}$ be the absolute value of the difference in ticks, at time t, of the prices between the i-th level and the mid-price on the ask and bid side respectively, and let $\alpha \geq 0$ a real number. The *Multi-level Order Book Volume Imbalance* $ovi^\alpha_t(l)$ of order l at time t is then defined as follow:

$$ovi^\alpha_t(l) = S_t \cdot \frac{C^\alpha_{A,t}(l) - C^\alpha_{B,t}(l)}{C^\alpha_{A,t}(l) + C^\alpha_{B,t}(l)}, \tag{3.2}$$

where

$$C^\alpha_{A,t}(l) = V^1_{A,t} + \sum_{i=2}^{l} \frac{V^i_{A,t}}{(\delta^i_{A,t})^\alpha}, \quad C^\alpha_{B,t}(l) = V^1_{B,t} + \sum_{i=2}^{l} \frac{V^i_{B,t}}{(\delta^i_{B,t})^\alpha}.$$

In the equation for $ovi^\alpha_t(l)$ in 3.2 the spread has a direct impact on the order book imbalance. Specifically, a larger spread corresponds to a greater order book imbalance.

This can be intuitively understood by considering that a larger spread provides more 'room' for price movement, thereby allowing the order book imbalance to exert a greater influence on price dynamics. Furthermore, the total volumes from the second level are weighted by the cube (we will use $\alpha = 3$ based on empirical results) of their distance (measured in ticks) from the mid-price. This implies that the further these volumes are from the mid-price, the less significant they become. This weighting scheme underscores the diminishing influence of distant orders on the overall order book imbalance and on the future price movements.

3.2 Empirical Forecast of Returns

After establishing a quantitative measure for the imbalance volume of the order book in the previous section, we will now explore how this measure can be employed to infer the trajectory of the mid-price in the next instants. This will be demonstrated by comparing the imbalance volume at a given time t with the return observed after a $50ms$ interval, thereby determining whether the return in the following $50ms$ is positive or negative. The return is defined as in Definition 2.1.1.

In order to evaluate these predictions, the same dataset of five days considered in the previous chapter (Chapter 2) will be used. This dataset will also be used in the next chapter to train and calibrate our models.

To assess the predictive power of the imbalance measure for subsequent returns, five distinct categories were established: *buy-heavy*, *buy-light*, *neutral*, *ask-light*, and *ask-heavy*. Each category represents a different range of the imbalance measure. The calculation for these categories is as follows: the *buy-heavy* class corresponds to the first 10% percentile of the data. The *buy-light* class encompasses the 10% to 35% percentile range. The *neutral* class spans from the 35% to 65% percentile. The *ask-light* class covers the 65% to 90% percentile range. Finally, the *ask-heavy* class includes all values from the 90% percentile up to the maximum value in the dataset. In this way, for example, if the volume imbalance of the limit order book at time t falls within the class of *buy-heavy*, we expect that with a higher probability the price will go up compared to the class *buy-light*, whose values are closer to the center of the interval. For each of these classes, it has been counted how many times the subsequent return is positive and negative, and the result is shown in the figures below. We will report the results both for $ovi_t^3(1)$ (Figure 3.1), and also for $ovi_t^3(5)$ (Figure 3.2).

Figures 3.1 and 3.2 reveal a clear pattern, corroborating the theoretical assumption that the five distinct classes of imbalances characterize a particular state in the limit order book, which is informative of the potential subsequent return (we didn't consider the cases where the return is null, signifying no alteration in the mid-price). During an imbalance condition, such as when the state is *buy-heavy*, the incidence of positive

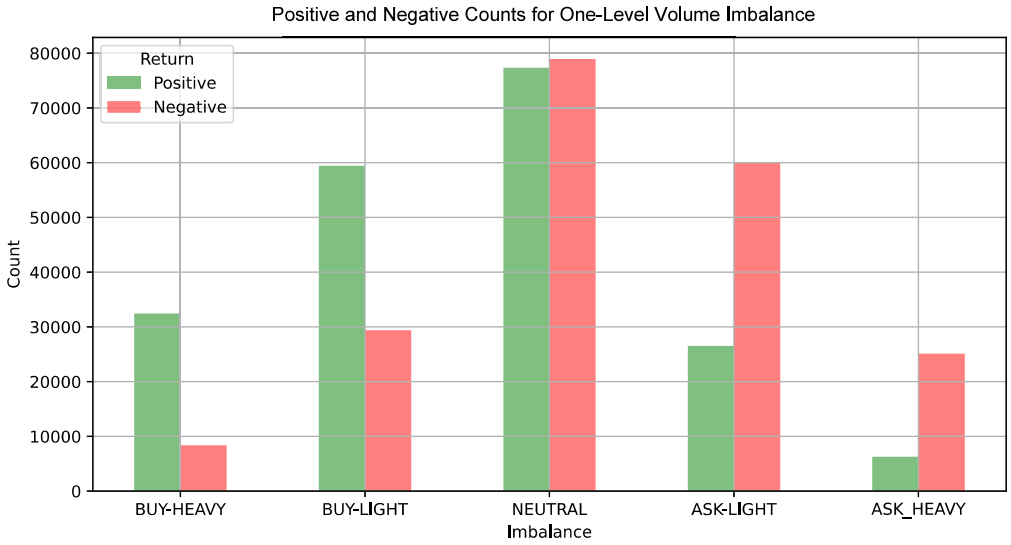


Figure 3.1: Frequency of positive and negative returns for each imbalance class for the one-level measure of imbalance ($ovi_t^3(1)$).

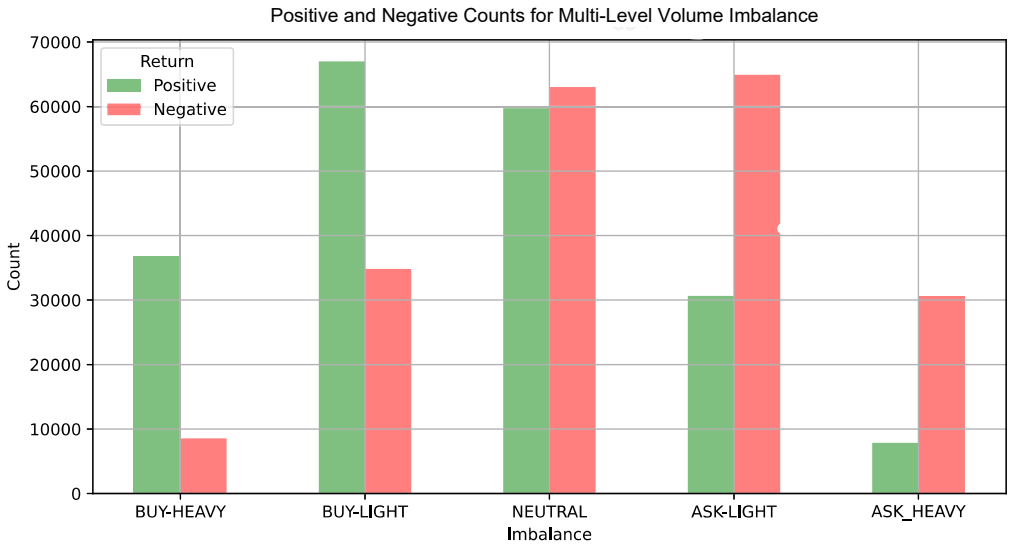


Figure 3.2: Frequency of positive and negative returns for each imbalance class for the multi-level measure of imbalance ($ovi_t^3(5)$).

returns (indicative of an increase in the mid-price) significantly outweighs that of negative returns. This observation highlights the predictive value of imbalance conditions

in forecasting price movements.

The frequency of the positive and negative returns are shown in Table 3.1.

	$ovi_t^3(1)$		$ovi_t^3(5)$	
	Positive	Negative	Positive	Negative
Buy-Heavy	79.5%	20.5%	81.2%	18.8%
Buy-Light	66.5%	33.5%	65.8%	34.2%
Neutral	49.5%	50.5%	48.7%	51.3%
Ask-Light	31.6%	68.4%	31.1%	68.9%
Ask-Heavy	21.0%	79.0%	20.2%	79.8%

Table 3.1: Frequency of positive and negative returns for each imbalance class, for $ovi_t^3(1)$ and $ovi_t^3(5)$.

Table 3.1 illustrates a compelling correlation: as the state of imbalance intensifies towards one side, the return's positivity or negativity becomes more pronounced. Moreover, when we incorporate the fifth-order imbalance measure, the predictive accuracy is better than the one of the first-order measure. This observation implies that factoring in multiple levels when calculating the volume of imbalance can enhance the precision of predictions regarding the mid-price's future movements. For this reason, we are going to consider for our model $ovi_t^3(5)$, since it can provide more accurate results. Furthermore, these charts demonstrate another significant point: the foundational intuition of this thesis is valid, and imbalances in the limit order book can indeed provide crucial information. When utilized with appropriate models, such information can help us develop an optimal execution strategy.

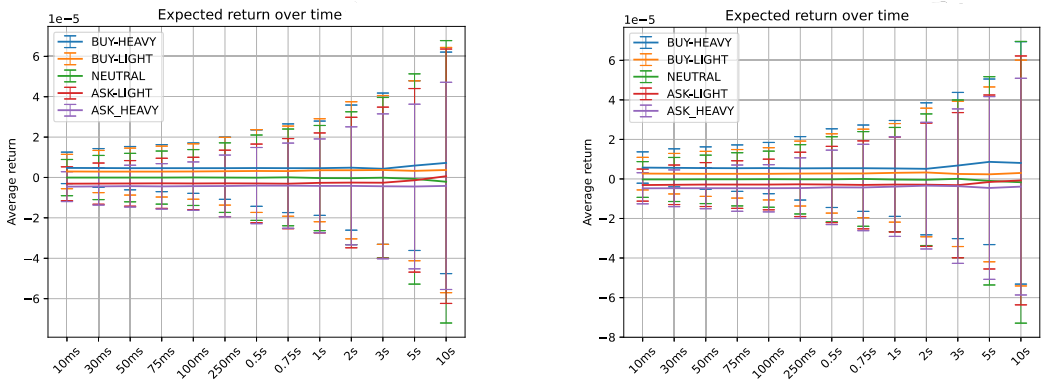
3.3 Expected Return as a Temporal Function

In Section 3.2, we explored the utility of order book imbalances as a tool for predicting the direction of mid-price movements in future instants, with a specific focus on the impact within a $50ms$ time-frame. We examined the frequency of such occurrences and whether these movements were positive or negative. In this section, our analysis will extend to study the expected return as a temporal function, thereby enhancing the generalizability of its predictive efficacy.

The expected return is defined as the expected value of the return, i.e. $\bar{R}_t = \mathbb{E}[R_t]$, which can be empirically computed as

$$\bar{R}_t = \frac{\sum_{i=1}^{n} R_{t_i}}{n}$$

. The analysis was conducted as follows: for each distinct class of imbalance (Buy-Heavy, Buy-Light, Neutral, Ask-Light, Ask-Heavy), the empirical expected return (considering only when there was a change in the mid-price) was calculated for the upcoming Δ seconds, where $\Delta \in [0s, 10s]$. The results for the first-order volume imbalance measure (3.3a) and for the fifth-order one (3.3b) are shown in Figure 3.3.



(a) First-order measure.

(b) Fifth-order measure.

Figure 3.3: Expected return (and standard deviation) as function of time for the five different imbalance classes.

From Figure 3.3, it can be observed that each imbalance class (for instance, Buy-Heavy or Ask-Heavy) has a distinct expected return, especially when considering smaller time intervals (e.g., within $500ms$). It is also noticeable that as the time interval increases, so does the standard deviation. This is understandable and can be attributed to the difficulty of making long-term predictions using only the order book volume imbalance. Nevertheless, the results are noteworthy. It can be observed that the outermost imbalance classes, such as Buy-Heavy or Ask-Heavy, respectively have the highest and lowest expected returns.

Finally, we will compare the results between the two different types of measures. The trends exhibited by the two charts are very similar, implying that both measures potentially possess robust predictive power for future mid-price movements. Up to one second, both measures maintain a constant expected value. However, a notable distinction can be observed in the case of the fifth-order measure (Figure 3.3b), where these values are more differentiated (the actual values of average and standard deviation of the expected returns are shown in Table 3.2, for the first order measure, and in Table 3.3, for the fifth order one). The extremal values, corresponding to the outermost classes such as Buy-Heavy and Ask-Heavy, are higher (and respectively, lower) than those of the first-order measure (Figure 3.3a). This observation suggests that incorporating

	Buy-Heavy		Buy-Light		Neutral		Ask-Light		Ask-Heavy	
	avg	std	avg	std	avg	std	avg	std	avg	std
10ms	0.47	0.80	0.28	0.87	0.00	0.92	-0.29	0.86	-0.44	0.78
30ms	0.46	0.98	0.28	1.06	0.00	1.11	-0.29	1.04	-0.44	0.95
50ms	0.45	1.09	0.28	1.18	0.00	1.22	-0.28	1.15	-0.43	1.06
75ms	0.45	1.18	0.28	1.28	0.00	1.34	-0.28	1.26	-0.44	1.15
100ms	0.45	1.27	0.28	1.39	0.00	1.41	-0.28	1.32	-0.43	1.22
250ms	0.45	1.56	0.29	1.69	0.00	1.76	-0.28	1.67	-0.42	1.56
500ms	0.46	1.56	0.30	1.70	0.00	1.76	-0.28	1.66	-0.42	1.55
750ms	0.45	2.22	0.30	2.25	0.00	2.44	-0.29	2.25	-0.40	2.13
1s	0.45	2.36	0.35	2.57	-0.03	2.65	-0.26	2.49	-0.41	2.34
2s	0.48	3.12	0.35	3.39	-0.03	3.34	-0.24	3.25	-0.41	2.94
3s	0.43	3.74	0.35	3.69	-0.01	3.98	-0.25	3.76	-0.43	3.62
5s	0.59	4.20	0.32	4.45	-0.07	5.20	-0.14	4.57	-0.44	4.07
10s	0.72	5.48	0.37	6.07	-0.21	7.00	-0.06	6.30	-0.41	5.13

Table 3.2: Average (avg) and standard deviation (std) of returns over different time intervals, for volume imbalance of 1st order ($ovi_t^3(1)$), for each class.

	Buy-Heavy		Buy-Light		Neutral		Ask-Light		Ask-Heavy	
	avg	std	avg	std	avg	std	avg	std	avg	std
10ms	0.58	0.76	0.29	0.80	-0.02	0.88	-0.33	0.79	-0.50	0.77
30ms	0.57	0.95	0.29	1.01	-0.02	1.09	-0.31	0.98	-0.50	0.91
50ms	0.56	1.05	0.28	1.12	-0.01	1.20	-0.30	1.09	-0.49	1.02
75ms	0.56	1.15	0.28	1.20	-0.02	1.31	-0.30	1.18	-0.49	1.14
100ms	0.56	1.26	0.28	1.28	-0.01	1.38	-0.30	1.24	-0.50	1.16
250ms	0.55	1.56	0.29	1.60	-0.02	1.71	-0.29	1.58	-0.49	1.49
500ms	0.56	1.95	0.30	1.96	-0.02	2.12	-0.30	1.89	-0.46	1.84
750ms	0.55	2.14	0.30	2.20	0.00	2.36	-0.31	2.19	-0.47	2.14
1s	0.54	2.38	0.33	2.45	0.03	2.59	-0.30	2.36	-0.41	2.48
2s	0.53	3.30	0.34	3.21	-0.04	3.29	-0.31	3.08	-0.37	3.16
3s	0.68	3.66	0.28	3.64	0.01	3.95	-0.31	3.63	-0.38	3.86
5s	0.87	4.14	0.25	4.37	-0.09	5.22	-0.17	4.35	-0.48	4.57
10s	0.81	6.13	0.30	5.71	-0.16	7.13	-0.07	6.30	-0.40	5.48

Table 3.3: Average (avg) and standard deviation (std) of returns over different time intervals, for volume imbalance of 5th order ($ovi_t^3(5)$), for each class.

multiple levels in the computation of volume imbalance can yield more reliable and

precise information, thereby enhancing the cost-effectiveness of the expected return.

Generally, from these tables and figures, it can be observed that as time increases, the standard deviation also increases, indicating a decrease in prediction accuracy. This highlights the necessity of striking an appropriate balance between the time period considered for analysis and the desired precision of the prediction.

4 Order Flow Imbalance

In Chapter 3, we have observed that the measure of order book volume imbalance can provide additional insights into future expected returns, and consequently, the movement of the mid-price in subsequent instances. In this chapter, we introduce another measure, the Order Flow Imbalance, denoted by OFI, as defined in [11] and in [20]. We will employ this measure and the methods proposed in the papers to evaluate its predictive power on returns. We will compare the results with the previously defined measure (multi-level order book volume imbalance), and adapt and extend the models for our use. Moreover, we will propose predictive models based on machine learning methods to use this information to predict the future return. This approach will allow us to further understand the dynamics of market behavior and improve our predictive capabilities.

4.1 Mathematical Definitions of Order Flow Imbalance

The Order Flow Imbalance (OFI) is a measure of the limit order book volume imbalance that also takes into account the imbalance in a past time interval relative to the one under consideration. Therefore, it can provide additional information on the flow of volume imbalance. This measure enhances our understanding of the temporal dynamics of volume imbalances and their implications for mid-price changes, which, according to [13], are driven by the OFI.

As described in [11], if we consider a time interval $(t-h, t]$, we can assign a numerical enumeration, denoted by n, to each observation of all order book updates. In this way, considering the venue (or the stock) i, and two consecutive order book states for it at $n-1$ and n, we can define the bid order flows ($OF^{m,i}_{n,Bid}$) and ask order flows ($OF^{m,i}_{n,Ask}$)

of venue i at level m and at time n as [11]:

$$OF_{n,Bid}^{m,i} = \begin{cases} V_{n,Bid}^{m,i}, & \text{if } B_n^{m,i} > B_{n-1}^{m,i}, \\ V_{n,Bid}^{m,i} - V_{n-1,Bid}^{m,i}, & \text{if } B_n^{m,i} = B_{n-1}^{m,i}, \\ -V_{n,Bid}^{m,i}, & \text{if } B_n^{m,i} < B_{n-1}^{m,i}, \end{cases} \tag{4.1}$$

$$OF_{n,Ask}^{m,i} = \begin{cases} -V_{n,Ask}^{m,i}, & \text{if } A_n^{m,i} > A_{n-1}^{m,i}, \\ V_{n,Ask}^{m,i} - V_{n-1,Ask}^{m,i}, & \text{if } A_n^{m,i} = A_{n-1}^{m,i}, \\ V_{n,Ask}^{m,i}, & \text{if } A_n^{m,i} < A_{n-1}^{m,i}, \end{cases} \tag{4.2}$$

where $B_n^{m,i}$ and $A_n^{m,i}$ denote the bid or ask price of venue i at level m at time n, and $V_{n,Bid}^{m,i}$ ad $V_{n,Ask}^{m,i}$ correspond to the total volume of the m level price, for venue i, at time n, for the bid and ask price [11].

Now, we will continue by the definition of *Best-level OFI*:

Definition 4.1.1 ([11], Best-level OFI)**.** Given a real number $h \geq 0$, and a time interval $(t-h, t]$, the *Best-level OFI* calculates the accumulative OFIs at best ask/bid side during that time interval, and it is defined as

$$OFI_{t,h}^{1,i} := \sum_{n=N(t-h)+1}^{N(t)} OF_{n,Ask}^{1,i} - OF_{n,Bid}^{1,i}, \tag{4.3}$$

where $N(t-h)+1$ and $N(t)$ are the indexes of the first and last limit order book updates during the time interval taken into consideration, and i denote the venue.

We can extend this definition to the deeper-level OFI:

Definition 4.1.2 ([11], Deeper-level OFI)**.** Given a real number $h \geq 0$, and a time interval $(t-h, t]$ and a level $m(m \geq 1)$, the *Deeper-level OFI* is

$$OFI_{t,h}^{m,i} := \sum_{n=N(t-h)+1}^{N(t)} OF_{n,Ask}^{m,i} - OF_{n,Bid}^{m,i}, \tag{4.4}$$

where $N(t-h)+1$ and $N(t)$ are the indexes of the first and last limit order book updates during the time interval taken into consideration, and i denote the venue.

Finally, we will use the average size of order volumes to scale OFIs, in order to deal with intraday pattern in limit order volume, as suggested in [11].

Definition 4.1.3 ([11], Scaled Deeper-level OFI)**.** Given a real number $h \geq 0$, and a time interval $(t-h, t]$ and a level $m(m \geq 1)$, the *Scaled Deeper-level OFI* is

$$\text{ofi}_{t,h}^{m,i} = \frac{OFI_{t,h}^{m,i}}{Q_{t,h}^{M,i}}, \tag{4.5}$$

where $Q_{t,h}^{M,i} = \frac{1}{2M\Delta N(t)} \sum_{m=1}^{M} \sum_{n=N(t-h)+1}^{N(t)} [V_{n,Ask}^{m,i} + V_{n,Bid}^{m,i}]$ is the average order book volume across the M levels, and $\Delta N(t) = N(t) - N(t-h)$ is the number of updates during that time interval.

In our research, we consider the top $M = 5$ levels of the limit order book, since they are all the available levels in the data.

The final issue to address is the aggregation of these values. As suggested by [11], we will employ Principal Component Analysis (PCA) to aggregate these values (for a comprehensive explanation of PCA, please see Appendix A). We will demonstrate that the first principal component accounts for a significant portion of the total variance among multi-level OFIs. For this reason, we proceed to define the Integrated Order Flow Imbalance (ofi^I). This approach allows us to condense the information from multiple variables into a unique one, simplifying our analysis while retaining the essential information about the original data.

Definition 4.1.4 ([11], Integrated ofi)**.** Given the vector of multi-level OFI $\mathbf{ofi}_{t,h}^{i} = (\text{ofi}_{t,h}^{1,i}, \ldots, \text{ofi}_{t,h}^{5,i})$, the *integrated ofi* is defined as

$$\text{ofi}_{t,h}^{I,i} = \frac{\mathbf{w}_1^T \mathbf{ofi}_{t,h}^{i}}{||\mathbf{w}_1||_1}, \tag{4.6}$$

where $\mathbf{w}_1$ is the first principal vector computed from historical data.

We can conclude this section by highlighting that our dependent variable that we are taking into consideration is the logarithmic return, defined as

Definition 4.1.5 ([11], logarithmic return)**.** Given a time interval $(t-h, t]$, we define the *logarithmic return* as

$$R_{i,t}^{(h)} = \log\left(\frac{M_{i,t}}{M_{i,t-h}}\right), \tag{4.7}$$

where $M_{i,t}$ is the mid-price at time t for venue i.

4.2 Forward-Looking Price and Cross Impact Models

In the previous section, we have explored the concept of OFI, including its definition and calculation methods. In this section, we will present several models that have demonstrated effective predictive power for future mid-price movements. These models, which were introduced in [11], provide a basis for constructing trading strategies.

Originally, these models were designed for the equity market. Therefore, part of our research involves examining their performance in the FX spot market. Additionally, the models were conceived for a set of different stocks, some of which are correlated with each other. In our case, instead of considering different stocks, we will consider different mid-prices originating from various venues, and we will analyze how one venue influences the others.

4.2.1 Model

The models we are about to present initially consider a single venue and analyze the price impact of the best-level OFI, referred to as FPI^1, and the Integrated OFI (FPI^I). Subsequently, different venues will be considered (for our experiments, the number of venues will be two), and the cross impact will also be taken into account by including the best-level OFI (FCI^1) (and, respectively, the Integrated OFI (FCI^I)) to aid in predicting future returns. These models are defined in the following way, according to [11]:

$$FPI^1: \quad R^{(f)}_{i,t+f} = \alpha^1_i + \sum_{k\in L} \beta^{1,k}_i \text{ofi}^{1,i}_{t,kh} + \epsilon^1_{i,t+f}, \tag{4.8}$$

$$FPI^I: \quad R^{(f)}_{i,t+f} = \alpha^I_i + \sum_{k\in L} \beta^{I,k}_i \text{ofi}^{I,i}_{t,kh} + \epsilon^I_{i,t+f}, \tag{4.9}$$

$$FCI^1: \quad R^{(f)}_{i,t+f} = \alpha^1_i + \sum_{j=1}^{N} \sum_{k\in L} \beta^{1,k}_{i,j} \text{ofi}^{1,j}_{t,kh} + \eta^1_{i,t+f}, \tag{4.10}$$

$$FCI^I: \quad R^{(f)}_{i,t+f} = \alpha^I_i + \sum_{j=1}^{N} \sum_{k\in L} \beta^{I,k}_{i,j} \text{ofi}^{I,j}_{t,kh} + \eta^I_{i,t+f}. \tag{4.11}$$

In the previous definition, f is the forecasting horizon of future returns, L represents the set of lags, i the venue, and N the number of venues (or stocks).

As an extension of the models and results studied in [11], we adopt the same methodology, but using the first and fifth order multi-level order book volume imbalance, as explained in the previous chapters. Therefore, the models will be:

$$FPovi^1: \quad R^{(f)}_{i,t+f} = \alpha^1_i + \beta^1_i \text{ovi}^3_{i,t}(1) + \epsilon^1_{i,t+f}, \tag{4.12}$$

$$FPovi^5: \quad R^{(f)}_{i,t+f} = \alpha^5_i + \beta^5_i \text{ovi}^3_{i,t}(5) + \epsilon^5_{i,t+f}, \tag{4.13}$$

$$FCovi^1: \quad R^{(f)}_{i,t+f} = \alpha^1_i + \sum_{j=1}^{N} \beta^1_{i,j} \text{ovi}^3_{j,t}(1) + \eta^1_{i,t+f}, \tag{4.14}$$

$$FCovi^5: \quad R^{(f)}_{i,t+f} = \alpha^5_i + \sum_{j=1}^{N} \beta^5_{i,j} \text{ovi}^3_{j,t}(5) + \eta^5_{i,t+f}. \tag{4.15}$$

Here, $\text{ovi}^3_{i,t}(l)$ represents the multi-level order book volume imbalance of venue i, at time t, of level l.

To compare the results, as also done in [11], we will compare the values of out-of-sample R^2 (OS R^2), where R^2 is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where y_i are the observed values, $\hat{y}_i$ are the predicted values from the model, $\bar{y}$ is the mean of the observed values. R^2 measures the proportion of variance in the dependent variable explained by the independent variables. A higher R^2 indicates that the model effectively captures the variance, aligning closely with actual data points. Consequently, a higher R^2 value signifies a more accurate and reliable model, demonstrating strong predictive power and generalizability to new data. This approach allows us to evaluate the predictive performance of our models in a manner consistent with the original study, providing a robust basis for comparison and further analysis.

4.2.2 Hyperparameter Tuning

Before presenting the results, it is important to discuss certain parameters that need to be selected, namely the past interval to consider for calculating the order flow imbalance and the future interval to use for predicting returns. These parameters are fundamental. The first parameter defines the historical window of the order book data that is significant for making informed predictions about future market behavior. In contrast, the second parameter determines the optimal time frame for forecasting future returns. The choice of these intervals directly impacts the accuracy and reliability of the predictive models, as they influence the relevance of past data and the timing of predictions, respectively. Therefore, a careful selection and justification of these parameters are essential for the robustness of the study.

To select these parameters, we have conducted a grid search, and the optimal values will be chosen based on the highest R^2 value. For our experiments, the value considered (in milliseconds) for f (in the future) are {30, 50, 70, 90, 110, 130, 150, 170, 190, 210, 230, 250, 270, 300}, while the value (in milliseconds) for h (in the past) are {50, 100, 150,

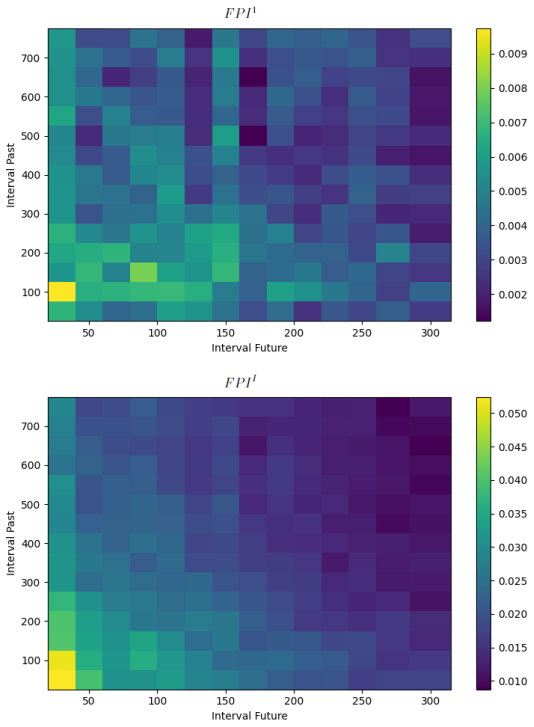
200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750}. In this study, we will focus on the results for Venue1 with the effect of Venue2 as a representative example. This approach prevents redundancy and avoids excessive repetition across similar scenarios. Given the substantial computational resources and time required to run simulations for one venue and the influence of the other, expanding our analysis to include the results for the same experiments, but with the swap venues, would significantly slow down the research process without providing proportionate additional insights.

In Figures 4.1 and 4.2, meshgrids are presented where the R^2 values are represented through colors. The y-axis indicates the past time interval used to calculate the order flow values, while the x-axis represents the future prediction interval. The different colors denote the average R^2 values obtained by fitting the model on a randomly constructed week. This week is created by selecting five days from different weeks and combining them to form a generic week without specific trends. This approach ensures generality of the results and avoids the computational burden of a larger dataset. The randomly chosen days taken into considerations are: January 17th, 2024; February 19th, 2024; March 5th, 2024; April 26th, 2024; May 16th, 2024. This is the dataset that will be used to test our models. It is also important to note that the available data consists of actual FX Spot Market limit order books for the Euro-Dollar exchange, including the limit order book states from two different venues, updated continuously. In order to implement the models, we used the *statsmodels.api.OLS* function in Python.

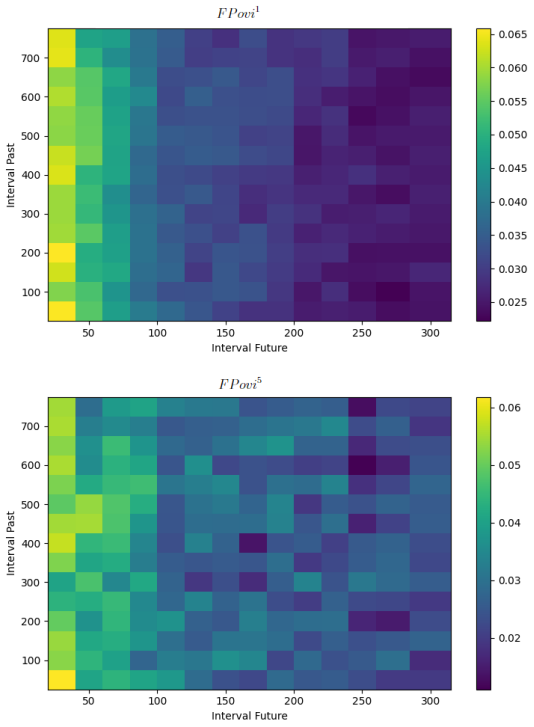
The models depicted include all those described in the previous section. As can be observed, there is a consistent trend where the further into the future the prediction is made, the weaker and less accurate it becomes. This outcome is expected, as forecasting the future movement of the mid-price is inherently challenging, and a longer future interval introduces greater uncertainty.

Let us now examine the figures in greater detail. First, we will consider the functions that account solely for the price impact of a single venue, as illustrated in Figure 4.1. Notably, the highest R^2 values are observed when using order book volume imbalance as features (Figure 4.1b). The results are comparable whether we consider only the first level ($FPovi^1$) or five levels of depth in the limit order book ($FPovi^5$). It is noteworthy that the models yielding the best R^2 values are consistently those with the shortest prediction time, specifically 30 milliseconds. However, predictions at 50 milliseconds also demonstrate satisfactory performance, with positive R^2 values, which contrasts with the findings reported in [11], where the values of R^2 were negatives.

Regarding models that incorporate order flow (Figure 4.1a), the optimal past time intervals are 50 milliseconds or 100 milliseconds, while longer past intervals are unnecessary. For models considering the order book volume imbalance, it is important to note that the past interval is not considered, hence each row should be similar to the others. To ensure homogeneity, averaging over a larger number of days would be ideal. However, for the purpose of this thesis and due to the computational cost associated



(a) R^2 values for FPI^1 (on top) and FPI^I (below)



(b) R^2 values for $FPovi^1$ (on top) and $FPovi^5$ (below)

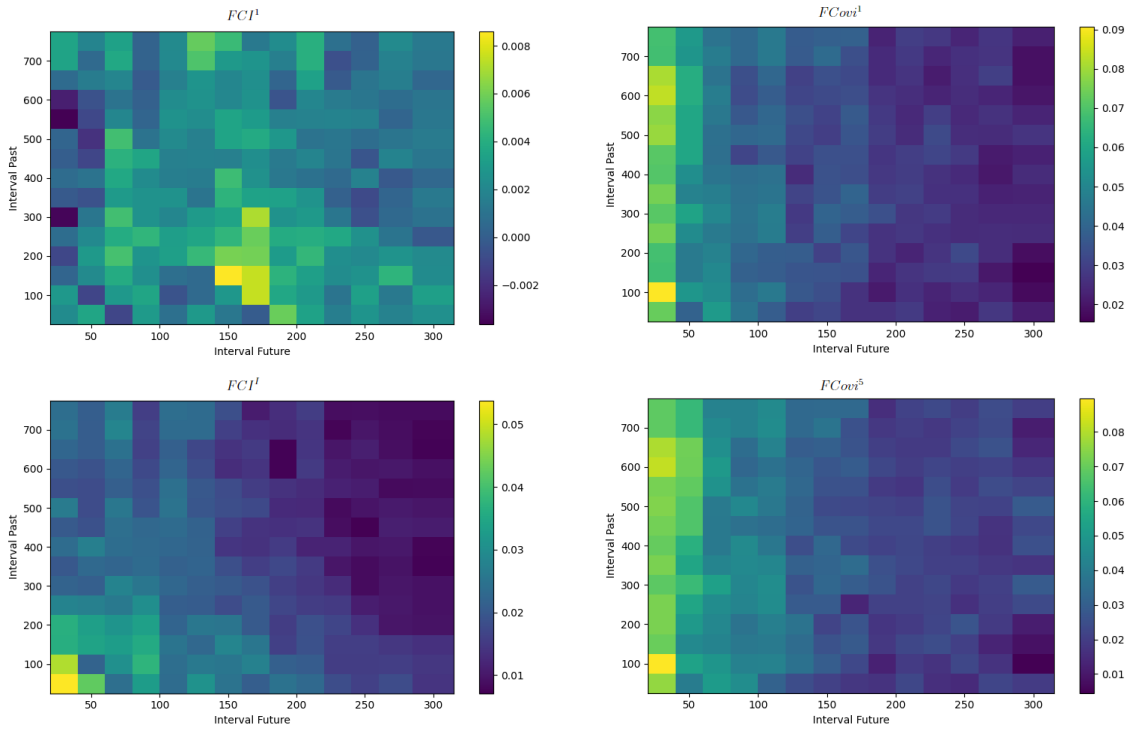
Figure 4.1: Different values of R^2 for different time in the past and in the future. The models considered are the Price Impact ones, and the venue taken into account is Venue1 .

with such an operation, the current approach is considered sufficient.

Let us now focus on the case where the impact of other venues on the future return movement is also included. These scenarios are depicted in Figure 4.2. As can be observed, even in this case, the highest R^2 values are obtained when considering the order book volume imbalance rather than the order flow imbalance. Similar observations to those previously discussed can be made.

Observing all these meshgrids of Figures 4.1 and 4.2, we can conclude that when considering order flow, it is consistently more advantageous to consider a greater number of levels in the limit order book (hence integrated ofi). Including more levels provides additional information, which enhances the model's predictive power. Conversely, when considering the limit order book volume imbalance, it appears unnecessary to include multiple levels, as the first level alone suffices.

To validate these observations, we will fit these models to the data. Based on our



(a) R^2 values for FCI^1 (on top) and FCI^I (below)

(b) R^2 values for $FCovi^1$ (on top) and $FCovi^5$ (below)

Figure 4.2: Different values of R^2 for different time in the past and in the future. The models considered are the Cross Impact ones, where the impact of the imbalances of Venue2 are included on the prediction of the return of Venue1.

findings here, we will adopt the following parameters for further analysis: for the past interval, we will use 100 milliseconds for the order flow, as this has been shown to provide robust predictive information. For the future interval, we will use 50 milliseconds. Although a 30-millisecond interval yields the highest R^2 values, it is not practical as the trading strategy can only update at a minimum of every 50 milliseconds. Thus, a 30-millisecond interval does not sufficiently cover the prediction window, whereas 50 milliseconds still provides good R^2 results and is therefore a more practical and acceptable choice.

These decisions are made to ensure that our models are not only theoretically sound but also practically viable within the constraints of trading operations.

4.2.3 Model Performance on Market Data

In this section, we will present and compare the results obtained from the models just described.

As observed in the previous section, the intervals we will use are $f = 50$ milliseconds for the future prediction and $h = 100$ milliseconds for the past interval. Although these are not the intervals yielding the highest R^2 values, they still provide good R^2 values and align well with the algorithm's operational cadence.

We will now examine the R^2 values obtained by fitting the models to the dataset, and we will present the plots for the best cases. Our focus will be on Venue1, considering the impact of Venue2 (Table 4.1).

Based on the results presented in Table 4.1, the best model is $FPovi^5$, which is slightly better than $FPovi^1$. This model includes the fifth-level order book volume imbalance. These numerical results indicate that including more levels generally enhances predictive capability. This is evident as the R^2 values for $FPovi^5$ are higher than those for $FPovi^1$, and similarly, FPI^I outperforms FPI^1.

When the impact of the other order book is added, an improvement is observed only if we consider the integrated ofi; otherwise, it is not beneficial. Nevertheless, the results remain good. Therefore, when focusing on Venue1, the best predictive model is $FPovi^5$, and, after the calibration of the model on data, its plot is shown in Figure 4.3. When $x = ovi^3(5)$ is known, the prediction will be governed by the line:

$$y = 7.13 \cdot 10^{-8} - 1.88 \cdot 10^{-6} x, \tag{4.16}$$

where x is $ovi^3(5)$ and y is the predicted log return.

Model	FPI^1	FPI^I	$FPovi^1$	$FPovi^5$	FCI^1	FCI^I	$FCovi^1$	$FCovi^5$
OS R^2	0.003	0.023	0.060	0.063	0.0002	0.033	0.056	0.052

Table 4.1: Out-of-Sample R^2 values for each model, using $f = 50ms$ and $h = 100ms$, on Venue1 (with the effect of Venue2).

To present the results of the model $FCovi^1$, we also display the regression plane in Figure 4.4. The parameters are described in the figure caption.

Regarding the execution time of the prediction, it takes less than a millisecond to calculate the prediction. This efficiency is due to the fact that the operations involved are simple and linear. Therefore, its use does not pose any issues with timing.

4.3 CNN - LSTM Models

We now turn our attention to a different approach to the model, specifically the application of deep learning techniques, which have been demonstrated in the literature to

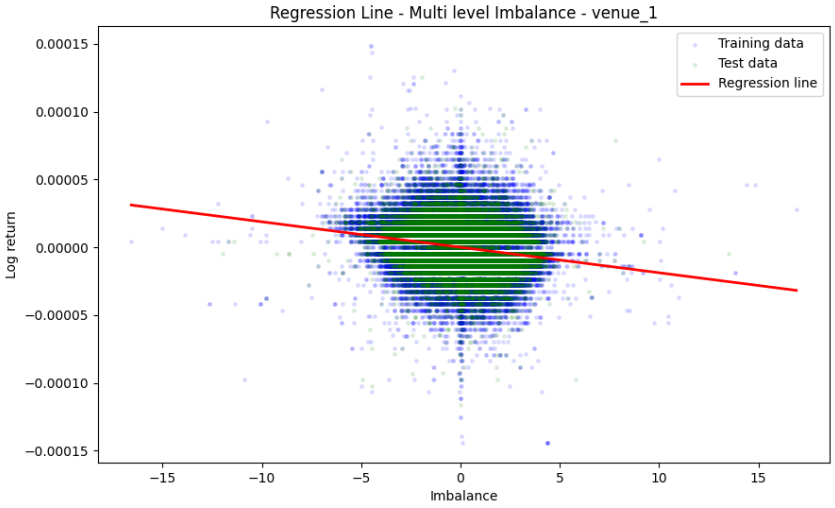


Figure 4.3: Plot of $FPovi^5$ for Venue1. The regression line is $y = 7.13{\cdot}10^{-8} - 1.88{\cdot}10^{-6}x$, where $ovi^3(5)$.

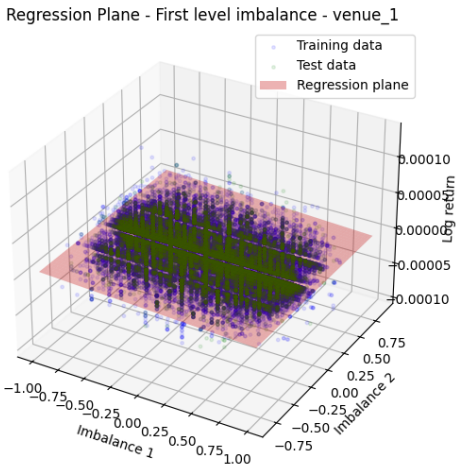


Figure 4.4: Plot of $FCovi^1$ for Venue1. The regression plane is $y = 9.83 \cdot 10^{-8} - 6.41 \cdot 10^{-6}x_1 - 8.26 \cdot 10^{-9}x_2$, where $x_1 = ovi^3(1)$ of Venue1, and $x_2 = ovi^3(1)$ of Venue2.

be effective in predictive capabilities (see for example [28]). The model we are about to present is based on the concept from [20]. However, we have implemented modifications to enhance the predictive capacity of the model and to allow for the consideration of

various venues (or stocks) in the prediction. These enhancements not only increase the model's accuracy but also accelerates its computation, as some features that unnecessarily complicated it have been removed.

4.3.1 Model Architecture

In our study, we have implemented, using Python, a Convolutional Neural Network combined with a Long Short-Term Memory network to efficiently model high-frequency trading data. The architecture of this network is inspired by [20]. However, in contrast to the original model, we have removed the temporal layers and the inception module. Temporal convolutions and the inception module, though beneficial for capturing complex patterns in data, significantly increase the computational complexity and runtime of the model. By omitting these components, we aim to reduce the computational cost without substantially sacrificing the model's performance. Furthermore, we also propose a new model where we have added a new layer in the CNN model to enable the neural network to extract information of the limit order book from multiple venues (which can be used also in the case of multiple stocks). This addition takes into account scenarios where other LOBs may influence or provide additional information about the movement of the mid-price of a specific price, as in our case. This modification does not result in a significant computational cost. This addition can lead to promising results, and we will use this extension to see if adding multiple venues or stocks can bring new information and improve the accuracy of predictions.

Rationale for Architecture

The use of a CNN followed by a LSTM network in our predictive model leverages the unique characteristics of these architectures, making them particularly suitable for addressing our prediction problem. CNNs excel in handling spatial relationships in data. In our case, they aggregate information across the various features of a limit order book—such as prices and volumes for both bid and ask directions, across all available levels. This capability allows the CNN to compute an order book volume imbalance as a derived feature, optimized through parameter tuning rather than predefined formulas. This aggregated output is then provided as input to an LSTM network, preserving the temporal sequence of the data.

LSTMs belong to the family of Recurrent Neural Networks (RNNs) and are adept at managing temporal data dependencies, which is crucial for our application. Unlike standard RNNs, LSTMs are designed to avoid the long-term dependency problem, allowing them to remember information for prolonged periods with less risk of vanishing gradient issues. This is achieved through their internal structure, which includes memory cells that store past information and gates that regulate the flow of information.

The gates—comprising the input, output, and forget gates—manage the cell's state and output flow for each timestep, selectively updating or forgetting information.

This combination of CNNs for spatial data processing and LSTMs for temporal data processing ensures a robust approach to modeling and forecasting future returns in financial markets. The CNN layers first extract and transform spatial features from the sequential input data, which are then effectively synthesized over time by the LSTM layers to make predictive insights.

For more technical information about the structure of the CNN and the LSTM, please refer to [17].

Specifications of the Architecture - One Venue

We will now focus into the details of the architecture of the CNN-LSTM model for only one venue. As previously mentioned, it is structured in two parts: a CNN component followed by a LSTM component. Before discussing each layer, let's define the structure of the CNN-LSTM input. First, we need to define the *State of a Limit Order Book*.

Definition 4.3.1 (State of a Limit Order Book)**.** Given a LOB, denoted by the index i, the *State of a Limit Order Book* $\mathbf{s}_t^i$ at time t is the vector of prices and volumes for ask and bid for each level, i.e.:

$$\mathbf{s}_t^i := (A_{t,i}^1, V_{A,t,i}^1, B_{t,i}^1, V_{B,t,i}^1, A_{t,i}^2, V_{A,t,i}^2, B_{t,i}^2, V_{B,t,i}^2, \dots, A_{t,i}^l, V_{A,t,i}^l, B_{t,i}^l, V_{B,t,i}^l), \tag{4.17}$$

where l is the number of level taken into consideration, $A_{t,i}^j, B_{t,i}^j$ are the ask and bid prices at the j-th level at time t, and $V_{A,t,i}^j, V_{B,t,i}^j$ are the corresponding volumes.

This will be the core of the input given to the CNN-LSTM. In fact, the final input will consist of states of consecutive times, so that the LSTM can capture the temporal dependence.

Let us show now how the model is structured in details (in Table 4.2 a summary of the architecture is shown).

Model	Layer	Description	Input-Output
CNN	**Convolution 1** One spatial convolution (1×2) Stride $= 1 \times 2$ Filters $= 32$	Combines prices and volumes for each side and level of the order book	$10 \times 4 \cdot l \times 1$ $\downarrow$ $10 \times 2 \cdot l \times 32$
	Convolution 2 One spatial convolution (1×2) Stride $= 1 \times 2$ Filters $= 32$	Combines imbalances across the two sides (ask and bid) for each level of the order book	$10 \times 2 \cdot l \times 32$ $\downarrow$ $10 \times l \times 32$
	Convolution 3 One spatial convolution $(1 \times l)$ Stride $= 1 \times l$ Filters $= 32$	Combines imbalances across all levels of the order book	$10 \times l \times 32$ $\downarrow$ 10×32
LSTM	**LSTM** 64 hidden units	Takes the data from the CNN and outputs the predicted return	10×32 $\downarrow$ 1×1

Table 4.2: Architecture of the CNN-LSTM.
l is the number of levels.

First of all, we will start by stating that all the layers use the *LeakyReLu* as activation function. Leaky Rectified Linear Unit (LeakyReLU) is preferred over the standard Rectified Linear Unit (ReLU) primarily to counteract the *dying ReLU* problem, where neurons become inactive and cease to contribute to the learning process. Unlike ReLU, which outputs zero for any negative input, LeakyReLU introduces a small, positive slope α in the negative part of its domain, thereby allowing a small, non-zero gradient when the input is negative. This modification ensures that all neurons remain active and continue to adapt during training, potentially leading to improved network performance and stability. LeakyReLU is defined as $f(x) = \max(\alpha x, x)$, where α is a small constant. In our case, we will use $\alpha = 0.01$.

The first convolutional layer in this network uses a spatial convolution with a kernel size of 1×2 and a stride of 1×2. It applies 32 filters to the input data. This layer's primary function is to aggregate prices and volumes for each side and level of the order book. By doing this, the layer effectively combines these features into a higher-level representation, reducing the spatial dimension from 4 times the number of levels to 2 times. Following the initial convolution, the second convolutional layer employs the same kernel size and stride, again with 32 filters. This layer focuses on combining the bid and ask sides of the order book to generate features that represent imbalances across these two sides for each level. The third convolutional layer uses a kernel size of $1 \times l$ (where l is the number of levels), covering all levels in a single stride. This broader kernel is designed to synthesize information across all levels, summarizing the entire depth of the market into a more compact form. After processing through the convolutional layers, the transformed data is fed into an LSTM layer with 64 hidden units. This LSTM layer is tasked with analyzing the temporal sequence of the features extracted by the CNN. Unlike the convolutional layers that primarily handle spatial features within the data, the LSTM can model long-term dependencies by learning from sequences of the market's evolution over time.

Multiple Venues

Now, we will show the extension of this model, a new contribution of this thesis. A summary of this new architecture is shown in Table 4.3.

Model	Layer	Description	Input-Output
CNN	**Convolution 1** One spatial convolution (1×2) Stride $= 1 \times 2$ Filters $= 32$	Combines prices and volumes for each side, level and venue of the order book	$10 \times 4 \cdot l \cdot n_v \times 1$ $\downarrow$ $10 \times 2 \cdot l \cdot n_v \times 32$
	Convolution 2 One spatial convolution (1×2) Stride $= 1 \times 2$ Filters $= 32$	Combines imbalances across the two sides (ask and bid) for each level and venue of the order book	$10 \times 2 \cdot l \cdot n_v \times 32$ $\downarrow$ $10 \times l \cdot n_v \times 32$
	Convolution 3 One spatial convolution $(1 \times l)$ Stride $= 1 \times l$ Filters $= 32$	Combines imbalances across all levels for each venue of the order book	$10 \times l \cdot n_v \times 32$ $\downarrow$ $10 \times n_v \times 32$
	Convolution 4 One spatial convolution $(1 \times n_v)$ Stride $= 1 \times n_v$ Filters $= 32$	Combines imbalances across all venues of the order book	$10 \times n_v \times 32$ $\downarrow$ $10 \times 1 \times 32$
LSTM	**LSTM** 64 hidden units	Takes the data from the CNN and outputs the predicted return	$10 \times 1 \times 32$ $\downarrow$ 1×1

Table 4.3: Architecture of the Multi-Venue CNN-LSTM.
l is the number of levels, n_v is the number of venues.

As already explained, this model, an extension of the previous one, also allows us to consider more venues, so that we can attempt to improve the prediction capability by including data from other limit order books that might be influential or significant.

To do so, we will indicate with n_v the number of venues (or assets) that we are taking into consideration. In this way, we can now define the *total state* as the concatenation of multiple states of limit order book taken from n_v venues.

Definition 4.3.2. Let $\mathbf{s}_t^1, \mathbf{s}_t^2, \ldots, \mathbf{s}_t^{n_v}$ be the state of n_v limit order book (taken from n_v venues) as defined in Definition 4.3.1, we define the *total state* as the concatenation of these states:

$$\mathbf{s}_t := (s_t^1, s_t^2, \ldots, s_t^{n_v}). \tag{4.18}$$

The main difference lies in the addition of the fourth layer (Convolution 4). This final convolutional layer uses a kernel size of $1 \times n_v$ (where n_v is the number of venues) and compresses across all venues, essentially reducing the data to a single feature set that captures inter-venue dynamics. This layer is crucial for models that aim to exploit variations across different trading venues, possibly reflecting different trading behaviors or liquidity patterns which could be predictive of future market movements.

In addition to this layer, obviously the input and output of all the above also changes, as the size includes the addition of n_v venues.

Further Extensions: Order Flow Imbalances Inputs

Up until now, the models presented have taken the raw limit order book data as input, without any prior aggregation. This approach allows the model to independently discover relationships among the various features, thereby providing greater flexibility and a more detailed description of the data. However, this also results in a higher number of parameters to train, leading to increased computation time and a greater risk of overfitting. For these reasons, we will now introduce further extensions to the model, based also on what it was proposed in [20]. These extensions encompass six additional new models that we have developed and evaluated, and they differ in the input they receive. Contrary to earlier models that utilized the LOB as input, these new models employ different inputs: the Order Flow, as defined in Equations 4.1 and 4.2, the OFI, as described in Definition 4.1.2, excluding the temporal sum, and the integrated ofi, as specified in Definition 4.1.4. To accommodate these changes, the architecture of the model has been adjusted, including the removal of specific layers. These modifications are detailed further in the subsequent descriptions.

- OF input: the models start from Convolution 2, and receive the OF as input (Equations 4.1 and 4.2) (both for the one venue and multi venue model);

- OFI input: the models start from Convolution 3, and receive the OFI as input (Definition 4.1.2) (both for the one venue and multi venue model);

- integrated ofi input: the one venue model starts directly from the LSTM layer with the integratedd ofi as input (Definition 4.1.4), while the multi venue model starts from Convolution 4.

The concept of utilizing OF or OFI as inputs for a single LOB was initially introduced in [20]. In our research, we extend this approach by incorporating the integrated ofi, a concept introduced in [11], as an input. Furthermore, our new models expand these scenarios to accommodate multiple venues, proposing new techniques that can enrich the scientific literature.

4.3.2 Model Performance on Market Data

We will now demonstrate the effectiveness of these models in predicting the future movement of the mid-price. The same dataset as previously used was employed to ensure the results are comparable, with predictions made for the next 50ms. As with any robust deep learning model, the dataset was divided into a training set and a test set to enable evaluation of the model on data not used during training. To implement these models, we first normalized the dataset used, then created samples for training and testing the model, taking 10 subsequent updates for each sample. A significant challenge arose when generating samples for models that encompass multiple venues. In such cases, it was necessary to first examine the timestamp associated with a sample in the target venue, and then retrieve the corresponding order book state from another venue at that same timestamp, in order to create a sample consisting of the subsequent 10 updates. This step is crucial to ensure that the samples across different venues are synchronized in time. Subsequently, we implemented the deep learning model using the Python library *Tensorflow.keras*, using the *layers.Conv2D*, *layers.LSTM*. Finally, the optimizer used is the Adam optimizer (*Tensorflow.keras.optimizers.Adam*).

To assess performance, Table 4.4 reports the in-sample R^2 (i.e., the value obtained using the training set data) and the out-of-sample R^2 (i.e., the value obtained using the test set). Additionally, we present the mean square error values for both the in-sample and out-of-sample data. The models we compare vary in the input they receive: in fact, as just explained in the previous sections, we will consider as input the raw LOB, the OF, the OFI, and the integrated ofi (all of these for both the cases of one venue and multiple venues, denoted by Multi).

As can be observed from the table, the best results are obtained when the input is the order flow for a single venue (row OF). This model achieves the highest out-of-sample R^2 value and the lowest MSE. It is also noteworthy that the models using limit

Model	IS R^2	OS R^2	Training MSE	Test MSE
LOB	0.31	0.30	0.0015	0.0015
Multi LOB	0.07	0.02	0.0012	0.0012
OF	0.49	0.44	0.0010	0.0012
Multi OF	0.061	0.053	0.0012	0.0012
OFI	0.35	0.33	0.0014	0.0014
Multi OFI	0.014	0.010	0.0013	0.0012
integrated ofi	0.11	0.11	0.0023	0.0023
Multi integrated ofi	0.009	0.009	0.0013	0.0012

Table 4.4: Comparison of In-Sample (IS) and Out-of-Sample (OS) R^2 values and Training and Test Mean Square Error for different models. The model considered differs based on the input data.

order book data or OFI for a single venue (rows LOB and OFI) perform well, though slightly worse than the OF model. In contrast, the results for the integrated ofi are not comparable to the previous ones.

The superior performance of the OF model can be attributed to the fact that the input data is still sufficiently raw for the model to further process and uncover relationships among different features, with only slight aggregation, primarily of prices and volumes. It is important to notice that the absolute prices at different levels are less significant than their relative distances and whether they are increasing or not. This characteristic allows the OF model to retain substantial information that has not been manually aggregated, while still achieving a reduction in dimensionality.

When considering the model that includes multiple venues (in our case, two), the results do not improve; in fact, they become worse. The rows in the table presenting the "Multi" model do not exhibit better R^2 values. This observation aligns with the findings in Section 4.2.3. Including multiple venues in our case does not lead to better price predictions. However, this is still a significant result to consider and does not imply that the model is incapable of providing more accurate predictions in other contexts.

What we can conclude is that, in the case of the FX Spot Market, including data from two different venues does not enhance the accuracy of the predictions. Therefore, the model that yields the best predictions in our case is the one using the Order Flow of only one venue as input, which we will refer to as the OF-CNN-LSTM model. This result also confirms the observations made in [20].

Future research could involve applying this model extension to different contexts, such as considering various stocks (e.g., in the case of equities), to explore how dynamics from other markets might influence the one being analyzed.

We will now visually examine the results, displaying how the predicted future returns

compare to the actual future returns (Figures 4.5 and 4.6). We will use the LOB and OF cases as examples to see how performance improves when considering OF as input rather than LOB. Ideally, we would want the blue points to lie on the red line. However, we recognize that in our case, it is impossible to predict future returns with such accuracy, as market dynamics include a significant non-deterministic component.

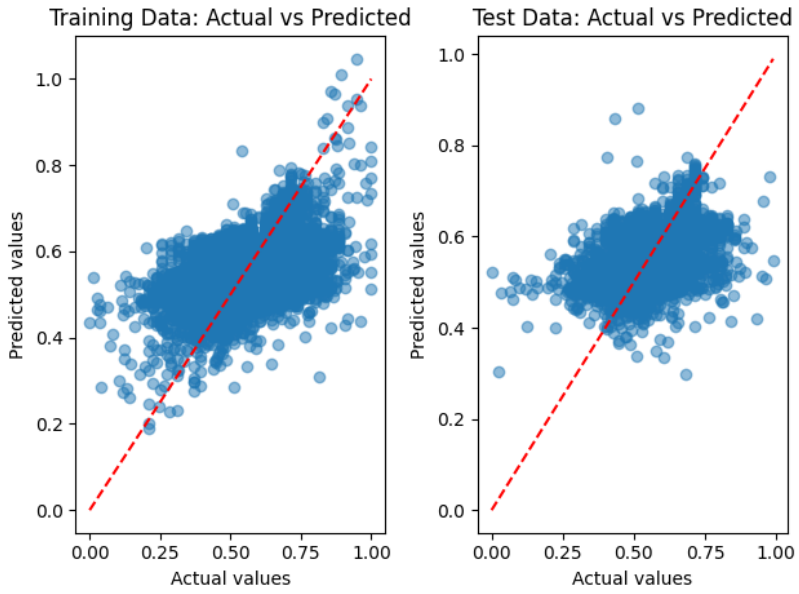


Figure 4.5: Actual values vs Predicted values for Training Data (on the left) and Test Data (on the right). The input is the Limit Order Book data (LOB).

As we can observe in Figure 4.5, the points are already reasonably distributed along the red line (which represents the case where the actual values and the predicted values are equal). However, when we consider the case of Figure 4.6 where the input is OF, the values are better distributed along the red line, indicating a higher predictive capacity, as also reflected by the R^2 values.

Regarding the execution time of the OF-CNN-LSTM model prediction, there is an important consideration. Numerical experiments were conducted on a local laptop[1], and the required time is around 35 milliseconds to make a prediction. This duration could pose a problem, as the market could potentially change within these 35 milliseconds. However, as noted in [20], a model of this nature, if implemented efficiently using a faster programming language like C++ and on more powerful machines commonly used in algorithmic trading, the execution time for predictions should be reduced to less

[1]Processor: 11th gen Intel(R) Core(TM) i5-1145G7 @ 2.60GHz 1.50 GHz. Installed RAM: 16.0 GB

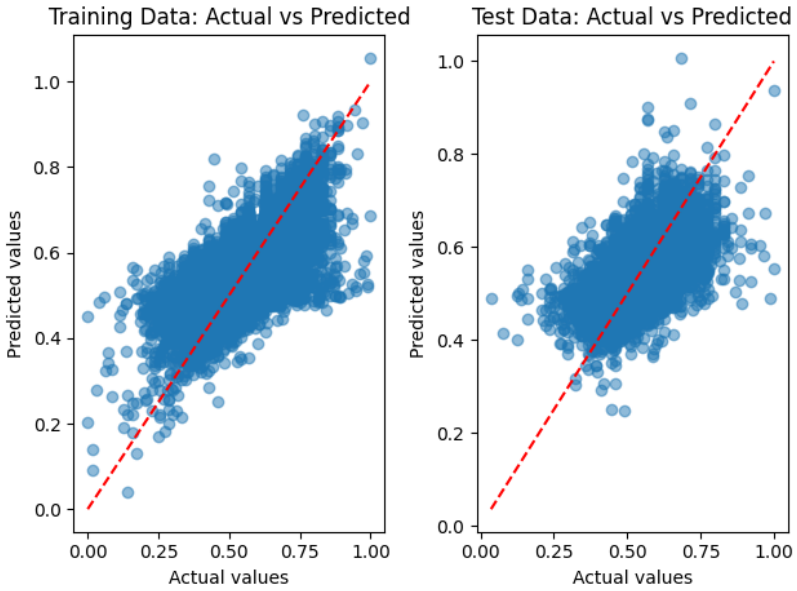


Figure 4.6: Actual values vs Predicted values for Training Data (on the left) and Test Data (on the right). The input is the Order Flow (OF).

than a millisecond. Therefore, it should not pose any issues even in real-time trading environments.

5 Optimal Execution Strategy

In this chapter, we will describe the mathematical model used, discuss key considerations regarding its selection, and explain the rationale behind this choice. Subsequently, we will outline the optimization problem, solve it, and present the analytical solution that constitutes our execution strategy. This analysis will specifically address the case of posting a sell limit order, noting that the approach for a buy limit order is analogous.

As already mentioned, the order book imbalances or the order flow can give important information about the dynamic of the mid-price. Consequently, we will extend current models by considering the effect of the order book volume imbalances on the dynamics of the mid-price. For this reason, we aim to develop comprehensive models capable of receiving signals from the limit order book and utilizing these signals to predict future movements of the mid-price and build an optimal execution strategy. These signals can be of various types, but for the purpose of our research, we will focus on signals related to order book volume imbalance or order flow.

5.1 The Optimization Problem

Consider the case where an agent wants to sell a number of $\mathcal{R}$ shares of an asset whose mid-price is M_t, whose dynamic follows the process described in Equation 5.1, where we denote with s_t a generic signal received from the data:

$$dM_t = g(\mathrm{s}_t)dt + \sigma_t dW_t \tag{5.1}$$

Here, $W = (W_t)_{0\leq t\leq T}$ is a standard Brownian motion, σ_t is the volatility, and $g(\mathrm{s}_t)$ represents the effect of a signal on the mid-price, where $g : \mathbb{R}^d \to \mathbb{R}$ is a generic predictive function capable of receiving an input (which could be a signal, but also the entire current order book), and returning as a result the prediction of future movement.

We can now define the agent's cash process $X^\delta = (X_t^\delta)_{0\leq t\leq T}$ as [8]

$$dX_t^\delta = (M_t + \delta_t)dN_t^\delta, \tag{5.2}$$

where $\delta = (\delta_t)_{0\leq t\leq T}$ is the depth at which the agent posts the sell limit order, and $N^\delta = (N_t^\delta)_{0\leq t\leq T}$ is the counting process corresponding to the number of times that the

agent's order is executed, whose intensity is $\lambda e^{-\kappa\delta}$, where $\lambda, \kappa \in \mathbb{R}$ are parameters to be calibrated on real data [8]. This means that we assume that a sell limit order posted at depth δ will be executed with probability $\lambda e^{-\kappa\delta}$. Moreover, T is the terminal time at which we would like the sell limit order to be executed. Finally, we can define the agent inventory which remains to be liquidated as

$$Q_t^\delta = \mathcal{R} - N_t^\delta. \tag{5.3}$$

The agent aims to maximize the profit from the liquidation of the inventory, while also ensuring that the order is executed within the prescribed time interval, namely by the terminal time T. If the agent holds any inventory at the end of the allowed time, it will be liquidated as if it were a market order, thus likely fetching a price inferior to the mid-price. For this reason, we impose a penalty if this situation occurs. Consequently, the agent's optimization problem becomes, based on [8], finding the value of

$$H(x,S) = \sup_{\delta\in\mathcal{A}} \mathbb{E}[X_\tau^\delta + Q_\tau^\delta(M_\tau - \gamma Q_\tau^\delta) \,|\, X_{0^-}^\delta = x, S_0 = S, Q_{0^-}^\delta = \mathcal{R}], \tag{5.4}$$

where $\gamma \geq 0$ is the liquidation penalty, and

$$\tau = \min\{T, \min\{t : Q_t^\delta = 0\}\} \tag{5.5}$$

is the minimum of T and the first time where the inventory is zero. Moreover, $\mathcal{A}$ is the admissible set of strategies δ. This function is interpreted as follows: the objective is to find the supremum, with respect to the depth δ, of the expected value of the cash that the agent possesses at the terminal time, denoted by X_τ^δ, plus a quantity representing the scenario where, if there remains an amount Q_τ^δ to be liquidated at the terminal time, it will be liquidated at a mid-price M_τ minus a penalty term that depends on the remaining amount to be liquidated, i.e., γQ_τ^δ. In this manner, the agent is not compelled to execute the order immediately, yet aims to avoid reaching the terminal time with remaining volume to be liquidated. Indeed, in such circumstances, the agent loses decision-making power, and therefore it is appropriate to impose a penalty in this scenario.

The corresponding value function, simplifying the notation based on [8], will be:

$$H(t,x,M,q) = \sup_{\delta\in\mathcal{A}} \mathbb{E}_{t,x,M,q}[X_\tau^\delta + Q_\tau^\delta(M_\tau - \gamma Q_\tau^\delta)], \tag{5.6}$$

where $\mathbb{E}_{t,x,M,q}[\cdot]$ is the expectation conditional on $x = X_{t^-}^\delta$, $M = M_t$, $q = Q_{t^-}^\delta$.

The dynamic programming principle (DPP) suggests that the value function in Equation 5.6 solves the following dynamic programming equation (DPE):

$$\begin{cases} \frac{\partial H}{\partial t} + g(\mathrm{s}_t)\frac{\partial H}{\partial M} + \frac{1}{2}\sigma_t^2\frac{\partial^2 H}{\partial M^2} + \\ + \sup_{\delta\in\mathcal{A}}\left\{\lambda e^{-\kappa\delta}\left[H(t, x+(M+\delta), M, q-1) - H(t,x,M,q)\right]\right\} = 0, \\ \\ H(T,x,M,q) = x + qM - \gamma q^2, \\ \\ H(t,x,M,0) = x. \end{cases} \tag{5.7}$$

This represents an optimal trading problem formulated as a non-linear partial integral differential equation (PIDE). Below, we provide an interpretation of the various terms:

- $g(\mathrm{s}_t)\frac{\partial H}{\partial M} + \frac{1}{2}\sigma_t^2\frac{\partial^2 H}{\partial M^2}$: corresponds to generator of the function H with respect to M.
- $\sup_{\delta\in\mathcal{A}}$: represents the ability of the agent to control the depth of the sell limit order.
- $\lambda e^{-\kappa\delta}$: corresponds to the fill probability, i.e. the probability that the sell limit order will be executed.
- $H(t, x+(M+\delta), M, q-1) - H(t,x,M,q)$: represents the change in the agent's value function when the sell limit order is executed. In this case, the cash increases by $(M+\delta)$, and the inventory decreases by 1.

The boundary condition with $q = 0$ represents the value function when the agent has liquidated everything, and hence the value is equal to the cash. Finally, the terminal condition $t = T$ corresponds to the case where the agent arrives at the terminal time, and hence the value function is the cash hat the agent has acquired up to that point (x), plus the value of liquidating what is left (q) at the penalized mid-price $(M - \gamma q)$.

5.2 Solving the Dynamic Programming Equation

The terminal and boundary conditions suggests that the ansatz for the value function that solves Equation 5.7 could be

$$H(t,x,M,q) = x + qM + h(t,q). \tag{5.8}$$

This ansatz consists of three components. The initial component represents the total cash gathered. The second term signifies the market value of the leftover stock, which is evaluated at the mid-price. Lastly, the function $h(t,q)$ illustrates the additional cash value gained by the agent from the optimal sale of the remaining shares. This last one will be the new function to be determined.

By substituting Equation 5.8 into the system 5.7, we discover that $h(t,q)$ must satisfy the following nonlinear system of Ordinary Differential Equations (ODEs):

$$\begin{cases} \frac{\partial h}{\partial t} + g(\mathrm{s}_t)q + \sup_{\delta\in\mathcal{A}}\left\{\lambda e^{-\kappa\delta}\left[\delta + h(t,q-1) - h(t,q)\right]\right\} = 0, \\ h(T,q) = -\gamma q^2, \\ h(t,0) = 0. \end{cases} \tag{5.9}$$

The optimal depth $\delta^*(t,q)$ can be determined in a feedback form by focusing on the first-order conditions for the supremum. This yields the subsequent results:

$$0 = \frac{\partial}{\partial\delta}\left(\lambda e^{-\kappa\delta}\left[\delta + h(t,q-1) - h(t,q)\right]\right) = \lambda e^{-\kappa\delta}\left(-\kappa\left[\delta + h(t,q-1) - h(t,q)\right] + 1\right).$$

By solving this equation we obtain that the optimal strategy δ^* in feedback control form must satisfy

$$\delta^*(t,q) = \frac{1}{\kappa} + [h(t,q) - h(t,q-1)]. \tag{5.10}$$

Let us now insert the optimal depth in feedback control form into 5.9, and we obtain the following non-linear system of ODEs for $h(t,q)$

$$\begin{cases} \frac{\partial h}{\partial t} + g(\mathrm{s}_t)q + \frac{\lambda}{e\kappa}\exp\left\{-\kappa[h(t,q) - h(t,q-1)]\right\} = 0, \\ h(T,q) = -\gamma q^2, \\ h(t,0) = 0. \end{cases} \tag{5.11}$$

In order to simplify the system of equation, let us make the substitution

$$h(t,q) = \frac{1}{\kappa}\log w(t,q), \tag{5.12}$$

where $w(t,q)$ has the same properties of $h(t,q)$. Here we assume that $w(t,q) > 0$. The first equation in 5.11 will satisfy

$$\begin{aligned} 0 &= \frac{\partial h(t,q)}{\partial t} + g(\mathrm{s}_t)q + \frac{\lambda}{e\kappa}\exp\left\{-\kappa[h(t,q) - h(t,q-1)]\right\} = \\ &= \frac{1}{\kappa}\frac{\partial_t w(t,q)}{w(t,q)} + g(\mathrm{s}_t)q + \frac{\lambda}{e\kappa}\exp\left\{-\kappa[\frac{1}{\kappa}\log w(t,q) - \frac{1}{\kappa}\log w(t,q-1)]\right\} = \\ &= \frac{1}{\kappa}\frac{\partial_t w(t,q)}{w(t,q)} + g(\mathrm{s}_t)q + \frac{\lambda}{e\kappa}\frac{w(t,q-1)}{w(t,q)}, \end{aligned}$$

hence

$$0 = \partial_t w(t,q) + g(\mathrm{s}_t)\kappa q\, w(t,q) + \frac{\lambda}{e} w(t,q-1). \tag{5.13}$$

In this way, the system 5.11 becomes

$$\begin{cases} \partial_t w(t,q) + g(\mathrm{s}_t)\kappa q\, w(t,q) + \frac{\lambda}{e} w(t,q-1) = 0, \\ \\ w(T,q) = e^{-\kappa\gamma q^2}, \\ \\ w(t,0) = 1. \end{cases} \tag{5.14}$$

In order to solve explicitly the system 5.14, we will start by computing $w(t,q)$ for $q = 1,2,3,4$ by explicit integration of 5.13. After that, we will generalize the solution and prove that it solves 5.14.

$\mathbf{q = 1}$

First, we will consider the case $q = 1$, which provides the equations:

$$\begin{cases} \partial_t w(t,1) + g(\mathrm{s}_t)\kappa\, w(t,1) + \frac{\lambda}{e} w(t,0) = 0, \\ \\ w(T,1) = e^{-\kappa\gamma}, \\ \\ w(t,0) = 1. \end{cases}$$

Let us focus only on the first equation, and using the boundary condition we obtain:

$$0 = \partial_t w(t,1) + g(\mathrm{s}_t)\kappa\, w(t,1) + \frac{\lambda}{e}.$$

This can be rewritten as

$$e^{g(\mathrm{s}_t)\kappa\, t}\partial_t w(t,1) + g(\mathrm{s}_t)\kappa\, e^{g(\mathrm{s}_t)\kappa\, t}\, w(t,1) = -\frac{\lambda}{e} e^{g(\mathrm{s}_t)\kappa\, t}.$$

We can now observe that the left-hand side can be written as

$$e^{g(\mathrm{s}_t)\kappa\, t}\partial_t w(t,1) + g(\mathrm{s}_t)\kappa\, e^{g(\mathrm{s}_t)\kappa\, t}\, w(t,1) = \partial_t \left[e^{g(\mathrm{s}_t)\kappa\, t} w(t,1) \right],$$

and hence

$$\partial_t \left[e^{g(\mathrm{s}_t)\kappa\, t} w(t,1) \right] = -\frac{\lambda}{e} e^{g(\mathrm{s}_t)\kappa\, t}.$$

The integration wrt t provides

$$\int_t^T \partial_u \left[e^{g(\mathrm{s}_t)\kappa\, u} w(u,1)\right] du = \int_t^T -\frac{\lambda}{e} e^{g(\mathrm{s}_t)\kappa\, u} du$$

$$e^{g(\mathrm{s}_t)\kappa\, T} w(T,1) - e^{g(\mathrm{s}_t)\kappa\, t} w(t,1) = -\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}(e^{g(\mathrm{s}_t)\kappa\, T} - e^{g(\mathrm{s}_t)\kappa\, t})$$
$$e^{g(\mathrm{s}_t)\kappa\, T} e^{-\kappa\gamma} - e^{g(\mathrm{s}_t)\kappa\, t} w(t,1) = -\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}(e^{g(\mathrm{s}_t)\kappa\, T} - e^{g(\mathrm{s}_t)\kappa\, t}),$$

and this let us conclude that

$$w(t,1) = e^{g(\mathrm{s}_t)\kappa\,(T-t)} \left[e^{-\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right] - \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \tag{5.15}$$

To continue the derivation of the solution, we need to derive the solutions for cases $q = 2, 3, 4$. These are shown in the Appendix B, as the derivation is long and the method is analogous to the $q = 1$ case.

General case

We now turn our attention to the generic case for q. Upon examining the Equations 5.15, B.2, B.4, B.6, we observe a recurring pattern, which can be generalized by the following solution:

$$w(t,q) = \sum_{i=0}^{q} (-1)^i e^{(q-i)g(\mathrm{s}_t)\kappa(T-t)} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!} \sum_{j=0}^{q-i} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!} e^{-\kappa\gamma(q-i-j)^2} \tag{5.16}$$

Thanks to this solution, we are now able to identify our optimal execution strategy, as outlined in the following proposition.

Proposition 5.2.1 (Optimal Execution Strategy)**.** *Let Equation 5.16 be the solution of the system 5.14, then the optimal execution strategy derived from system 5.7 is given by*

$$\delta^*(t,q) = \frac{1}{\kappa}\left[1 + \log \frac{\sum_{i=0}^{q} (-1)^i e^{(q-i)g(s_t)\kappa(T-t)} \left(\frac{\lambda}{g(s_t)\kappa e}\right)^i \frac{1}{i!} \sum_{j=0}^{q-i} \left(\frac{\lambda}{g(s_t)\kappa e}\right)^j \frac{1}{j!} e^{-\kappa\gamma(q-i-j)^2}}{\sum_{i=0}^{q-1} (-1)^i e^{(q-1-i)g(s_t)\kappa(T-t)} \left(\frac{\lambda}{g(s_t)\kappa e}\right)^i \frac{1}{i!} \sum_{j=0}^{q-1-i} \left(\frac{\lambda}{g(s_t)\kappa e}\right)^j \frac{1}{j!} e^{-\kappa\gamma(q-1-i-j)^2}}\right] \tag{5.17}$$

Proof. Recall Equation 5.12, and that the optimal strategy δ^* in feedback control form is Equation 5.10, we can derive that the analytical formula for the optimal strategy is

$$\delta^*(t,q) = \frac{1}{\kappa}\left[1 + \log\frac{w(t,q)}{w(t,q-1)}\right],$$

or

$$\delta^*(t,q) = \frac{1}{\kappa}\left[1 + \log\frac{\sum_{i=0}^{q}(-1)^i e^{(q-i)g(\mathrm{s}_t)\kappa(T-t)}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!}\sum_{j=0}^{q-i}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!}e^{-\kappa\gamma(q-i-j)^2}}{\sum_{i=0}^{q-1}(-1)^i e^{(q-1-i)g(\mathrm{s}_t)\kappa(T-t)}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!}\sum_{j=0}^{q-1-i}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!}e^{-\kappa\gamma(q-1-i-j)^2}}\right]$$

□

In order for Proposition 5.2.1 to hold true, we need the following theorem, the *Verification Theorem*:

Theorem 5.2.2 (Verification Theorem)**.** *Given a function $g : \mathbb{R}^d \to \mathbb{R}$, a signal $s_t \in \mathbb{R}^d$, parameters $\lambda, \kappa \in \mathbb{R}$, a terminal time $T > 0$, with $T \in \mathbb{R}$, and a function $w : \mathbb{R} \times \mathbb{N} \to \mathbb{R}^+$, then Equation 5.16 solves the system 5.14.*

Proof. To demonstrate that 5.16 solves the system 5.14, we proceed to verify the terminal and boundary conditions, and confirm that this solution indeed satisfies the ODE.

It is straightforward to observe that

$$w(t,0) = 1,$$

since, if $q = 0$, then also $i = 0$ and $j = 0$. Moreover,

$$w(T,q) = \sum_{i=0}^{q}(-1)^i\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!}\sum_{j=0}^{q-i}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!}e^{-\kappa\gamma(q-i-j)^2},$$

because, if $t = T$, we have that

$$e^{(q-i)g(\mathrm{s}_t)\kappa(T-t)}\big|_{t=T} = e^{(q-i)g(\mathrm{s}_t)\kappa(T-T)} = e^0 = 1.$$

Now, rewriting the terms of the sums, we get

$$w(T,q) = \sum_{i=0}^{q}\sum_{j=0}^{q-i}(-1)^i\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^{i+j}\frac{(i+j)!}{i!\,j!}\frac{1}{(i+j)!}e^{-\kappa\gamma(q-i-j)^2}.$$

If we now call $n = i + j$, we derive

$$w(T,q) = \sum_{n=0}^{q} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^n \frac{1}{n!} e^{-\kappa\gamma(q-n)^2} \sum_{i=0}^{n} (-1)^i \binom{n}{i}.$$

Now, using the *Binomial theorem*, we can observe that

$$\sum_{i=0}^{n} (-1)^i \binom{n}{i} = (1-1)^n,$$

which is equal to 0, $\forall n \neq 0$. So, we will consider only the case where $n = 0$, and we obtain that

$$w(T,q) = e^{-\kappa\gamma q^2},$$

as in 5.14. Finally, we will prove that Equation 5.16 is indeed a solution of the ODE in system 5.14 by substituting it into the equation and verifying that it sums to zero. By substituting Equation 5.16 into the ODE of the system 5.14, we obtain

$$0 = \sum_{i=0}^{q} (-1)^{i+1}(q-i) g(\mathrm{s}_t)\kappa e^{(q-i)g(\mathrm{s}_t)\kappa(T-t)} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!} \sum_{j=0}^{q-i} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!} e^{-\kappa\gamma(q-i-j)^2} + \tag{5.18}$$

$$+ \sum_{i=0}^{q} (-1)^{i} q g(\mathrm{s}_t)\kappa e^{(q-i)g(\mathrm{s}_t)\kappa(T-t)} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!} \sum_{j=0}^{q-i} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!} e^{-\kappa\gamma(q-i-j)^2} + \tag{5.19}$$

$$+ \sum_{i=0}^{q-1} (-1)^{i} \frac{\lambda}{e} e^{(q-1-i)g(\mathrm{s}_t)\kappa(T-t)} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!} \sum_{j=0}^{q-1-i} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!} e^{-\kappa\gamma(q-1-i-j)^2}, \tag{5.20}$$

where 5.18 is the term $\partial_t w(t,q)$, 5.19 is $q g(\mathrm{s}_t)\kappa w(t,q)$, and 5.20 is $\frac{\lambda}{e} w(t,q-1)$. The first line (5.18) can be written as

$$- \sum_{i=0}^{q} (-1)^{i} q g(\mathrm{s}_t)\kappa e^{(q-i)g(\mathrm{s}_t)\kappa(T-t)} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!} \sum_{j=0}^{q-i} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!} e^{-\kappa\gamma(q-i-j)^2} + \tag{5.21}$$

$$\sum_{i=0}^{q} (-1)^{i} i g(\mathrm{s}_t)\kappa e^{(q-i)g(\mathrm{s}_t)\kappa(T-t)} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!} \sum_{j=0}^{q-i} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!} e^{-\kappa\gamma(q-i-j)^2} \tag{5.22}$$

and 5.21 vanishes with 5.19. Moreover, if $i = 0$, 5.22 will be zero. Therefore, 5.22 is equivalent to Equation 5.23. With these considerations, we need to prove that

$$\sum_{i=1}^{q}(-1)^i i g(\mathrm{s}_t)\kappa e^{(q-i)g(\mathrm{s}_t)\kappa(T-t)}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!}\sum_{j=0}^{q-i}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!}e^{-\kappa\gamma(q-i-j)^2}+ \tag{5.23}$$

$$+\sum_{i=0}^{q-1}(-1)^i \frac{\lambda}{e}e^{(q-1-i)g(\mathrm{s}_t)\kappa(T-t)}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^i \frac{1}{i!}\sum_{j=0}^{q-1-i}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!}e^{-\kappa\gamma(q-1-i-j)^2} \tag{5.24}$$

sums to zero.

Let us now focus on 5.23. By making a change of indices, we find that 5.23 is equal to

$$-\sum_{i=0}^{q-1}(-1)^i e^{(q-1-i)g(\mathrm{s}_t)\kappa(T-t)}\left(\frac{\lambda}{e}\right)^{i+1}\left(\frac{1}{g(\mathrm{s}_t)\kappa}\right)^i \frac{1}{i!}\sum_{j=0}^{q-1-i}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^j \frac{1}{j!}e^{-\kappa\gamma(q-1-i-j)^2},$$

which vanishes with 5.24. Hence we have proven that 5.16 is indeed the solution of the system 5.14. □

Remark: Special Case with No Drift Term

We have just demonstrated an analytical strategy derived through stochastic control techniques. This strategy allows for the inclusion of a drift term in the dynamics of the mid-price, obtained through predictive models, and provides an indication of where the mid-price will move. This enables the development of more intelligent strategies that are capable of perceiving signals from the market and using them to their advantage. As a comparative strategy, we will consider the same theoretical model, but without the drift term (i.e., when $g(\mathrm{s}_t) = 0$). In this way, the mid-price M_t will have the following dynamic:

$$dM_t = \sigma_t dW_t, \tag{5.25}$$

where, as before, σ_t is the volatility, and $W = (W_t)_{0\leq t\leq T}$ is a standard Brownian motion. With this setup, [9] shows that the optimal execution strategy for a sell limit order is

$$\delta^*(t,q) = \frac{1}{\kappa}\left[1 + \log\frac{\sum_{n=0}^{q}\frac{\tilde{\lambda}^n}{n!}e^{-\kappa\alpha(q-n)^2}(T-t)^n}{\sum_{n=0}^{q-1}\frac{\tilde{\lambda}^n}{n!}e^{-\kappa\alpha(q-1-n)^2}(T-t)^n}\right] \tag{5.26}$$

This strategy will be used as a benchmark and, therefore, will serve as a basis for comparison, as the execution strategy presented in this research (Equation 5.17) is an extension of this model. This strategy will be referred to as the No-Drift Strategy.

5.3 Theoretical Comparison of Different Strategies

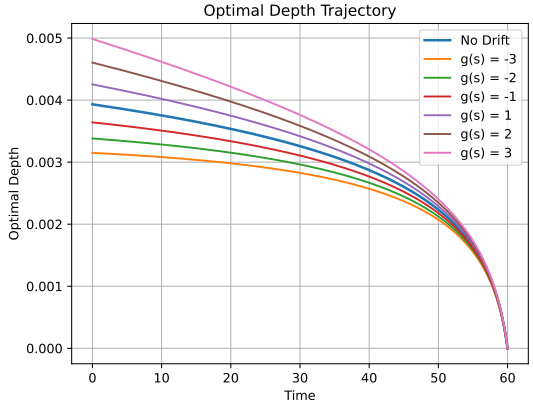
We have introduced two strategies: the strategy with drift, as given by Equation 5.17 (the main contribution of this thesis to the scientific research), and the strategy without drift (Equation 5.26). We now proceed to theoretically examine their behavior. For this purpose, we assume that the terminal time is $T = 60$ seconds, and we consider the possible values of q to be 1 and 2 (this is to relate to the placement of child orders, and thus we assume that the unit of measure is €500,000). We will use two different values for γ, namely 0.001 and 0.0001, and finally, we assume that κ is equal to 1000, and λ is equal to 50/60. These parameters were chosen based on what was chosen in [9], and thus to show an appropriate theoretical comparison. The values that $g(\mathrm{s}_t)$ can assume range from -3 to 3, and we use these to illustrate what happens and how the new strategy behaves in comparison with the strategy without drift. The theoretical simulations are shown in Figure 5.1, where the depth is shown on the y-axis as a function of time.

As illustrated in Figure 5.1, the value of the optimal depth of the limit order exhibits an interesting behavior, aligning well with theoretical intuitions. Depending on the type of signal perceived from the data, the strategy provides a different optimal price. Moreover, it ultimately converges to the value obtained without drift (calculated using Equation 5.26), thereby demonstrating the correctness of the model extension approach. It is crucial to note that as the value of $g(\mathrm{s}_t)$ varies, the price also changes significantly. The more negative $g(\mathrm{s}_t)$ is, the lower the price compared to other strategies. Conversely, the more positive $g(\mathrm{s}_t)$ is, the higher the price will be.

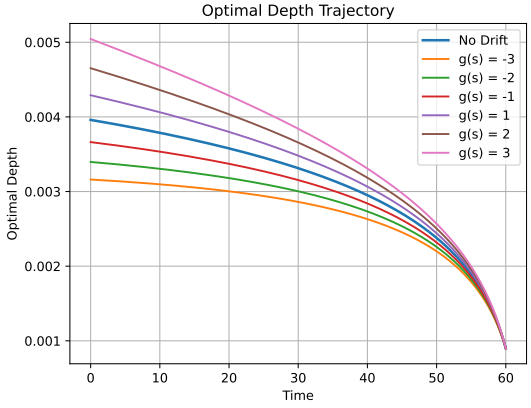
This observation aligns well with theoretical expectations. If, based on our predictive model, the current data from the limit order book suggests that the mid-price is likely to rise, we prefer to post the limit order at a higher price. This strategy allows us to wait and potentially achieve a higher profit. On the other hand, if our predictions indicate that the mid-price is declining, we would post the limit order at a more aggressive price to avoid missing the execution opportunity.

It is also important to observe how the values change with variations in q and γ. In this analysis, we are currently showing values for $q = 1, 2$, but this can be generalized. As the value of q increases, the strategy tends to select a more aggressive price because there is still a significant volume to be executed (as evidenced by the differences between Figures 5.1a, 5.1c and Figures 5.1b, 5.1d). Furthermore, as γ varies, we can observe that the aggressiveness of the strategy changes over time. Specifically, if γ is higher (as depicted in Figures 5.1a and 5.1c)—indicating a greater penalty for reaching the terminal time without execution—the final optimal depth will be more aggressive to ensure the order is executed as soon as possible. Conversely, if γ is lower (as shown in Figures 5.1b and 5.1d), the final optimal depth will be less aggressive due to the reduced penalty for reaching the terminal time without execution.

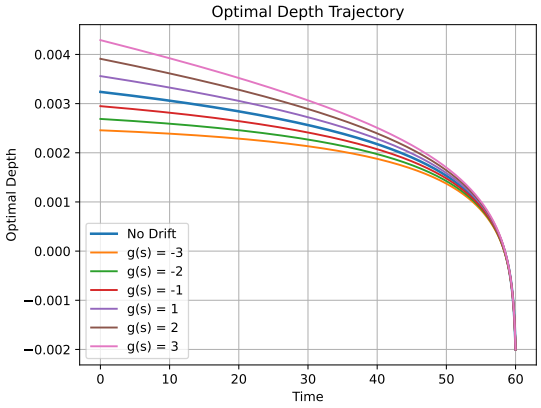
This dynamic adjustment mechanism highlights the adaptive nature of the strategy



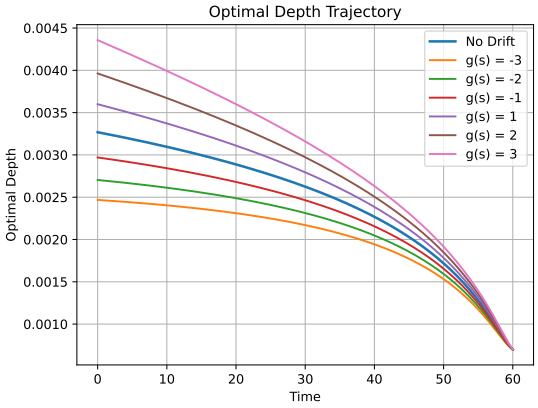
(a) $q = 1$, $\gamma = 0.001$



(b) $q = 1$, $\gamma = 0.0001$



(c) $q = 2$, $\gamma = 0.001$



(d) $q = 2$, $\gamma = 0.0001$

Figure 5.1: Optimal depth at which the agent posts the sell limit order as function of time. The common parameters are $T = 60$, $\lambda = 50/60$, $\kappa = 1000$.

in response to varying market signals. By adjusting the limit order price according to the predicted market trend, the strategy optimizes execution and profitability. The model's ability to converge to the no-drift value further validates its robustness and reliability. Such behavior underscores the importance of incorporating predictive signals into trading strategies.

There is indeed a significant challenge in utilizing this strategy, namely the selection of parameters. This challenge is also pertinent to the development of this thesis, as parameter selection heavily relies on the available data. For parameter calibration to be more efficient, historical limit order book data should be available (for the purposes

of this research, historical limit order book data for the FX Spot Market is available). However, it is equally important to have access to data describing the market orders entering the market. Such data is essential for understanding their frequency, volume, and order type. Unfortunately, this information is not available for the purposes of this thesis, necessitating assumptions or simulations to best calibrate the model with the required data. Consequently, in this chapter, we have decided to present theoretical values not calibrated to the data, to effectively demonstrate how the strategy theoretically behaves, given the calibration difficulties due to data limitations.

Additionally, the strategy is highly sensitive to parameter variations. Significant changes in parameters result in changes to the strategy's shape. However, what remains consistent is that if the value of $g(\mathrm{s}_t)$ becomes negative (or positive), the optimal depth value will be lower (or higher) compared to the value without drift.

Despite these challenges, demonstrating the theoretical behavior of the strategy provides valuable insights. It allows us to observe how the strategy would perform under ideal conditions and underscores the importance of accurate data for practical implementation. The consistent response of the optimal depth to changes in the signals further validates the model's theoretical foundations, even though practical calibration remains challenging.

Ultimately, addressing these data limitations would significantly enhance the model's applicability and robustness. Future research should focus on obtaining comprehensive market order data to improve parameter calibration and strategy performance. This would bridge the gap between theoretical models and real-world trading scenarios, leading to more effective and reliable trading strategies.

6 Empirical Experiments

In previous chapters, we have explored predictive models used to forecast the future movement of the mid-price (Chapter 4), as well as methods for integrating these predictions into our stochastic model, which describes the dynamics of mid-price movement (Chapter 5). Subsequently, we developed an optimal strategy utilizing stochastic optimal control techniques. This process allowed us to derive an analytical formula for our strategy, as shown in Equation 5.17. As previously explained, this represents an extension of the strategy that does not incorporate any type of prediction, as demonstrated in Equation 5.26. The new strategy, by including the prediction of the movement of the mid-price, can better identify a more advantageous price for placing sell limit orders.

In this chapter, we will explain how we built a backtesting environment that we will use to determine which strategy yields the greatest profit under identical conditions. We will compare the optimal strategy derived from this research with the No-Drift strategy to verify that such an approach can generate tangible benefits.

Additionally, we will use the WM/Reuters (WMR) FX benchmark as another benchmark. The WMR FX benchmarks are essential financial tools widely used to determine the relative values of different currencies globally. These benchmarks are provided by the WM company in partnership with Reuters, commonly referred to as the "WMR rates" or "4 PM fix" [26]. The WMR FX benchmarks aim to offer standardized, objective, and reliable exchange rates that reflect market conditions at a specific time of day. They are calculated from the trades and order rates during a five-minute window around 4:00 PM London time, a period of high trading and liquidity, thus believed to provide representative pricing.

6.1 Backtesting Environment

To simulate the strategies and observe their behavior on historical data, it is now necessary to construct a backtesting environment. This environment should be capable of replaying the market based on historical data, effectively reconstructing past market conditions and evaluating the performance of our strategies. However, this task is filled with challenges, as accurately simulating past events using only LOB data is complex.

With the available LOB data, we can determine the state of the order book at any given moment. However, it is difficult to reconstruct the exact causes of changes in the LOB state. For instance, changes in the LOB could result from the arrival of a market order, which would alter the best ask or best bid. Alternatively, these changes could be due to order cancellations or modifications. Another significant difficulty lies in incorporating the market impact of our limit orders. When our strategy places a limit order, it can influence the market by altering the LOB and potentially affecting subsequent market orders. Accurately modeling this impact is essential for realistic simulations but is inherently complex.

Despite these challenges, constructing a robust backtesting environment is important for accurately assessing trading strategies. Future research focusing on these areas could significantly refine backtesting accuracy. Steps such as acquiring more comprehensive data, refining simulation assumptions, and developing sophisticated models to assess market impact are essential for more reliable and insightful backtesting outcomes.

6.1.1 Market Order Simulation

We now turn our attention to simulating market orders. As noted earlier, direct market order data is not available for this study, even though such data is critical for understanding market dynamics due to its role in executing limit orders. Therefore, we must rely on assumed frequencies for the arrival of market orders, based on values derived from [22], which are detailed in Table 6.1.

We will consider the average of those frequencies, hence the arrival frequencies of market orders for our research are shown in Table 6.2.

It is important to note that the frequency of market orders increases as the spread decreases. This occurs because the bid and ask prices are closer, making it more attractive for traders to execute market orders due to reduced transaction costs and increased liquidity. A smaller spread implies less price slippage and higher execution probability, encouraging higher trading activity.

For the simulation of market order arrivals, we assume they follow a Poisson process, which is explained in more details in Appendix C. The Poisson process is a common model for describing the random occurrence of events over time, characterized by a constant average rate of occurrence. It is particularly suitable for modeling market order arrivals in our case due to several reasons:

- **Simplification for Modeling:** Using a Poisson process simplifies the mathematical and computational modeling of order arrivals, making it easier to integrate into the backtesting framework.
- **Historical Precedent:** Prior research and practical implementations in financial markets have successfully utilized Poisson processes to model order arrivals,

Table 6.1: Arrival rate per second of market orders for each spread size Δ in ticks.

(a) 7-6-2021 - 11-6-2021.

Δ (in ticks)	Sell	Buy	Total
1	0.21	0.27	0.48
2	0.09	0.08	0.17
3	0.05	0.04	0.09
4	0.04	0.03	0.07
5	0.04	0.03	0.07

(b) 14-6-2021 - 18-6-2021.

Δ (in ticks)	Sell	Buy	Total
1	0.24	0.30	0.54
2	0.11	0.11	0.22
3	0.05	0.05	0.09
4	0.04	0.04	0.07
5	0.04	0.03	0.07

(c) 21-6-2021 - 25-6-2021.

Δ (in ticks)	Sell	Buy	Total
1	0.27	0.27	0.54
2	0.11	0.10	0.22
3	0.05	0.05	0.11
4	0.05	0.04	0.09
5	0.04	0.05	0.09

(d) 28-6-2021 - 2-7-2021.

Δ (in ticks)	Sell	Buy	Total
1	0.16	0.17	0.34
2	0.10	0.09	0.19
3	0.07	0.06	0.13
4	0.05	0.05	0.11
5	0.08	0.05	0.13

providing a validated approach, such as in [9] or [8].

By assuming that market orders follow a Poisson distribution, we can simulate their arrival times more realistically. The key parameter in this process is the rate parameter f, which represents the average number of market orders arriving per unit of time.

Incorporating a Poisson process into our simulation allows us to approximate the random yet statistically quantifiable nature of market order arrivals. However, it is important to recognize the limitations of this approach. The assumption of a constant arrival rate may not always hold, particularly during periods of market turbulence or significant events that cause sudden spikes in trading activity. Therefore, future research could explore more sophisticated models that account for time-varying arrival rates and other factors influencing market order dynamics, such as Hawkes processes ([29], [19]).

Once the bid-ask spread is established and thus the frequency of incoming market orders is known, it becomes important to integrate this information within the back-testing environment. Consequently, each time a limit order is placed, the simulator will examine the spread and calculate the arrival time of the next market order. Whenever the price of the limit order is updated, the spread will be recalculated. If a change in the spread occurs, a new arrival time for the subsequent market order will be determined, and the final time will be calculated as the average of the two. This approach accounts for the potential impact of the preceding order on the market dynamics.

Table 6.2: Average arrival rate per second of market orders for each spread size Δ in ticks.

Δ (in ticks)	Sell	Buy	Total
1	0.22	0.2525	0.475
2	0.1025	0.095	0.2
3	0.055	0.05	0.105
4	0.045	0.04	0.085
5	0.05	0.04	0.09

It is important to note that for our simulations, we assume that the market orders always have a constant volume. This assumption is plausible, as typically, based on [22], the volume of market orders is either 500,000 euros or 1,000,000 euros. Given that we consider the volume of child orders to be 1,000,000 euros. This strategy involves breaking down a larger "parent order", which is the entire quantity to be traded, into smaller, less conspicuous child orders that match common market transaction sizes [22]. By doing so, a child order can either be executed in one go if a market order matches the size of 1,000,000 euros, or it may require two transactions of 500,000 euros each. This approach ensures that the execution of large orders doesn't visibly affect market dynamics or alert other traders to the strategy being employed, thereby facilitating smoother, more stealthy market operations.

6.1.2 Execution of Limit Orders

Upon simulating the arrival of market orders, it is essential to understand the dynamics of the simulator. The process begins at the initial time when a limit order is intended to be placed in the limit order book. The price of this order is calculated based on the selected trading strategy. Once determined, the simulator assesses the time until the next market order based on the current spread, and this time is recorded.

Subsequently, the limit order book is scanned to verify if there exists a limit order in the opposite direction that matches the trader's own limit order, leading to its execution. This verification continues until there is a price update, which in our case will happen once every 50 milliseconds. This quantity will also be utilized in our execution strategy. Indeed, the price will be updated every 50ms, as this not only serves to keep the price aligned with market developments but also represents the minimum acceptable interval for updating the price. In the venues we consider for our simulations, there is a maximum limit of 20 limit orders that can be placed per second. Upon updating the price, the arrival of the next market order is recalculated, and the simulator persistently checks for the execution of the trader's limit order by an opposing limit order.

At the point when the time for the market order arrives, the simulator checks whether the trader's limit order is within the current spread. If it is, one unit of the order is executed. This process continues until the order is entirely executed, either by market orders or by a limit order in the opposite direction.

6.2 Execution Strategy Experimentation with Market Data

The backtesting system described in the previous section is complex, yet it is capable of simulating the FX Spot Market dynamics fairly realistically using historical market data provided by MN. In this section, we will present the results of the strategy comparison and determine which performs better relative to the others. For our simulations, we will assume that an agent wishes to place a sell limit order, as this serves as the example for our thesis. The various execution strategies will be used under the same conditions in the following manner: 10 different days throughout the year will be selected, and from these, 50 times will be randomly chosen, totaling 500 simulations. For each of these times, the execution strategies will be independently initiated in the backtesting environment. Once the order is executed, the final execution price will be recorded and used as the value for comparison.

6.2.1 Hyperparameters tuning

Firstly, in Chapter 5, we outlined the theoretical strategies. It is now crucial to explain how in this research the parameters for these strategies were calibrated. It is also important to note that this phase is highly complex as it heavily relies on the available data, and choosing these parameters appropriately is not straightforward. Additionally, a further challenge arises from the lack of available data on market orders, requiring some parameters to be based on simulations conducted by us. Future research could focus on improving the calibration of these strategies.

Recalling how we modeled the problem of an agent wanting to post a sell limit order in Chapter 5, we need to identify the following parameters: T, which is the terminal time; λ and κ, which are part of the intensity parameter of counting process $N^\delta = (N_t^\delta)_{0\leq t\leq T}$; q, which is the agent inventory to be liquidated; γ, which is the liquidation penalty.

Some of these parameters are straightforward to choose, as they depend on how we wish to execute the order. For instance, the terminal time T should be selected based on our execution requirements and the amount of time available for each child order. For our simulations, we will choose $T = 10$ seconds. The parameter γ also depends on our requirements. Based on its usage in the literature, such as in [9] and [8], we will choose

γ to be 0.001. Also q depends on our requirements. Typically, a child order is selected with a volume that is designed to be "invisible" in the market, meaning it does not alter market conditions or have a market impact. Moreover, it usually has a standard volume used by all participants to prevent identification of the order's originator. In our case, this standard volume is 1 million euros. Our strategy, however, accommodates significantly larger quantities, q, in units. Drawing on analyses by [22] and to illustrate scenarios where the q units to be liquidated exceed 1, we will assume that the volume unit is 500,000 euros. Consequently, incoming market orders are also of 500,000 euros. This adjustment results in our q being equal to 2.

The parameters λ and κ, part of the intensity of the counting process N^δ with an intensity of $\lambda e^{-\kappa\delta_t}$, are more challenging to determine because they represent the probability that our sell limit order will be executed. Therefore, this quantity represents the fill probability. Considerable research has been conducted on this probability, and it is generally accepted in the literature that it can be represented by an exponential function [8].

To calibrate these parameters, as previously mentioned, it is necessary to gather information about market orders entering the market. [8] does not explicitly provide details on how these parameters were chosen; thus, in this research, we propose a method to determine them via simulation. We utilized Monte Carlo simulations to compute these fill probabilities: for each potential depth δ in the limit order book, across several days, we randomly selected a total of 10,000 initial times. For each of these starting points, we observed whether the placed sell limit order at that depth was executed within the next 50 milliseconds. This sell limit order can be executed in two ways: either the best bid reaches this order and, if the volume is sufficient, executes the sell limit order, or it is executed via the arrival of a market order. Additionally, our strategy includes a condition that if the price of the sell limit order is set higher than the best ask, it is then placed directly at the best ask (similarly for the best bid). This is because it has been observed that orders placed outside the spread are unlikely to be executed [22], and it also simplifies some considerations for simulating the market and our trading venues.

Moreover, our simulations assume that the arrival of market orders is dependent of the spread, which implies that we take as frequency of the arrival of market orders the values shown in Table 6.1. For each depth of the limit order, the spread is calculated, which will vary depending on the state of the limit order book of the simulation under consideration, and based on that, the frequency will be determined.

Upon completing these simulations on historical data, we fitted an exponential function of the form $\lambda e^{-\kappa\delta}$. Both the simulation results of the execution probabilities for each depth and the fitted exponential function are displayed in Figure 6.1. The parameters found are $\lambda = 0.027$, and $\kappa = 1.38$.

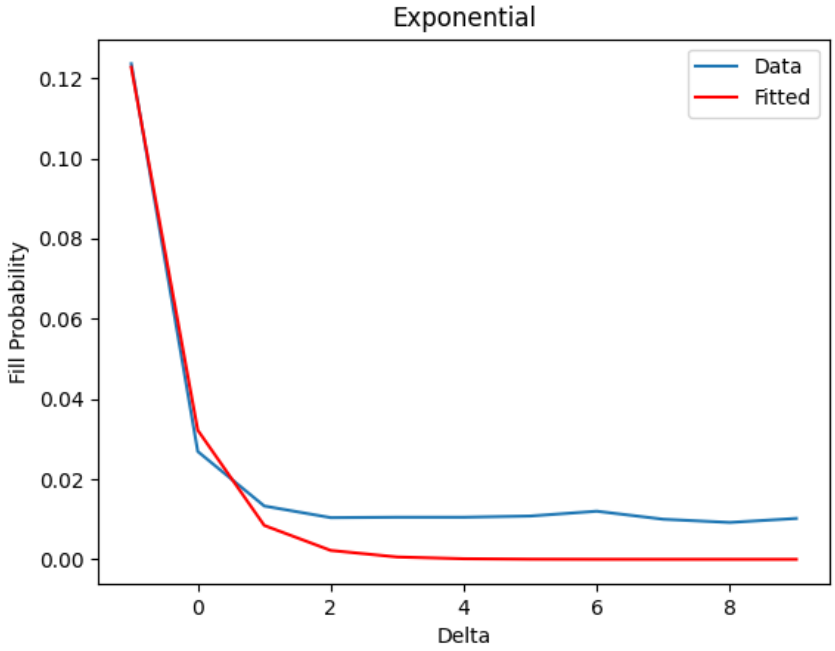


Figure 6.1: This plot visualizes the results from Monte Carlo simulations alongside the fitted exponential decay function $\lambda e^{-\kappa\delta}$, illustrating the probability of sell limit order execution within 50 milliseconds at varying depths δ.

6.2.2 Comparison of Execution Strategies' Performance

We will now examine the performance of the execution strategies. To do this, we consider three different execution strategies, chosen based on the models discussed in this thesis:

- OF-CNN-LSTM: this is the execution strategy where the $g(\mathrm{s}_t)$ is the Order-Flow-CNN-LSTM model, as explained in 4.3, and the optimal depth of the sell limit order is given by Equation 5.17.
- FPovi^5: this is the execution strategy where the $g(\mathrm{s}_t)$ is the Forward-looking Price Impact model, with ovi^5, as explained in 4.2, and the optimal depth of the sell limit order is given by Equation 5.17
- No-Drift: this is the execution strategy where $g(\mathrm{s}_t) = 0$, and the optimal depth is given by Equation 5.26.

These are the three execution strategies that we have discussed throughout the thesis and have chosen to consider for this research. Using Monte Carlo simulations,

after being simulated 500 times as previously explained, we calculated the average difference in execution prices (in ticks) between the various execution strategies, as well as the standard deviation. The results are shown in Table 6.3.

Strategy 1	**Strategy 2**	**Average Difference Execution Price Strategy 1 - Strategy 2**	**Standard Deviation**
OF-CNN-LSTM	No-Drift	1.2	3.6
FPovi^5	No-Drift	0.95	3.0
OF-CNN-LSTM	FPovi^5	0.25	2.7

Table 6.3: Comparison of execution strategies.

Table 6.3 displays the three strategies, presenting the average difference (and the standard deviation) in execution prices among them. Notably, the OF-CNN-LSTM strategy performs best, as on average, a sell limit order placed using this strategy is executed at a price that is 1.2 ticks higher than the execution price of a sell limit order placed using the No-Drift strategy. Additionally, using the FPovi^5 strategy, the average execution price of a sell limit order is 0.95 ticks higher compared to one executed using the No-Drift strategy. The third row is related to the first two in the table.

Regarding the standard deviation, it is relatively high. However, it still shows good values, and on average, execution strategies that include the effect of the drift term, which models the mid-price prediction based on the limit order book imbalance, tend to execute their sell limit orders at higher prices.

Additionally, to better illustrate the performance of the strategies, Figure 6.2 shows the dynamics of price evolution as determined by the strategies.

The figure features five different lines. The red line represents the Best Ask price from the limit order book during that time interval, while the purple line represents the Best Bid. These values are based on historical limit order book data, hence they are real. The other three lines represent the sell limit order price determined by the three different strategies previously discussed: the blue line represents the FPovi^5 strategy, the orange line represents the No-Drift strategy, and finally, the green line represents the OF-CNN-LSTM strategy.

Let us now focus on the individual lines representing the sell limit order prices obtained with the three strategies. Starting with the green line: as we can observe, the strategy initially sets a price, and then, based on the data from the limit order book, decides to lower it. This movement allows the order to be executed at point A at a price of 1.05322. This execution occurs because the sell limit order matches the Best Bid, effectively becoming a market order. This execution also happens due to a sudden increase in the Best Bid, which matched the sell limit order placed by our agent.

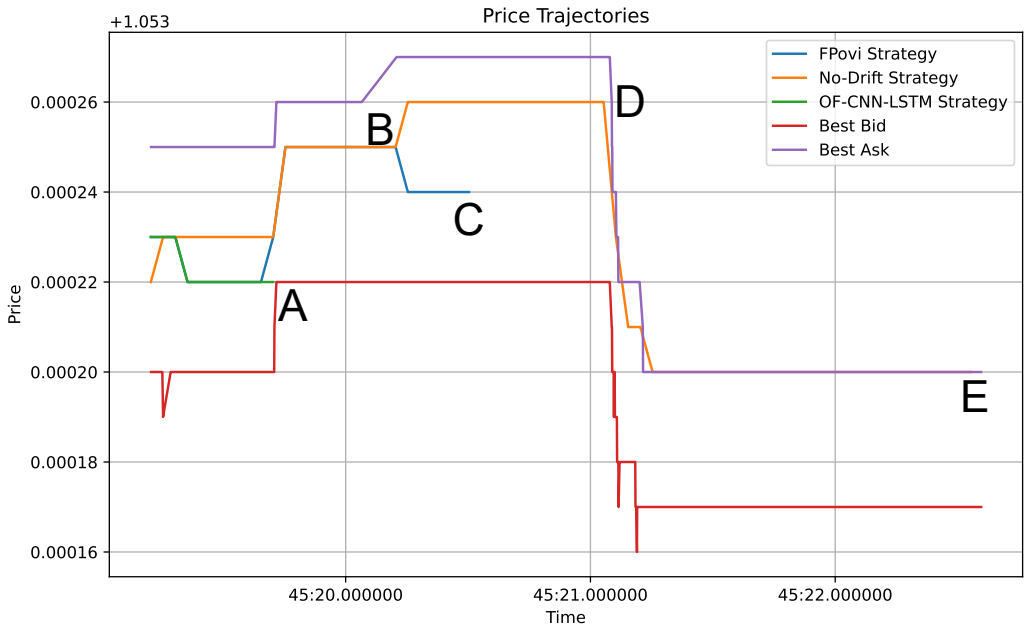


Figure 6.2: Price evolution of the three strategies, in the same condition.

Now let us have a look at the blue line, which represents the FPovi[5] strategy. The price undergoes movements similar to the green line but then rises to 1.05325, aligning with the price on the red line, i.e. the Best Bid. However, at point B, it decides to lower the price to 1.05324, by one tick, making the order more aggressive. This adjustment is based on the data from the limit order book, as at that moment the imbalance had changed, and the strategy suggested lowering the price, presumably because the average price was expected to drop. This caused a narrower spread, thus increasing the frequency of market orders, leading to the execution of the sell limit order at point C at a price of 1.05324 by a market order.

Finally, examining the orange line, representing the No-Drift strategy, the situation is different. This strategy does not make any predictions, hence it follows the price trend. It was unable to anticipate a price drop, so it follows this drop at point D, remaining at a price equal to the Best Ask until it is executed by a market order at point E at a price of 1.05320. This results in the final execution price obtained with the No-Drift strategy being lower, thus yielding less profit compared to the execution prices obtained with the other strategies.

The execution prices are shown in Table 6.4, and, as we can see, the No-Drift execution price is the worse, given by the fact that it was not able to predict the

movement of the mid-price in advance.

Strategy	Execution Price
OF-CNN-LSTM	1.05322
FPovi[5]	1.05324
No-Drift	1.05320

Table 6.4: Final execution prices of the three strategies.

An important final consideration concerns the execution time of these strategies. As noted in Chapter 5, having found an analytical solution to the optimization problem ensures that the optimal depth can be determined very quickly (less than a millisecond), which represents a significant achievement of this research. If we also consider the time required to predict mid-price movements using the FPovi[5], the total execution time remains on average under one millisecond. However, in the case of using the OF-CNN-LSTM model, the total time is around 35 milliseconds, which, as previously noted, is due to the machine and programming language used. This timing, as observed in Chapter 4, can be reduced to about a millisecond with the use of appropriate machines and techniques. Thus, these strategies are viable in a real-world setting as they are fast enough.

6.2.3 WMR Benchmarks

The WM/Reuters (WMR) FX benchmarks are financial tools used globally to determine the exchange rates for currencies at specific times throughout the trading day. These rates are important for valuing portfolios, measuring performance, and executing substantial currency trades at fair, transparent rates, since they are commonly used in industry [26].

The WMR FX benchmarks are calculated using a methodology that ensures accuracy and fairness in rate determination. The most referenced WMR rate, the 4 PM London fix, is based on actual transactions and order rates sourced from global forex markets within a defined five-minute window (two minutes and thirty seconds before and after 4 PM London time). By taking a median of all eligible trades and order rates during this period, the benchmark reflects a consistent and representative exchange rate for each currency pair.

These benchmarks are widely utilized in the industry, and therefore, we found it pertinent to assess how our strategy performs in comparison with these benchmarks. To do this, we considered various randomly chosen days (January 17th, February 6th, February 19th, March 5th, March 21st, April 11th, April 23rd, April 26th, June 17th, July 1st, July 2nd, July 3rd) and executed 30 child orders of 1 million each, posted

every 10 seconds starting from two and a half minutes before 4 PM London Time, using the OF-CNN-LSTM execution strategy. We then calculated the average execution price for each day to obtain our average exchange rate between the euro and the dollar at the time of posting a sell limit order, which we then compared to the benchmarks.

However, the WMR benchmarks are not highly precise as they are approximated to the fifth decimal place to either 0 or 5, thus losing some pricing flexibility by a few ticks. This makes it impossible to directly compare our execution price with the benchmarks, as a difference of 5 ticks is considerable, especially since our analysis considers ticks as the minimal variation. To address this issue and to obtain benchmarks comparable to the WMR, we decided to use the average of the mid-prices over those five minutes as our WMR benchmarks. Therefore, we will compare our average sell limit order price over five minutes with the average mid-price, calculate the difference, and finally, analyze the average over the various days considered.

On the days considered, after utilizing the backtesting environment described earlier and running our strategy within those five minutes, the average difference between the execution price of the sell limit orders and the empirically calculated average mid-price for each day is $5 \cdot 10^{-7}$, with a standard deviation of $1.2 \cdot 10^{-6}$. This indicates that our strategy is capable of executing sell limit orders at a price, on average, higher than our empirical WMR benchmarks, thereby facilitating a favorable profit.

7 Conclusions

Algorithmic trading is increasingly prevalent, and many companies, including MN which collaborated on this thesis, utilize it to execute trades at favorable prices. Specifically, MN has developed ALGO, an autonomous execution algorithm that operates in the FX Spot Market. The purpose of this thesis is to identify a novel optimal execution strategy for limit orders. This is based on the information of the limit order book (LOB), which is used to make prediction of the movements of the mid-price using data-driven models.

To achieve this, we first defined a new way of measuring the order book volume imbalance, which incorporates multiple price levels of the LOB in a weighted manner. This measure provides clearer indications of future mid-price movements compared to a measure that considers only the first level, i.e., just the best ask and best bid.

Subsequently, this research proposes methods for utilizing information from the limit order book and its imbalances (not only through the order book volume imbalance but also through the order flow, which takes into account the temporal evolution of the imbalances) to attempt to predict future mid-price movements. The main models fall into two categories: the first is based on a model similar to regression (the Forward-Looking Price and Cross Impact model), where the features considered are the imbalances of one's own order book, but also those of others that are related to the one being considered. Tested on real LOB data from two different venues, we conclude that in our case including the cross effect of two venues does not aid in the prediction of movement, and that including multiple levels of the limit order book provides more accurate predictions. Subsequently, the second type of predictive models we consider is based on deep learning models. The base model is a Convolutional Neural Network (CNN) (capable of aggregating LOB data together, implicitly defining a measure of imbalance), followed by a Long-Short Term Memory network (which takes the aggregated data from the CNN and uses it, considering their temporal relationship, to make future predictions of return). This research extends this concept to the use of multiple venues by introducing a new and innovative method: a new layer is added to the CNN model, enabling the utilization of multiple venues. Here again, the results are promising, however, it is not necessary to include the data from the other order book as it does not provide more accurate predictions.

Once our predictive models are constructed, the research continues with the stochastic modeling of the mid-price and the actions of an agent who wants to post a sell limit order. This thesis proposes a new stochastic model, which to our knowledge has not yet been proposed in the literature, that includes in the drift term the predictions made by our models: upon receiving a signal from the limit order book (for instance, an imbalance), the model makes a prediction of future return based on historical data, and this is used as the drift term of our stochastic model for the mid-price. Subsequently, a stochastic optimization problem is defined which is solved analytically, and leads to an explicit solution, which corresponds to our optimal execution strategy. This new strategy effectively reflects the predictions of the mid-price: if the forecast suggests that the price will fall (or rise), and we wish to sell, the strategy will choose a lower (or higher) price for the limit order, aiming to execute the order as soon as possible and secure a higher profit.

Finally, in this research, we have developed a backtesting environment that allows for the simulation of the market using historical data to test our strategies. The results are excellent: the new strategy achieves a greater profit compared to a strategy that does not include the drift term and is therefore unable to adapt to current market conditions. Furthermore, the best predictive model to use in the strategy is the one based on deep learning. Additionally, a final comparison is made with the WMR benchmarks, which were empirically determined in this thesis. This strategy enables sell limit orders to be executed on average at a higher price than these benchmarks.

However, this research also has its limitations. Firstly, all models were trained and calibrated over a one-week data interval to simplify both computational costs and the amount of data used. This provides a good degree of generalizability, which, however, could be made more robust with the use of a larger dataset. Additionally, the strategy was tested only on the FX Spot Market and not on other types of data. Moreover, the predictions of the models are good, but not excellent. Obviously, we cannot expect to predict future returns with high accuracy, due to the influence of numerous unpredictable factors, including economic news, investor decisions, market volatility, and geopolitical events, which complicate the accurate modeling of its dynamics. In addiction, we assumed that the arrival of market orders follows an independent Poisson process, which is a good assumption, but it could be made more general by considering the arrivals not as independent but as dependent on other arrivals. For this, one could utilize alternative models, such as Hawkes processes. Another limitation is the execution time of the strategies. As observed, this should not be a problem since the maximum execution time is expected to be around one millisecond. However, this should be empirically tested and verified using more powerful machines and more efficient programming languages. A final limitation concerns the backtesting environment where all the strategies were tested. This environment was constructed based on certain market assumptions, and many other factors could be included in simulations to

make them more realistic. Since simulations are not entirely true to life, the actual performance of these strategies might differ in reality. However, this does not pose a problem for comparing strategies, as they were executed under the same conditions and within the same environments.

This study lays the groundwork for many potential future investigations. First and foremost, the strategy allows for the incorporation of any predictive function, thus opening the door to exploring various types of functions, including those based on time series analysis, or other forms of machine learning and deep learning models. Additionally, there is scope to research alternative methods for calibrating the parameters used in the optimal strategy, as well as to enhance the backtesting environment to make it more realistic and thus have more truthful simulation results.

Another potential area of research is the application of this strategy across different markets, such as equities, since the methodology should be transferable. Lastly, there is an opportunity to further extend and improve the deep learning model that operates across multiple venues (or stocks). This could also be adapted for use in contexts involving different assets, not just different venues, potentially broadening the applicability of the strategy.

Bibliography

[1] Hervé Abdi and Lynne J Williams. "Principal component analysis". In: *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010), pp. 433–459.

[2] Robert Almgren and Neil Chriss. "Optimal execution of portfolio transactions". In: *Journal of Risk* 3 (2001), pp. 5–40.

[3] Bank for International Settlements. *Triennial Central Bank Survey of Foreign Exchange and Over-the-counter (OTC) Derivatives Markets in 2022*. Bank for International Settlements. 2022. URL: `https://www.bis.org/statistics/rpfx22_fx.htm` (visited on 07/24/2023).

[4] Dimitris Bertsimas and Andrew W Lo. "Optimal control of execution costs". In: *Journal of financial markets* 1.1 (1998), pp. 1–50.

[5] Francesco Capponi and Rama Cont. "Multi-asset market impact and order flow commonality". In: *Available at SSRN 3706390* (2020).

[6] Alvaro Cartea, Ryan Donnelly, and Sebastian Jaimungal. "Enhancing trading strategies with order book signals". In: *Applied Mathematical Finance* 25.1 (2018), pp. 1–35.

[7] Álvaro Cartea and Sebastian Jaimungal. "Incorporating order-flow into optimal execution". In: *Mathematics and Financial Economics* 10 (2016), pp. 339–364.

[8] Álvaro Cartea and Sebastian Jaimungal. "Optimal execution with limit and market orders". In: *Quantitative Finance* 15.8 (2015), pp. 1279–1291.

[9] Álvaro Cartea, Sebastian Jaimungal, and José Penalva. *Algorithmic and high-frequency trading*. Cambridge University Press, 2015.

[10] Álvaro Cartea, Sebastian Jaimungal, and Leandro Sánchez-Betancourt. "Deep reinforcement learning for algorithmic trading". In: *Available at SSRN 3812473* (2021).

[11] Rama Cont, Mihai Cucuringu, and Chao Zhang. "Cross-impact of order flow imbalance in equity markets". In: *Quantitative Finance* 23.10 (2023), pp. 1373–1393.

[12] Rama Cont and Arseniy Kukanov. "Optimal order placement in limit order markets". In: *Quantitative Finance* 17.1 (2017), pp. 21–39.

[13] Rama Cont, Arseniy Kukanov, and Sasha Stoikov. "The price impact of order book events". In: *Journal of financial econometrics* 12.1 (2014), pp. 47–88.

[14] Saeid Fallahpour et al. "Pairs trading strategy optimization using the reinforcement learning method: a cointegration approach". In: *Soft Computing* 20 (2016), pp. 5051–5066.

[15] Luca Ferretti. "Algorithmic FX trading: a new backtesting approach for the venue selection". Master's thesis. TU Delft, 2023.

[16] The Ducth Authority for the Financial Markets (AFM). *AFM Market Watch Edition 8 | Algorithmic Trading*. 2023. URL: `https://www.afm.nl/en/sector/themas/beurzen-en-effecten/afm-market-watch` (visited on 08/03/2024).

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[18] Fabien Guilbaud and Huyen Pham. "Optimal high-frequency trading with limit and market orders". In: *Quantitative Finance* 13.1 (2013), pp. 79–94.

[19] Alan G Hawkes. "Hawkes processes and their applications to finance: a review". In: *Quantitative Finance* 18.2 (2018), pp. 193–198.

[20] Petter N Kolm, Jeremy Turiel, and Nicholas Westray. "Deep order flow imbalance: Extracting alpha at multiple horizons from the limit order book". In: *Mathematical Finance* 33.4 (2023), pp. 1044–1081.

[21] Charles-Albert Lehalle, Othmane Mounjid, and Mathieu Rosenbaum. "Optimal liquidity-based trading tactics". In: *Stochastic Systems* 11.4 (2021), pp. 368–390.

[22] Felix Lokin and Fenghui Yu. "Fill Probabilities in a Limit Order Book with Applications to Foreign Exchange Spot Markets". In: (2024). URL: `http://resolver.tudelft.nl/uuid:55b62810-8dc4-453b-b7df-4092935d5d1e`.

[23] MN. URL: `https://www.mn.nl/` (visited on 08/03/2024).

[24] David S Moore and George P McCabe. *Introduction to the practice of statistics.* WH Freeman/Times Books/Henry Holt & Co, 1989.

[25] Brian Ning, Franco Ho Ting Lin, and Sebastian Jaimungal. "Double deep q-learning for optimal execution". In: *Applied Mathematical Finance* 28.4 (2021), pp. 361–380.

[26] FTSE Russell. *WMR FX Benchmarks*. `https://www.lseg.com/en/ftse-russell/benchmarks/wmr-fx-benchmarks`. [Online; accessed 14-July-2024].

[27] Sasha Stoikov and Rolf Waeber. "Optimal asset liquidation using limit order book information". In: *Available at SSRN 2113827* (2012).

[28] Avraam Tsantekidis et al. "Using deep learning for price prediction by exploiting stationary limit order book features". In: *Applied Soft Computing* 93 (2020), p. 106401.

[29] Ekaterina Vinkovskaya. *A point process model for the dynamics of limit order books*. Columbia University, 2014.

[30] Christopher JCH Watkins and Peter Dayan. "Q-learning". In: *Machine learning* 8 (1992), pp. 279–292.

[31] Ke Xu, Martin D Gould, and Sam D Howison. "Multi-level order-flow imbalance in a limit order book". In: *Market Microstructure and Liquidity* 4.03n04 (2018), p. 1950011.

[32] Pengfei Yu and Xuesong Yan. "Stock price prediction based on deep neural networks". In: *Neural Computing and Applications* 32.6 (2020), pp. 1609–1628.

A Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique utilized to emphasize variation and identify strong patterns in a dataset. It is frequently employed to reduce dimensions while preserving as much of the data's variability as possible. This chapter will explain what is PCA, based on [1].

A.1 Mathematical Foundations

PCA essentially transforms the original variables into a new set of variables, which are linear combinations of the original ones. These new variables, called *principal components*, are orthogonal and arranged in such a way that the first few retain most of the variation present in all of the original variables.

The mathematical foundation of PCA is based on the eigendecomposition of matrices. Consider a data matrix X of size $n \times p$, where n represents the number of observations and p the number of variables. PCA starts with the covariance matrix C of X, defined as:

$$C = \frac{1}{n-1}(X - h\bar{x}^T)^T(X - h\bar{x}^T)$$

where C is a $p \times p$ symmetric matrix, $\bar{x}$ is the mean vector of the columns of X, and h is a $n \times 1$ column vector of all 1s. The eigendecomposition of C is given by:

$$C = V\Lambda V^T$$

where V consists of the eigenvectors of C, and Λ is a diagonal matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ of C.

The principal components are obtained by projecting the original data onto the eigenvectors. The first principal component Z_1 is obtained by:

$$Z_1 = Xv_1$$

where v_1 is the eigenvector associated with the largest eigenvalue λ_1. This component captures the maximum variance in the data. The subsequent components are calculated in a similar manner, under the constraint that they are orthogonal to the previous components.

PCA reduces the dimensionality of the data by retaining only those components that significantly contribute to the variance. The number of components retained, k, is much smaller than p. This reduction is accomplished by keeping only the first k eigenvectors corresponding to the largest k eigenvalues.

B Derivation of Execution Strategy. Cases q = 2, 3, 4

In this appendix, we will continue with the derivation of the analytical solution of the execution strategy described in Chapter 5. Here, the cases $q = 2, 3, 4$ will be shown.

q = 2

We now proceed on calculating $w(t, 2)$ (i.e. the case $q = 2$),

$$\begin{cases} \partial_t w(t,2) + 2g(\mathrm{s}_t)\kappa\, w(t,2) + \frac{\lambda}{e} w(t,1) = 0, \\ w(T,2) = e^{-\kappa\gamma 4}, \\ w(t,0) = 1. \end{cases} \tag{B.1}$$

Similarly to the case $q = 1$, we obtain

$$\partial_t \left[e^{2g(\mathrm{s}_t)\kappa\, t} w(t,2) \right] = -\frac{\lambda}{e} e^{2g(\mathrm{s}_t)\kappa\, t} w(t,1),$$

and using Equation 5.15 and integrating wrt t, we obtain

$$\begin{aligned} e^{2g(\mathrm{s}_t)\kappa\, T} w(T,2) - e^{2g(\mathrm{s}_t)\kappa\, t} w(t,2) &= -\frac{\lambda}{e} \int_t^T e^{g(\mathrm{s}_t)\kappa\,(T+u)} \left[e^{-\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right] du + \\ &\quad + \frac{\lambda}{e} \int_t^T e^{2g(\mathrm{s}_t)\kappa\, u} \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} du = \\ &= -\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \left[e^{-\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right] \left(e^{2g(\mathrm{s}_t)\kappa\, T} - e^{g(\mathrm{s}_t)\kappa\,(T+t)} \right) + \\ &\quad + \frac{1}{2} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^2 \left(e^{2g(\mathrm{s}_t)\kappa\, T} - e^{2g(\mathrm{s}_t)\kappa\, t} \right) \end{aligned}$$

So, using the terminal condition,

$$e^{2g(\mathrm{s}_t)\kappa\, t}w(t,2) = e^{2g(\mathrm{s}_t)\kappa\, T}e^{-\kappa\gamma 4}+ \\ + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\left[e^{-\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right]\left(e^{2g(\mathrm{s}_t)\kappa\, T}-e^{g(\mathrm{s}_t)\kappa\,(T+t)}\right)+ \\ -\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2\left(e^{2g(\mathrm{s}_t)\kappa\, T}-e^{2g(\mathrm{s}_t)\kappa\, t}\right).$$

We can conclude that

$$w(t,2) = e^{2g(\mathrm{s}_t)\kappa\,(T-t)}e^{-\kappa\gamma 4}+ \\ + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\left[e^{-\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right]\left(e^{2g(\mathrm{s}_t)\kappa\,(T-t)}-e^{g(\mathrm{s}_t)\kappa\,(T-t)}\right)+ \\ -\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2\left(e^{2g(\mathrm{s}_t)\kappa\,(T-t)}-1\right),$$

which can be written as

$$w(t,2) = e^{2g(\mathrm{s}_t)\kappa\,(T-t)}\left[e^{-4\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\,e^{-\kappa\gamma}+\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2\right]+ \\ -\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\,e^{g(\mathrm{s}_t)\kappa\,(T-t)}\left[e^{-\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right] \\ +\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2. \tag{B.2}$$

q = 3

Let us now consider the case $q = 3$:

$$\begin{cases} \partial_t w(t,3) + 3g(\mathrm{s}_t)\kappa\, w(t,3) + \frac{\lambda}{e}w(t,2) = 0, \\ \\ w(T,3) = e^{-\kappa\gamma 9}, \\ \\ w(t,0) = 1. \end{cases} \tag{B.3}$$

Repeating the steps of the previous cases, we get

$$\partial_t\left[e^{3g(\mathrm{s}_t)\kappa\, t}w(t,3)\right] = -\frac{\lambda}{e}e^{3g(\mathrm{s}_t)\kappa\, t}w(t,2).$$

Using Equation B.2 and integrating wrt t, we obtain

$$
\begin{aligned}
e^{3g(\mathrm{s}_t)\kappa T}w(T,3) - e^{3g(\mathrm{s}_t)\kappa t}w(t,3) &= -\frac{\lambda}{e}\int_t^T e^{3g(\mathrm{s}_t)\kappa u}w(u,2)du \\
&= -\frac{\lambda}{e}\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^2\int_t^T e^{3g(\mathrm{s}_t)\kappa u}du + \\
&\quad -\frac{\lambda}{e}\left[e^{-4\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa e}e^{-\kappa\gamma} + \frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^2\right]e^{2g(\mathrm{s}_t)\kappa T}\cdot \\
&\quad \cdot\int_t^T e^{g(\mathrm{s}_t)\kappa u}du + \\
&\quad +\frac{\lambda}{e}\frac{\lambda}{g(\mathrm{s}_t)\kappa e}e^{g(\mathrm{s}_t)\kappa T}\left[e^{-\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right]\int_t^T e^{2g(\mathrm{s}_t)\kappa u}du \\
&= -\frac{1}{6}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^3\left(e^{3g(\mathrm{s}_t)\kappa T} - e^{3g(\mathrm{s}_t)\kappa t}\right) + \\
&\quad -\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\left[e^{-4\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa e}e^{-\kappa\gamma} + \frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^2\right]\cdot \\
&\quad \cdot\left(e^{3g(\mathrm{s}_t)\kappa T} - e^{g(\mathrm{s}_t)\kappa(2T+t)}\right) + \\
&\quad +\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^2\left[e^{-\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right]\left(e^{3g(\mathrm{s}_t)\kappa T} - e^{g(\mathrm{s}_t)\kappa(T+2t)}\right).
\end{aligned}
$$

So, we can conclude, by using the terminal condition, that

$$
\begin{aligned}
w(t,3) = e^{3g(\mathrm{s}_t)\kappa(T-t)}&\left[e^{-9\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa e}e^{-4\kappa\gamma} + \frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^2 e^{-\kappa\gamma} + \frac{1}{6}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^3\right] + \\
&-\frac{\lambda}{g(\mathrm{s}_t)\kappa e}e^{2g(\mathrm{s}_t)\kappa(T-t)}\left[e^{-4\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa e}e^{-\kappa\gamma} + \frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^2\right] + \\
&+\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^2 e^{g(\mathrm{s}_t)\kappa(T-t)}\left[e^{-\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right] + \\
&-\frac{1}{6}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa e}\right)^3.
\end{aligned}
\tag{B.4}
$$

q = 4

Finally, let us now consider the case where $q = 4.$,

$$\begin{cases} \partial_t w(t,4) + 4g(\mathrm{s}_t)\kappa\, w(t,4) + \frac{\lambda}{e} w(t,3) = 0, \\ \\ w(T,4) = e^{-16\kappa\gamma}, \\ \\ w(t,0) = 1. \end{cases} \tag{B.5}$$

Following the idea of the previous cases that

$$\partial_t \left[e^{4g(\mathrm{s}_t)\kappa\, t} w(t,4) \right] = -\frac{\lambda}{e} e^{4g(\mathrm{s}_t)\kappa\, t} w(t,3),$$

by utilizing Equation B.4 and performing integration with respect to t, we have that

$$\begin{aligned} e^{4g(\mathrm{s}_t)\kappa\, T} w(T,4) - e^{4g(\mathrm{s}_t)\kappa\, t} w(t,4) &= -\frac{\lambda}{e} \int_t^T e^{4g(\mathrm{s}_t)\kappa\, u} w(u,3) du \\ &= \frac{1}{6}\frac{\lambda}{e} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^3 \int_t^T e^{4g(\mathrm{s}_t)\kappa\, u} du - \frac{\lambda}{e} \Big[e^{-9\kappa\gamma} + \\ &\quad + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} e^{-4\kappa\gamma} + \frac{1}{2} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^2 e^{-\kappa\gamma} + \frac{1}{6} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^3 \Big] \cdot \\ &\quad \cdot e^{3g(\mathrm{s}_t)\kappa\, T} \int_t^T e^{g(\mathrm{s}_t)\kappa\, u} du + \\ &\quad + \frac{\lambda}{e} \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \left[e^{-4\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} e^{-\kappa\gamma} + \frac{1}{2} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^2 \right] \cdot \\ &\quad \cdot e^{2g(\mathrm{s}_t)\kappa\, T} \int_t^T e^{2g(\mathrm{s}_t)\kappa\, u} du + \\ &\quad - \frac{1}{2}\frac{\lambda}{e} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^2 \left[e^{-\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right] e^{g(\mathrm{s}_t)\kappa\, T} \int_t^T e^{3g(\mathrm{s}_t)\kappa\, u} du \\ &= \frac{1}{24} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^4 \left(e^{4g(\mathrm{s}_t)\kappa\, T} - e^{4g(\mathrm{s}_t)\kappa\, t} \right) + \\ &\quad - \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \left[e^{-9\kappa\gamma} + \frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} e^{-4\kappa\gamma} + \frac{1}{2} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^2 e^{-\kappa\gamma} + \right. \\ &\quad \left. + \frac{1}{6} \left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e} \right)^3 \right] \left(e^{4g(\mathrm{s}_t)\kappa\, T} - e^{g(\mathrm{s}_t)\kappa\, (3T-t)} \right) + \end{aligned}$$

$$+\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2\left[e^{-4\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}e^{-\kappa\gamma}+\right.$$
$$\left.+\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2\right]\left(e^{4g(\mathrm{s}_t)\kappa\, T}-e^{2g(\mathrm{s}_t)\kappa\,(T+t)}\right)+$$
$$-\frac{1}{6}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^3\left[e^{-\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right]\left(e^{4g(\mathrm{s}_t)\kappa\, T}-e^{g(\mathrm{s}_t)\kappa\,(T+3t)}\right).$$

Hence, we can conclude that

$$w(t,4)=e^{4g(\mathrm{s}_t)\kappa\,(T-t)}\left[e^{-16\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}e^{-9\kappa\gamma}+\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2e^{-4\kappa\gamma}+\right.$$
$$\left.+\frac{1}{6}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^3e^{-\kappa\gamma}+\frac{1}{24}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^4\right]-\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}e^{3g(\mathrm{s}_t)\kappa\,(T-t)}\left[e^{-9\kappa\gamma}+\right.$$
$$\left.+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}e^{-4\kappa\gamma}+\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2e^{-\kappa\gamma}+\frac{1}{6}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^3\right]+$$
$$+\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2e^{2g(\mathrm{s}_t)\kappa\,(T-t)}\left[e^{-4\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}e^{-\kappa\gamma}+\frac{1}{2}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^2\right]+$$
$$-\frac{1}{6}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^3e^{g(\mathrm{s}_t)\kappa\,(T-t)}\left[e^{-\kappa\gamma}+\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right]+$$
$$+\frac{1}{24}\left(\frac{\lambda}{g(\mathrm{s}_t)\kappa\, e}\right)^4. \tag{B.6}$$

C Poisson Process and Event Arrival Simulation

The Poisson process is a stochastic process that models the occurrence of events happening randomly over a fixed period of time.

Definition of the Poisson Process

A Poisson process is characterized by the following properties:

1. **Independent Increments:** The number of events occurring in disjoint time intervals are independent.
2. **Stationary Increments:** The probability of a certain number of events occurring in a time interval depends only on the length of the interval, not on its position in time.
3. **No Simultaneous Events:** The probability of more than one event occurring at exactly the same time is zero.

Mathematically, a Poisson process with rate parameter $\lambda > 0$ is defined such that the number of events $N(t)$ occurring in a time interval of length t follows a Poisson distribution:

$$P(N(t) = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}, \quad k = 0, 1, 2, \ldots \tag{C.1}$$

Here, λ represents the average rate of event occurrences per unit time.

Interarrival Times

An important property of the Poisson process is that the interarrival times (the times between consecutive events) are exponentially distributed. If T_i denotes the time between the $(i-1)$-th and i-th event, then T_i follows an exponential distribution with parameter λ:

$$f_T(t) = \lambda e^{-\lambda t}, \quad t \geq 0. \tag{C.2}$$