Agli affetti di casa, alle persone fedeli,

a quel qualcosa

che mai si assopisce né dorme

Contents

1 Introduction **1**

1.1 Motivations 1

1.2 Research Questions 2

1.3 Contributions 3

2 Literature Review **5**

2.1 General Overview 5

2.2 Relevance of LOBs 6

2.3 Prices as Interaction of Demand and Supply 7

2.4 Statistical Approaches to LOBs 8

2.5 Machine Learning Approaches to LOBs 9

2.6 Summary 12

3 Background Theory **14**

3.1 Trading Protocol at NYSE 14

3.1.1 General Overview 14

3.1.2 Market Structures 15

3.2 Limit Order Book 16

3.2.1 General Overview 16

3.2.2 Mathematical Representation 17

3.2.3 Dynamics of LOBs 19

3.2.4 Market By Order Data 21

3.3 Artificial Neural Networks 21

3.3.1 Single Layer Perceptron 22

3.3.2 Convolutional Neural Networks 24

3.3.3 Recurrent Neural Networks 26

4 Dataset **30**

4.1 Data Collection 30

4.2 Dataset Generation 32

4.2.1 Data Cleaning 32

4.2.2 Data Labelling 33

4.2.3 Data Normalisation 34

5 Data Analysis **36**

5.1 Deep Learning Model: DeepLOB 36

5.1.1 Convolutional Layers 37

5.1.2 Inception Module 38

5.1.3 Long-Short Term Memory Unit 39

5.2 Practical Implementation 39

6 Results **41**

6.1 Discussion 42

6.2 Limits and Improvements 45

7 Conclusion **47**

Bibliography **49**

Chapter 1

Introduction

1.1 Motivations

This empirical project aims at investigating the predictability of asset prices in a new context of financial markets. With the heavy diffusion of order-driven markets, new interesting data are arising to predict the price pattern of financial securities. In particular, nowadays, more than half of the markets use electronic Limit Order Books (LOBs), and so we have access to all the limit orders placed each day for a given security in a given stock exchange. Since these are high-frequency trading data points, we could imagine extrapolating some helpful information from this data source. Many researchers have been attracted and fascinated by the functioning of Limit Order Markets. However, they mainly focused on describing its mechanisms from an economic point of view or developing minimal statistical models. Hasbrouck in [1] underlined in his famous book that, at his time, there were not comprehensive and realistic models for LOBs. Nevertheless, with the increasing computational power and new techniques based on big data, exploring this kind of data seems more appealing. Only in the last two years, some exciting papers have started to appear in this context; thus, the topic covered in this thesis is relatively new and promising.

Moreover, studying these data is exciting because we could predict price patterns in the

short horizon by analysing the mechanism directly behind price movements: the fluctuations in demand and supply. Moreover, the intuition is not only to focus on the order flow but also to capture the existence of imbalances in volumes demanded and supplied. In this context, this project aims at applying modern machine learning techniques to extract the information mentioned above. The idea of relying on these ML techniques is driven by the complexity of the data we are dealing with. Classical time-series models are too limited to handle billions of data points. Besides working with a considerable amount of data, we should also consider that these time-series are non-stationary, with stochastic components and characterised by different unobservable elements such as the possibility that orders could be cancelled or replaced and the fact that there exist dark pools, which add even more complexity to the analysis. In addition to this, the idea to explore and apply these methodologies to LOBs data is also driven by the possibility to use them in real algorithmic trading strategies and, anticipating the direction of price variations, profit from them. As a final remark, it could be useful to think of LOB data as very rich sources of information, in fact, the type of order itself suggests the patience of the trader and the way orders are posted could reveal the underlying beliefs of investors on a given security.

1.2 Research Questions

This project has different objectives, all related to the underlying goal of understanding if the usage of LOB data could be something worth spending resources on and thus if it is informative about future price movements. However, before specifying the research questions, it is essential to highlight the collocation of this work within the range of papers that apply machine learning techniques to LOB data. Thus, this is not about the analysis of LOBs and how agents act, so this is less connected to the broad literature of market microstructure theory and related models. The idea of the thesis is to focus on the most suitable time-series models to use for a very specific and complex time-series; which is the one that at any given time captures the state of the order book considering second-level data, that is, ten price levels for both ask and bid side. In other words, the time-series

will be multi-dimensional in the sense that each observation at any given point in time is a matrix where the columns correspond to bid/ask prices and bid/ask volumes, while the rows are the different levels of the LOB. Similarly, if market by order data (MBO) are used, then each event will be represented by a vector, and so the time-series generated could be seen as a collection of random vectors.

Having clear in mind this concept, the first objective of this work is to answer the following research questions:

1. Is it possible to use the evolution of the states of the Limit Order Book to **predict price variations (directions)** over the short-horizon (intraday)?
2. Are Limit Order Books **informative**? Can events outside the equilibrium suggest something about the future equilibrium reached by the market?

In order to answer these two research questions, this work considers instruments traded on NYSE using raw data for a period of time never investigated. Thus, with this thesis, the discussion on this topic will be enriched and also a specific data set will be created. Working with high-frequency trading data is itself a complex task since it is not simple to obtain them, and even more complex is to handle and analyse them. Besides this, this thesis applies a model developed only recently and tested on a limited dataset, thus working with different data could be useful to check the goodness of this algorithm.

1.3 Contributions

Given the above framework, the idea of this research project is to follow the work done by Zhang, Zohren and Roberts in [2]. There are different elements in which our work could be differentiated from the cited one. In particular, the model could be modified and improved, for example, investigating more in-depth the Bayesian development or the techniques suggested by the authors. Besides this, another essential element that could distinguish our work is the different choice of data. We could consider different financial

instruments instead of stocks, and we can also choose less liquid stocks or assets traded in different markets. Moreover, we could use as a train set a more significant sample and more recent data to understand if the model works well in a high volatility context. These are some of the elements that could be used to understand universal features in the order book. Since this is an applied work, the exciting part is understanding if the model could be used practically to implement a trading strategy. Thus, some possible implementations and practical issues could be considering trading fees and relaxing the assumption that trading occurs at mid-price, as suggested by the authors. A final consideration is that we could also try to employ the model to predict price movements intra-day and so during a single trading session, using price and volume data recursively as new orders are placed.

In this context, the main point of differentiation is the usage of a different dataset that covers a more recent historical period. While the model remains the same, leaving other elements as possible future developments for this work. To be more precise, data used for this thesis refers to stocks traded at NYSE while authors of [2] used stocks negotiated on LSE. As highlighted in the following chapters, stocks considered for this work are more frequently traded and observations for each day are almost double with respect to stocks traded at London Stock Exchange.

Chapter 2

Literature Review

2.1 General Overview

One of the most dominant topics in financial economics is the discussion regarding the predictability of stock returns. The central paradigm remains today, the Efficient Market Hypothesis, according to which there is no way to predict stock returns since prices incorporate all available information at any given time. In particular, since past information is independent of the unpredictable new ones, it turns out that prices and returns cannot be predicted. However, many works empirically contradict this thesis and, starting from the 80s, many papers highlighted that returns are in part predictable (refer to [3]). Given these recent developments, many scientists have tried to find a good model to forecast this variable; some relied on the usage of LOB data but not too many implemented machine learning methodologies on this kind of data set. Moreover, there are no comprehensive and realistic, both statistical and economic, models for LOBs as underlined in [1].

In order to introduce the topic of stock prediction using LOB data, it could be helpful to refer to [4] in which a comprehensive overview of the state-of-art techniques in this field are described. According to these authors, information about a stock is incomplete, complex, uncertain, and vague, so it is impossible to forecast its evolution. They start from the well-known distinction between technical and fundamental approaches to stock

prediction, highlighting the underlying assumptions behind them. In this context, the relevant point is that this thesis, which can be categorised neither within the fundamental nor the technical paradigm, makes no assumptions about the true price of a stock. This is the critical novelty and main point of this work. In fact, the key idea is to investigate the forces directly behind price fluctuations. Given these underpinnings, the most recent and interesting developments in this field are concentrated on applying machine learning methodologies to solve predictability in financial markets.

In the following sections, the discussion about the main contributions relevant for this work is analysed, distinguishing between statistical approaches and machine learning approaches to this problem, as highlighted in [2]. In particular, a subset of researchers focuses on the definition of models that start from the financial market microstructure theory and try to model the order flow accordingly, finding the probability law that governs the underlying process starting from the observed sample. On the other hand, ML methods are used by the researcher being agnostic about the mechanism that relates each variable to the others, as specified in [5], so they are directly applied to the raw time-series generated by a Limit Order Market.

2.2 Relevance of LOBs

Before entering into the details of different approaches used to model LOB data, it is essential to understand whether or not the information content of the Limit Order Book is relevant. In particular, in [6] the authors argue that LOBs are helpful when studying future price changes. They analyse 144 randomly selected securities between 1990 and 1991 from the TORQ database, arguing that traders who know the order book trade more successfully than those who do not, as in the case of specialists (or dealers). The intuitive reason behind this conclusion is that LOB data permit them to discover imbalances between demand and supply which ultimately is the unique driver of price movements. In fact, Hopman (2007) in [7] found that order flow imbalances can explain 70% of stock prices' changes. This conclusion is achieved by analysing 34 stocks for four years between

1995 and 1999. Thus, the author established the causality going from orders to prices and not the other way around. On this issue, another interesting reference is [8], in which the authors underlined that LOB data has explanatory power for price prediction and its contribution to the price discovery process accounts for 22%.

2.3 Prices as Interaction of Demand and Supply

Another prerequisite, before discussing the details of some works that have specifically applied machine learning methods to LOB data, is to highlight some papers that study the supply and demand in financial markets. This topic is interesting since the intuition behind this thesis is that ultimately the price variation is just the result of the interaction of these two forces. In [9], the authors underline that order-flow exhibits long-memory which practically can be translated as the fact that events from the distant past can have an impact on the present. Furthermore, there is another crucial element that drives price movements that is liquidity. The authors of [10] show that this factor is the primary explanation behind significant price variations. Again, this concept is related to the LOB since the authors highlight that a high density of limit orders per price results in high liquidity for market orders and thus, best quotes do not move or move only marginally. To better understand this concept, it is important to think of liquidity as the ability of the market to absorb new orders. In fact, if outstanding limit orders are small in size, then a market order will have a more significant impact since it changes the best quotes. For completeness, notice that this paper is based on the analysis of 16 stocks traded at LSE from 1999 to 2002 (which is a period that includes the Dot-com bubble). A final reference on this topic is [11] in which the authors state that price movements are the result of two factors: the flow of orders and response of prices to individual orders. The interesting element is that these researchers also highlight that the order flow can be seen as a highly correlated long-term memory process, and this feature is mainly because large investors split their huge orders into smaller ones. The fact that the order flow exhibits some statistical properties is particularly interesting for this thesis since the idea is precisely to use the evolution of the order flow to forecast which orders will arrive in the near future.

2.4 Statistical Approaches to LOBs

In this context, different researchers started to investigate the relationship between order flow history and price movements. In [12], the authors point out the presence of a universal way according to which markets react in a similar way to different perturbations induced by all market participants. This universality is particularly interesting, and also the concept expressed by these authors that "trading affects prices" is an incentive to look at order flows to predict price variations. Related to this idea, one of the first relevant works done is [13] which investigates the specific case of the Paris Bourse. To be precise, the authors of this paper are interested in understanding how the order flow reacts to the state of the order book and informational events in the marketplace. They analyse 40 stocks considering five levels of the book for each side (ask and bid), using a pilot period of 6 days and then 19 days as a sample period (the historical time of reference is 1991). Using this kind of data, they reconstruct the supply and demand, noticing that orders placement away from the quotes decreases monotonically. This is a critical element to be taken into account for this thesis since one could argue that order book data are informative but only what regards the best quotes, since what happens away from them is less relevant, as stated in [14]. However, this thesis's objectives are specifically to prove that this is not entirely true. Besides the analysis implemented in [13], which should be considered slightly out-of-date, an element of interest is the comment about hidden orders. In fact, one of the most extensive limits in this thesis and in general with the analysis of LOB data is the presence of hidden orders, which does not permit to fully appreciate the real demand and supply at any given time. However, the authors underline that this issue is limited because this kind of order should be theoretically inferred from changes of the order book different from what we should expect given a specific market event.

A more technical paper that defines a statistical model for the Limit Order Book is [15] in which they implement a simplified and parsimonious representation of the LOB. It is important to highlight that only data at the first level (best quotes) are considered in this work. In particular, they focus on events like cancellations, MOs and new LOs that change the actual state of the book. These events are modelled like Poisson processes

with different rates/parameters. Then they use their model to compute conditional probabilities of price changes given the current state of the book. A final remark is that this model is based on strict assumptions, and this is done to achieve analytical tractability.

Another interesting reference is [16] in which the authors are interested in LOB data that could be modelled as a queuing system. They argue that limit order markets had spread out during the last years influencing all the major exchanges. However, the interesting point is that they highlight that High Frequency Trading (HFT) data are of particular interest since they show statistical regularities; they are available in a huge amount and result from a precise mechanism of execution. Moreover, the analysis of this kind of data is practically useful for market makers, optimal execution, and for developing methods for short-term predictions. The authors offer some valuable insights for further analysis. They highlight that trading activity is not uniform but shows peaks at the initial and final phases of the trading day; thus, it could be useful to filter them since they are not stationary when analysing intraday data. Another element to consider is the cancellation of orders. In this work, it is said that up to 80% of orders for liquid stocks at NYSE and NASDAQ are cancelled within a few seconds. This could represent an issue for this thesis and the empirical analysis.

2.5 Machine Learning Approaches to LOBs

While statistics has a long-standing focus on inference, machine learning concentrates on prediction using general-purpose learning algorithms. The work [17] by Sirignano is one of the few that develops a machine learning model for the LOB. In fact, the author himself underlines that he has developed a novel neural network, called spatial neural network, that works well in multi-dimensional spaces and so can be applied to LOB data. The advantage of this model is that it uses information more efficiently contained at deeper levels of the book. Moreover, it is characterised by low computational costs and greater interpretability. They consider a sample of 489 stocks traded on the NASDAQ between 2014 and 2015 for one year, data have been collected from the LOBSTER data set, and 50

levels on each side of the book have been taken into account. As a final remark, without entering into the details of the model developed, it is interesting to notice that they model the joint distribution of the best bid and ask prices conditional to the current state of the book. Thus, they do not focus just on the expected price change. Another work that applies recurrent neural networks is [18], in which it is said that ML techniques are handy for this problem since LOB data are very complex and high-dimensional.

One of the most recent and exciting papers about machine learning applied to the order book data is [2]. The attention dedicated to this work will be higher since it represents the ultimate state-of-art model for this kind of problem, and thus this thesis is mainly based on this research and try to apply it to a different dataset. Note that the authors have also developed some adjustments to the original model, called DeepLOB, introducing bayesian components in [19], the model is now called BdLOB, and focusing on the market by order data in [20] and finally also developing a multi-horizon model in [21].

The goal of these authors is to develop a large-scale deep learning model to forecast price movements using LOB data. The major focus is obviously on the definition of their model, which represents a novelty with respect to past research. In fact, they implement a network architecture made by three major components: convolutional layers, an Inception Module and a Long-Short Term Memory layer. Moreover, they follow the LIME method proposed by [22] that allows to understand which components of LOBs are important in predicting the stock price. Regarding the empirical analysis, they use both a benchmark LOB data set called FI-2010, which is made by ten trading days at NASDAQ Nordic for five stocks and data regarding five liquid stocks traded on LSE over one year. Besides this dataset, they also consider five more stocks on which the previously trained model is applied only to test it, this is done in order to understand if there exist universal features in the dynamics of the LOB that can be learnt and transmitted to out-of-sample securities. In particular, they consider the first ten levels for each side of the order book obtaining multi-dimensional objects. To be more precise, each object can be represented as: $x_t = [p_a^{(i)}(t), v_a^{(i)}(t), p_b^{(i)}(t), v_b^{(i)}(t)]_{i=1}^{n=10}$, where p_i represents the price (a = ask, b = bid) while v_i represents the volume/size. Once data have been obtained, there

are two important elements that the authors underline: normalisation and definition of price change's direction. Regarding the former, it is key in order to correctly apply the machine learning algorithm. They choose a dynamic normalisation, called z-score since financial data are not suitable for a static one. The latter element is also essential and not immediately intuitive. In fact, no trades occur at the mid-price (p_t), even if it is often used as a proxy for the current price, thus the direction is determined not comparing p_t with p_{t+k}, but instead, it is firstly determined the mean of the previous and next k mid-prices, and then the percentage change (l_t) of the mid-price is used to determine the direction. Thus, if $l_t > \alpha$ the price is up while it is down if $l_t < \alpha$, where α is the threshold.

The paper's final part implements this technique in a trading simulation computing profits but assumes some strong assumptions like the absence of fees. However, results are encouraging and highlight that this Deep Learning algorithm applied to LOB could also be used by practitioners for algorithmic trading. Then, they compare results obtained using alternative methods computing different metrics such as precision, recall, accuracy and F1-score. Among the models cited, some interesting ones are important to mention. For example, in [23], the authors implement a logistic regression and a lasso logistic regression for variable selection. Without entering into the details of the model, the authors argue that LOB data are informative in particular volumes that turn out to be more valuable than prices. Unlike Zhang et al., they use the 40 stocks of CAC40 traded for one month (April 2011). This data set is also divided into morning and afternoon trading sessions. Another important paper is [24] where a two-hidden-layer network is developed to analyse and predict from order book data. It is interesting the attention given to Recurrent Neural Networks (RNNs), which are specifically suited for streaming data. In particular, they focus on a so-called Sequence-RNN applied to the standard data set FI-2010. Besides these models, DeepLOB is also compared to the one that uses Bag of Features (BoF) in [25] and to the models of deep learning developed by [26]. The latter is interesting because the authors apply a RNN called LSTM (Long Short Term Memory) that overcomes the problem of overfitting of RNNs (refer to the theoretical background for more details).

Moreover, Zhang et al. extend the DeepLOB model introducing Bayesian elements in what they call BdLOB. Without entering into the details, it is interesting to notice that they are the first ones to use a Bayesian framework for this kind of problem, through which they can quantify uncertainty. In particular, they introduce a so-called variational dropout after the Inception Module in DeepLOB, which has been proven to be a good proxy for the posterior distribution in a bayesian network. For completeness, it is important to highlight the most recent work of Zhang et al. in [21], where they do not focus on predicting the price variation at a specific future point in time, but instead, they model a multi-horizon forecasting method to obtain a multi-horizon path for stock prices. Finally, they also try to use market by order data to predict price movements. These data are even more granular than LOB ones since the Limit Order Book is the result of aggregating these market by order (MBO) data which represent every single order submission (message) sent to the exchange, and so the authors believe that this will constitute the ultimate frontier in stock prediction models.

2.6 Summary

When approaching a new research project, the literature review is key in order to capture the current state-of-art of a certain topic. Given the above discussion, the most important takeaways are summarized below.

1. There is an **increasing attention** to the Limit Order Book, and to the even more granular data of Market by Orders. In particular, the functioning of financial market transactions and trading sessions has attracted researchers other than economists, like physicists, mathematicians and statisticians. In fact, this kind of dataset captures the most microscopic dynamics of financial markets and studying it is considered a promising frontier in the field of financial economics.

2. The main reason for this broad interest is that LOB data represents a very **huge source of big data** produced daily and so it is favourable to use them for the application of the growing techniques of statistical learning and machine learning.

Moreover, only recently some interesting results start to appear given the evolution of models and also computational power. In this context, the main focus is on Deep Learning models like CNNs and RNNs.

In order to have a synthetic overview of the main models developed and presented above, a brief recap is given below. An interesting reference to capture this overall framework is [27]. These authors compare different models using the same dataset obtained from LOBSTER. In particular, they consider a multinomial logistic regression as a baseline, then a Feedforward Neural Network, RNNs in the form of LSTM and at the end Convolutional Neural Networks. Notice that also [2] compares different models on the same dataset, as already underlined, with the conclusion that Deep Learning and Reinforcement Learning are promising frontiers in this research field.

Chapter 3

Background Theory

The aim of this chapter is to briefly introduce the organizational structure of financial markets and the functioning of the Limit Order Books. In particular, the former topic refers to the trading procedures and the order routing system behind the functioning of the New York Stock Exchange, while the latter topic will be introduced to give a formal and mathematical representation of a LOB. Then, a concise presentation of neural networks is given in order to be able to correctly explain the model used for this analysis.

3.1 Trading Protocol at NYSE

3.1.1 General Overview

Since this work relies on data regarding the state of the order book of stocks traded on the New York Stock Exchange, it could be important to present how the biggest trading venue in the world works. Overall, the importance of the market structure is that it affects the level of trading transparency and the information spread. Moreover, it impacts agents' trading strategies, their actions and market quality as a whole. In introducing the trading protocol that governs the NYSE, a basic distinction is between an order-driven market and the quote-driven one. The former indicates a market where agents' orders interact directly while the latter refers to a market where an intermediary must fulfil the

contract. In other words, and referring to [28], quote-driven markets are characterized by the presence of a dealer in each trade that occurs in the market, while this is not true in an order-driven market. Given these two broad definitions, NYSE combines both systems and it is called hybrid market. In general, order-driven markets are the most common ones and, within this framework, limit order books are particularly interesting.

3.1.2 Market Structures

In order to give a precise presentation of the key elements characterizing the NYSE, the main references used are [29] and the website of the stock exchange.

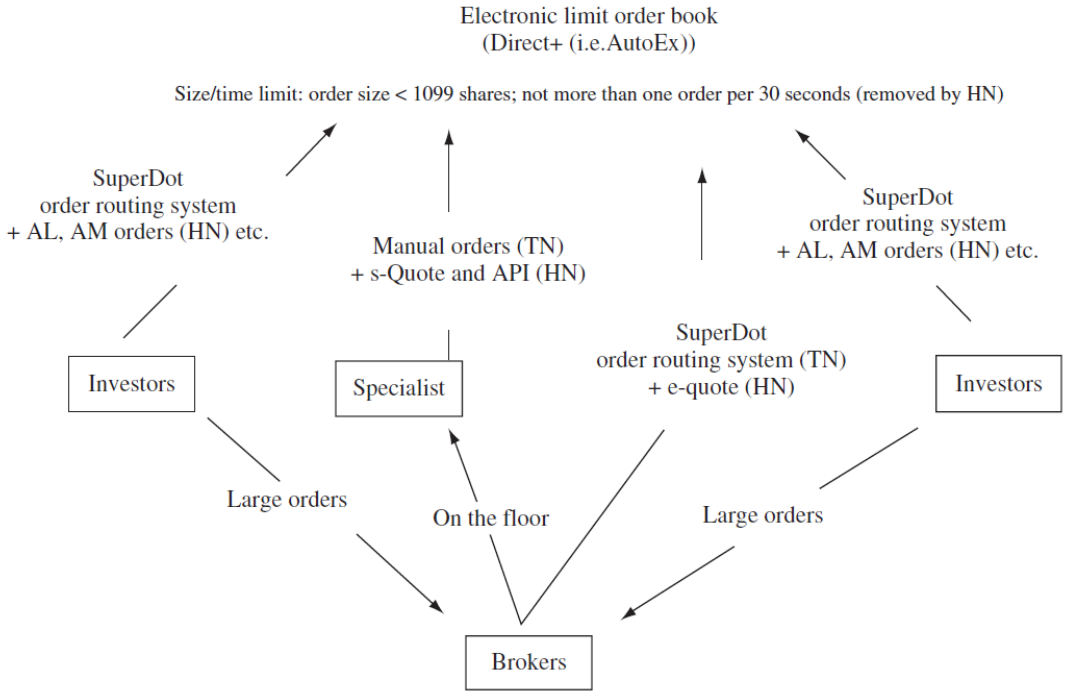

Figure 3.1: Hybrid Market Structures at NYSE

Generally speaking, hybrid markets are organized by specialists who provide liquidity and manage the order book. Then, investors can transmit their orders to them both using automated sorting systems or floor brokers. In this sense, orders exhibited by specialists include both their quotes but also limit orders transmitted by final investors or brokers. Thus, liquidity at NYSE is ensured by both specialists and other investors. In fact, professional market-makers compete with the electronic limit order book. The

illustration in 3.1 represents the functioning of NYSE both in traditional terms (TN) and with the transformation that occurred in 2006 that transformed it into a hybrid market (HN). The working of the traditional NYSE is important to understand some key aspects of the underlying trading mechanism. When a market participant wants to execute an order, he/she can post it through a floor broker or the SuperDot Direct+. Once the order has been posted, it is not executed automatically but the specialist evaluates it and then it is publicized among floor traders and if they are not interested then it can be executed by the specialist who acts as counterpart. Thus, brokers interact directly with the specialists but at NYSE they can also send orders to the limit order books. Moreover, LOB was only visible to specialists but starting from 2002 the so-called NYSE Open Book was introduced. A final remark is about the priority rules, at NYSE price priority is absolute while then priority is given to orders transmitted through SuperDot or floor brokers, only after these two rules time priority is applied. Finally, with the Hybrid Market Initiative, limit orders enjoyed direct and immediate execution.

3.2 Limit Order Book

3.2.1 General Overview

From a general point of view, order books are simply databases held by brokers, dealers and exchanges that record open orders that they cannot yet fill. They are specifically called limit order books (LOBs) and contain all active orders in a market (meaning for specific security) at any specific point in time. These orders are mainly limit orders, but there could also be stop orders and other kinds of orders. A final relevant distinction is between open book markets and closed book markets which intuitively refers to the fact that the order book is fully displayed and available or not.

It is important to specify that an order is an instruction sent by investors to brokers and it must contain information regarding the instrument traded, the direction of the trade and the quantity. Optionally, it could also contain a limit price and other information like the

duration of the validity of the order. The fact that the price is specified or not, classify the order into a limit order or a market order. This distinction is interesting because it also suggests the behaviour of the traders and thus the information they have and so the information they implicitly share when posting the order. In fact, a market order (MO) absorbs liquidity and is used by impatient traders while patient traders who have some information and what to anticipate the market by positioning themselves in the book to have precedence over others can give liquidity by posting limit orders (LOs).

3.2.2 Mathematical Representation

In this section, the functioning and elements of a Limit Order Book are presented rigorously using the most common mathematical notation. In particular, this paragraph is highly indebted to the work of Gould et al. ([30]) which represents the main reference used when papers about LOBs are developed.

It has been previously said that the Limit Order Book (LOB) at time t contains all the active orders in a given market, it is usually indicated by $\mathcal{L}(t)$. Moreover, each order is specified as a vector $x = (p_x, \omega_x, t_x)$, where t_x represents the time at which the order is submitted, ω_x is the quantity and p_x is the limit price. In particular, with $\omega_x > 0$ indicating an order to sell at a price no lower than p_x, while if $\omega_x < 0$ this is an order to buy at a price no greater than p_x. Thus, it follows that $\mathcal{L}(t)$ can be partitioned into two subsets: $\mathcal{A}(t)$, made by all sell orders with $\omega_x > 0$, and $\mathcal{B}(t)$ containing all buy orders with $\omega_x < 0$.

Both the volume and the price are discrete quantities that cannot take every value but must be a function of the smallest units permissible. Specifically, the lot size σ of $\mathcal{L}(t)$ is the smallest amount of an asset that can be traded and so $\omega_x \in \{\pm k\sigma \mid k = 1, 2, ...\}$. On the other hand, the tick size π is the smallest permissible price interval.

Given these general definitions, it is possible to define the bid price $b(t)$ as the highest price among active buy orders and the ask price $a(t)$, which is the same for sell orders. More formally:

$$b(t) := \max_{x \in \mathcal{B}(t)} p_x \qquad a(t) := \min_{x \in \mathcal{A}(t)} p_x$$

It follows that the bid-ask spread is given by $s(t) := a(t) - b(t)$, while the mid-price is defined by $m(t) := [b(t) + a(t)]/2$. These two quantities are both relevant for the analysis since the first one is an approximation of the cost of liquidity and the second one is often used as a proxy of the asset's price. From these definitions, it is possible to express the price of a new order as a function of the bid and ask prices. This is useful because it is sometimes used in some models and because it captures how far from the best quotes the order is. In particular, for the order $x = (p_x, \omega_x, t_x)$, the relative price is:

$$\delta^x := \begin{cases} \delta^b(p_x) & \text{if the order is a buy order} \\ \delta^a(p_x) & \text{if the order is a sell order} \end{cases}$$

$$\text{where} \quad \delta^b(p) := b_t - p \quad \delta^a(p) := p - a_t$$

A final definition is about the market depth, which is intuitively the volume at each t for a given level of price available in the market. In particular, it is interesting to define the depth available both in the bid and ask sides for a given level of relative price. Formally,

$$N^b(p,t) := \sum_{\{x \in \mathcal{B}(t) | \delta^x = p\}} \omega_x$$

$$N^a(p,t) := \sum_{\{x \in \mathcal{A}(t) | \delta^x = p\}} \omega_x$$

Given all these main elements of a LOB, it is possible to pool them together and give a graphical representation of it before studying how the arrival of new orders (both market and limit) changes the current state of the book.

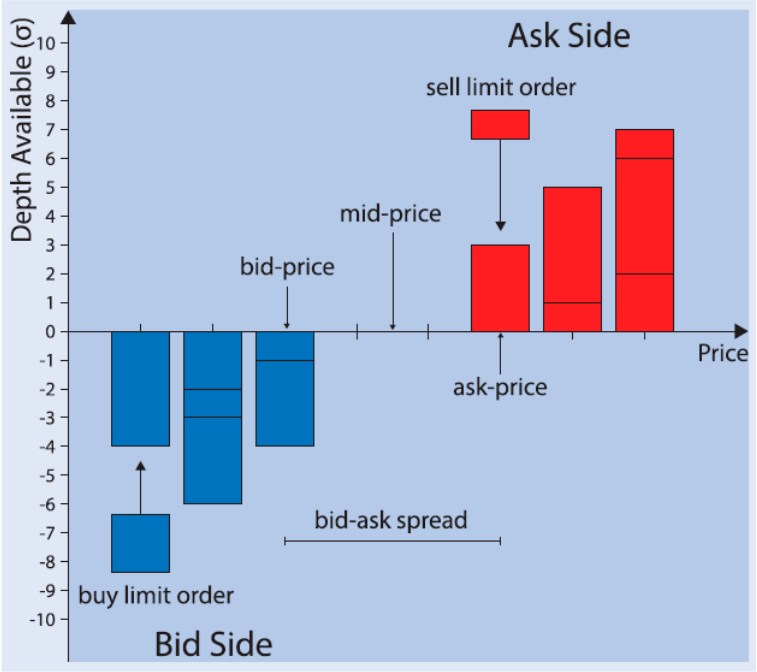

Figure 3.2: Graphical Representation LOB

3.2.3 Dynamics of LOBs

The aim of this paragraph is to describe formally the impact of a new order that reaches the book. In particular, the analysis is about two different types of orders: market orders which result in their immediate execution and limit orders which are instead not immediately executed because the conditions attached to them are maybe not satisfied and so they become active orders in the book. Note that with the term limit order we refer to this broad category of orders not immediately executed, then they could be distinguished into more detailed categories according to their validity and features.

Case I. Consider a new buy order $x = (p_x, \omega_x, t_x)$ that arrives in the market, there could be three different scenarios:

- If $p_x \leq b(t)$ then it is a limit order that does not change the value of $b(t)$ and it becomes an active order
- If $b(t) < p_x < a(t)$ then this is a limit order which changes the value of the bid price which becomes $b(t_x) = p_x$
- If $p_x \geq a(t)$ then it is a market order which is immediately executed and the change

of the state of the LOB depends on ω_x and the depth $n^a(a(t), t)$. To be more precise the new ask price will become:

$$min(p_x, q) \quad \text{where} \quad q = arg\min_{k'} \sum_{k=a(t)}^{k'} n^a(k, t) > |\omega_x|$$

Case II. Consider the opposite case of a new sell order x sent to the market, the above three scenarios can be rewritten as:

- If $p_x \geq a(t)$ then it is a limit order that does not change the value of $a(t)$ and it becomes an active order
- If $b(t) < p_x < a(t)$ then this is a limit order which changes the value of the ask price which becomes $a(t_x) = p_x$
- If $p_x \leq b(t)$ then it is a market order which is immediately executed and the change of the state of the LOB depends on ω_x and the depth $n^b(b(t), t)$. To be more precise the new bid price will become:

$$max(p_x, q) \quad \text{where} \quad q = arg\max_{k'} \sum_{k=k'}^{b(t)} |n^b(k, t)| > \omega_x$$

From a practical point of view the following table (3.3) is useful to understand these different situations. In particular, it contains both buy (sell) orders indicated by $\omega_x < 0$ ($\omega_x > 0$) expressed as a function of σ.

Arriving order x	Values after arrival (USD)			
	$b(t_x)$	$a(t_x)$	$m(t_x)$	$s(t_x)$
Initial values	1.50	1.53	1.515	0.03
($1.48, $-3\sigma, t_x$)	1.50	1.53	1.515	0.03
($1.51, $-3\sigma, t_x$)	1.51	1.53	1.52	0.02
($1.55, $-3\sigma, t_x$)	1.50	1.54	1.52	0.04
($1.55, $-5\sigma, t_x$)	1.50	1.55	1.525	0.05
($1.54, $4\sigma, t_x$)	1.50	1.53	1.515	0.03
($1.52, $4\sigma, t_x$)	1.50	1.52	1.51	0.02
($1.47, $4\sigma, t_x$)	1.48	1.53	1.505	0.05
($1.50, $4\sigma, t_x$)	1.49	1.50	1.495	0.01

Figure 3.3: Arrival of new order in the LOB

3.2.4 Market By Order Data

Even if LOB data could be seen as the most microscopic, this is not completely true since they are the result of an aggregation process. In fact, for every price level, it is not possible to know who and when traders and investors have sent those orders, in other words in the LOB all orders are aggregated at each price level and so only the cumulative quantity at that price is visible. However, this aggregation process starts from all the messages that trading participants send to the exchanges directly or through brokers, these are called Market By Order (MBO) data and they really represent the most granular and microscopic information about the decisions taken by each participant at any given point in time. The following table 3.4 summarises typical information of records of these single events.

Time stamp	**ID**	**Type**	**Side**	**Action**	**Price**	**Size**
2018-01-02 09:21:15.717500766	462805645163273214	1	N/A	2	N/A	N/A
2018-01-02 09:21:18.585446702	462805645163298476	1	1	1	68.54	8334.0
2018-01-02 09:21:20.680552032	462805645163297649	1	1	0	68.56	3227.0
2018-01-02 09:21:20.944574722	462805645163297649	1	N/A	2	N/A	N/A
2018-01-02 09:21:20.945483443	462805645163298567	1	2	1	68.59	5100.0

Figure 3.4: Example of Market by Order raw data

3.3 Artificial Neural Networks

The aim of this paragraph is to present some key aspects of the underlying theory behind the model used for this analysis and that will be explained in the next chapter. In particular, the general intuition behind Neural Networks is presented using a single hidden layer, feed-forward neural network for simplicity. Then the aspects that differentiate Convolutional and Recurrent Neural Networks are introduced as well. Main references used include [31] and [32]. Intuitively, artificial neural networks could be seen as the artificial counterpart of a human brain where the computational units are called neurons and they are connected to each other using weights, that simulate the strength of synaptic connections in biological organisms. An artificial neural network computes a function of

the inputs by propagating the computed values from the input neurons to the output neuron and using the weights as intermediate parameters. Learning occurs by changing the weights connecting the neurons. Moreover, Artificial Neural Networks (ANNs) refers to a class of models that implement a non-linear approach to the analysis of different types of datasets like time-series data. Sometimes they are considered black-box models but there is a lot of attention on them as potentially useful to handle complex and unstructured datasets like LOB data considered in this thesis.

3.3.1 Single Layer Perceptron

As stated above, the aim of this chapter is obviously not to present all the detailed aspects related to this class of models but just to introduce its simplest version in order to understand the underlying intuition. In particular, the starting point is the so-called Single Layer Perceptron. The basic idea is to connect a set of inputs (X) to some outputs (Y) using intermediary nodes, also called hidden layers of neurons (Z), in order to introduce non-linearity in the model. Consider the following standard representation:

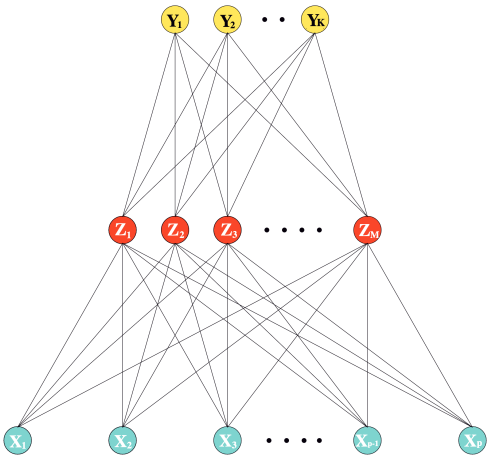

Figure 3.5: Neural Network Diagram: single hidden layer

Even if the above figure represents the simplest model, the logic behind is always the same. Complexity is added by just considering more hidden layers. Using the same notation of [31], we could just think to extract some features from inputs, by applying an activation function to a linear combination of them, and then these features are combined to get the

output. Mathematically,

$$\begin{aligned} Z_m &= \sigma(\alpha_{0m} + \alpha_m^T X) \quad m = 1, ..., M \\ T_k &= \beta_{0k} + \beta_k^T Z \quad k = 1, ..., K \\ f_k(X) &= g_k(T) \quad k = 1, ..., K \end{aligned} \tag{3.1}$$

$$\text{where} \quad \sigma(v) = \frac{1}{1 + e^{-v}} \quad \text{and} \quad g_k(T) = \frac{e^{T_k}}{\sum_{j=1}^{K} e^{T_j}}$$

The above functions are usually used in a standard neural network but researchers can appropriately change them. For example, a very common activation function used is the Rectified Linear Unit function:

$$ReLu(a) = max(0, a)$$

A final remark, related to the above diagram 3.5, is about the introduction of a bias unit in the hidden and output layer. It is possible to think about it as a threshold that requires that the linear combination of inputs must be at least above that value. More formally,

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T X - BIAS) \quad m = 1, ..., M$$

To conclude this brief introduction, it is fundamental to discuss the key topic in machine learning models; that is the learning procedure of the algorithm. First of all, parameters that must be learnt are weights and biases and they are estimated through a process of trial and error. Intuitively, different values for these parameters are tried in order to minimize a cost function where generally the cost can be thought as the difference between the true output value and the one estimated. More formally it could take the form of the sum-of-squared errors in (3.2) or the cross-entropy in (3.3):

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k(x_i))^2 \tag{3.2}$$

$$R(\theta) = -\sum_{k=1}^{K} \sum_{i=1}^{N} y_{ik} \log f_k(x_i) \tag{3.3}$$

To minimize $R(\theta)$ we cannot implement standard solutions given the complexity of the problem at hand. In this situation, solutions are found through the gradient descent approach using the back-propagation algorithm. Considering the cost function (3.2), this algorithm, as illustrated by [31], works as follows:

$$\frac{\partial R_i}{\partial \beta_{km}} = -2(y_{ik} - f_k(x_i))g_k'(\beta_k^T z_i)z_{mi} = \delta_{ki} z_{mi} \tag{3.4}$$

$$\frac{\partial R_i}{\partial \alpha_{m\ell}} = -\sum_{k=1}^{K} 2(y_{ik} - f_k(x_i))g_k'(\beta_k^T z_i)\beta_{km}\sigma'(\alpha_m^T x_i)x_{i\ell} = s_{mi} x_{i\ell} \tag{3.5}$$

It follows that δ_{ki} is the error from the model used at the output units while s_{mi} is the error at the hidden layer units. It can be easily shown that:

$$s_{mi} = \sigma'(\alpha_m^T x_i) \sum_{k=1}^{K} \beta_{km}\delta_{ki} \tag{3.6}$$

Finally, given these elements, a gradient descent update is given by:

$$\beta_{km}^{(r+1)} = \beta_{km}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{km}^{(r)}} \tag{3.7}$$

$$\alpha_{m\ell}^{(r+1)} = \alpha_{m\ell}^{(r)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i}{\partial \alpha_{m\ell}^{(r)}} \tag{3.8}$$

Thus, once estimated outputs $\hat{f}_k(x_i)$ are calculated using (3.1), then δ_{ki} are computed and after this step they are back-propagated to obtain s_{mi}. Once both errors have been computed, they are used to restart the procedure at r+1.

3.3.2 Convolutional Neural Networks

Given the above framework, it is now possible to enter into the details of a convolutional neural network. This kind of architecture is usually used for image classification and recognition. Thus, when considering an image it can be seen as a 2-dimensional object where every single unit is a pixel. However, instead of taking a single pixel as input,

a patch is considered and to it a kernel (or filter) is applied to detect certain features. Basically, the filter is a matrix of weights and once they are multiplied by the patch, an activation function is applied. To be more precise, a CNN is made by three crucial steps:

1. **Convolution**: application of filters to patches of inputs generating features maps;
2. **Non-linearity**: application of a non-linear activation function like ReLU;
3. **Pooling**: downsampling operation summarising infomation contained on each feature map.

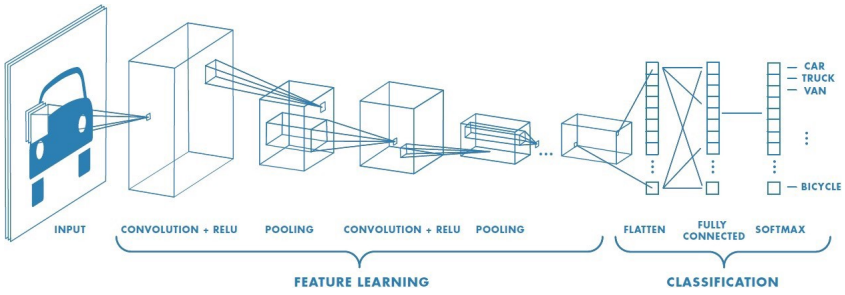

Figure 3.6: Graphical Representation of a Convolutional Neural Network

Given the above introduction, mathematically the convolutional and pooling steps are the key ones. Moreover, refers to 3.6 to get the intuition behind CNN. Consider a kernel **K** applied to a 2D signal (an image) I, then the convolution operation can be expressed mathematically as:

$$(\mathbf{K} * I)(i,j) = \sum_{m,n} \mathbf{K}(m,n) I(i+n, j+m)$$

Graphically, the above equation is equivalent to:

With reference to 3.7, the input is called patch, and in image detection problems can be seen as a matrix of pixels, while the number of steps with which the kernel moves is called the stride, which is 1 in this example. Finally, once an element of the feature map is obtained, an activation function ϕ is applied. In particular, given the kernel **K** (k x k) and **x** (k x k patch), it is activated using $z(\mathbf{x}) = \phi(K * x + b)$ where b stands for the bias. To conclude this brief presentation, a final remark regards the pooling step

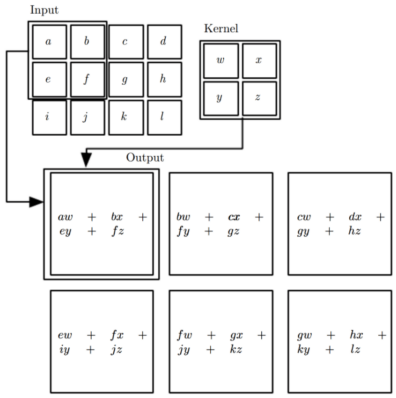

Figure 3.7: Convolution Operation in a CNN

which simply consists in summarizing and downsampling the set of features extracted. A common procedure for this step is the max-pooling method which consists in taking the maximum value of a certain patch and again moving over different patches according to the steps specified by the stride. Consider the following for a better understanding:

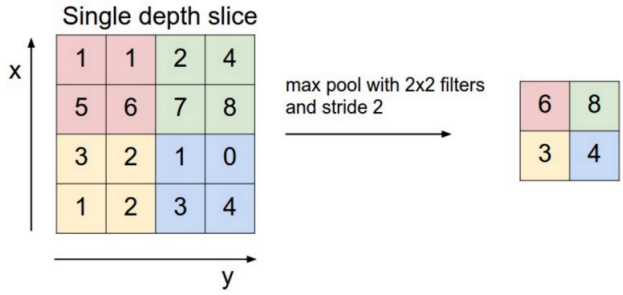

Figure 3.8: Application of a maxpooling

3.3.3 Recurrent Neural Networks

Another important class of ANNs is a Recurrent Neural Network. It is generally used with sequential and time-series data in order to capture time dependencies existing in the dataset. The intuition is that, differently from the CNN where inputs and outputs are fixed and from the same timestamp, now there is a temporal component in the learning process and this is introduced by considering neurons with a so-called recurrence. In fact, even if at each timestamp there is the corresponding output, that output could depend not only on the input at the same timestamp but also on inputs from previous timestamps.

Thus, the central topic of this paragraph is to understand how to modify a neural network in order to be able to handle sequential data. Considering the basic perceptron with only one hidden neuron, this goal is achieved by basically introducing a recurrent cell as follows:

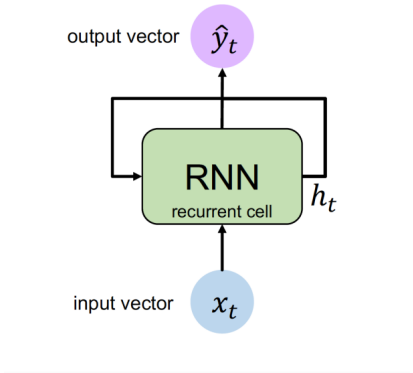

Figure 3.9: Basic Structure of a Recurrent Neural Network

To link inputs and computations at different timestamps, the solution is given by using a so-called internal memory (h_t) that is transmitted from timestamp to timestamp. In this way , the output vector can be expressed as $\hat{y}_t = f(x_t, h_t)$ and $h_t = f_W(x_t, h_{t-1})$. Note that this internal memory component (called cell state) can be seen as the state variable that is recurrently updated at each timestamp as the sequence is processed. Mathematically, starting with an input x_t, the update of the hidden cell state and the output are computed as follows:

$$h_t = \sigma(\mathbf{W}_{hh}^T h_{t-1} + \mathbf{W}_{xh}^T x_t) \tag{3.9}$$

$$\hat{y}_t = \mathbf{W}_{hy}^T h_t \tag{3.10}$$

At this point is important to introduce the notion of the vanishing gradient of which RNN suffers. The idea is that standard RNNs are not efficient if the gap between the output and what is relevant increases, so one solution is to consider more complex recurrent units called gated cells, where the Long Short Term Memory unit is the main example. Without entering into the mathematical details of this issue and the algorithm of backpropagation through time (BPTT) used to train RNNs, the attention is given to the architecture of these LSTM units that are also present in the model applied for this thesis, which was

introduced by [33]. Intuitively, they are simply an extension of the standard recurrent network presented above but with a more complex structure to detect more effectively relevant information. In particular, a LSTM contains computational blocks to control the flow of information as represented below:

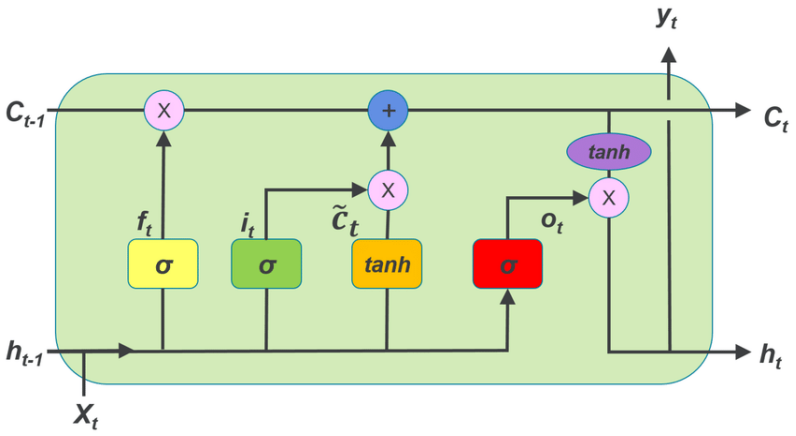

Figure 3.10: Long-Short Term Memory Unit

Just to give the intuition behind the functioning of this unit consider the following steps:

- **Step 1.** Irrelevant information are forgotten in this step, thus looking at h_{t-1} and x_t, the output of this first operation is 0 or 1 in the cell state C_{t-1}, obtained by applying a sigmoid function. Mathematically this step involves the computation of f_t as follows:

$$f_t = \sigma(\mathbf{W}_f \cdot (h_{t-1}, x_t) + b_f)$$

- **Step 2.** Relevant information are now stored in the new cell state. In particular, a two-step procedure is implemented. Firstly, a sigmoid layer decides which values must be updated and a tanh function creates a vector of new values $\widetilde{C}_t$ that could eventually be added to the state. Referring to figure 3.10, this means computing:

$$i_t = \sigma(\mathbf{W}_i \cdot (h_{t-1}, x_t) + b_i) \tag{3.11}$$

$$\widetilde{C}_t = tanh(\mathbf{W}_C \cdot (h_{t-1}, x_t) + b_C) \tag{3.12}$$

- **Step 3.** Now C_{t-1} is updated into C_t and this is done by considering the results obtained in the previous two steps. In particular,

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$

- **Step 4.** This is the output gate step in which the outputs of the LSTM unit are decided. This is done applying a sigmoid function to decide which parts of the cell state should be preserved and then this is multiplied to the tanh applied to C_t. With reference to the LSTM unit diagram above, this coincides with the computation of:

$$o_t = \sigma(\mathbf{W}_o \cdot (h_{t-1}, x_t) + b_o) \tag{3.13}$$

$$h_t = o_t * tanh(C_t) \tag{3.14}$$

Chapter 4

Dataset

The study of limit order books is not so common given also the difficulty of obtaining the appropriate data, since their access is generally limited or specific subscriptions to financial data providers are required. Moreover, they are high-frequency data and so their amount and dimension are extremely huge. The scope of this paragraph is to briefly introduce the key steps that lead to the creation of the dataset then used for the analysis. In particular, two fundamental aspects investigated are normalisation and data labelling.

4.1 Data Collection

Generally speaking, there are three alternatives available to collect data regarding the states of the Limit Order Book and they are the benchmark dataset called FI-2010 (the reference is [34]), reconstructed Limit Order Books (i.e. LOBSTER) or it is possible to use common data providers that give access to the LOBs for a given security traded at a given stock exchange (i.e. Thomson Reuters or Bloomberg). For completeness, it is important to recall that the benchmark dataset is made by observations from Nasdaq Nordic regarding five stocks traded over a period of 10 consecutive days. This is important since it is used to compare the performance of different models developed. However, for this thesis data have been directly obtained from Thomson Reuters Tick History that gives access to Level 2 data for stocks traded on NYSE. Initially, five stocks have

been selected: J.P.Morgan (Ticker: JPM), Johnson & Johnson (Ticker: JNJ), Berkshire Hathway (Ticker: BRK.A), Walmart (Ticker: WMT) and Pfizer (Ticker: PFE). However, the first instrument analysed was Pfizer leaving the others for further analysis or to use them to check the existence of universal properties in the Limit Order Book, training the model on Pfizer and then testing it on a different, out-of-sample, stock. Even if available data cover a period of time of one year, the analysis focused on a few days. The reason behind this choice is due to computational constraints. However, other research works, as underlined in the literature review, consider a dataset made by only a few days. Nevertheless, the objectives of this thesis are still pursued even with a limited sample. In fact, the idea is to check how the deep learning model works for NYSE data and to investigate the informativeness of LOBs to predict price movements.

Specifically, the analysis has taken into account three consecutive days of trading for Pfizer from 4th to 6th September 2019. This choice has been made in order to consider each day for training, validation and test sets. For each day all states of the Limit Order Book are recorded and information includes 40 features for each state; ten levels both for bid and ask side regarding both price and volume. In total, this leads to almost 40 million data points. To give an overview of the dataset used for this analysis, some basic descriptive statistics are presented below. In particular, for each day, the number of daily recordings and the average distance between two consecutive states of the LOB are presented in table 4.1.

	09/04/2019	**09/05/2019**	**09/06/2019**
Number of Obs	323,891	381,108	348,763
Avg Time Interval (seconds)	0.072	0.061	0.067

Table 4.1: Features Datasets Analysed

These statistics are helpful because they firstly highlight the huge dimension of HFT data. Then, the interesting element is the time interval (also indicated with k in other parts of the thesis) because it underlines that considering a forecasting horizon of k=100 means trying to predict the price movement direction after 7 seconds, which is a very short period

of time. Besides this, to give a full presentation of the dataset analysed, the following table contains some information regarding prices, volumes and spreads over these three days. In particular, the mean and standard deviation of price and volume are then used to normalise data. Notice also that these statistics do not distinguish between the bid side and the ask side.

	09/04/2019	**09/05/2019**	**09/06/2019**
Mean Price	35.84	36.28	36.38
St.Dev. Price	0.11	0.13	0.10
Mean Volume	3,265	2,845	2,950
St.Dev. Volume	2,827	3,305	2,543
Mean Spread	0.0105	0.0104	0.0103
St.Dev. Spread	0.0023	0.0022	0.002

Table 4.2: Descriptive Statistics about price, volume and spread

Note that both tables 4.1 and 6.3 refer to datasets already cleaned as explained below.

4.2 Dataset Generation

4.2.1 Data Cleaning

A very first step is to separate and create a different file for each day considered, this is necessary because preliminary analysis must be performed for each day separately. Once this computation has been performed, data are cleaned in the sense that only observations within normal trading hours are considered valid and auction trading sessions are deleted. For NYSE this means considering LOB states lying between 9.30 am and 4 pm. Besides this, an important note is that raw data are expressed in terms of Greenwich Mean Time (GMT) and so this must be taken into account when specifying the conditions to accept or not an observation in the Python script. However, for the period of time considered this is not a big issue but when a longer dataset is analysed the changing hour during the

year must be taken into account too. Since these three separate files on which analyses are performed are already the equivalent of the training set, validation set and test set, no further manipulation is needed.

4.2.2 Data Labelling

The final two steps of this pre-processing analysis are the most important ones because they are related to the creation of labels (that is the outputs) and the normalization of the dataset, both needed to correctly apply the machine learning algorithm. As regards the creation of the labels, these are categorical variables used to classify inputs and assigning to each state of the LOB a label expressing an upward trend of the price (Label used is 1), downward (Label used is 3) or a stationary trend (Label used is 2). Now, since we are dealing with high-frequency data and minimal variations occur every millisecond, the computation of price variation is such that it takes into account the nature of the data. Firstly, the mid-price is computed as follows:

$$p_t = \frac{p_b(t) + p_a(t)}{2}$$

In fact, it is known that there is no existence of a single price in stock markets but only ask and bid prices are available, thus for convenience, the mid-price is used as a proxy. Then the percentage change of this mid-price is defined as follows:

$$l_t = \frac{m_+(t) - m_-(t)}{m_-(t)} \tag{4.1}$$

$$m_-(t) = \frac{1}{k}\sum_{i=0}^{k} p_{t-i} \tag{4.2}$$

$$m_+(t) = \frac{1}{k}\sum_{i=1}^{k} p_{t+i} \tag{4.3}$$

Note that the above quantities depend on the forecasting period k, which is 100 in this case, where recall that this k refers to the time interval between two consecutive states of the LOB which are irregularly spaced. Then, the value computed in 4.1 is compared

to the value of a threshold α through which the label is determined. What follows holds true:

$$\begin{cases} \text{Label 1} & \text{if } \ l_t > \alpha \\ \text{Label 2} & \text{if } \ -\alpha \leq l_t \leq \alpha \\ \text{Label 3} & \text{if } \ l_t < -\alpha \end{cases}$$

A final remark regards the choice of α. This is not trivial and the underlying intuition behind the choice of this parameter for this thesis is based on the idea to capture the minimum percentage change of the price in order to be able to consider it an upward or downward variation. Furthermore, since the tick size at NYSE is 0.01 it follows that Pfizer stock price increases if it changes by at least 0.01 and so dividing this quantity for the average price over the sample period an α of 0.00028 is obtained. In the following plot, the mid-price is represented over a period of time of k = 3000 and labels are associated using green areas (1), red regions (3) and uncoloured ones (2):

Figure 4.1: Labels constructed using k=100 for Pfizer

4.2.3 Data Normalisation

The final step for the dataset generation is the normalisation. This computation is needed in order to efficiently apply machine learning algorithms, and, according to [2], a dynamic

normalisation procedure is more desirable and in fact, they implement a z-score normalisation. In particular, the following formula has been used:

$$x_{norm} = \frac{x - \overline{x}}{\sigma_x}$$

$$\text{where} \quad \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{and} \quad \sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

The authors compute mean and standard deviation using the observations of the five previous days and so the first five days of the dataset are then deleted from the analysis. Moreover, these stats are computed separately for price and volume features but without distinguishing between the bid and ask sides, since this seems more reasonable. However, for this thesis the same procedure cannot be followed since only three days are analysed. Then, z-score normalisation is still implemented but the mean and the standard deviation for both price and volume are computed considering each day and used to normalise that day.

Chapter 5

Data Analysis

The aim of this paragraph is to illustrate how results have been obtained and the model applied. In the chapter about the literature review a consistent analysis about models used to analyse LOB data has already carried out. Thus, this chapter goes deeper into the DeepLOB Model developed by [2], which is used for this thesis. After the investigation of the model architecture and the three main components involved, a bit of attention will be dedicated also to the practical implementation of it. In fact, computational issues are important to be discussed when these kinds of algorithm are studied.

5.1 Deep Learning Model: DeepLOB

From a general point of view, the model called DeepLOB is a machine learning algorithm that combines a Convolution Neural Network component to a Recurrent Neural Network one. In particular, the latter is represented by the Long-Short Term Memory units, while the former is given by the combination of two elements: standard convolutional layers and an Inception Module. In this way, the convolutional component aims at extracting the main features of the raw data, while the LSTM component is used to capture time dependencies in the dataset, but, as the authors underline, short-time dependencies are captured by the convolutional components as well. In a nutshell, diagram 5.1 summarizes the model architecture. Recall that inputs are simply raw data without any pre-processing

manipulation, the only element to notice is that every single input includes 100 states of the LOB. Thus, the dimension of a single input taken into account in this model is (100 x 40) where 40 are the features considered at each timestamp. In particular, it takes the following form:

$$\{p_a^{(i)}(t), v_a^{(i)}(t), p_b^{(i)}(t), v_b^{(i)}(t)\}_{i=1}^{n=10}$$

And, referring to 5.1, the Input unit will have the following dimension: N x 100 x 40 x 1. On the other hand, outputs are categorical variables representing the direction of the price movements over a forecasting horizon of k steps (refer to the previous chapter for a complete description of them). And so the dimension of the outcome of the model will be: N x 3

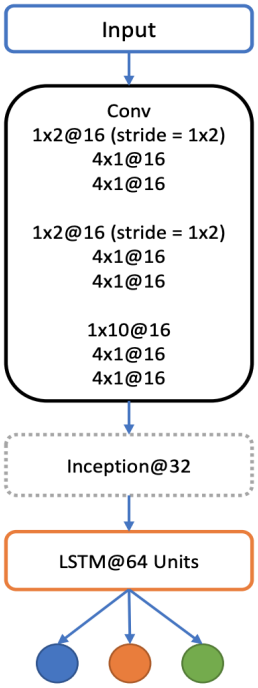

Figure 5.1: DeepLOB Model Architecture

5.1.1 Convolutional Layers

The first convolutional filter has dimension (1 x 2) and the stride applied is (1 x 2). It follows that this first filter summarises information between price and volume at each level of the book. Then a similar filter is used to extract features when information across

different order book levels are integrated. The final filter has a bigger dimension of (1 x 10) through which all information is combined. From 5.1 it is possible to notice that after these three main filters also other filters of dimension (4 x 1) are applied. To correctly interpret them, notice that they are used to capture relationships over four-time steps. A final remark regards the activation function used which is a so-called Leaky Rectifying Linear Units (Leaky-ReLU) that takes the following form:

$$\text{Leaky-ReLU}(x) = max(a \cdot x, x)$$

To conclude, the authors of [2] highlight that standard pooling layers could be treated with caution when working with time-series data instead of the classical problem of image recognition. This is the reason why this pooling unit is not inserted in this CNN.

5.1.2 Inception Module

This component of the model is used to combine together multiple convolutions to extract information about the behaviour over multiple timescales. Consider the following representation provided by the authors:

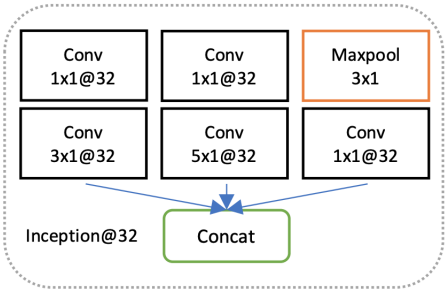

Figure 5.2: Inception Module Structure of the DeepLOB Model

The underlying idea of this unit is that (1 x 1) convolutions are applied to the input to obtain smaller representations of it, these elements are then transformed using different kernels of dimension (3 x 1) and (5 x 1) and these outputs are finally merged. The last step consists of a max-pooling layer.

5.1.3 Long-Short Term Memory Unit

As regards LSTM units, these are exactly as described in the previous theoretical chapter. The only two elements to underline are that 64 units are used and that the last output layer of this entire DeepLOB Model consists in a softmax function in order to obtain the probability of each price movement class at each time step.

5.2 Practical Implementation

Given the above framework, it is important also to quickly go through the code underlying the implementation of the model. This data analysis step could be divided into two sub-steps: the first one regards the pre-processing of raw data that leads to a manageable dataset, and then it is used in another code to actually apply the model, that is firstly trained and then tested. All the theoretical elements behind these processes have already been discussed, thus the aim of this paragraph is to understand better how they are implemented using Python. First of all, notice that the final sample analysed is made by three consecutive days of negotiations for Pfizer and so each day has been separated in a single `csv` file in order to manipulate them separately. This partition has been carried out manually using `Excel`. After this, each file is processed by three different codes that perform data cleaning, labels generation and data normalisation respectively. All these three steps are carried out using the Python programming language and some common libraries used in data science like `Pandas` and `Numpy`. More emphasis should be given to the code related to the DeepLOB Model, it is implemented using `Keras` and `Tensorflow`. In particular, the former is an open-source library used in Python and it provides an interface for artificial neural networks and among others it supports `Tensorflow` which is a library developed by Google to follow the entire building process of a machine learning algorithm. The model itself is defined as a function and inside it, each layer is inserted using built-in components of the above libraries. To be more precise, `Conv2D` is used to insert convolutional layers specifying the number of filters, their dimensions and the dimension of the stride. The same component but with different hyperparameters is used

to build the Inception Module. Then, the Long-Short Term Memory units are added in the model using `LSTM` built-in module, again the number of hidden layers must be specified. The final building block is the output layer in which the optimizer and the cost function are defined. Once the overall model has been built, it is used firstly to train the model, using the `fit` class, and then it is applied to the test set using the class `predict`. A final remark regards the choice about the number of epochs and the batch size. Intuitively, the first parameter represents the number of times the entire dataset goes through forward and backward the neural network, while the latter relates to the idea that feeding the algorithm directly with the entire dataset is demanding, thus it is common to divide it into batches with specific batch size. So, these two hyperparameters are set equal to 200 and 132 respectively in this model. Besides these elements, another parameter set is a in the Leaky-ReLU function which is equal to 0.01. The loss function chosen is the categorical cross-entropy and the optimizer is called the Adaptive Moment Estimation algorithm (ADAM) with the learning rate equals to 0.01.

To sum up, from a technical point of view, these are the main key elements to underline. Computationally, the programming part is manageable while another big issue relates to computational power. In fact, dealing with high-frequency data means working with very huge files and training a very complex model, with a lot of parameters to estimate, which requires a lot of computations. For future works, it is worth underlying that train, validation and test sets have a dimension of almost 1GB in total. The `csv` file contains almost one million rows and 45 million data points. Given these numbers, which only refers to three days and one stock, it is also important to highlight that computations require several days. In particular, for this thesis, a common laptop with 16GB of RAM and Intel i-7 Core as CPU, is used. It takes three days for the pre-processing step (one day for each day of data) and seven days to train and test the model, where each epoch takes more or less 40 minutes to train given 142,435 parameters to estimate.

Chapter 6

Results

To recap the key points of the empirical analysis, recall that Pfizer has been considered as instrument traded on NYSE. In particular, three days are taken into account where recordings for each day are used as train, validation and test respectively. The model implemented is DeepLOB by [2], and among the hyperparameters, the number of epochs chosen is 200. Moreover, the analysis has been conducted considering k=100, meaning that both train and test sets are processed considering the label built using this forecasting horizon. Results about the performance of the model are expressed using the following metrics:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6.1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6.3}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.4}$$

In these equations TP and TF indicate true positives and true negatives, meaning that the model has predicted correctly the true and negative classes. While FN and FP mean false negatives and positives, that is the model predicts incorrectly the outcomes. Intuitively, higher values for these statistics indicate a better performance of the model. In particular, accuracy is interesting because it captures the percentage of predictions that the algorithm got right.

6.1 Discussion

Firstly, consider the values for the metrics presented above about the goodness of the model. They are reported in the following table:

	Accuracy %	**Precision %**	**Recall %**	**F1 %**
DeepLOB	97.21	95.74	97.21	96.35

Table 6.1: Experiment Results for Pfizer

It is possible to notice that all values are particularly high. The accuracy metric is around 97%, meaning that the algorithm associates the correct label in 97 cases out of 100. However, this metric alone cannot capture the whole goodness of the model because it tends to not work well when the dataset is imbalanced. Thus, it could be helpful to consider also the value of F1 which is still particularly high. To sum up, the model performs very well and it is able to correctly predict the direction of the stock price after 100 states of the LOB. However, these interesting results have a drawback which is related to the nature of the labels as discussed in the next section.

Another element to consider in discussing the performance of the DeepLOB model is the loss and accuracy curves for the train set. These trends are shown below:

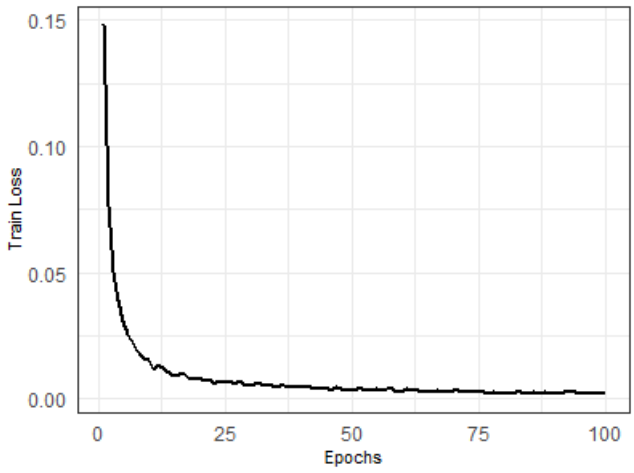

Figure 6.1: Evolution of Train Loss

From plots 6.1 it is possible to notice that the train loss converges to almost 0 very quickly and it reaches that value after more or less 100 epochs.

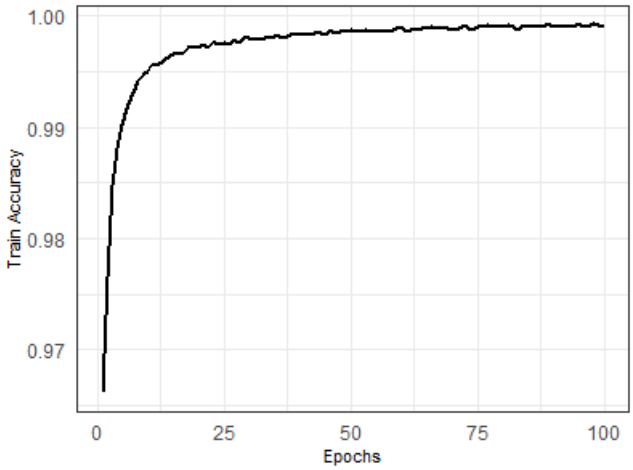

Figure 6.2: Evolution of Train Accuracy

Similarly, the train accuracy, in 6.2, is still very high from the beginning and tends to 1 as number of the epochs increases, reaching almost that value again after 100 epochs.

After this first analysis, the model trained on Pfizer is also applied to a comparable stock traded on NYSE and belonging to the pharmaceutical sector, that is Abbott Laboratories (ABT.N). The idea of this further application of the DeepLOB to an out-of-sample instrument is to check the universality of LOB dynamics. Firstly, some basic statistics and a graph (6.3) with labels over a period of k=3000 are reported below. In particular, the period considered to test the algorithm is only one day and it is the 6th of September, to maintain coherence with the analysis done for Pfizer. An interesting element to underline is that observations are one third of Pfizer ones and also the average time interval is higher, almost three times the time interval in Pfizer dataset. This is a key aspect since results obtained from the model are in part related also to this.

	09/06/2019
Number of Obs	117,135
Avg Time Interval (seconds)	0.199

Table 6.2: Main Features of Abbott Labs dataset

	09/06/2019
Mean Price	86.05
St.Dev. Price	0.2
Mean Volume	322
St.Dev. Volume	478
Mean Spread	0.0177
St.Dev. Spread	0.0122

Table 6.3: Descriptive Statistics about price, volume and spread for Abbott Labs

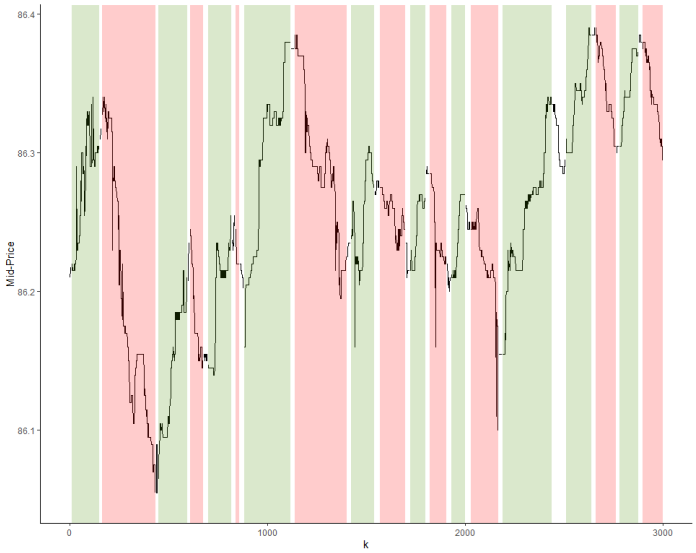

Figure 6.3: Labels constructed using k=100 for Abbott Labs

Thus, given this general framework, results obtained from the model are not as high as the ones got for Pfizer. In fact, accuracy is just 53%, and also other metrics are all only around 50%. While there is this discrepancy between the instruments, the results for Abbott are more similar to the ones obtained by [2], considering also that here only one day is taken into account. Thus, these results, very promising for Pfizer while more standard for Abbott, suggest that further analysis should be implemented but intuitively one of the principal reason is related to the fact that the train set used for Pfizer, and so to train the model, was not well balanced in the sense that labels where not uniformly distributed but mainly concentrated around Label 2.

6.2 Limits and Improvements

The analysis implemented in this thesis shows interesting results but it is only the starting point of a deeper investigation about this topic. In particular, there are two main limits that are given by the horizon of time covered by the dataset and the label construction process. As regards the former drawback, this is related to the fact that only a few days are taken into account while a more complete investigation should rely on more data like considering one year. Nevertheless, this choice is just driven by computational constraints and the analysis can be extended, with few adjustments, to a longer period of time just making sure to have a higher level of computational power. The second negative point of the analysis is the way in which labels have been created, in fact, this represents a key aspect that could have influenced heavily the results obtained. To be more precise, the key element is the correct choice of α which can be seen as the threshold value that indicates a percentage change in upward/downward trend. The main issue about this element is that data explored are high-frequency data and capturing changes in the mid-price is very difficult also because, as already highlighted, the interval time between two consecutive states of the LOB is around 0.07 seconds, meaning that analysing k=100 is equivalent to check what happens after 7 seconds and the result is that most of the time the stock price is stationary (Label 2). Thus, a sensitivity analysis about the choice of α is an aspect that should be investigated in future works.

To conclude this paragraph it is also important to highlight some aspects that could be improved in future analyses. Besides what is already stated above, there are different elements that could be taken into account in other works. Firstly, the model should be investigated deeper in order to understand which components can be improved. In this sense, different layers could be considered in the model, or a Bayesian approach should be implemented and also reinforcement learning is another interesting class of machine learning algorithm that could be studied to handle LOB data. A second aspect regards the dataset chosen. In fact, this thesis already takes into consideration stocks traded at NYSE, rarely analysed, but the period of time is limited. Future works should not only consider more observation but also investigate other financial instruments. Related to

this, another important aspect is to consider more deeply and maybe look for specific data sources, about dark pools and hidden orders that do not permit to completely appreciate the demand and supply for a certain instrument. Finally, the really fascinating aspect in studying the order flow is to use the model formulated to create real trading strategies and improve market making and HFT algorithms, choosing the right moment to send orders to sell/buy or predicting the direction of price movements and profit from them.

Chapter 7

Conclusion

The aim of this thesis was mainly to check the informativeness of Limit Order Book data and intuitively how the order flow and trade imbalances could predict future price movements direction. The main problem is related to computational constraints and so a limited sample has been analysed to support the initial thesis. However, the idea of looking directly at demand and supply as the ultimate element determining the price of financial instruments is something very interesting from our point of view, since imagining a static situation or like an opening auction, the idea to predict market forces and then determine the price is something practically interesting and true. Obviously, the velocity and dynamism of trading sessions make very difficult to predict price variations just looking at supply and demand since they are difficult to compute.

However, moving in this direction is very interesting and further efforts should be done and also practitioners could agree about the usefulness of this research. In particular, in this work a very recently-developed model has been used to pursue the above-cited goal, that is a deep learning algorithm that combines both a recurrent and a convolutional component, this was developed by [2] and the interesting element lies in precisely the combination of these two units that permit to extract relevant features from the dataset and then retrieve temporal dependencies on them. Obviously, this approach has the limit that it is a sort of black-box since it is not exactly possible to know which element contributes exactly to the prediction of outcomes. Moreover, from a practical point of view, it could be argued that

looking at LOB is limited since it does not permit to fully appreciate the entire demand and supply since orders could be also hidden/iceberg orders and also dark pools make the narrative even more complicated.

To conclude and also answer the two initial research questions, it is possible to argue that Limit Order Book is partly informative even if a more complete analysis should be done to understand better which elements contribute the most in predicting price movement directions. As regards universality, a clear conclusion cannot be derived due to the limited analysis implemented. Nevertheless, results are interesting and further research is desirable.

Bibliography

[1] Joel Hasbrouck. *Empirical market microstructure: The institutions, economics, and econometrics of securities trading*. Oxford University Press, 2007.

[2] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.

[3] Ping Zhou et al. *Nonlinear modelling of high frequency financial time series*. John Wiley & Sons Incorporated, 1998.

[4] JG Agrawal, V Chourasia, and A Mittra. State-of-the-art in stock prediction techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(4):1360–1366, 2013.

[5] David Easley, Marcos López de Prado, Maureen O'Hara, and Zhibai Zhang. Microstructure in the machine age. *The Review of Financial Studies*, 2019.

[6] Lawrence E Harris and Venkatesh Panchapagesan. The information content of the limit order book: evidence from nyse specialist trading decisions. *Journal of Financial Markets*, 8(1):25–67, 2005.

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

[8] Charles Cao, Oliver Hansch, and Xiaoxin Wang. The information content of an open limit-order book. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, 29(1):16–41, 2009.

[9] Fabrizio Lillo, Szabolcs Mike, and J Doyne Farmer. Theory for long memory in supply and demand. *Physical review e*, 71(6):066122, 2005.

[10] J Doyne Farmer 5, Laszlo Gillemot, Fabrizio Lillo, Szabolcs Mike, and Anindya Sen. What really causes large price changes? *Quantitative finance*, 4(4):383–397, 2004.

[11] Jean-Philippe Bouchaud, J Doyne Farmer, and Fabrizio Lillo. How markets slowly digest changes in supply and demand. In *Handbook of financial markets: dynamics and evolution*, pages 57–160. Elsevier, 2009.

[12] Bence Toth, Zoltan Eisler, Fabrizio Lillo, Julien Kockelkoren, J-P Bouchaud, and J Doyne Farmer. How does the market react to your order flow? *Quantitative Finance*, 12(7):1015–1024, 2012.

[13] Bruno Biais, Pierre Hillion, and Chester Spatt. An empirical analysis of the limit order book and the order flow in the paris bourse. *the Journal of Finance*, 50(5):1655–1689, 1995.

[14] Marc Potters and Jean-Philippe Bouchaud. More statistical properties of order books and price impact. *Physica A: Statistical Mechanics and its Applications*, 324(1-2):133–140, 2003.

[15] Rama Cont and Adrien De Larrard. Price dynamics in a markovian limit order market. *SIAM Journal on Financial Mathematics*, 4(1):1–25, 2013.

[16] Rama Cont. Statistical modeling of high-frequency financial data. *IEEE Signal Processing Magazine*, 28(5):16–25, 2011.

[17] Justin A Sirignano. Deep learning for limit order books. *Quantitative Finance*, 19(4):549–570, 2019.

[18] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Using deep learning to detect price change indications in financial markets. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2511–2515. IEEE, 2017.

[19] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Bdlob: Bayesian deep convolutional neural networks for limit order books. *arXiv preprint arXiv:1811.10041*, 2018.

[20] Zihao Zhang, Bryan Lim, and Stefan Zohren. Deep learning for market by order data. *arXiv preprint arXiv:2102.08811*, 2021.

[21] Zihao Zhang and Stefan Zohren. Multi-horizon forecasting for limit order books: Novel deep learning approaches and hardware acceleration using intelligent processing units. *arXiv preprint arXiv:2105.10430*, 2021.

[22] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[23] Ban Zheng, Eric Moulines, and Frédéric Abergel. Price jump prediction in limit order book. *arXiv preprint arXiv:1204.1381*, 2012.

[24] Dat Thanh Tran, Alexandros Iosifidis, Juho Kanniainen, and Moncef Gabbouj. Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE transactions on neural networks and learning systems*, 30(5):1407–1418, 2018.

[25] Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Temporal bag-of-features learning for predicting mid price movements using high frequency limit order book data. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(6):774–785, 2018.

[26] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Using deep learning for price prediction by exploiting stationary limit order book features. *Applied Soft Computing*, 93:106401, 2020.

[27] Antonio Briola, Jeremy Turiel, and Tomaso Aste. Deep learning modeling of the limit order book: A comparative perspective. *Available at SSRN 3714230*, 2020.

[28] Larry Harris. *Trading and exchanges: Market microstructure for practitioners*. OUP USA, 2003.

[29] Frank De Jong and Barbara Rindi. *The microstructure of financial markets*. Cambridge University Press, 2009.

[30] Martin D Gould, Mason A Porter, Stacy Williams, Mark McDonald, Daniel J Fenn, and Sam D Howison. Limit order books. *Quantitative Finance*, 13(11):1709–1742, 2013.

[31] Jerome H Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. springer open, 2017.

[32] Chris Chatfield and Haipeng Xing. *The analysis of time series: an introduction with R*. Chapman and hall/CRC, 2019.

[33] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[34] Adamantios Ntakaris, Martin Magris, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37(8):852–866, 2018.