University of Padova

Department of Department of Mathematics

Master Thesis in Data Science

Modeling Stock Order Book Dynamics with Marked Hawkes Processes

Supervisor
Massimiliano Caporin
University of Padova

Master Candidate
Darko Ivanovski

Academic Year
2023-2024

To my parents, Tomo and Svetlana, without whose sacrifices nothing would have been possible.

Abstract

I introduce a model used to describe the fluctuation of tick-by-tick financial time series. The model, based on market point process, allows to incorporate in a unique process the time between different transactions and their volume. The model was already used for the foreign exchange market, and I try to extend it to the stock market, using data sampling in millisecond. The main motivation for the model is the fact that the ”excitation” of the market is different in periods of time with low exchanged volume and high volume exchanged.

Contents

Listing of figures

Listing of tables

Listing of acronyms

HFT . . . . . High-Frequency Trading

HFD . . . . . High-Frequency Data

NYSE . . . . . New York Stock Exchange

TAQ . . . . . Trades and Quotes

ETF . . . . . Exchange-traded fund

MLE . . . . . Maximum Likelihood Estimation

BFGS . . . . . Broyden–Fletcher–Goldfarb–Shanno

SGD . . . . . Stochastic Gradient Descent

Introduction

At the end of the last century, the financial markets were first exposed to electronic trading. Year after year the number of trades performed via software exploded and the main difference between market participants became the speed of execution of those trades. At the beginning of 2000s, the execution time was measured in seconds, but only 10 years later it was measured in milliseconds and now even microseconds.

Since all of the trades became electronic, the data available exploded and new trading strategies were implemented: HFT, or High-Frequency Trading. Many started implementing models and strategies that analyze the trading data in milliseconds, such as volume, price, bid and ask.

In this thesis, we will analyze HFT data with millisecond precision in order to model the order book of the JNJ stock (Johnson & Johnson) between the 2nd January 2020 and the 30th March 2023. The main objective is to build a model that explains and can reproduce the order book, that is the available price to buy or sell the stock, and then investigate the correlation of the model's variables to macro-variables such as daily volatility.

The thesis is organized as follows. In Chapter 1 there is an introduction to high-frequency trading data and how they are organized in our case study, showing some of the first preliminary analysis and describing the necessary cleaning process.

In Chapter 2 there is a theoretical description of the Hawkes Processes that are used in order to model the data, starting from the basic definition and expanding it to the marked Hawkes Process that is the core of the analysis.

Finally, in Chapter 3 are presented the statistical analysis performed and the results of the model.

Concluding remarks are reported in Chapter 4.

1 High-Frequency Data

1.1 Characteristics of High-Frequency Data

The last few decades have seen an important transformation in the financial markets. Since the late 70s', when the first electronic order-routing system went into action on the NYSE, technology has improved tremendously and just a few decades later basically anyone can trade remotely in real time through different devices. As a result, the number of buying and selling actions that take place within a stock exchange on a daily basis has increased tremendously.

Moreover, not only the possibility to trade remotely was introduces, but also, and probably most importantly, the speed of execution has increased exponentially. That led to the creation of a new way of trading and a bunch of new strategies: High Frequency Trading, which is a practice that entrusts algorithms with the task of trading online. These algorithms are capable of analyzing and deciding which trades to make in the time frame of a few milliseconds, or even less, by going after even the smallest market opportunities. This automated method of trading has obviously increased the number of financial trades that take place in a single day.

The other important consequence of technological development is the creation of very powerful databases capable of storing an enormous amount of data. Decades ago, one was fortunate enough if he was able to obtain daily or hourly time series, and often only summary data such as

opening and closing prices were available. Now, one can have extremely detailed information on all the transactions that have taken place. There are now databases such as the NYSE's TAQ (Trades and Quotes), which contains information on all transactions that have occurred on the NYSE, NASDAQ, and other U.S. regional exchanges since 1993. These types of intraday data are called high-frequency data.

High Frequency Data (HFD) in financial markets capture trading activities at extremely short intervals, typically in milliseconds or microseconds. Understanding HFD is essential for analyzing market dynamics, price movements, and trading strategies. Unlike the fluidity of continuous time seen in theoretical models, buy and sell orders are executed not in continuous time but in discrete intervals known as ticks. HFD records trades and orders in a tick-by-tick fashion. This means that every transaction, whether it be a trade or an order update, is timestamped and recorded at the precise moment it occurs. The granularity of tick-by-tick data allows for detailed analysis of market micro-structure and price movements.

Central to the architecture of high-frequency trading are the various order types that govern market interactions. Market orders, executed at the prevailing market price, prioritize immediacy, facilitating swift transactions. In contrast, limit orders introduce a layer of price discretion, stipulating the desired price or better at which a trade is to be executed. This interplay between market and limit orders underscores the strategic decisions made by traders in navigating market liquidity and price dynamics.

A crucial element in market mechanics is the order book, a real-time ledger that organizes and displays all buy and sell orders for a particular security. This tool provides a snapshot of market liquidity, showing the demand and supply at various price levels. The order book is constantly updated, allowing traders to assess where they can place their orders and how price changes are likely to unfold based on the flow of market orders.

In parallel, the concept of the bid-ask spread becomes essential. The best bid represents the highest price that a buyer is willing to pay for a security, while the best ask is the lowest price a seller is willing to accept. The difference between the two is known as the bid-ask spread. This spread serves as a proxy for liquidity and transaction cost, with narrower spreads indicating more liquid markets, while wider spreads suggest lower liquidity and potentially higher transaction costs.

Another important relationship exists between trading volumes and transaction times. In markets with high liquidity, where trading volumes are substantial, transaction times are typically much faster. The abundance of buyers and sellers ensures that orders can be matched quickly. In contrast, lower liquidity often correlates with slower transaction times due to fewer market participants and orders available at a given price. This is true also when considering the changes in trading volumes and the difference in transaction times: when there is a sudden increase in the exchanged volume, there is a decrease in the time between two different transactions. This dynamic plays a critical role in the profitability of high-frequency trading, where even fractions of a second can make a significant difference.

1.2 Analysis of High-Frequency Data: Johnson & Johnson (JNJ) stock

The data that we are going to analyze are the data relative to the transactions of the Johnson & Johnson (ticker: JNJ) stock between 02/01/2020 and 30/03/2023, meaning 13 quarters of trade data. In the following figure, we can see how the dataset is organized before assigning the column names, cleaning it with various rules, and before any analysis.

```
> print(data)
             V1           V2       V3     V4     V5     V6
1   01/02/2020 04:10:16.275 146.0000 145.78 146.35      1
2   01/02/2020 04:10:16.275 146.1100 145.78 146.35     23
3   01/02/2020 04:10:16.275 146.2500 145.78 146.35     26
4   01/02/2020 04:34:12.650 146.2500 146.02 146.36     50
5   01/02/2020 04:42:31.748 146.2500 145.97 146.34     20
6   01/02/2020 04:47:02.667 146.2500 145.92 146.31     50
7   01/02/2020 04:47:57.996 146.0300 145.75 146.69     70
8   01/02/2020 04:48:03.770 146.0300 145.75 146.69     30
9   01/02/2020 05:51:53.310 146.0000 145.85 146.21     11
10  01/02/2020 05:51:53.310 146.0000 145.85 146.21     34
```

Figure 1.1: The first 10 rows of the dataset before any manipulation and cleaning

We can easily recognize the columns that our dataset has: there is the date and the time of the transaction, the price at which the transaction occurred, the bid and the ask at that time,

and the volume exchanged.

We need to perform a few data quality checks before using the dataset for our model and our statistical analysis. First of all, we know that the stock market is open between 9:30:00 and 16:00:00, Eastern Time. That means that we cannot have real transactions before 9:30 or after 16:00 (with 9:30:00 included and 16:00:00 excluded). One can see that those data exists in the figure 1.2.
We found out that of our original dataset of 68'103'149 observations, 1'054'811 were outside of the official trading hours, meaning that those are erroneous data and we need to remove them. That is a deletion of almost 1,55% of our dataset.

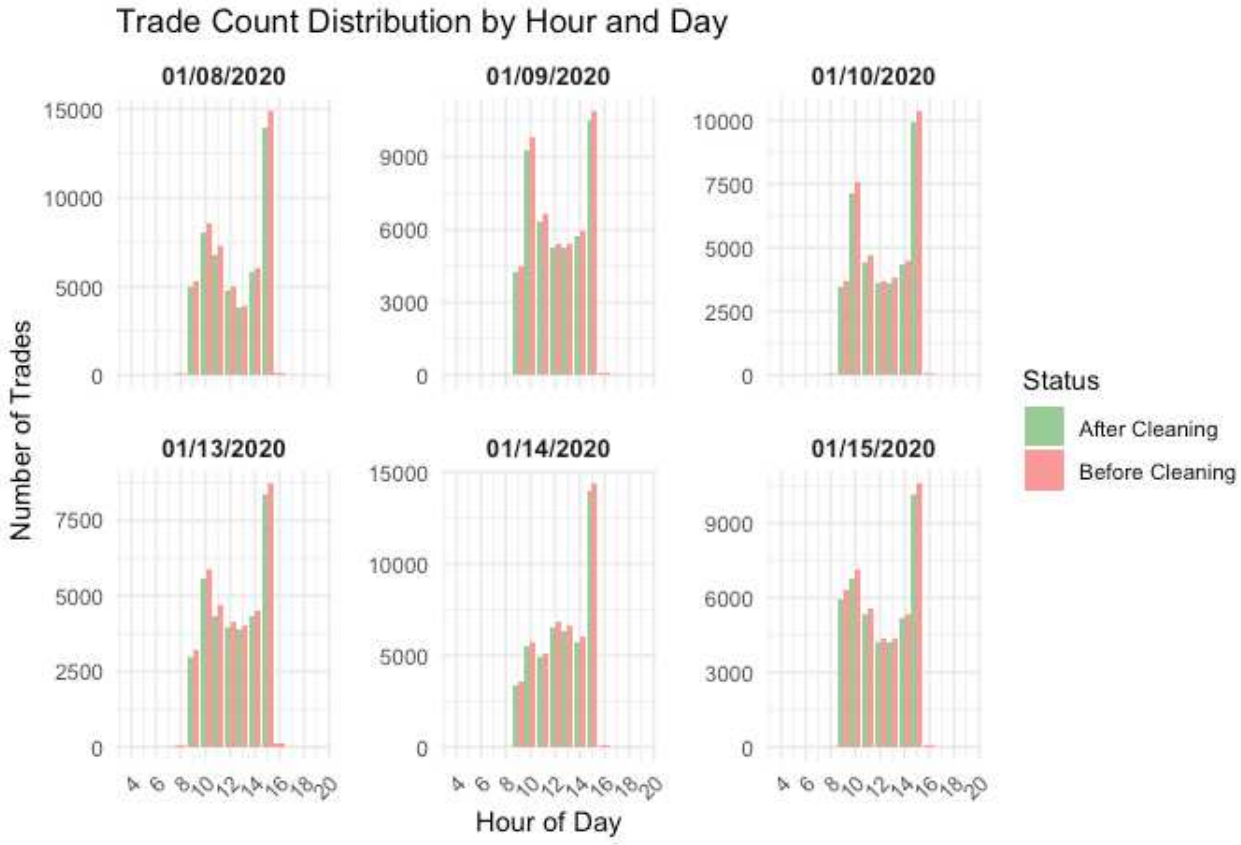

Figure 1.2: The first 6 days of the dataset visualized by trading hour. It's visible that there were data outside of the trading hours. The After cleaning data consists of all of the cleanings, i.e. trading hours, bid-ask-price consistency, order type determination.

Next, we need to check that the relationship between the transacted price, bid and ask is re-

spected. Given the definition of bid, which is the highest price that a buyer is willing to pay, and ask, which is the lowest price a seller is willing to accept, it is obvious that the bid is smaller than the ask and, moreover, the transacted price must be between the bid and the ask.
For example, if we have a transacted price greater than the ask price of a given moment, the buyer could have paid less for the same security given that a seller was willing to sell at a lower price. That is absurd given how the order book and the market makers work, so it is a data quality error that we must remove. One can see the existence of these types of data in the figure 1.3.

So, after checking for $bid \leq price \leq ask$ and removing the data that doesn't satisfy the relation, we are left with 63'425'697 data. That is approximately 93,13% of the original data.

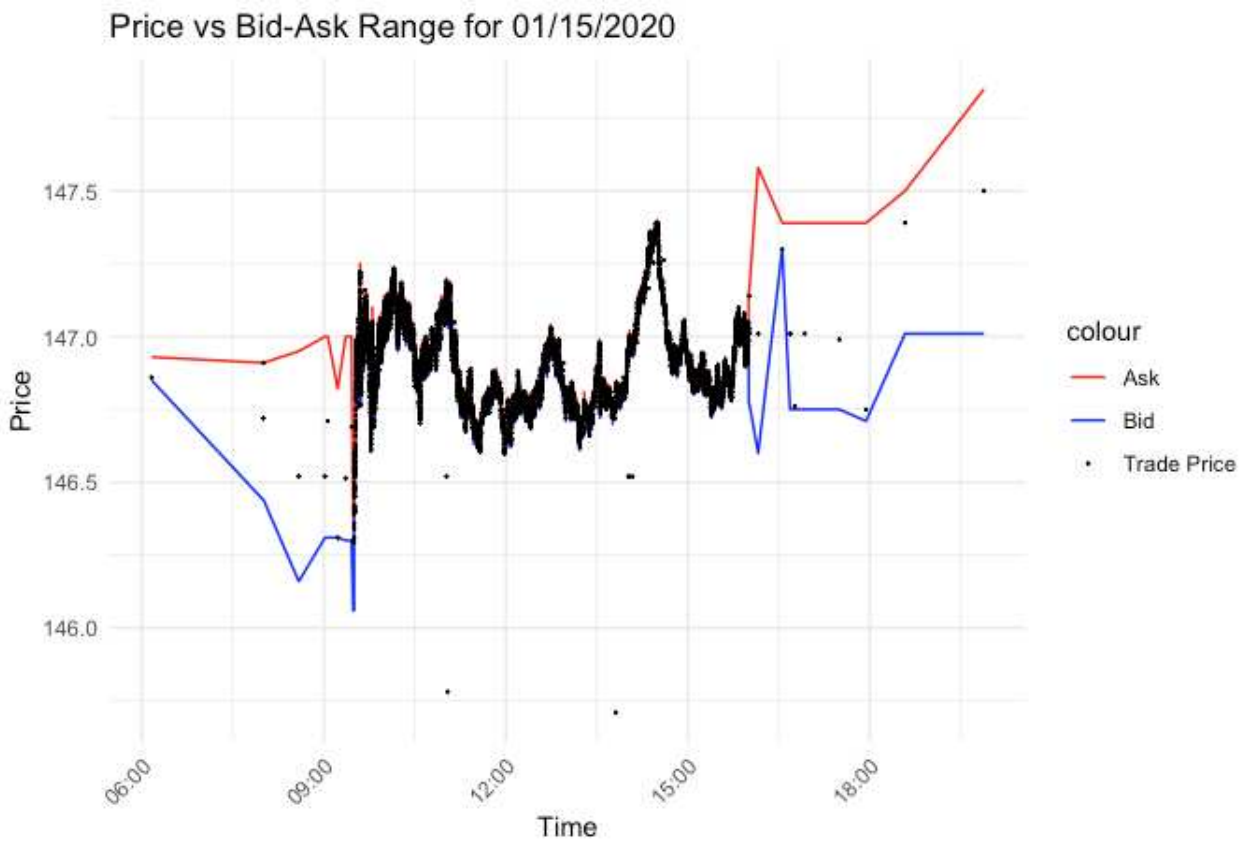

Figure 1.3: The bid-ask range and the price on 01/15/2020. It is visible the existence of prices outside the bid-ask, which is impossible.

After these straightforward data quality checks, we can infer the type of the transaction. That means we can assign a new column to our dataset, called *OrderType*, in which we can determine if a given transaction was the result of a buy order or a sell order. That is doable since, intuitively, if a new buy order arrives in the market and results in an actual transaction, it shifts the order book up: the new order was above the last bid price (if not the previous bid would have been a transaction) and exercise a buying pressure, moving the best bid up (possibly even the best ask if it fills all of the available shares at the previous ask).

To perform this classification, one can simply use the so-called tick test, where a transaction at a price higher than the previous transaction is a buy and reciprocally for a sell, or one can use the more robust Lee-Ready algorithm [1], which has an estimated accuracy between 88% and 92%.
Specifically, the Lee-Ready algorithm considers both the last trade price and the bid-ask spread from the order book. If the trade price is greater than the mid-point of the bid-ask spread, the trade is classified as a buy (buyer-initiated), since the buyer was willing to pay a price above the midpoint. However, if the trade price is less than the mid-point of the bid-ask spread, the trade is classified as a sell (seller-initiated), since the seller was willing to accept a lower price. If the trade price is exactly at the midpoint of the bid-ask spread, it performs a tick test with the previous transaction: if the current price is higher then the transaction was a buy, if it is lower then the transaction was a sell. If the price is the same again, the algorithm goes back again until it can determine if it was a buy or a sell, stopping at (usually) 10 previous transactions. If after 10 previous transactions it is still impossible to decide on the order type, we exclude that transaction from our dataset.

Applying the Lee-Ready algorithm to our dataset left 322'812 entries that cannot be classified, representing less than 0,5% of the original dataset. In addition, this classification tells us that approximately 50,9% of all of the transactions are Buy orders and 49,1% are Sell orders.

The reasons to perform this classification before applying our model to the dataset are multiple. First of all, as we will see in the next chapter, the order type influences our model. In principle, one can develop a model without using the order type. However, it is pretty intuitive that the direction of the trade has a huge impact on the order book, and trying to model the order book dynamics without the order type will give a worse result. Given that the deleted data points are less than 0,5% of the initial dataset, we decided to keep the order type as a key

element of our model.

Moreover, now that we have the order type of all transactions, we can aggregate different transactions that happen in the same timestamp. It is not uncommon to have multiple data points with the same timestamp, given that our dataset has a precision of milliseconds and there exist datasets with a microsecond precision. If two different transactions occur at less than a millisecond distance, our dataset is not able to assign them a different timestamp. To further clean our dataset, we can aggregate orders with the same timestamp and with the same order type. That means that we can still have transactions at the same time, but one Buy transaction and one Sell transaction.

In aggregating the transactions that have the same Date and Time we need to choose which price, bid, and ask to assign to the aggregated transaction. Since the objective of our work will be to model the order book and determine the bid and the ask, we decided to keep the last one that occurs in the dataset. The last data point represents the last bid and the last ask available, which is exactly the data that we are interested in. For the volume, however, we sum the volume of each transaction. So, in conclusion, when we find transactions with the same Date and Time for a given Order Type, we substitute those transactions with a single transaction that has the sum of the volume of each one as volume, the last price as price, the last bid as bid, and the last ask as ask.

After this last cleaning process, our database now has 38'105'623 observations, which is 55,95% of the original dataset dimension. The magnitude of the deleted and/or aggregated observations denotes the importance of analyzing and cleaning every dataset before applying models and statistics to it.

A summary of the operations performed on the dataset is presented in Table 1.1. It is worth noting that a given data point can have multiple inconsistencies and appear in multiple cleaning processes: for example, a data point that is out of trading hours can also have a price smaller than the bid.

In the figure 1.4, we can see how the dataset looks after the cleaning process and before using it to model the order book using the Hawkes processes, which we will now study in the next chapter from a theoretical point of view before applying it to our data.

Description of data	Number of data points	% to total	% to section
Whole dataset	68103149	100%	-
Out of trading hours	1054811	1,55%	-
Bid, Price, Ask order inconsistent	3711299	5,45%	-
Impossible to determine order type	322812	0,47%	-
Cleaned data	63102885	92,66%	100%
Buy orders	32115649	47,16%	50,89%
Sell orders	30987236	45,50%	49,11%
Aggregated data	38105623	55,95%	100%
Aggregated buy orders	19740268	28,99%	51,80%
Aggregated sell orders	18365355	26,97%	48,20%

Table 1.1: Summary of the data cleaning process

```
> print(allAggregated)
                Date         Time Volume   Price    Bid    Ask OrderType
              <fctr>       <char>  <int>   <num>  <num>  <num>    <char>
       1: 01/02/2020 09:30:00.304      3 145.870 145.51 146.09       Buy
       2: 01/02/2020 09:30:00.304      2 145.690 145.51 146.09      Sell
       3: 01/02/2020 09:30:00.609 134565 145.870 145.80 145.87       Buy
       4: 01/02/2020 09:30:00.663    100 145.870 145.80 145.87       Buy
       5: 01/02/2020 09:30:00.879      4 145.850 145.85 145.88      Sell
      ---
38105619: 03/30/2023 15:59:59.978    103 153.460 153.44 153.46       Buy
38105620: 03/30/2023 15:59:59.990    100 153.445 153.44 153.45      Sell
38105621: 03/30/2023 15:59:59.991    169 153.440 153.44 153.45      Sell
38105622: 03/30/2023 15:59:59.992      2 153.440 153.44 153.45      Sell
38105623: 03/30/2023 15:59:59.998      1 153.430 153.43 153.45      Sell
```

Figure 1.4: The first 5 and last 5 rows of the dataset after the cleaning process

2

Marked Hawkes Process

2.1 Modeling the order book

As we saw in Chapter 1, the main variables present in the high-frequency data are: Volume, Price, Bid, Ask. In the high-frequency trading world, the main goal is to find and implement some algorithms to predict a financial product's price in the near future, sometimes just a few milliseconds are enough to have a profitable strategy.

However, buy and sell orders did not arrive at continuous times, and counterparts do not meet them at any time, hence, the fluctuations of the stock market cannot evolve continuously. The data points are intrinsically irregular and discontinuous. In this scenario, the idea of using point processes to describe irregularities appears as evidence.

A point process defined on the non-negative real half-line, usually used to represent time, consists of a strictly increasing sequence of random times $(T_i)_{i\geq 1}$ with no accumulation points. Each time T_i can be interpreted as the time at which event i occurs, and consequently, T_i is called the "time of event i."

Equivalently, one can define a counting process N_t, where N_t is a function defined for $t \geq 0$ that takes only non-negative integer values. Its value represents the number of events in the

point process that have occurred before time t. Ultimately, N_t counts the number of events up to time t and is uniquely determined by the random sequence of times T_i in the point process. One can write:

$$N_t := \sum i > 0 \mathbb{1}[t \geq T_i] \tag{2.1}$$

where $\mathbb{1}[t \geq T_i]$ is a function that equals 1 if the condition $t \geq T_i$ is met and 0 otherwise. It can be observed that, naturally, $N_0 = 0$.

Between the point processes, models can be distinguished based on whether the occurrence of future events is influenced by past events or not. One of the simplest models that accounts for the dependency between events is the Hawkes process. Hawkes introduced such a model as a self-exciting point process, meaning that the occurrence of an event increases the probability of the next occurrence. This model has been useful for studying and modeling earthquakes and, more recently, in the study of spike trains in neurons.

The idea here is to try to model the buy and sell orders as Hawkes processes. The scientific literature related to high-frequency data analysis usually proposes to use self-exciting point processes that naturally correspond to the frequency of the arrival times of orders. However, we also want to include the volumes of the orders in our model, since we saw in Chapter 1 that an increase in the exchanged volume corresponds to a decrease in the time between two different transactions. Adding the volume is a "mark", so we will use the so-called Marked Hawkes process. This idea was developed and applied to the Forex market in [2].

Moreover, looking at the market dynamics and how an event, i.e. an order, influences the prices, the idea is to model the bid and the ask, not the price.
As we know, two principal types of orders can move the price: the market order and the limited order. A market order will immediately be executed at the best available price and at the relevant time. A buy market order will be executed to the best ask price, while a sell market order will be executed to the best bid price. The best ask (bid) price is understood as the lowest price at which an agent can buy (sell) an asset. The volume offered corresponding to the best ask (bid) price needs to be completely bought to see the best ask (bid) price move upward (downward).
A limited order is an order to buy (sell) an asset to a specific price, lower (higher) than the best ask (bid) price. Then, a new offered volume inside the bid-ask spread will move the price down-

ward or upward depending if it is proposed on the ask side or the bid side.

In summary, a market order will decrease the amount of shares available on the market while a limited order inside the bid-ask spread will increase them. Thus, it is the exchange volume that affects the price and the financial fluctuation. If we are able to model the order book, and in particular the best bid and the best ask, we should be able to reconstruct the price given a new future order.

2.2 Notion of Marked Hawkes Process

Let $N(t)$ be a d-dimensional point process, $N = (N_1, N_2, ..., N_d)$, with $N_i(t_i), 1 \leq i \leq d$ the cumulative number of events for the i^{th} component a time t_i.
An important characteristic of the process N is its conditional intensity, which is given by

$$\lambda(t \mid \mathcal{F}_t) = \lim_{\delta t \to 0} \frac{1}{\delta t} \mathbb{E}\big[N(t + \delta t) - N(t) \mid \mathcal{F}_t\big]. \tag{2.2}$$

The $\mathcal{F}_t$-intensity characterizes the evolution of the process N(t) with respect to this history $\mathcal{F}_t$. We could interpret that as the conditional probability at time t to observe a new event at the next time $t + \delta t$.

The Hawkes process is a particular class of the point process, defined by its intensity function. Normally, the intensity function for a Hawkes process is the following:

$$\lambda_i(t \mid \mathcal{F}_t) = \mu_i + \sum_{j=1}^{d} \alpha_{ij} \int_{-\infty}^{t} h_i(t - s) N_j(ds) \tag{2.3}$$

where the function h is called the kernel and satisfies the condition $h_i(t) \geq 0$ for all i. The parameters $\{\alpha_{ij}\}_{i,j=1,...,d}$ are referred to as the branching coefficient, which quantifies the ability of an event i to trigger an event of type j. Thus, we see the mutually exciting structure of the process since event $j \neq i$ can affect the conditional intensity $_i$. The self-exciting part is of course the case $j = i$, past and current events will induce a response in their own intensity process and therefore, on the corresponding point process. The constants $\{\mu_i\}_{i=1,...,d}$ are understood as the rate of instantaneous events.

Regarding the kernel, the integral in the equation can be also written as:

$$\int_{-\infty}^{t} h_i(t-s)N(ds) = \sum_{k|t_k<t} h_i(t-t_k), \tag{2.4}$$

where $\{t_k\}_{k=1,...,n}$ represents the arrival times of events. One can see that the kernel should be a decreasing function, meaning that as time passes an event has less and less impact. Since the kernel is continuous and non-negative, a classical choice for a kernel is the exponential function.

As mentioned before, we want to include the marks, i.e. the volume of the orders, in our model. A mark is an additional value attached to each point and brings some new information about the points. Consequently, we will have marked intensities, denoted in the sequel $\lambda(t, v_t \mid \mathcal{F}_t)$ where v_t represents the mark and will model the volume of the orders at time t. The marked intensity takes the form:

$$\lambda_i(t, v \mid \mathcal{F}_t) = \mu_i + \sum_{j=1}^{d} \alpha_{ij} \int_{(-\infty,t)\times\mathbb{R}^+} h_i(t-s)g_j(v)N_j(ds \times dv). \tag{2.5}$$

The function $g_j, j = 1, ..., d$ is the so-called impact function of marks; in other words, it characterizes the impact of the volume on the fluctuation of financial assets.
A standard choice for the impact function can be the power law or the exponential or the linear impact, $g(x) = x^\eta$ or $g(x) = e^{\eta x}$ or $g(x) = \eta x$, with $\eta > 0$.

Now, since our goal is to model the order book, meaning creating a model that can explain and reproduce the bid-ask spread, and since the order book is influenced by buy orders or sell orders, our idea is to diffuse conjointly the buys and the sells as two different processes. That means, we will have a 2-dimensional point process $N = (N_1, N_2)$, where N_1 is the Buy process and N_2 is the Sell process. In that way, we will be able to simulate the time t and the volume v of new orders, and starting from the order book at time $t = 0$ it should be possible to reconstruct the order book in the future.

The model will be multivariate, meaning that the arrival of a new buy order also influences the sell process. Intuitively, if there is a big buy pressure on the financial asset and a lot of new buy orders, that should also influence the probability of receiving a new sell order.
Taking in consideration all of this, and expanding the kernel and the impact function, we can write our model as:

$$
\begin{cases}
\lambda_1(t) = \mu_1 + \displaystyle\int_{(-\infty,t)\times\mathbb{R}^+} (\alpha_{11} + g_{11}(t,v))e^{-\beta_{11}(t-s)}N_1(ds \times dv) \\
\qquad\qquad + \displaystyle\int_{(-\infty,t)\times\mathbb{R}^+} (\alpha_{12} + g_{12}(t,v))e^{-\beta_{12}(t-s)}N_2(ds \times dv), \\
\lambda_2(t) = \mu_2 + \displaystyle\int_{(-\infty,t)\times\mathbb{R}^+} (\alpha_{21} + g_{21}(t,v))e^{-\beta_{21}(t-s)}N_1(ds \times dv) \\
\qquad\qquad + \displaystyle\int_{(-\infty,t)\times\mathbb{R}^+} (\alpha_{22} + g_{22}(t,v))e^{-\beta_{22}(t-s)}N_2(ds \times dv).
\end{cases}
\tag{2.6}
$$

The parameters of the model are the matrices μ, α, β. Moreover, depending on which functions we use as the impact function g, we will have a matrix of parameters that we can call η. For example, if we have a linear impact, we will have $g_{11}(t,v) = \eta_{11} * v$ and so on.
Summing up, our parameters are:

$$
\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}, \quad \eta = \begin{bmatrix} \eta_{11} & \eta_{12} \\ \eta_{21} & \eta_{22} \end{bmatrix} \tag{2.7}
$$

However, we can, of course, make a few simplifications to reduce the number of parameters in the model and make it more tractable. The first and simplest simplification is to use a single parameter, η, for the impact function. This approach is justified by the absence of any particular reasons why a mark should influence the buy process differently from the sell process, or why a process should be influenced differently depending on whether the mark originates from a buy order or a sell order. In principle, the two types of orders and processes should be equivalent, with no inherent bias in either direction.

Using the same reasoning, we can also simplify the kernel decay parameter to a single β. There is no strong justification for different parts of the processes to exhibit distinct decay behavior over time.

Regarding the μ variables, which represent the rate of instantaneous events and are constant, not depending on the history of the process, one could theoretically argue that a difference

might exist between the buy and sell processes. This is because the stock market, particularly the U.S. stock market, exhibits an inherent bias towards greater buy pressure than sell pressure. This bias arises from various factors, such as passive investing through instruments like ETFs and the regular inflow of capital from pension funds, which are legally mandated to deploy contributions received each month. However, while these dynamics may influence data on a weekly or monthly scale, they have no significant impact on the timeframes considered in this work. Therefore, we can reasonably set $\mu_1 = \mu_2 = \mu$.

Lastly, regarding the α parameters, which represent the branching coefficients, we can make two reasonable assumptions. First, the self-exciting processes for buys and sells should be identical, based on the earlier reasoning that there is no inherent bias between the two. Second, we can assume that the mutually exciting components of the process are symmetric. This means that the effect of a buy order on the sell process is equivalent to the effect of a sell order on the buy process. These two assumptions allow us to set $\alpha_{11} = \alpha_{22}$ and $\alpha_{12} = \alpha_{21}$.

After those simplifications, our model now has 5 parameters to estimate: $\mu, \alpha_{11}, \alpha_{12}, \beta, \eta$.

2.3 Estimation Procedure

To estimate the parameters of the multivariate marked Hawkes process, the common choice is to use the Maximum Likelihood Estimation (MLE) method. This technique allows us to estimate the parameters that maximize the likelihood of observing the given data, under the assumption that the data follows the specified process. The general form of the likelihood function for a point process is based on the conditional intensity function, which represents the rate at which new events are expected to occur, conditioned on the history of past events.

For a multivariate marked Hawkes process with multiple prices (e.g., ask and bid prices), let Θ be the parameters set which depend on mark distribution $f = (f_1, \ldots, f_d)$, also consider the impact function $g = (g_1, \ldots, g_d)$, the decay kernel $h = (h_1, \ldots, h_d)$, the branching matrix $\alpha = \{\alpha_{ij}\}_{i,j=1,\ldots,d}$ and $\mu = (\mu_1, \ldots, \mu_d)$. Let $I = [T^-, T^+]$ be interval containing all arrival times. With these notation, the likelihood function is given by:

$$L(\{t_i, v_i\}; \Theta) = \prod_{j=1}^{d} \int_{I \times \mathbb{R}} \lambda_j(t, v(t)|F_t) N_j(dt \times dv) e^{(-\Lambda_j(T^+))}, \tag{2.8}$$

where $\Lambda_j(T)$ is the compensator, or integrated intensity given by

$$\Lambda_j(T) = \int_{-\infty}^{T} \lambda_j(t, v|F_t)dt \times dv, \quad j \in \{1, \ldots, d\}. \tag{2.9}$$

where $\lambda_j(t, v(t)|F_t)$ is the intensity function for price j at time t and volume $v(t)$.

The log-likelihood function is derived from the likelihood by taking the logarithm of the product, which simplifies to:

$$\log L(t_i, v_i; \Theta) = \sum_{j=1}^{d} \sum_{k=1}^{N_j} \log \lambda_j(t_k, v(t_k)|F_{t_k}) - \sum_{j=1}^{d} \Lambda_j(T^+). \tag{2.10}$$

In our case, where we have $d = 2$ and $N_j = (N_1, N_2)$, this function is maximized using optimization algorithms to find the parameter set $\Theta = \{\mu, \alpha_{11}, \alpha_{12}, \beta, \eta\}$, which includes the baseline intensity μ, the branching coefficients α_{11} and α_{12}, the decay parameter β, and the impact parameter η.

The impact function $g(v)$ quantifies the effect of transaction volume v on the asset price fluctuation. We choose a linear impact function, but a power law impact function is also possible, as it provides a good fit to the data and an easy to understand meaning. The linear impact function is given by:

$$g(v) = \eta v,$$

where $\eta > 0$. This choice of impact function reflects the fact that large trading volumes have a more significant impact on price movements than smaller ones.

Once the likelihood function is defined, we compute the log-likelihood function, as seen in 2.10, and use numerical optimization methods, such as the BFGS algorithm, to maximize the function and obtain the parameter estimates.

3

Model implementation and analysis

3.1 Parameter estimation

When modeling an order book, it is essential to recognize that the model represents the order book of a single broker. Each broker maintains their own order book, along with unique agents and market participants. Thanks to arbitrageurs and market makers, the last traded price typically aligns across brokers. Instances where bid and ask prices differ between brokers—creating arbitrage opportunities—are usually identified and resolved swiftly.
This distinction is crucial for anyone looking to apply a model in the real world or implement a trading strategy based on it. However, for the purposes of our work, this consideration is not particularly relevant. We can proceed with the available data without any significant concerns.

To estimate our five free parameters, we first need to determine the most effective approach given the amount of data available. With data spanning 817 trading days, training a single model that incorporates all of it would be computationally prohibitive—at least with the resources currently at our disposal.
Therefore, we present results obtained by training the model on both daily and weekly data. Additionally, we examine in detail the 2020 COVID-induced market crash period to identify any notable differences compared to more typical market conditions.

Another important factor to consider is that, given the large dataset size, performing meaningful statistical significance tests becomes challenging. With a large dataset, all parameters are likely to appear statistically significant, as their standard deviations σ become relatively small. This occurs because, in general, $\sigma \propto \frac{1}{\sqrt{n}}$, where n represents the size of the dataset. As σ approaches zero, the test statistic becomes larger, causing the p-value to approach zero as well.

Our parameters were estimated using the Maximum Likelihood Estimation (MLE), and the optimization process was performed with the classical BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm. This algorithm has complexity $\mathcal{O}(n^2)$, which is great for an optimization algorithm and is a standard choice in this kind of fit.
We present, in the figure 3.1 below, the result of the fit for the first day we have in our dataset, January 2nd 2020.

```
--------------------------------------------
Maximum Likelihood estimation
BFGS maximization, 89 iterations
Return code 0: successful convergence
Log-Likelihood: -14751.84
5  free parameters
Estimates:
          Estimate Std. error  t value Pr(> t)
mu1      4.269e-01  2.965e-03   143.99  <2e-16 ***
alpha1.1 3.815e+01        NaN      NaN     NaN
alpha1.2 8.383e+00        NaN      NaN     NaN
beta1.1  2.310e+02  1.921e-02 12025.97  <2e-16 ***
eta1     9.028e-02  2.284e-03    39.53  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
--------------------------------------------
```

Figure 3.1: Fit results for data on 02/01/2020

We can see that the convergence is successful and we have the first estimate for our parameters, all of them with a high t-value and thus high statistical significance. We can also see that the estimation of the parameters α_{11} and α_{12} produces NaN values, likely because the standard error becomes too small to be reliably calculated.

Having a successfully convergent model is essential to continue our analysis, so after training the model for January 2nd 2020 we trained a new separate model for each day of 2020. There

were 253 trading days in 2020, so we produced 253 different sets of our 5 free parameters. Additionally, we also included the training time needed to perform the estimate on our machine (an Apple M3 chip). Below are the charts of the parameters for each day of 2020.

The first thing that we can see, and it is extremely visible looking at the chart 3.3 for example, is that there are differences in the parameters during the turbulence of the market between the end of February 2020 and the middle of April 2020. To have a reference, we also include a chart of the stock price of JNJ in 2020 in figure 3.2.
As we can see, during the COVID crash, the price fell sharply, with almost a -30% from top to bottom in the span of 50 days, and the traded volumes during the days increased, almost doubling, from an average of 7.5-8 million shares traded normally to an average of 15 million shares traded between the highlighted period.

Figure 3.2: Stock price of JNJ during 2020, with the covid induced turbulence visible as reference

It is perfectly normal then to see in Figure 3.8 that the training time "exploded" during those days. The average training time between February 21 and April 23 was around 280 seconds, while the average training time outside that period was around 164 seconds.
As expected, the more data we have, i.e. the more transactions are registered in a single day, the more time our model needs to perform the estimate.

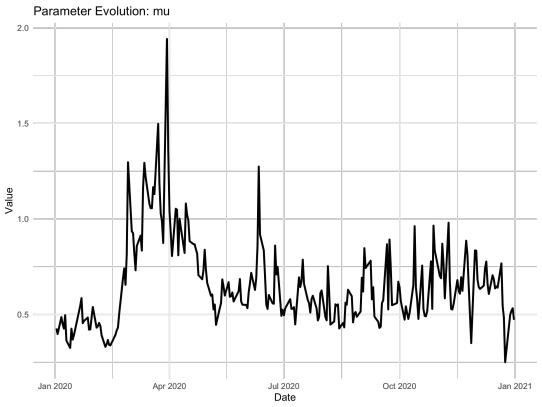

Figure 3.3: Estimates of μ for daily training during 2020

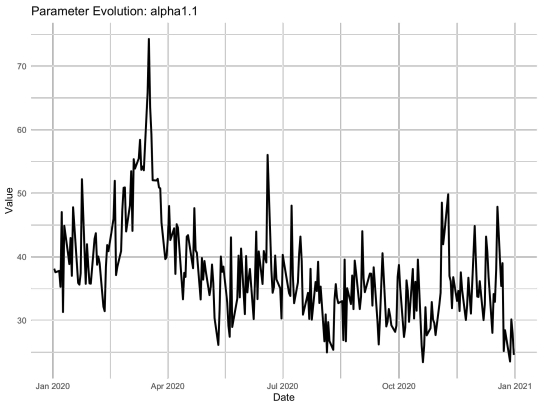

Figure 3.4: Estimates of α_{11} for daily training during 2020

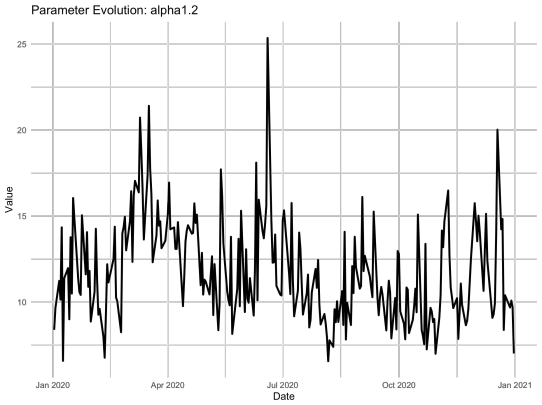

Figure 3.5: Estimates of α_{12} for daily training during 2020

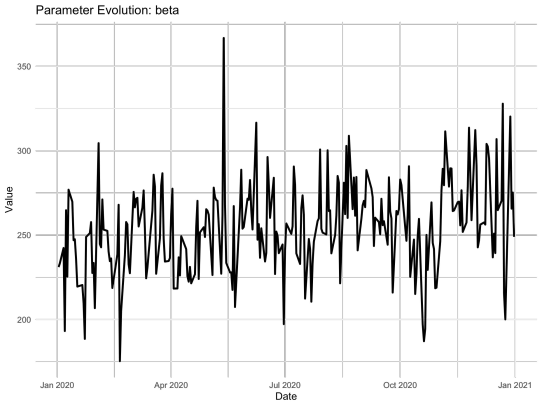

Figure 3.6: Estimates of β for daily training during 2020

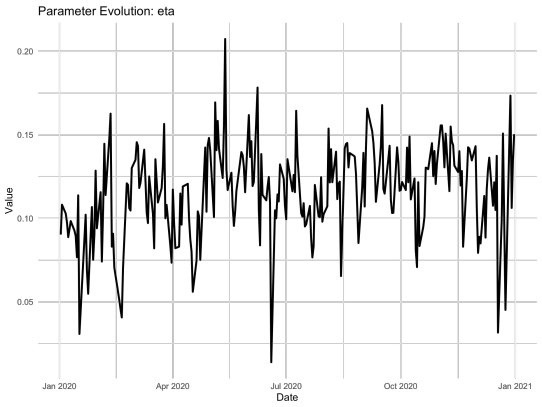

Figure 3.7: Estimates of η for daily training during 2020

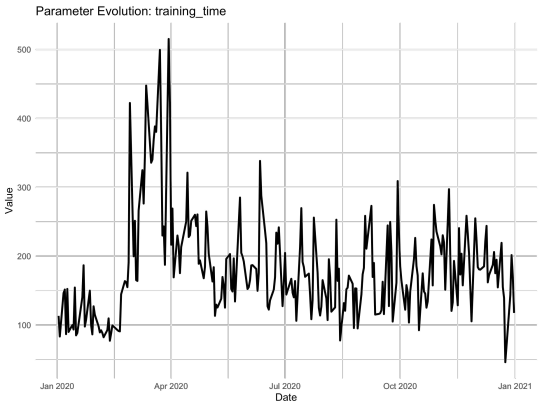

Figure 3.8: Training time for daily training during 2020

It is interesting to analyze the behavior of our variables across the different estimations. For instance, the variables β and η, visible in the charts 3.6 and 3.7, appear to be unaffected by the broader market trends, as there is no significant difference between periods of market turbulence and normal market conditions.
Given the meanings of these parameters, we can conclude that the time decay parameter β and the impact parameter η are not directly influenced by market conditions. This suggests that the rate at which the influence of a past trade decays remains constant across different market conditions, and that the volume of a transaction exerts the same impact regardless of whether the market is in a period of turbulence or normalcy.

Regarding the mutually exciting components of our model, which are summed with the parameter α_{12}, we see in Figure 3.5 that there appears to be a limited impact from the market conditions, but definitely not as visible as in Figure 3.4 regarding the self-exciting part of the process.
This can be intuitively understood: during periods of market turbulence, stock price movements often exhibit self-reinforcing behavior. When the stock is rising, increased exuberance typically leads to more buy orders, which in turn drives the price even higher. Conversely, during market crashes, a significant drop in price often triggers additional selling, further driving the decline. These self-reinforcing dynamics are influenced not only by human psychology but also by market structure factors, such as leverage and options. For example, margin calls and short squeezes can exacerbate these movements. And, in our example, during the market crash of 2020 it is well-documented that a significant number of margin calls occurred, for example in [3].

Lastly, the spontaneous events, represented by the variable μ shown in Figure 3.3, reveal a strong correlation between volatility and μ. In Figure 3.9, we have plotted the daily price range of JNJ, which is the difference between the highest and lowest prices observed during each trading day in 2020. Additionally, Figure 3.10 illustrates the intraday volatility, calculated as the standard deviation of intraday returns. These returns are defined as the percentage change in price between two consecutive trades.

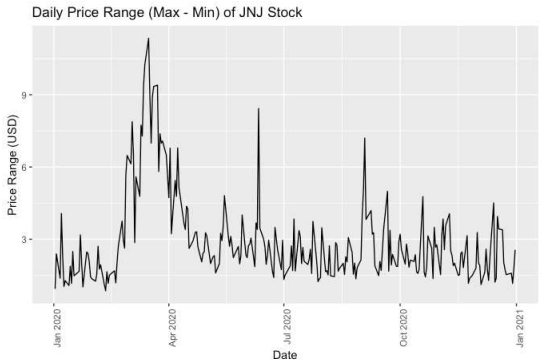

Figure 3.9: Daily price range (max-min) of JNJ stock during 2020

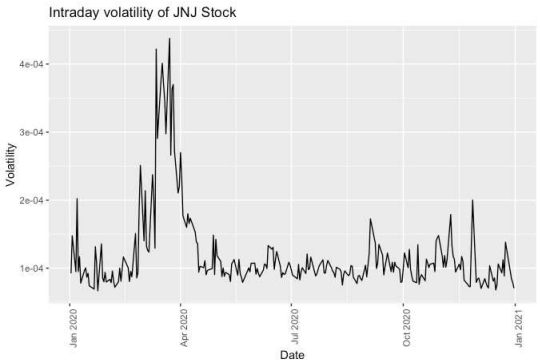

Figure 3.10: Intraday Volatility of JNJ stock price during 2020

To confirm our intuition, we calculated the correlation matrix for all of the variables presented here. We can see in the Table 3.1 the results: our μ variable is strongly correlated with the daily range and the intraday volatility. Our α_{11} has a moderate correlation to the intraday volatility and the daily range but a high correlation to α_{12}, while α_{12} is weakly correlated to volatility and range.
Also, we can clearly see that β and η are completely uncorrelated to to the other variables, with just a moderate correlation between them.
Lastly, as expected, the training time is very strongly correlated with μ and strongly correlated to the intraday volatility and the daily range.

We can now analyze what happens when training the model using weekly data instead of the daily ones used until now. The results are presented in the charts below from 3.11 to 3.16.

	Volatility	**Range**	μ	α_{11}	α_{12}	β	η	**Training**
Volatility	1.0000	0.7897	0.6346	0.5862	0.3989	-0.0547	0.0027	0.6207
Range	0.7897	1.0000	0.7683	0.5899	0.4454	-0.0431	0.0323	0.7161
μ	0.6346	0.7683	1.0000	0.4350	0.3943	-0.0088	0.0166	0.9004
α_{11}	0.5862	0.5900	0.4350	1.0000	0.8038	0.1838	-0.2522	0.3921
α_{12}	0.3990	0.4454	0.3943	0.8038	1.0000	0.2717	-0.2727	0.3340
β	-0.0547	-0.0431	-0.0088	0.1838	0.2717	1.0000	0.5620	0.0333
η	0.0027	0.0323	0.0166	-0.2522	-0.2727	0.5620	1.0000	0.0971
Training	0.6207	0.7161	0.9004	0.3921	0.3340	0.0333	0.0971	1.0000

Table 3.1: Correlation matrix for our variables calculated with daily estimates.

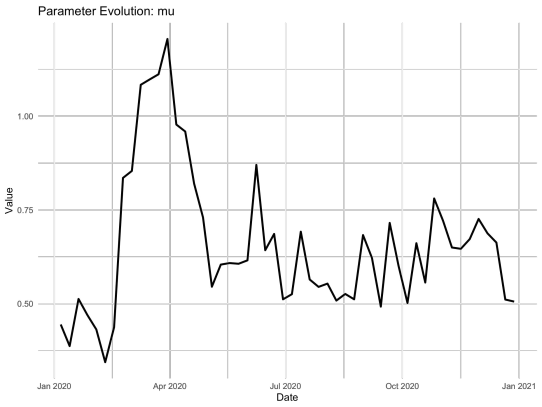

Figure 3.11: Estimates of μ for weekly training during 2020

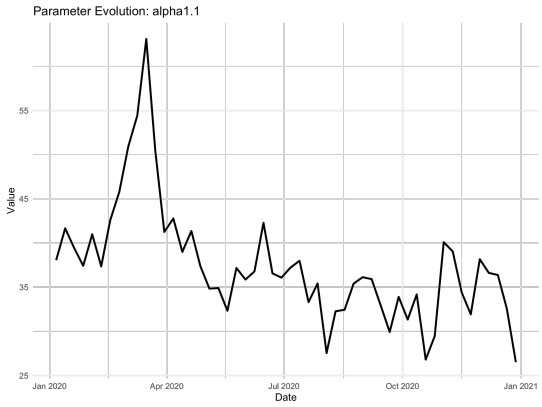

Figure 3.12: Estimates of α_{11} for weekly training during 2020

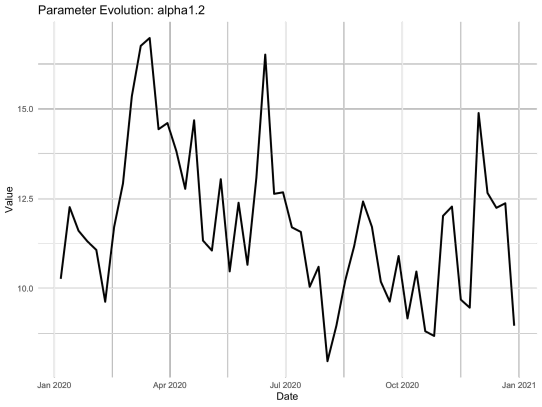

Figure 3.13: Estimates of α_{12} for weekly training during 2020

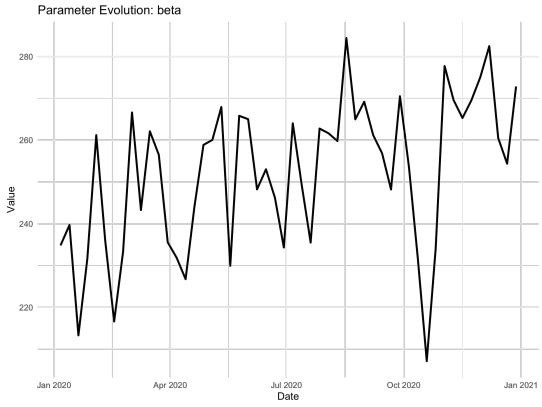

Figure 3.14: Estimates of β for weekly training during 2020

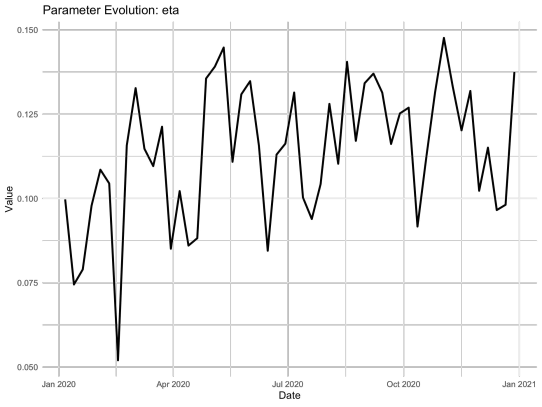

Figure 3.15: Estimates of η for weekly training during 2020

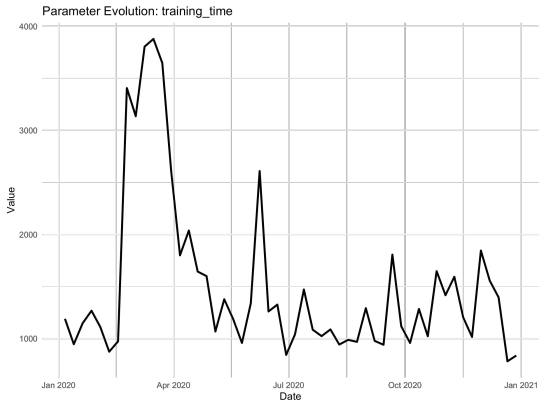

Figure 3.16: Training time for weekly training during 2020

We trained the model for each week in 2020 starting from Monday to Friday, so the first week was between January 6 and January 10, while the last one was between December 28 and December 31.
However, since we didn't find any meaningful significance in the intraday volatily when also having the daily range, to perform our analyses on the weekly data we considered only the weekly range, i.e. the difference between the minimum and the maximum price of that week. An intraweek volatility can also be introduced but it's not as commonly used. The chart of the weekly range is presented in 3.17.

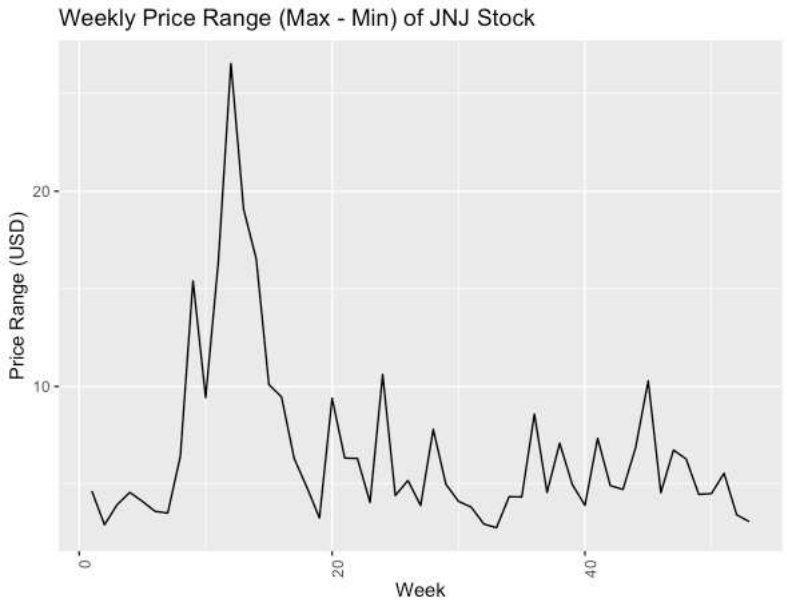

Figure 3.17: Weekly price range (max-min) of JNJ stock during 2020

The correlation matrix is presented in Table 3.2. As expected, the strong correlation between μ and the price range and between α_{11} and the range remains. Actually, it slightly increased, but that can be due to the fact that there are less estimates (52 vs 253).
Moreover, all of the considerations done for α_{12}, β and η remain valid. At first glance, there aren't major differences between the weekly estimates and the daily ones.

	Weekly Range	μ	α_{11}	α_{12}	β	η	**Training**
Weekly Range	1.0000	0.8070	0.7471	0.6151	-0.0435	0.0225	0.8611
μ	0.8070	1.0000	0.5861	0.6424	-0.0209	0.0053	0.8568
α_{11}	0.7471	0.5861	1.0000	0.8286	-0.0414	-0.2339	0.7744
α_{12}	0.6151	0.6424	0.8286	1.0000	0.0476	-0.2316	0.6724
β	-0.0435	-0.0209	-0.0414	0.0476	1.0000	0.6498	-0.0458
η	0.0225	0.0053	-0.2339	-0.2316	0.6498	1.0000	-0.0079
Training	0.8611	0.8568	0.7744	0.6724	-0.0458	-0.0079	1.0000

Table 3.2: Correlation matrix for our variables calculated with weekly estimates.

An interesting aspect to explore when comparing the weekly and daily estimates is the training time. As mentioned earlier, the average training time for daily estimates outside of the market crash was approximately 164 seconds, while during the market crash, it increased to about 280 seconds.
For the weekly estimates the average training time outside the crash was around 1218 seconds and, during the crash, it rose significantly to around 2885 seconds. Since each week typically consists of 5 trading days (with some weeks having only 4 days due to holidays), it's important to note that the training time is not linear. The ratio of training time between weekly estimates and daily estimates is more than 7 times higher under normal conditions, and this ratio increases to over 10 times during the market crash.

This outcome is not surprising, considering that our model is nonlinear and the optimization algorithm has a complexity of $\mathcal{O}(n^2)$. Since no clear advantages were observed from training the model with weekly data, we can conclude that training the model with daily data is sufficient.

3.2 Limitation to our model

Before concluding, it is important to address potential limitations of our model and suggest areas for future improvements.
Firstly, while the assumptions made to reduce our parameters to five are reasonable, they may not fully capture the complexity of order book dynamics. For instance, the inherent bias towards buy orders, driven by passive investing and pension fund flows, could influence the mar-

ket, challenging our assumption that buy and sell orders behave identically. If buy and sell orders do indeed exhibit different characteristics, it is plausible that the impact function and decay rates should be treated separately. While we believe that such differences do not significantly affect high-frequency trading (HFT) models, it remains a theoretical consideration.

Another important consideration is market conditions. While we analyzed the data during both normal periods and turbulent events like the COVID crash, the model may not fully capture emerging market trends or sudden shifts in trading behavior. As such, while the model is theoretically sound and could be applied in real-time, it requires continuous updates to ensure that any new market trends or behaviors do not impact its accuracy.

Finally, we must acknowledge the computational complexity of the model. Despite reducing the number of parameters, training the model on large datasets still requires significant time. This could pose a limitation for real-world, real-time applications, where faster and more flexible models are needed.
For example, regarding the optimization algorithm used to estimate the model parameters, we employed the BFGS (Broyden–Fletcher–Goldfarb–Shanno) algorithm, which, while effective, may not be the most efficient choice for the complexity of our model, particularly when dealing with large datasets. Although BFGS is widely used due to its relatively good convergence properties, there could be alternative optimization methods that offer better performance, especially in terms of computational efficiency or the ability to handle non-convexity in the parameter space.
For example, algorithms like stochastic gradient descent (SGD) or more advanced techniques such as the Adam optimizer, commonly used in machine learning, could potentially provide faster convergence with large datasets. Future work could investigate the applicability of these algorithms to see if they can improve the model's efficiency or accuracy in real-time applications.

An interesting open question arises regarding the temporal granularity of the data used to train the model. In this work, we trained the model using data from a full trading day, or using a whole week of data, but what would happen if we trained the model on shorter time periods, such as half a trading day or even a few hours? It is possible that the dynamics of the order book behave differently over shorter time scales, and such a modification could potentially improve the model's responsiveness to intra-day fluctuations. Exploring this possibility could reveal

whether training on shorter intervals could enhance predictive accuracy or adapt the model to fast-changing market conditions.

4 Conclusion

In this thesis, we developed a model to analyze the dynamics of the stock order book using Marked Hawkes Processes. The objective was to better understand high-frequency trading (HFT) data and its impact on price movements, specifically for the Johnson & Johnson (JNJ) stock. The model was applied to a large dataset spanning from 2020 to 2023, and specifically focusing on 2020, and key parameters were estimated to understand the behavior of buy and sell orders.

The results demonstrated that our model could effectively capture the self and mutually exciting nature of the market, with both buy and sell orders influencing future market activity. In particular, the model was able to capture key market phenomena during both normal market conditions and periods of high volatility, such as the COVID-induced market crash of 2020.

This thesis tries to provide a theoretical model that could be adapted for real-time market analysis and prediction. While our model was focused on JNJ stock, it is likely applicable to other financial instruments and could be used by traders or financial institutions to better understand order book dynamics and make more informed trading decisions.

Although the model provides valuable insights, certain limitations were identified, such as the assumption that buy and sell orders behave identically and the computational challenges posed by large datasets. Additionally, while the model showed promising results, it is impor-

tant to continuously revise the assumptions to ensure that new market trends or behaviors are accounted for.

Future works could focus on refining the model by exploring the impact of shorter training periods, such as half a trading day, to better capture intra-day fluctuations. Additionally, testing the model with different stocks or other financial instruments could further validate its robustness. Further exploration into alternative optimization techniques and the inclusion of external factors, such as macroeconomic data, could enhance the model's applicability in real-time trading scenarios.

This thesis presents a valuable approach to modeling high-frequency trading dynamics using Marked Hawkes Processes, contributing to the understanding of order book behavior and its impact on price movements. While the model offers insights into market activity, its real-world application could be further enhanced by addressing the limitations discussed and incorporating more adaptive techniques. As financial markets continue to grow more complex, advancing models that capture their evolving dynamics will be crucial for both theoretical development and practical applications in trading strategies.

References

[1] C. M. C. Lee and M. J. Ready, "Inferring trade direction from intraday data," *The Journal of Finance*, vol. 46, no. 2, pp. 733–746, 1991.

[2] A. Fauth and C. A. Tudor, "Modeling first line of an order book with multivariate marked point processes," *SAMM, Université Paris 1 Panthéon-Sorbonne*, 2021, presented June 17, 2021.

[3] I. Basel Committee on Banking Supervision, CPMI, "Bcbs-cpmi-iosco finalise analysis of margining practices during the march 2020 market turmoil," https://www.bis.org/press/p220929.htm, September 2022.

Acknowledgments

I am profoundly grateful to my supervisor, Prof. Massimiliano Caporin, for the extreme patience during this work and for his expert guidance and sharp insights during the little conversations that we had.

I'm really proud to master from the University of Padova. Even if the last few years weren't so profound and my mind was already over it, the first years profoundly changed me and had a decisive impact in who I am today. I'll always be happy and grateful to have studied in UNIPD.

To my friends, who are the essence of life, that were part of this big chapter of my life. I'm grateful to have friends with so many different backgrounds and lifestyles, having met them in various different phases of my youth.

To my family, for whom this thesis and degree hold greater meaning than they do for me, but without whom everything would be empty and pointless.

To all the people who hold and will always hold a place in my heart: those from the past, from the present and from the future.

To me and the ability to close chapters. To the future me and his never ending ambition.