Predicting Extreme Price Movements in Technology Stocks: A Study of High-Frequency Trading Dynamic

Auteur : Flas, Martin
Promoteur(s) : Hambuckers, Julien
Faculté : HEC-Ecole de gestion de l'Université de Liège
Diplôme : Master en ingénieur de gestion, à finalité spécialisée en Financial Engineering
Année académique : 2024-2025
URI/URL : http://hdl.handle.net/2268.2/22417

Predicting Extreme Price Movements in Technology Stocks:

A Study of High-Frequency Trading Dynamic

Jury:

Supervisor:
Julien HAMBUCKERS
Reader:
Philippe HÜBNER

Master thesis by

Martin Flas
For a Master's degree in Business Engineering, Financial Engineering
Academic year 2024-2025

Contents

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, **Julien Hambuckers**, for his invaluable guidance, support, and expertise throughout the development of this master thesis. His constructive feedback and insightful suggestions have been crucial in helping me navigate this challenging yet rewarding academic journey. I am sincerely grateful for his dedication and encouragement, which have greatly contributed to the completion of this work.

I would also like to extend my heartfelt thanks to my reader, **Philippe Hübner**, for their thorough evaluation of my thesis and their thoughtful comments, which have helped refine and improve the quality of my work.

Both **Julien Hambuckers** and **Philippe Hübner** had a profound influence on my academic path. Through the course, *Applied Financial Instruments*, during the third year of my bachelor's studies, you inspired me to pursue a master's degree in Financial Engineering. Furthermore, your guidance and expertise throughout this master's program, including this final project, have provided me with the strong foundational knowledge needed for a future professional career in the field.

Finally, I would like to thank my family, friends, and colleagues for their unwavering support, encouragement, and understanding during the completion of this thesis. Their belief in my abilities and their patience have been a constant source of motivation.

This thesis represents the culmination of an important chapter in my academic life, and it would not have been possible without the invaluable contributions and support of all the individuals mentioned above. To them, I extend my sincerest appreciation.

Chapter 1

Introduction

Financial markets operate as intricate systems where price fluctuations are influenced by a myriad of factors, ranging from macroeconomic indicators to trading dynamics. Among these, the advent of High-Frequency Trading (HFT) has introduced unprecedented complexities and challenges. Leveraging cutting-edge technology, HFT has transformed market operations by executing trades at ultra-high speeds. While this evolution enhances liquidity and efficiency, it also raises significant concerns regarding market stability, particularly during periods of extreme price movements (EPMs). These sudden and significant changes in asset prices, although rare, pose substantial risks to market participants and financial systems alike.

The motivation for this thesis stems from the critical need to understand, predict, and potentially mitigate the risks associated with EPMs under normal market conditions. Unlike periods of financial crises or flash crashes, EPMs occurring during stable market phases often go unnoticed yet can have far-reaching implications. By exploring machine learning models, this study aims to uncover predictive patterns that can serve as early warning signals for such events.

This thesis contributes to the existing body of knowledge by employing a diverse set of analytical techniques, ranging from traditional logistic regression to advanced deep learning models like Long Short-Term Memory (LSTM) networks. Each methodology is rigorously evaluated to determine its effectiveness in forecasting EPMs. The study also integrates novel covariates derived from high-frequency trading data, enhancing the granularity and relevance of the predictions.

Beyond methodological innovation, this work addresses the inherent difficulties of EPM prediction, such as class imbalance and the dynamic nature of financial markets. It emphasizes the importance of feature engineering and model interpretability, particularly in the context of financial decision-making, where transparency is paramount.

The structure of this thesis is as follows: Chapter 2 presents a comprehensive review of the literature, focusing on HFT, EPM identification, and their interplay. Chapter 3 details the data sources, preprocessing steps, and new covariates introduced for analysis. Chapter 4 outlines the models employed, providing theoretical foundations and implementation specifics. Chapter 5 discusses empirical results, evaluating model performance against established metrics. Finally, Chapter 6 synthesizes findings, discusses limitations, and proposes future research directions.

In navigating the complexities of predicting EPMs, this thesis aspires to bridge gaps in both theoretical understanding and practical application, offering valuable insights for market practitioners, regulators, and academics. By advancing predictive modeling in this domain, it contributes to the broader goal of enhancing market resilience and stability.

Chapter 2

Litterature review

2.1 High-Frequency Trading

High-frequency trading (HFT), a sophisticated subset of algorithmic trading, relies on advanced technology to execute large volumes of transactions at unprecedented speeds. While HFT has only recently become a focal point in financial discourse, its foundational technologies and practices have been developing for several decades. This section explores the historical evolution of computerized trading, tracing its progression through pivotal milestones that have shaped the financial markets of today.

The computerization of financial markets began in the 1970s with the New York Stock Exchange's (NYSE) Designated Order Turnaround (DOT) system, introduced in 1976, and its successor, Super-DOT, in 1984. These systems facilitated the electronic routing of orders to the appropriate trading post, where specialists could execute trades via an electronic workstation known as the "display book", as noted by Markham and Harty (2007). These innovations marked the decline of manual, floor-based trading, where participants had relied on slower, more expensive communication methods, such as ticker tape and telephonic exchange, as highlighted by Brummer (2008).

The establishment of NASDAQ in 1971 as the world's first electronic stock market further accelerated this transformation. Unlike the NYSE's specialist-driven auction system, Markham and Harty (2007) found that NASDAQ employed competing market makers through an electronic quotation system, capturing 42% of total U.S. market share volume by 1992. The rise of program trading in the 1980s—defined by the NYSE as orders involving 15 or more stocks valued at over $1 million—further exemplified the role of computers in executing complex strategies. Techniques like stock index arbitrage leveraged these systems to capitalize on pricing discrepancies between equity and futures markets, though they faced criticism for contributing to market instability, such as during the 1987 crash, as detailed by Furbush (1989).

The late 1990s saw the emergence of electronic communication networks (ECNs), following the U.S. Securities and Exchange Commission's (SEC) authorization under Regulation Alternative Trading Systems (Reg. ATS) in 1998. ECNs, such as Island and Instinet, offered significant advantages over traditional exchanges, including cost reduction, operational efficiency, and lower error rates, as reported by Liebenberg (2002). The widespread adoption of ECNs fueled the proliferation of algorithmic trading strategies, which laid the groundwork for the eventual rise of HFT.

The SEC's 2001 mandate for decimalization further reshaped the market landscape by reducing the minimum tick size to $0.01 per share. This structural change increased market liquidity and spurred the adoption of algorithmic execution methods by institutional traders seeking optimal pricing strategies. Subsequently, Regulation National Market System (Reg. NMS), introduced in 2005, established a unified framework for modernizing equity markets, including mandates for best-price execution under the Trade-Through Rule. These measures facilitated the rise of HFT firms, which capitalized on advance-

ments in market access and order execution speed, as described by Liebenberg (2002).

The convergence of technological innovation, regulatory reform, and market evolution has positioned high-frequency trading at the forefront of modern financial markets. The historical developments outlined here provide critical context for examining the operational mechanics and implications of HFT, particularly in the realm of technology stocks.

2.1.1 Defining High-Frequency Trading

High-frequency trading (HFT) remains a term lacking a universally accepted definition, as noted by key regulatory authorities such as the U.S. Securities and Exchange Commission (SEC, 2010, 2014) and the Netherlands Authority for the Financial Markets (NAFM, 2010). This definitional ambiguity extends beyond semantics, affecting research, policy-making, and market oversight. Generally, HFT is characterized by the use of advanced computer algorithms and infrastructure to execute large volumes of trades at ultra-fast speeds. However, while HFT is frequently associated with automated trading, electronic trading, and algorithmic trading (AT), these terms are not interchangeable and only partially capture the essence of HFT.

The SEC (2010) describes HFT as the activity of professional traders who utilize proprietary algorithms to engage in strategies that generate a high volume of daily trades. Similarly, the NAFM (2010) views HFT as a form of automated trading that employs advanced mathematical models for implementing short-term trading strategies, although it refrains from classifying HFT as a distinct trading strategy itself. In line with this perspective, the Australian Securities and Investments Commission, ASIC (2010) emphasizes the low-latency technology infrastructure underpinning HFT, distinguishing it as a specialized form of AT rather than a standalone method of trading.

The lack of a precise definition creates significant challenges. As the NAFM highlights, it complicates efforts to estimate HFT's market share and influence. Furthermore, confusion between HFT and AT has led to unwarranted attributions, such as blaming HFT for events like the 2010 flash crash, as explained by Moosa and Ramiah (2015). Without a clear understanding of HFT, regulatory interventions risk being inefficient or even detrimental, potentially imposing significant costs across various market participants, from retail investors to large institutional funds.

Efforts to delineate HFT's characteristics reveal commonalities among definitions provided by different authorities. These characteristics typically include the use of high-speed, sophisticated computer programs (SEC, CESR, MiFID II), extremely short holding periods (SEC, NAFM, ASIC, CESR), and high order amendment or cancellation rates (SEC, NAFM, ASIC, MiFID II). HFT practitioners also rely on low-latency infrastructure, such as colocation services, proximity hosting, and direct market data feeds, to minimize delays in accessing market information and executing trades (SEC, NAFM, ASIC, CESR, MiFID II). Additionally, HFT often involves strategies aimed at achieving small profit margins across large trading volumes and closing the trading day with minimal open positions (NAFM, ASIC, CESR).

Regulatory frameworks such as MiFID II have attempted to formalize the concept of HFT. For instance, MiFID II (2014) defines HFT techniques as a subset of algorithmic trading that relies on infrastructure designed to minimize latency. However, even with such efforts, trading platforms face difficulties distinguishing HFT from other forms of AT, as quantifying the market shares of various trading strategies remains technologically challenging, as detailed by NAFM (2010).

The defining features of HFT extend beyond speed and volume. The London Stock Exchange Group, LSE (2010) highlights that HFT strategies are diverse, encompassing electronic market-making, arbitrage, and liquidity provision. These strategies are underpinned by state-of-the-art infrastructure that processes vast amounts of market data in milliseconds, enabling participants to execute trades with unparalleled efficiency. According to Harris (2013) and Manahov (2016), unlike human traders, algorithmic

systems operate without emotional biases, allowing them to adhere strictly to preprogrammed strategies.

Despite the lack of a unified definition, a consensus exists on certain technical and operational characteristics that distinguish HFT from traditional and algorithmic trading. These include:

- Ultra-fast execution: Trades are executed in milliseconds or even microseconds.
- High order volumes with low profit margins: Strategies focus on generating numerous small gains across a large number of trades.
- Short holding periods: Positions are typically held for seconds or minutes.
- End-of-day flat positions: Practitioners close out positions by the end of the trading day.
- Sophisticated infrastructure: This includes colocation services, low-latency data feeds, and high-frequency algorithms.

The ongoing evolution of financial technology continues to shape the nature and perception of HFT. As markets become increasingly electronic, the role of HFT is likely to grow, necessitating continued scrutiny and refinement of its definition to ensure effective regulation and understanding.

2.1.2 Historical Growth and Market Penetration of High-Frequency Trading

By the early 2000s, high-frequency trading (HFT) had emerged as a dominant force in financial markets, fundamentally altering the dynamics of trading across various asset classes. Its rapid rise was driven by advancements in computational technology, regulatory changes, and market demands for increased efficiency. Empirical evidence highlights HFT's substantial share in trading activity, particularly in equity and futures markets, underscoring its transformative impact.

In equity markets, HFT quickly became a significant contributor to trading volume. Brogaard et al. (2010) in 2008 and 2009 estimated that HFTs participate in 77% of the dollar-volume traded in U.S. equities. Similarly, in the work of Popper (2012), HFT activity accounted for approximately 51% of total trading volume in U.S. equity markets by 2012 and stabilized around 60.5 % by 2016, as mentioned by Bazzana and Collini (2020). As of the December 2020, data by the European Securities and Markets Authority suggests that HFT represents roughly 50% of trading volume in US equity markets. European markets also witnessed substantial HFT participation, with an estimated 40% share of total trading volume in 2009, as explained by Haldane (2010) and between 24% and 43% percent of trading volume in 2020, as highlighted by Breckenfelder (2020). According to Hagströmer and Nordén (2013), Swedish large-cap stocks experienced HFT participation ranging between 25% and 50% during the 2011–2012 period. In Canada, HFT represented 46% of total equity market volume in 2010–2011, as mentioned by Boehmer et al. (2018).

The prominence of HFT extends beyond equities into futures markets, where its influence is even more pronounced. Miller and Shorter (2016) reported that HFT accounted for approximately 80% of the trading volume in U.S. foreign exchange futures and 66% in both interest rate and Treasury 10-year futures markets. These findings highlight the versatility of HFT strategies in adapting to the unique liquidity and volatility profiles of futures contracts.

Globally, the increasing reliance on HFT has presented both opportunities and challenges. On the positive side, HFT enhances market efficiency by narrowing bid-ask spreads, improving liquidity, and accelerating price discovery. These benefits are particularly valuable in markets with high transaction volumes and significant information flow. However, the concentration of trading activity among a small number of HFT firms raises concerns about systemic risks, market fairness, and the potential for destabilization during periods of heightened volatility.

The evolution of HFT from a niche technological advancement to a cornerstone of modern financial markets reflects its profound impact on trading efficiency and market structure. While HFT has undoubtedly improved the operational dynamics of trading, its widespread adoption necessitates a careful balance between leveraging its benefits and mitigating associated risks. As HFT continues to expand across geographies and asset classes, ongoing research and regulatory adaptation are essential to ensuring its alignment with broader market objectives.

2.2 Extreme Price Movements

Extreme Price Movements (EPMs) refer to sudden and pronounced changes in asset prices that exceed the bounds of typical market fluctuations. These events, while rare, carry significant implications due to their potential to destabilize markets. They are frequently linked to market corrections, economic shocks, or crises, occurring both during routine market conditions and extraordinary periods of heightened volatility. Early discussions of EPMs in financial literature, as noted by Longin (2000), framed them as critical moments ranging from ordinary corrections to extraordinary events such as stock market crashes, bond market collapses, or foreign exchange crises. Earlier foundational works by Longin (1999) emphasized their occurrence in the extreme tails of return distributions, challenging traditional risk models that often inadequately capture their predictive nuances. These insights underscore the unique nature of EPMs, which demand specific methodologies for their identification and analysis.

2.2.1 Identification of EPMs

To identify EPMs, Brogaard et al. (2018) developed three distinct methodologies, each aimed at isolating significant deviations in price movements. These approaches are designed to address specific challenges in financial market data, focusing on midquote returns from the National Best Bid and Offer (NBBO) rather than trade prices to minimize biases from bid-ask bounces. While the underlying data preparation involves extensive filtering and adjustments, such as removing anomalies and interpolating missing data, the specifics of these steps will be detailed in a later section.

The first approach relies on identifying intervals with absolute midquote returns that fall within the 99.9th percentile for each stock. This method is straightforward and effective for capturing extreme values relative to the stock's typical trading activity. However, its simplicity introduces certain limitations, such as an implicit assumption that all stocks have equal susceptibility to extreme price movements. As a result, it may over-represent periods of low volatility while under-representing high-volatility events. This method also lacks the ability to distinguish between stock-specific and broader market-driven price movements.

The second approach refines the identification process by incorporating a residual-based methodology. A short-term market model is used to estimate expected returns for each stock based on historical data and market-wide influences, such as the performance of the S&P 500 ETF. The residuals from this model, representing unexplained price changes, are then used to identify intervals that belong to the 99.9th percentile. By isolating stock-specific price changes from predictable patterns, this method improves accuracy and addresses some of the limitations of the absolute threshold approach. However, its reliance on prior-day estimation introduces a lag in adapting to rapidly changing market dynamics, which may be a drawback in highly volatile conditions.

The third approach employs the volatility-adjusted methodology developed by Lee and Mykland (2012). This technique adjusts thresholds for contemporaneous local volatility, ensuring that intervals are classified as extreme based on their deviation from the prevailing market conditions. By normalizing price changes relative to current volatility, this method is particularly effective in mitigating the oversampling of high-volatility periods observed in the other approaches. It provides a dynamic and context-aware framework for identifying EPMs, although its computational intensity and sensitivity to precise volatility estimates can be challenging in practice.

Each of these methodologies offers unique advantages and limitations. The absolute returns-based threshold provides a simple and intuitive baseline, while the residual-based approach enhances precision by accounting for market-wide and stock-specific factors. The volatility-adjusted method stands out for its ability to incorporate real-time market conditions, making it the most robust in volatile environments. Despite these differences, the results obtained by Brogaard et al. (2018) demonstrate consistency across the three methods, with the residual-based approach serving as the primary framework for analysis. This comprehensive approach to EPM identification ensures a balanced treatment of diverse market conditions and enables a nuanced understanding of extreme price dynamics. Further exploration of data cleaning and preparation will follow in subsequent sections to fully contextualize these methodologies.

2.2.2 EPM and Flash crash

Large price movements, are triggered by sudden and significant disruptions in market conditions. The literature identifies two primary types of events that can instigate these movements: information arrival and trade imbalances.

First, information arrival refers to events where new and impactful information enters the market, such as earnings reports, macroeconomic announcements, or geopolitical developments. These events lead to rapid adjustments in prices as market participants assimilate the new information, resulting in permanent changes in price levels that reflect updated fundamental values.

Second, trade imbalances occur when disproportionate buy or sell orders are placed, often by impatient traders or institutions executing large trades. These imbalances push prices away from their fundamental levels temporarily, creating transitory price movements that are typically followed by reversals as the market corrects itself.

2.2.3 The Flash Crash of May 6, 2010: A Comprehensive Analysis

Among the many instances of transitory EPMs in modern financial history, the 2010 Flash Crash stands out for its scale, impact, and the insights it offers into high-frequency trading (HFT) and market dynamics.

On May 6, 2010, U.S. financial markets experienced a dramatic short-term crash, later named the "Flash Crash." This event has since been extensively analyzed, becoming a benchmark case for understanding the vulnerabilities and behaviors of modern financial systems under stress. Both the Commodity Futures Trading Commission (CFTC) and the Securities and Exchange Commission (SEC) conducted in-depth investigations, culminating in joint reports that highlighted the event's underlying mechanisms and regulatory implications.

The Flash Crash of May 6, 2010, is widely regarded as one of the most significant transitory extreme price movements (EPMs) in financial history. On this day, the U.S. financial markets experienced a sudden and severe downturn, with the Dow Jones Industrial Average plunging nearly 1,000 points within minutes. The trading day began under the shadow of global uncertainty, largely driven by the European debt crisis, which heightened market volatility. By mid-afternoon, the S&P 500 volatility index had surged by 22.5% compared to its opening level, signaling increased investor anxiety. The Dow Jones Industrial Average reflected this sentiment, falling by approximately 2.5% by 2:30 p.m. Central Standard Time. In response to this environment, Waddell& Reed, a prominent fundamental trader, initiated a large-scale sell algorithm designed to offload 75,000 E-mini (S&P 500 futures) contracts to hedge its equity positions, as reported by CFTC & SEC (2010a) and discussed by Lash and Spicer (2010). While the initial market response absorbed this sell order through high-frequency traders and other intermediaries, the subsequent interaction of market participants caused a cascade of effects across trading platforms.

High-frequency traders initially accumulated significant long positions in E-mini contracts, but the rapid

escalation of selling pressure led them to reduce inventories aggressively, contributing to a cycle of intense buying and selling activity. Over 33% of the day's total E-mini trading volume—about 140,000 contracts—was attributed to high-frequency traders, as demonstrated by CFTC & SEC (2010b). This surge in activity amplified market volatility, creating an environment where long-term traders retreated, further exacerbating liquidity challenges. The result was a "hot potato" effect, where high-frequency traders exchanged positions repeatedly among themselves, driving the E-mini price down by 3% in just four minutes. This rapid decline spilled over into the equities market through cross-market arbitrage, leading to a parallel 3% drop in the S&P 500 SPDR (SPY).

The situation worsened when an additional 1.7% drop in the E-mini occurred within a mere 15 seconds, marking an intraday low of 1,056 points. This triggered the Chicago Mercantile Exchange's Stop Logic Functionality, halting E-mini trading for five seconds at 2:45:28 p.m. When trading resumed at 2:45:33 p.m., longer-term and opportunistic traders began re-entering the market, stabilizing and eventually recovering E-mini prices. This recovery extended to SPY prices, but broader market volatility persisted.

Between 2:40 p.m. and 3:00 p.m., approximately 2 billion shares, valued at over $56 billion, were traded. Despite the stabilization of most securities within 10% of their pre-crash values, certain stocks experienced extreme anomalies. For instance, Accenture plc (ACN) saw its price plummet from $30 to down as low as a penny in a matter of seconds before recovering just as quickly. These aberrations were linked to orders executed against stub quotes—artificially placed quotes far removed from prevailing market prices, intended solely to fulfill quoting requirements, but not designed for execution, as pointed out by CFTC & SEC (2010a).

Overall, the flash crash resulted in over 20,000 trades involving 5.5 million shares across 300 securities and exchange-traded funds at prices at least 60% away from their 2:40 p.m. levels. By 3:00 p.m., most securities had reverted to their rational valuations. In the aftermath, regulatory bodies, including the SEC and FINRA, invoked "clearly erroneous" trade rules, classifying such trades as "broken" and excluding them from official records. Notably, the flash crash underscored the nuanced role of high-frequency traders. Kirilenko et al. (2017) explained that while these traders did not directly initiate the event, their automated systems, operating under extraordinary selling pressure, contributed to the rapid escalation of volatility by aggressively competing for liquidity. This behavior, marked by repeated buying and selling, created a hot-potato effect that amplified price declines and destabilized the market.

Others studies have offered a balanced perspective on high-frequency trading. Brogaard et al. (2018) argued that high-frequency traders often provide liquidity during periods of extreme price movements. However, when simultaneous disruptions occur, such as in the flash crash, they may shift roles and become liquidity demanders. This dual behavior has drawn critical attention from regulators, particularly the CFTC & SEC (2010b), for its potential to exacerbate short-term market volatility. The flash crash of May 6, 2010, remains a pivotal event in the discourse surrounding high-frequency trading and its implications for market stability.

Cespa and Foucault (2014) argue that a reversal of a market equilibrium characterized by low illiquidity and high price informativeness can precipitate a liquidity crash akin to the Flash Crash. This theoretical insight highlights the fragility of market conditions under certain stress scenarios. Easley et al. (2012b) further explore this dynamic by suggesting that higher-order flow toxicity—where toxic order flows dominate trading activity—can lead to short-term periods of heightened volatility, such as the Flash Crash. Flow toxicity, which measures the imbalance between informed and uninformed trading, has the potential to force liquidity providers, including high-frequency traders (HFTs), to withdraw from the market, amplifying price instability. Easley et al. (2011) also reference the CFTC-SEC report on the Flash Crash, underscoring the role of HFTs, which liquidated positions and retreated from the market when flow toxicity intensified. This withdrawal, they note, exacerbated the volatility observed during the event.

Hautsch et al. (2017) and Kang et al. (2020) investigate how HFTs respond to extreme market events, finding that they tend to align their positions with those of other market participants. Kang et al. (2020) provide further nuance by distinguishing between domestic and foreign HFTs, analyzing how their trading activities contribute differently to order flow toxicity. They note that, while these groups exhibit variations in trading behavior under normal conditions, both generate toxic orders during periods of market stress. This finding reinforces the argument that HFT activity can exacerbate systemic risks in highly volatile environments.

Golub et al. (2012) focus on the phenomenon of Mini Flash Crashes—smaller-scale but disruptive events that occurred in the U.S. markets between 2006 and 2011. Their analysis attributes these events to market fragmentation and the existing regulatory framework, while also identifying HFTs as a key factor in the adverse impact on liquidity during these crashes. Similarly, Johnson et al. (2013) link Mini Flash Crashes to the interaction of various automated trading algorithms, illustrating how their competition can destabilize markets.

Jacob Leal et al. (2016) take a modeling approach, using an agent-based framework to show that HFTs play a critical role in generating flash crashes by widening bid-ask spreads and synchronizing on the sell side of the limit order book. More recently, Leal and Napoletano (2019) construct a theoretical model demonstrating how HFTs can trigger flash crashes. Their findings reveal a complex trade-off: policies targeting HFT behavior may reduce the frequency of flash crashes but could also slow the recovery process following such events.

Together, these studies emphasize the intricate relationship between market microstructure, algorithmic trading, and systemic risk. The role of HFTs during flash crashes—whether as liquidity providers or liquidity demanders—remains a focal point for research. While HFTs are often credited with enhancing market efficiency during stable periods, their behavior during extreme market events underscores the need for further investigation into how market design and regulatory frameworks can mitigate the risks of flash crashes in increasingly fragmented and automated trading environments.

2.3 Liquidity Indicators

The concept of liquidity is pivotal in financial markets, embodying the ability to execute trades swiftly, at low cost, and with minimal impact on market prices. It is essential for the proper functioning of markets, ensuring efficiency, stability, and attractiveness to participants. As Demsetz (1968) originally noted, a liquid market minimizes the costs of a round trip for any quantity of security, which is crucial for fostering participation and sustaining market vibrancy. Liquidity, or its absence, has wide-reaching implications for traders, investors, exchanges, regulators, and financial intermediaries alike, influencing market design, asset pricing, and portfolio management strategies.

Measuring liquidity has proven to be a complex and evolving endeavor, particularly in high-frequency trading (HFT) environments. Early studies focused predominantly on quote-driven markets and utilized simple metrics such as the bid–ask spread and trading volume. Over time, the scope of liquidity measurement expanded to encompass order-driven markets, multiple asset classes, and high-frequency dynamics. Despite extensive literature, no universal liquidity measure exists, and researchers continually debate the most effective methodologies, as highlighted by Aitken and Comerton-Forde (2003) and Sarr and Lybek (2002).

Liquidity can be classified into five key dimensions, as outlined by Kyle (1985): tightness, depth, resilience, breadth, and immediacy. Tightness refers to the cost of turning over a position in a short time, often captured by the bid–ask spread. Depth measures the market's capacity to absorb large orders without significant price changes, commonly linked to the limit order book (LOB). Resilience reflects the speed at which prices recover from shocks. Breadth denotes the overall traded volume, and immediacy addresses the time required to execute trades of varying sizes at acceptable costs. High-frequency trad-

ing, with its emphasis on speed and precision, amplifies the importance of measuring these dimensions at granular levels.

A wide array of liquidity measures has been developed to capture these dimensions, categorized broadly by data frequency, dimensional focus, and computational complexity. Early low-frequency measures, such as the Roll (1984) measure, trading frequency, and turnover ratios, provided foundational insights but fell short in capturing the intraday variability and real-time shocks critical in HFT contexts. The advent of high-frequency data has spurred the creation of new metrics that leverage intraday transaction records, such as tick-by-tick trade and quote data. Among the most widely used high-frequency measures are those based on the bid–ask spread, order submissions, and LOB dynamics.

2.3.1 Bid–ask spread

Spread-based measures are central to understanding liquidity in financial markets, focusing on the bid–ask spread as a key indicator of transaction costs. The bid–ask spread represents the difference between the highest price a buyer is willing to pay and the lowest price a seller is willing to accept. This metric has been extensively analyzed in market microstructure research, as it reflects costs stemming from order processing, inventory holding, and adverse selection arising from information asymmetry among market participants.

In modern order-driven markets, liquidity is predominantly supplied through limit orders rather than designated market makers. This structure implies that the bid–ask spread is shaped directly by order submissions, cancellations, and resubmissions. Researchers like Foucault et al. (2005) have highlighted the relationship between the spread's width and the behavior of patient limit order traders. The ability to swiftly adjust or cancel limit orders in response to changing market conditions, facilitated by electronic trading systems, adds dynamism to the spread. Harris (2002) identified critical factors influencing the spread in such markets: the degree of information asymmetry, the agility of traders in canceling orders, and the security's price volatility.

High-frequency analysis of spread-based measures provides granular insights into market liquidity. Metrics like Percent Quoted Spread (PQS) evaluate liquidity at specific points, offering an immediate view of transaction costs. However, PQS assumes a static perspective and does not account for complexities such as large trades walking through the book or hidden orders. To address these limitations, the Percent Effective Spread (PES) and Percent Price Impact (PPI) have been developed. PES incorporates the trade price, providing a more dynamic reflection of liquidity conditions and accommodating for execution costs outside the quoted spread. Meanwhile, PPI measures price adjustments following trades, capturing the impact of information on liquidity providers' pricing.

The use of these spread-based measures, while advantageous for their precision and ease of computation at high frequencies, is not without challenges. For instance, accurately assessing liquidity over extended periods requires comprehensive high-frequency data, which may not be available for historical analysis. Additionally, for securities with high tick sizes, spreads often remain fixed at one tick, leading to potential biases in liquidity estimation. Researchers have proposed proxies using lower-frequency data, such as the Roll measure, to address these issues. However, these proxies frequently underperform in capturing the nuanced dynamics of intraday spreads, particularly in high-frequency contexts.

Spread-based measures have also proven valuable in analyzing liquidity responses to specific market events. Studies by Lee et al. (1993) and Ranaldo (2005) have demonstrated how bid–ask spreads react sharply to earnings announcements and real-time news, revealing short-term liquidity shifts. These observations emphasize the importance of high-frequency measures in capturing the immediate and transient effects of market information.

2.3.2 Limit order book

In order-driven markets, the liquidity available is largely dictated by the flow of orders and the configuration of the limit order book (LOB). When orders submitted by traders do not immediately match opposing orders, they are either stored in the LOB based on price and time priority or canceled. The LOB serves as a dynamic repository that reflects all dimensions of liquidity, including transaction costs, immediacy, spread, quantity, and price impact. As a result, measures derived from order flow and the LOB have become fundamental tools for assessing liquidity, particularly in high-frequency trading environments where rapid fluctuations in these dimensions occur.

Among the most recognized LOB-based measures is Average Depth (AD), which quantifies the average volume available for trading at the best bid and ask prices. This measure captures the depth dimension of liquidity, offering an immediate snapshot of the market's ability to handle transactions without significant price movements. AD is a forward-looking metric that shows liquidity at a specific point in time, making it especially useful for evaluating market conditions in real-time trading scenarios.

Another prominent measure is Weighted Order Value (WOV), which combines the weighted values of bid and ask orders across various price levels within the LOB. WOV incorporates both the size and the probability of execution of orders, providing a multidimensional view of market liquidity. By integrating execution probabilities, WOV not only reflects the quantities available but also accounts for the likelihood of these orders being fulfilled. This makes it a robust indicator of the structural balance between liquidity supply and demand.

The dynamic nature of the LOB also allows for more specialized metrics that incorporate time-sensitive dimensions of liquidity. For instance, duration-based measures such as Weighted Durations (WD) and Autoregressive Conditional Durations (ACD) assess the time required to trade a fixed volume of shares or capital. These measures capture the immediacy of the market, highlighting how quickly liquidity can be accessed under varying conditions. Similarly, metrics like the Cost of a Round Trip (CRT) evaluate the aggregated cost of executing a simultaneous buy and sell order of a specific size, thereby encompassing dimensions such as tightness, depth, and resiliency.

LOB-based measures like AD, WOV, and CRT are invaluable in high-frequency trading contexts for their ability to dissect the granular aspects of market liquidity. They rely on detailed order and trade data, enabling traders and market participants to develop nuanced strategies that respond to real-time changes in the market. These measures also provide actionable insights for regulators and exchanges aiming to ensure fair and efficient markets. However, the computational demands and data intensity required to calculate such metrics pose significant challenges, particularly in historical analyses or markets with limited data access.

2.3.3 Volume-Synchronized Probability of Informed Trading

The Volume-Synchronized Probability of Informed Trading (VPIN) emerged as a sophisticated evolution of the Probability of Information-based Trading (PIN) model, addressing the dynamic and high-frequency nature of modern financial markets. Introduced by Easley et al. (2012b), VPIN builds upon the foundational principles of PIN, adapting them to capture the complexities of intraday trading activity and high-frequency trading (HFT). While PIN revolutionized the study of information asymmetry by quantifying the likelihood of informed trading, its static structure and reliance on low-frequency data limited its applicability in fast-paced trading environments. VPIN bridges this gap by incorporating time-varying measures, making it a crucial tool for analyzing market microstructure in the HFT era.

The PIN model, developed by Easley et al. (1996)), introduced a statistical approach to measure information asymmetry between informed and uninformed traders. It was grounded in the sequential trade model proposed by Easley and O'hara (1992), which linked security price formation to the arrival of private information. PIN estimates the probability of informed trading based on market microstructure

parameters, derived through a maximum likelihood function that reflects the interaction between information events and trade flow dynamics.

PIN has been extensively utilized in studies examining information risk and market behavior. For instance, Easley et al. (1997) explored the relationship between trade size and PIN, demonstrating that uninformed traders often mimic others' actions, complicating the detection of informed trades. Easley et al. (2002) linked PIN to asset pricing, documenting that stocks with higher PIN values exhibit higher expected returns due to increased information risk. Subsequent research, as validated by Vega (2006), confirmed PIN's correlation with market characteristics like spreads and volatility, further solidifying its role in understanding the effects of information asymmetry.

Despite its contributions, the PIN model has faced several criticisms. One challenge lies in its reliance on a static framework, which does not account for the time-varying nature of trading activity. Additionally, according to Boehmer et al. (2007), empirical studies have identified biases in PIN estimation caused by trade misclassification and boundary issues during parameter optimization, as specified by William Lin and Ke (2011). Moreover, Hughes et al. (2007) questioned PIN's relevance for asset pricing, suggesting that information risk does not significantly influence returns in large economic settings.

To address the limitations of the PIN model in high-frequency trading (HFT) environments, Easley et al. (2011) introduced the VPIN metric. This innovative measure extended the static framework of PIN into a dynamic context, leveraging intraday transaction data and adapting to the rapid pace of modern financial markets. VPIN redefined the measurement of informed trading by synchronizing trade activity with volume rather than time, offering a more responsive and precise approach to understanding market microstructure.

The development of VPIN was influenced by earlier works that highlighted the time-varying nature of informed trading. Lei and Wu (2005) demonstrated that time-varying probabilities better captured the dynamics of bid-ask spreads and market information asymmetry. Building on this, Easley et al. (2008) incorporated a dynamic GARCH framework into the PIN model, introducing time-varying arrival rates for informed and uninformed trades. This groundwork laid the foundation for the rolling-window structure of VPIN, which tracks informed trading probabilities over discrete volume intervals rather than fixed time periods.

VPIN gained attention for its ability to signal market stress and predict volatility, particularly in high-frequency contexts. One of its most notable applications was its performance during the Flash Crash of May 6, 2010. Easley et al. (2011) demonstrated that VPIN achieved its maximum level hours before the event, providing a clear warning of elevated market risk. This showcased its potential as an early warning indicator for market instability and adverse selection risk.

The core innovation of VPIN lies in its "order flow toxicity" metric, introduced by Easley et al. (2012b). Order flow toxicity represents the risk that market makers face when they unknowingly provide liquidity at a loss due to adverse selection by informed traders. VPIN quantifies toxicity through a function of volume imbalances and trade intensity, capturing the likelihood of informed trading during periods of significant market activity. High VPIN values signal heightened toxicity, prompting market makers to adjust their strategies, such as reducing liquidity provision or exiting the market altogether.

Researchers have also explored VPIN's effectiveness in forecasting market volatility. Bethel et al. (2011) confirmed its potential as an early risk warning system, while Abad and Yagüe (2012) validated its robustness in measuring adverse selection risk in the Spanish market. These studies highlighted VPIN's adaptability to various market settings, particularly in high-frequency environments.

However, VPIN is not without its critics. Andersen and Bondarenko (2014)) argued that VPIN lacks predictive power for short-term volatility, noting that its highest values occurred after the Flash Crash

rather than before it. Their critique focused on the trade classification algorithms used in VPIN calculations, which they deemed flawed. Easley et al. (2012a) defended VPIN, asserting that Andersen and Bondarenko's analysis misinterpreted the model and underestimated its utility in identifying periods of order flow toxicity.

2.4 Impact of high frequency trading on the market

2.4.1 Liquidity

Liquidity, a cornerstone of market quality, has been significantly influenced by the advent of high-frequency trading (HFT). As ultra-fast traders, HFT firms play a dual role in shaping liquidity dynamics, offering substantial benefits under stable conditions while raising concerns during periods of market stress. On the positive side, HFT has been credited with narrowing bid-ask spreads and enhancing market depth, two critical measures of liquidity. Studies such as Brogaard et al. (2010) highlight that HFT firms frequently provide the most competitive bid and ask prices, thereby reducing trading costs and facilitating cost-effective transactions. Hasbrouck and Saar (2013) similarly find that HFT contributes to market depth by maintaining a robust supply of limit orders at multiple price levels, ensuring that market participants can execute trades efficiently without causing significant price disruptions. Jarnecic and Snape (2014) further emphasize HFT's role in stabilizing prices by populating order books across various levels, which minimizes market impact and ensures smooth trading operations.

Another key contribution of HFT lies in its ability to lower informational asymmetries, enhancing market efficiency and liquidity. By processing and acting upon new information at unprecedented speeds, HFT enables prices to quickly reflect fundamental and market-related developments. Baldauf and Mollner (2020) underscore that this rapid adjustment reduces adverse selection risks faced by slower traders, fostering confidence in price fairness. In diverse market environments, the liquidity-enhancing effects of HFT are evident across asset classes and geographies. Jain (2017) observe that platforms like Japan's Arrowhead system, designed to reduce latency, have resulted in significant improvements in traditional liquidity measures such as tighter spreads and increased order book depth. Similarly, Menkveld (2013) highlights that HFTs effectively act as quasi-market makers, ensuring consistent order flow even in fragmented markets, thereby substituting traditional liquidity providers and stabilizing trading activities.

However, these benefits are not without limitations. During periods of market stress or heightened volatility, HFTs tend to reduce their liquidity provision, prioritizing risk management over market stability. Kirilenko et al. (2017) illustrate this dynamic during the May 2010 Flash Crash, where HFT liquidity dried up, exacerbating market instability. Furthermore, excessive competition among HFT firms can lead to fleeting liquidity and heightened quoting activity, as observed by Malceniece et al. (2019). These behaviors, while technically maintaining liquidity, may strain market infrastructure and erode confidence among slower market participants. In emerging markets with lower HFT participation, such as those studied by Ersan et al. (2021), HFT's impact can even deteriorate liquidity due to limited competition and inadequate technological infrastructure.

In sum, HFT has undeniably reshaped liquidity in modern financial markets. By narrowing spreads, enhancing depth, and reducing transaction costs, it has fostered greater market efficiency and accessibility under stable conditions. Yet, its limitations during periods of stress and competitive imbalances highlight the importance of a nuanced approach to integrating HFT into market ecosystems. Effective regulation and robust market structures are essential to harnessing HFT's benefits while mitigating its potential risks to liquidity and overall market quality.

2.4.2 Volatility

The relationship between high-frequency trading (HFT) and market volatility has been extensively studied, with mixed findings highlighting its complex nature. Theoretical models have suggested that short-

term traders, including HFTs, may amplify volatility through herding behavior. Froot et al. (1992) propose that short-term traders often rely heavily on a single source of information, even if it lacks a connection to a firm's fundamental values, thereby causing price dislocations and excess volatility. Their model shows that such traders may prioritize transient signals over long-term fundamentals, leading to inefficiencies in price discovery. This is especially evident among technical analysts or chartists who, according Froot et al. (1992), employ forecasting methods that introduce noise and hinder market efficiency. Jarrow and Protter (2012) extend this perspective, theorizing that high-frequency traders' simultaneous, independent reactions to common signals can mimic the effects of a large trader, exerting excessive pressure on prices and increasing volatility.

Empirical evidence further elucidates HFT's impact on market volatility. Breckenfelder (2024) examines the influence of HFT competition on NOMX-St index stocks (OMX S30) between June 2009 and January 2010. The findings reveal that while HFT competition significantly increases intraday volatility, it has little impact on interday volatility, consistent with the tendency of HFTs to close their positions by the end of each trading day. The study also highlights the speculative nature of HFTs, which intensifies when their market presence grows alongside heightened competition, leading to greater liquidity deterioration and short-term volatility. Zhang (2010) corroborates these findings, demonstrating a positive correlation between HFT activity and price volatility, even after controlling for fundamental stock characteristics and other explanatory variables. Notably, the effect is pronounced among the largest 3,000 stocks by market capitalization, for stocks with high institutional holdings, and during periods of heightened market uncertainty, raising concerns about HFT's role in hindering the integration of fundamental values into asset prices.

Shkilko and Sokolov (2020) provide a unique perspective by examining how weather-induced disruptions to HFTs' microwave networks affect volatility. HFTs frequently use microwave towers for faster data transmission compared to fiber-optic cables, with the technology accessible to only a limited number of firms. However, precipitation, such as rain or snow, impairs the speed advantage of these networks, creating a differential in trading speeds on days with poor weather. Their analysis shows that volatility decreases during heavy precipitation, implying that higher levels of HFT activity are directly linked to increased market volatility under normal conditions.

In contrast, certain studies suggest that specific categories of HFT strategies, such as market making, may help dampen volatility. Boehmer et al. (2018) analyze S&P/TSX 60 Index stocks from June 2010 to March 2011 and reveal that high-frequency traders engaged in market making reduce short-term volatility. Their results indicate that competition among HFTs employing market-making strategies mitigates both the permanent and temporary price impacts of trades, leading to lower short-term volatility. These findings align with Hagströmer and Nordén (2013), who demonstrate that market-making HFTs can stabilize intraday volatility through their liquidity-supplying activities. Nonetheless, Boehmer et al. also caution that the highly correlated strategies of HFTs, combined with their dominant market presence, can exacerbate price movements during certain periods, reinforcing concerns about their potential to amplify short-term volatility.

Overall, the evidence underscores the dual and sometimes conflicting role of HFTs in influencing market volatility. While their activities can enhance liquidity provision and reduce volatility under stable conditions, the rapid and collective responses of HFTs to external signals may lead to significant price fluctuations, particularly during periods of market stress or heightened competition. Future research must delve deeper into these dynamics to establish a clearer framework for understanding and managing HFT-induced volatility.

Chapter 3

Data Description and New Covariates

3.1 Data Description

For this master's thesis, my goal was to study EPM prediction under normal market conditions, focusing on companies with the largest market capitalizations. The reason for selecting companies with the largest market cap is to avoid potential liquidity issues, as larger firms tend to have higher trading volumes and more stable order books, ensuring more reliable and representative data. As a result, I have excluded events such as the 2010 flash crash and stressful periods like the COVID-19 crisis. I have chosen to analyze the stocks of Facebook, Nvidia, Google, Microsoft, and Apple over a six-month period, from January 22, 2018, to July 20, 2018.

Ticker	**Name**	**Market Cap (in Billion USD)**
MSFT	Microsoft Corporation	724.07
NVDA	NVIDIA Corporation	145.76
GOOG	Google Inc.	816.92
FB	Facebook, Inc.	552.12
AAPL	Apple Inc.	861.82

Table 3.1: Market Capitalization for Selected Companies (2018-01-22)

The data provider that I have used is LOBSTER (Limit Order Book System - The Efficient Reconstructor). It is an advanced tool specifically designed to reconstruct and analyze the dynamics of the Limit Order Book (LOB) within financial markets, with a primary focus on the NASDAQ stock exchange. Serving as a crucial resource for academic research, LOBSTER facilitates an in-depth examination of high-frequency trading (HFT) and market microstructure, allowing researchers to address the complexities associated with large-scale financial data analysis.

Developed by Huang and Polak (2011) in collaboration with Humboldt-Universität zu Berlin, LOBSTER was created to overcome the challenges researchers encounter when dealing with vast datasets. It extracts data from NASDAQ's TotalView-ITCH message streams, which provide high levels of granularity and reliability. These message streams encompass all visible order activities, including submissions, cancellations, and executions, thereby enabling the precise reconstruction of the Limit Order Book. The system supports LOB reconstruction up to 200 depth levels, providing data from June 2007 to the present. By utilizing highly efficient algorithms, LOBSTER ensures the accurate replication of the LOB state at any given time, adhering to the specifications of NASDAQ's matching engine. Furthermore, the modular design of LOBSTER allows for its adaptation to other order-driven markets with minimal adjustments, extending its utility beyond NASDAQ.

LOBSTER offers comprehensive access to reconstructed LOB data for all NASDAQ-listed stocks, capturing crucial information such as order submissions, cancellations, and executions. This feature is essential for examining market dynamics and liquidity in a detailed and meaningful way. The system's robust design allows for the on-the-fly processing of extensive datasets, optimizing both time and computational resources. Additionally, LOBSTER's user-friendly web-based platform provides global researchers with convenient access to the data, along with tools for analysis and visualization using R and MATLAB. While primarily intended for academic purposes, LOBSTER's features cater to the needs of empirical research, enabling precise studies of trading behavior and market mechanisms.

Since its introduction in 2013, LOBSTER has become a significant tool in market microstructure research, particularly due to its ability to reconstruct the LOB from raw ITCH data. This level of detail far exceeds that offered by traditional snapshot datasets, providing valuable insights into phenomena such as liquidity provision, order flow dynamics, and the impact of high-frequency trading on financial markets. In this context, LOBSTER not only serves as a vital dataset for academic studies but also sets a standard for LOB reconstruction methodologies, fostering deeper exploration into the complexities of order-driven markets.

3.1.1 Message File

The `message file` generated by the LOBSTER system plays a critical role in capturing the events that drive the evolution of the limit order book for a specific security. Unlike the `orderbook file`, which records the static state of the order book at each point in time, the `message file` tracks the dynamic updates triggered by market participants' actions, such as submitting, modifying, or deleting limit orders. Each event in the `message file` corresponds to a change in the order book that needs to be reflected for accurate order book reconstruction and analysis.

The structure of the `message file` is organized as a matrix of size (N $\times$ 6), where N represents the number of recorded events. Each row in this matrix corresponds to an individual event and contains six variables that describe the nature of the event, its timing, and its impact on the order book. These variables are:

1. **Time**: This column records the timestamp of the event, expressed as the number of seconds elapsed since midnight, with decimal precision. Depending on the requested data resolution, this precision can extend to nanoseconds. For example, a timestamp of 34200.014 corresponds to 9:00:00.014 AM (or 34200 seconds after midnight), allowing for high-frequency analysis of trading activity.

2. **Event Type**: The event type indicates the nature of the action that caused an update to the order book. The possible values are:

   - 1: Submission of a new limit order
   - 2: Partial cancellation of a limit order
   - 3: Total deletion of a limit order
   - 4: Execution of a visible limit order
   - 5: Execution of a hidden limit order
   - 6: Cross trade (e.g., an auction trade)
   - 7: Trading halt indicator

3. **Order ID**: This is a reference number assigned to each order by the exchange. The `Order ID` helps to trace the lifecycle of an order, from submission to execution or cancellation. Notably, for hidden limit orders and trading halt events, the `Order ID` is set to 0.

4. **Size**: The `Size` column records the number of shares associated with the order or execution event. This provides insight into the volume of shares being traded, modified, or canceled at each event.

5. **Price**: The `Price` represents the dollar price of the security involved in the event, multiplied by 10,000. This scaling factor ensures that price values are recorded as integers, which is particularly useful for minimizing precision errors. For example, a stock price of $91.14 would be recorded as 911400.

6. **Direction**: This column indicates the direction of the limit order. A value of 1 denotes a buy limit order, and a value of -1 denotes a sell limit order. This is crucial for distinguishing between buy and sell side activity in the order book. Importantly, the execution of a buy (or sell) limit order corresponds to a seller (or buyer) initiated trade.

The following table presents a simplified example of the `message file` structure:

Time	Event Type	Order ID	Size	Price	Direction
34200.014	4	4201405	25	1725100	-1
34200.016	4	4201405	200	1724300	-1

Table 3.2: Example of the message file structure

In this example, at `Time` 34200.014, a visible limit order with order ID 4201405 is executed with a size of 25 shares at a price of $172.51. At `Time` 34200.016, a similar order is executed with a size of 200 shares at a price of $172.43.

It is important to note that the `message file` does not record individual trades in the traditional sense. Instead, it records limit order executions, which may involve multiple smaller executions to fulfill a larger order. For instance, if a market participant wishes to sell 100 shares, there could be several executions that cumulatively satisfy this quantity, such as 90 shares sold to one participant and 10 shares to another. These individual executions are captured separately in the `message file`.

Trading Halts and Special Events

LOBSTER incorporates trading halt messages in the order book. When trading halts, the *message* file logs a type-7 event with placeholders for prices and directions. Upon resumption, another type-7 event indicates the state reset. Table 3.3 shows the typical structure during such events.

Time (sec)	Event Type	Order ID	Size	Price	Direction
34713.685	7	0	0	-1	-1
34719.193	7	0	0	0	-1
34801.333	7	0	0	1	-1

Table 3.3: Example of Trading Halt Messages

3.1.2 Order Book File

The order book file records the prices and associated volumes of all buy and sell orders submitted for a specific security. As shown in Table 3.4, in a limit order book (LOB), buy and sell orders are referred to as *Bid* and *Ask*, respectively. A *Bid* is the highest price a buyer is willing to pay for a specified number of shares, while an *Ask* is the lowest price a seller is willing to accept. Each *level* in a LOB represents a unique price point and displays the quantity of shares available at that price.

Ask Price 1	Ask Size 1	Bid Price 1	Bid Size 1	Ask Price 2	Ask Size 2	Bid Price 2	Bid Size 2
1725100	1475	1723900	100	1726000	200	1723300	100
1725100	1275	1723900	100	1726000	200	1723300	100

Table 3.4: Example of an Order Book File

The columns in an order book file depend on the number of levels selected. For an n-level order book, there are $4n$ columns: n ask prices and volumes, and n bid prices and volumes. Prices are denoted in

fractions of a cent (0.01), while sizes represent the number of shares available.
Whenever a new order with an existing price in the LOB is submitted, its size is added to the already available quantity. Prices and sizes dynamically change based on submissions, cancellations, or executions of orders. For instance, in the first row of Table 3.4, the best ask price (*Ask Price 1*) is 1725100 with an available quantity of 1475 shares (*Ask Size 1*). A trade occurs when a buy order matches a sell order's price. If a buy order matches the best ask price or vice versa, the corresponding quantity is updated.

In LOBSTER's implementation, unoccupied price levels are populated with placeholder values to maintain symmetry. Extra bid prices are assigned -9999999999, extra ask prices 9999999999, and their volumes are set to 0.

3.2 Data Treatment

The initial step in the data treatment process involved selecting only executed trades (Type 4 events) were retained, while non-executed orders, hidden orders, and trading halts were excluded to focus on visible market activity. Additionally, trades occurring in the first five minutes after the market opening and the last five minutes before the market closing were excluded. These periods are characterized by heightened volatility and irregular activity due to order imbalances, overnight news reactions, or end-of-day adjustments, making them less representative of typical trading behavior.

To maintain consistency across trading days, the timeline was adjusted to create a continuous sequence, aligning time variables to eliminate gaps between sessions. The data was then segmented into uniform intervals of ten seconds, enabling the calculation of key covariates and capturing short-term dynamics within each trading session.

Among the key covariates calculated, the **bid-ask spread** measured the difference between the best ask and bid prices, providing an indicator of market liquidity and transaction costs. A narrower spread generally signals higher liquidity, while a wider spread can indicate uncertainty or lower trading activity. The **bid-ask midpoint**, defined as the average of the best bid and ask prices, reflected the perceived fair value of the security based on the current order book.

Log returns were calculated for each 10-second interval:

$$\text{Log Return}_t = \ln(P_t) - \ln(P_{t-1}) \tag{3.1}$$

The study also examined **volatility**, calculated as the standard deviation of log returns over a rolling window, to assess the magnitude of price fluctuations within short intervals. This metric offers crucial insights into market risk and the potential for abrupt price movements. Furthermore, **kurtosis** and **skewness** were computed to capture the shape characteristics of the return distribution. Higher kurtosis values indicated the presence of extreme price movements, while skewness revealed directional tendencies in the data, such as whether extreme gains or losses were more prevalent.

To ensure the quality of the covariates, a correlation analysis was conducted to identify highly correlated variables. Covariates with excessive correlation were removed to prevent redundancy and multicollinearity issues, which could impact the reliability of the models. Efforts were made to retain the same set of covariates for each stock to ensure consistency in the analysis. This approach facilitated meaningful comparisons and helped assess the impact of covariates across different models.

To validate the preprocessing steps and gain a contextual understanding of price dynamics, the log returns for the first day of each month were plotted for each stock. This visual examination highlighted trends and potential anomalies over time, ensuring the dataset's reliability for further analysis(see in the appendix 4).

These preprocessing steps and covariate calculations ensured the dataset was robust and relevant for the objectives of this study. By focusing on visible limit orders, removing irregular time intervals, and

incorporating key metrics such as bid-ask spread, volatility, kurtosis, and skewness, the data treatment process provided a solid foundation for meaningful and accurate results.

3.3 Liquidity indicators

3.3.1 Average Depth (AD)

Average Depth (AD) originates from the broader concept of market depth, a critical dimension of liquidity. Theoretical foundations for this measure can be traced back to Kyle (1985), who introduced the idea of depth as one of the core properties of market liquidity, alongside tightness and resiliency. Market depth measures the capacity of a market to absorb trades without significant price disruption, a concept further explored in the works of Mann and Ramanlal (1996). These studies emphasized the importance of order book dynamics in understanding liquidity, particularly in high-frequency trading environments. AD specifically quantifies the total volume available for trade at the best bid and ask prices, aggregated over a defined period.

The mathematical representation of AD is given as:

$$AD_t = \frac{\sum_{i=1}^{N}(Depth_{bid,i} + Depth_{ask,i})}{N}, \tag{3.2}$$

where $Depth_{bid,i}$ and $Depth_{ask,i}$ denote the volumes at the best bid and ask prices, respectively, at observation i, and N represents the number of observations within the chosen timeframe. By capturing these volumes, AD provides a snapshot of the market's robustness and its ability to handle substantial trading activity.

In the context of high-frequency trading, Average Depth serves as a crucial indicator of liquidity, highlighting the market's capacity to accommodate large or rapid transactions without inducing substantial price movements. A high AD value suggests a resilient market with sufficient liquidity to absorb trades efficiently, while a low AD value indicates vulnerability, where large trades might result in significant price impacts or slippage. These characteristics make AD an invaluable tool for traders seeking to optimize execution strategies, as well as for market analysts monitoring liquidity conditions.

AD is extensively applied in volatility analysis, enabling the identification of periods of diminished market stability during which low depth exacerbates price fluctuations. It is also integral to market impact studies, which investigate the effects of large trades by institutional participants on liquidity and price dynamics. Furthermore, regulators leverage AD as a benchmark to assess market robustness, particularly under stressed trading conditions.

3.3.2 Percent Effective Spread (PES)

The Percent Effective Spread (PES) is a widely used liquidity measure that captures the realized cost of executing a trade relative to the mid-point of the quoted bid-ask prices. It originates from market microstructure theory, where the bid-ask spread is central to understanding transaction costs and price formation. PES builds on foundational work by Huang and Stoll (1997), who developed methods to evaluate trading costs and their relationship to market efficiency.

The mathematical formula for PES is expressed as:

$$PES_t = 2 \times \left| \frac{P_t - P_m}{P_m} \right|, \tag{3.3}$$

where P_t represents the transaction price, and P_m is the mid-point of the bid and ask prices ($P_m = \frac{P_a + P_b}{2}$), with P_a and P_b being the ask and bid prices, respectively. This measure quantifies the deviation of the execution price from the mid-quote, effectively reflecting the cost incurred by traders.

PES is particularly valuable in high-frequency trading contexts as it provides a precise ex-post measure of liquidity. Unlike static measures such as the Percent Quoted Spread (PQS), PES accounts for the actual prices at which trades are executed, incorporating the effects of hidden liquidity and price impact. This makes it more representative of the true cost of trading in dynamic market environments.

The importance of PES lies in its ability to assess execution quality. A lower PES indicates more efficient trading conditions with minimal costs, whereas a higher PES suggests adverse trading conditions, often due to increased volatility or lower market depth. These attributes make PES a critical metric for evaluating trading strategies, understanding market conditions, and monitoring transaction costs over time.

PES is frequently applied in studies investigating the effects of market structure changes, liquidity shocks, and news events. For example, it has been used to measure the impact of macroeconomic announcements on trading costs and to analyze the performance of execution algorithms under varying liquidity conditions. Furthermore, PES is instrumental for regulators and market participants aiming to enhance market transparency and efficiency.

3.3.3 Percent Quoted Spread (PQS)

The Percent Quoted Spread (PQS) is a foundational liquidity measure that captures the instantaneous cost of a round-trip trade based on the quoted bid and ask prices. It originates from the seminal work of Demsetz (1968), who introduced the concept of the bid-ask spread as a fundamental measure of transaction costs in financial markets. PQS is widely used in market microstructure research due to its simplicity and direct representation of market tightness.

The mathematical formula for PQS is given by:

$$PQS_t = \frac{P_a - P_b}{P_m}, \tag{3.4}$$

where P_a and P_b are the ask and bid prices, respectively, and P_m is the mid-point of the bid and ask prices, calculated as $P_m = \frac{P_a+P_b}{2}$. This measure reflects the relative cost of executing a trade at the best available prices, expressed as a percentage of the mid-quote.

PQS is an ex-ante measure, meaning it represents the potential transaction cost at a given point in time without considering the actual execution price. This characteristic makes PQS particularly useful for assessing liquidity in static market conditions and comparing liquidity across different securities or time periods.

PQS is frequently used to analyze the tightness of the market and its response to changing conditions. A lower PQS indicates tighter spreads and more liquid markets, while a higher PQS reflects wider spreads, suggesting reduced liquidity. These attributes make PQS a critical metric for evaluating market efficiency and the behavior of market participants.

The simplicity of PQS also facilitates its application in real-time monitoring systems and algorithmic trading strategies. By providing an instantaneous snapshot of liquidity, PQS enables traders to make informed decisions about order placement and execution. Additionally, its ease of calculation makes it suitable for high-frequency data analysis, where computational efficiency is essential.

3.3.4 Weighted Order Value (WOV)

The Weighted Order Value (WOV) is a liquidity measure designed to capture the value of orders placed in the market, weighted by their likelihood of execution. This measure originates from the analysis of order book dynamics in limit order markets, where the structure and distribution of orders significantly impact market liquidity. WOV was first introduced in the context of high-frequency trading by Aitken and Comerton-Forde (2003) as part of their research into order-driven markets.

The mathematical representation of WOV is:

$$WOV_t = \sum_{i=1}^{N} w_i \cdot O_i, \tag{3.5}$$

where O_i represents the value of the i^{th} order, and w_i is the weight assigned to that order based on its probability of execution. The weights w_i are typically determined by the relative position of the order in the limit order book, with higher weights assigned to orders closer to the best bid or ask prices.

WOV provides a dynamic view of liquidity by accounting for both the volume and the quality of orders in the market. It reflects the distribution of liquidity supply and demand, emphasizing the availability of executable orders. This makes WOV particularly useful for understanding the depth and immediacy of the market.

In high-frequency trading, WOV is often used to analyze order book imbalances and predict short-term price movements. A high WOV on the buy side, for instance, may indicate strong demand and potential upward price pressure, while a high WOV on the sell side suggests the opposite. These insights are critical for designing algorithmic trading strategies and managing execution risk.

The computational complexity of WOV is higher than simpler liquidity measures like Percent Quoted Spread (PQS), as it requires detailed data on order book states and the execution probabilities of individual orders. However, its ability to incorporate the nuanced structure of the order book makes it an invaluable tool for market participants and researchers seeking to understand liquidity in depth.

3.3.5 Quote Slope (QS)

The Quote Slope (QS) is a liquidity measure that quantifies the steepness of the order book, reflecting the price impact of marginal trades. This measure was first introduced by Hasbrouck and Seppi (2001) as part of their work on market microstructure, where the shape of the limit order book is used to infer liquidity characteristics. QS is particularly relevant for assessing the resilience of a market, as it indicates how quickly prices change in response to incoming orders.

The mathematical representation of QS is:

$$QS_t = \frac{\Delta P}{\Delta Q}, \tag{3.6}$$

where ΔP is the change in price and ΔQ is the change in the quoted depth (sum of the bid and ask volumes) at a given level of the order book. This ratio captures the sensitivity of price to changes in the available liquidity, providing a measure of market elasticity.

QS is an ex-ante measure of liquidity, meaning it assesses the potential price impact before trades are executed. A steeper slope (higher QS) indicates that small changes in order quantities result in significant price movements, reflecting a less resilient and more illiquid market. Conversely, a flatter slope (lower QS) suggests that the market can absorb larger trades with minimal price disruption, indicating greater depth and resiliency.

In high-frequency trading, QS is particularly valuable for analyzing intraday liquidity dynamics and identifying periods of heightened volatility. Traders and market makers use QS to adjust their strategies, such as setting tighter spreads in stable markets or widening spreads during volatile conditions. Additionally, QS provides insights into order book imbalances, helping traders anticipate short-term price movements driven by liquidity constraints.

The computation of QS requires granular order book data, including detailed information on price levels and associated volumes. Despite its higher data requirements, QS is widely used in empirical research and trading applications due to its ability to capture the dynamic interplay between order flow and price movements.

3.3.6 The VPIN Metric and Flow Toxicity

The Volume-Synchronized Probability of Informed Trading (VPIN) metric is a novel approach to measuring order-flow toxicity, a concept central to understanding liquidity in high-frequency markets. Flow toxicity occurs when informed traders exploit uninformed market participants, adversely selecting market makers who unknowingly provide liquidity at a loss. VPIN was developed by Easley et al. (2012b) to address the challenges of quantifying toxicity in the high-frequency world.

Unlike traditional metrics based on clock time, VPIN operates in volume time, which aligns with the rate of information flow in high-frequency trading environments. This allows for a more accurate representation of market dynamics, as periods of high trade intensity are more likely to capture significant informational events. VPIN is particularly effective in predicting short-term, toxicity-induced volatility and has been linked to major market disruptions, such as the May 6, 2010, "flash crash."

The VPIN metric is calculated using the following formula:

$$VPIN = \frac{\sum_{i=1}^{n} |V_i^B - V_i^S|}{nV}, \tag{3.7}$$

where V_i^B and V_i^S represent the buy and sell volumes in the ith volume bucket, respectively, n is the number of buckets, and V is the fixed volume of each bucket. By synchronizing observations with traded volume, VPIN minimizes the impact of volatility clustering and heteroscedasticity commonly observed in time-based measures.

VPIN's real-time nature and predictive capabilities make it an invaluable tool for market participants and regulators. For market makers, VPIN serves as a risk management metric, signaling periods of elevated toxicity that may warrant adjusting their trading strategies or reducing their market exposure. Regulators can use VPIN to monitor liquidity conditions and implement preemptive measures, such as trading halts, to prevent market breakdowns.

Empirical studies have demonstrated VPIN's ability to predict significant market events. For instance, during the "flash crash," VPIN levels indicated increasing order-flow toxicity well before the market collapse, providing early warnings of impending liquidity withdrawal. This makes VPIN a critical component in the toolkit for understanding and managing liquidity in modern, high-frequency trading ecosystems.

3.4 Extreme Price Movement Identification

Identifying EPMs is crucial for understanding and predicting rare, high-impact events in financial markets. In this study, I draw inspiration from the methodology proposed by Brogaard et al. (2018) to establish an effective framework for EPM detection. Their techniques emphasize robustness in identifying significant deviations in price movements while addressing common challenges in financial data analysis.

Brogaard et al. proposed three approaches for EPM identification, focusing on absolute returns, residuals from market models, and volatility-adjusted thresholds. While their 99.9th percentile threshold for midquote returns is a well-established standard, this study adjusts the threshold to the 99th percentile to accommodate the high-frequency nature of the dataset and the six-month time frame. This modification ensures sufficient identification of EPMs while maintaining a manageable number of events for analysis.

The identification process in this study begins by calculating the log returns of midquote prices. This measure eliminates the potential biases associated with bid-ask bounces and focuses on fundamental price changes. The resulting log returns are then evaluated against the 99th percentile threshold to isolate intervals exhibiting extreme movements. This percentile-based method provides a straightforward yet effective approach to capturing significant deviations from typical price behavior.

Chapter 4

Models

4.1 Logistic Regression

Logistic regression (logit model) is a widely employed statistical method for binary classification problems, making it particularly suitable for analyzing financial markets where extreme price movements (EPM) are rare but impactful. Originating from Verhulst's logistic function and later formalized for statistical modeling, the logit model has been extensively used in financial research to predict binary outcomes such as market crashes, order imbalances, or price shifts. Its simplicity and interpretability, coupled with its ability to incorporate both continuous and categorical predictors, have made it a foundational tool in financial econometrics.

The utility of logistic regression in financial markets has been highlighted in several seminal works. Easley et al. (2019)) utilized logistic regression to quantify order-flow toxicity, demonstrating its role in anticipating liquidity withdrawal during periods of market stress. Similarly, Hasbrouck (2007) explored its application in predicting price changes based on high-frequency trading data, showcasing its effectiveness in linking liquidity and market dynamics to binary outcomes. Logistic regression has also been applied to study the relationship between bid-ask spreads, order flow, and price volatility, as discussed in Aitken and Comerton-Forde (2003). Despite its widespread adoption, traditional logistic regression faces limitations, particularly in handling high-dimensional datasets and multicollinearity among predictors.

To address these challenges, this study builds on the standard logistic regression model by integrating modern advancements such as LASSO regularization, balanced class weights and optimized classification thresholds. LASSO (Least Absolute Shrinkage and Selection Operator), introduced by Tibshirani (1996), is a regularization technique that shrinks less significant coefficients to zero, effectively performing variable selection and reducing overfitting. This enhancement is particularly valuable in high-frequency trading environments where the dataset contains numerous predictors, many of which may be highly correlated.

The logistic regression model used in this study predicts EPMs, defined as log returns below the 1st percentile of the return distribution in the next time interval. The model is expressed as:

$$\text{logit}(P(y=1)) = \beta_0 + \sum_{i=1}^{k} \beta_i X_i, \tag{4.1}$$

where $P(y=1)$ is the probability of an EPM, X_i are the predictor variables, β_0 is the intercept, and β_i are the regression coefficients estimated via maximum likelihood.

4.1.1 Evaluation Metrics: ROC Curve, AUC, and Precision-Recall Curve

In evaluating the performance of logistic regression models, two key tools are used: the Receiver Operating Characteristic (ROC) curve with its Area Under the Curve (AUC) metric, and the precision-recall (PR)

curve. These metrics provide complementary perspectives on the model's ability to distinguish between classes, particularly in imbalanced datasets.

ROC Curve and AUC

The ROC curve is a graphical representation of the trade-off between the true positive rate (TPR) and the false positive rate (FPR) across various cut-off thresholds. It provides insights into the model's overall discriminative ability, regardless of class imbalance.

The TPR and FPR are calculated as:

$$\text{TPR} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}},$$

$$\text{FPR} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}}.$$

The Area Under the Curve (AUC) summarizes the performance of the model into a single value ranging from 0 to 1. AUC is calculated as the integral of the ROC curve:

$$\text{AUC} = \int_0^1 \text{ROC}(x)\, dx \tag{4.2}$$

where higher AUC values indicate better discriminative power. AUC is particularly useful as it remains invariant to changes in the class distribution, making it an ideal metric for balanced datasets. However, in cases of significant imbalance, it may overstate the model's effectiveness.

Precision-Recall Curve

For datasets with a skewed class distribution, the precision-recall (PR) curve is a more informative metric. The PR curve plots precision against recall at various cut-off thresholds. These metrics are defined as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}},$$

$$\text{Recall} = \text{TPR} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}.$$

Unlike the ROC curve, the PR curve focuses exclusively on the minority class, highlighting the trade-off between precision and recall. This is particularly valuable for applications where correctly identifying the minority class is critical.

Practical Implications of Metrics

- **ROC Curve and AUC:** Provide a broad view of the model's classification ability, useful for comparing models on balanced datasets.
- **Precision-Recall Curve:** Offers deeper insights into performance for imbalanced datasets, emphasizing the model's capability to handle rare events.

By combining these evaluation metrics, a comprehensive understanding of the model's strengths and limitations is achieved. In this analysis, both the ROC and PR curves are plotted to assess the model's ability to predict flash crashes effectively.

4.1.2 Cut-off Selection

In logistic regression, selecting an appropriate cut-off threshold is crucial for converting predicted probabilities into binary classification outcomes. The cut-off directly influences the balance between precision, recall, and overall model performance. For this analysis, three cut-off values are considered: 0.5 (default threshold), the optimal threshold derived from the G1 score, and the optimal threshold derived from the G2 score.

Default Cut-off (0.5)

A threshold of 0.5 is commonly used in logistic regression, as it assigns class 1 to instances where the predicted probability exceeds 50%. While simple and interpretable, this threshold may not be optimal for imbalanced datasets, as it assumes equal costs for false positives and false negatives.

G1 Score and Cut-off

The G1 score is used to find a cut-off that balances sensitivity (true positive rate, TPR) and specificity (true negative rate, TNR). It is defined as:

$$G1 = \sqrt{\text{TPR} \cdot \text{TNR}}. \tag{4.3}$$

The optimal cut-off for G1 is identified by maximizing the G1 score. This threshold aims to achieve a balance between the model's ability to correctly identify both positive and negative instances, making it particularly useful when sensitivity and specificity are equally important.

G2 Score and Cut-off

The G2 score is used to identify a cut-off that prioritizes the trade-off between sensitivity (TPR) and precision. It is defined as:

$$G2 = \sqrt{\text{TPR} \cdot \text{Precision}}. \tag{4.4}$$

The optimal cut-off for G2 is determined by maximizing the G2 score. This threshold emphasizes the model's ability to correctly identify positive instances while minimizing false positives, making it valuable for applications where precision is a priority.

Practical Implications of Cut-offs

Each cut-off has its own implications for model performance:

- **Cut-off at 0.5:** Provides a baseline for comparison but may underperform in imbalanced datasets.
- **Cut-off from G1:** Offers a balanced approach, suitable when both sensitivity and specificity are critical.
- **Cut-off from G2:** Prioritizes precision, making it ideal for scenarios where false positives carry significant costs.

4.1.3 Model Comparison Metrics: AIC, BIC, and Log-Likelihood

To evaluate and compare logistic regression models, robust metrics that balance model fit and complexity are essential. The Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and log-likelihood are commonly used for this purpose. These metrics provide a systematic framework for selecting the most suitable model while avoiding overfitting.

Akaike Information Criterion (AIC)

The AIC assesses the trade-off between the goodness-of-fit of a model and its complexity. It is defined as:

$$\text{AIC} = -2 \cdot \log\left(\hat{L}\right) + 2k, \tag{4.5}$$

where $\hat{L}$ is the likelihood of the model given the data, and k represents the number of parameters in the model. Models with lower AIC values are generally preferred, as they indicate a better balance between fit and simplicity. The AIC penalizes models with more parameters to discourage overfitting.

Bayesian Information Criterion (BIC)

The BIC extends the principles of AIC by incorporating a stronger penalty for model complexity, particularly for large sample sizes. It is calculated as:

$$\text{BIC} = -2 \cdot \log\left(\hat{L}\right) + k \cdot \log(n), \tag{4.6}$$

where n is the sample size. The BIC is often more conservative than the AIC in selecting models, as it imposes a higher penalty for additional parameters. This makes BIC especially useful in contexts where parsimony is prioritized.

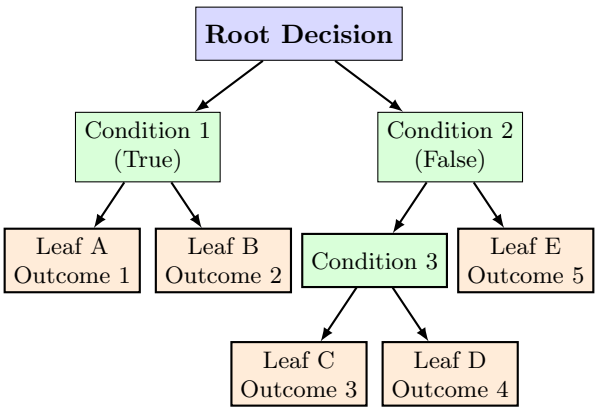
Log-Likelihood

The log-likelihood quantifies how well the model explains the observed data by measuring the likelihood of the observed outcomes under the estimated parameters. Higher log-likelihood values indicate a better fit of the model to the data. This metric provides foundational insight into the model's performance and is a component of both AIC and BIC calculations.

4.2 Decision Tree

Decision trees are a supervised machine learning algorithm used for classification tasks, offering a straightforward and interpretable method to predict Extreme Price Movements (EPMs). These models utilize a flowchart-like structure, where internal nodes evaluate specific features, branches represent decision outcomes, and leaf nodes provide final predictions. This interpretability makes decision trees particularly valuable in financial contexts, where understanding the rationale behind predictions is crucial. However, the inherent imbalance in datasets related to EPMs poses challenges, as the majority class typically dominates. To address this issue, the ROSE (Random Over-Sampling Examples) method is employed to balance the training dataset by generating synthetic samples for the minority class, ensuring that the model effectively captures the characteristics of rare events.

Decision Tree Visualization



Colored nodes represent the hierarchy of decisions and outcomes.

Figure 4.1: A general decision tree

This study employs two types of decision tree models: a simpler tree and a more complex tree. The simpler tree is constructed with higher values for the complexity parameter (`cp`) and stricter minimum requirements for node splitting (`minsplit`) and leaf node size (`minbucket`). These constraints encourage

a shallow structure, reducing overfitting and enhancing interpretability. Conversely, the more complex tree uses lower `cp` values and relaxed splitting criteria, allowing deeper exploration of the data. To mitigate the overfitting risk associated with deeper trees, cross-validation is used during training to identify the optimal `cp` value, which is subsequently applied for pruning. Pruning retains only the splits that contribute significantly to model performance, balancing complexity with generalization.

Model performance is evaluated using a comprehensive set of metrics. Accuracy, while providing an overall measure of correctness, is insufficient for imbalanced datasets. Sensitivity (recall) is critical, as it measures the model's ability to correctly identify EPMs, ensuring that rare events are detected. Specificity assesses the model's capacity to correctly classify normal instances, avoiding excessive false positives. Precision indicates the proportion of predicted EPMs that are actual EPMs, reflecting the reliability of positive predictions. The F1 score combines sensitivity and precision, offering a balanced perspective when both false negatives and false positives are important. Balanced accuracy, the average of sensitivity and specificity, provides a robust metric for imbalanced datasets.

Feature importance is another critical aspect of this study. The decision tree algorithm assigns importance scores to features based on their contributions to reducing impurity at each split. In this context, features such as spread, log returns, short-term volume, and skewness are expected to play significant roles in predicting EPMs. These scores not only guide the interpretation of model predictions but also provide insights into the underlying factors driving extreme market movements. By analyzing the importance scores, the study identifies the most influential predictors, offering a deeper understanding of the drivers of EPMs.

The final selection between the two models depends on their performance metrics and interpretability. The simpler tree is expected to offer greater transparency and robustness, making it suitable for scenarios requiring clear decision-making processes. The more complex tree, while potentially harder to interpret, may uncover subtle patterns that improve predictive accuracy, particularly in detecting rare events. By combining the evaluation of performance metrics, feature importance, and model complexity, this analysis identifies the most appropriate decision tree model for forecasting extreme price movements in financial markets.

4.3 Random Forest

Developed by Breiman (2001), Random Forest is a powerful ensemble learning technique that builds on the foundation of decision trees. Unlike a single decision tree, which is prone to overfitting and may not generalize well to unseen data, Random Forest constructs multiple trees and combines their predictions to create a more robust model (that can be found in Figure 1). This ensemble approach enhances the predictive accuracy and stability of the model, particularly in scenarios where datasets are complex and noisy, as often encountered in financial markets when predicting Extreme Price Movements.

The Random Forest algorithm introduces two key elements to improve the performance of decision trees: bootstrapping and feature randomness. Bootstrapping involves generating multiple subsets of the training data by sampling with replacement. Each decision tree is trained on a different subset, ensuring diversity among the trees. Feature randomness, on the other hand, limits the number of features considered for each split within a tree, further promoting variation in tree structure. By aggregating the predictions of these diverse trees through majority voting for classification tasks, Random Forest achieves high accuracy while mitigating overfitting.

The ability of Random Forest to handle imbalanced datasets is particularly beneficial for this study. Financial data related to EPMs often exhibits a significant imbalance between the frequency of normal and extreme events. This imbalance can lead to biased models that favor the majority class. Random Forest addresses this challenge effectively by enabling balanced sampling techniques and incorporating methods to weigh the contributions of individual trees based on their performance.

In addition to its predictive power, Random Forest provides valuable insights into the importance of features. During training, the algorithm evaluates how much each feature contributes to reducing uncertainty or impurity in the data at various splits. Features such as spread, log returns, and short-term volume, among others, emerge as critical predictors in this context. Understanding these contributions not only improves model interpretability but also sheds light on the underlying factors influencing EPMs, offering actionable insights for financial analysts and practitioners.

Performance evaluation of the Random Forest model relies on a comprehensive set of metrics tailored to the imbalanced nature of EPM datasets. Sensitivity, or the ability to detect rare extreme events, is a priority, as missing such events could have significant implications. Specificity, which measures the model's accuracy in identifying normal market conditions, ensures reliability and avoids false alarms. Metrics such as the F1 score and balanced accuracy combine these considerations, providing a holistic view of model effectiveness. Furthermore, the Receiver Operating Characteristic (ROC) curve and its Area Under the Curve (AUC) are used to evaluate the model's ability to distinguish between classes across various thresholds.

By combining the strengths of multiple decision trees and leveraging advanced sampling techniques, Random Forest emerges as a versatile and reliable tool for predicting Extreme Price Movements. Its balance of interpretability, accuracy, and robustness makes it a crucial component of the machine learning framework employed in this study.

4.4 Neural Network

Neural networks are a class of machine learning algorithms inspired by the structure and function of the human brain. They consist of layers of interconnected nodes, or "neurons," designed to learn and model complex patterns in data. Neural networks excel at tasks where traditional machine learning models might struggle, such as handling high-dimensional, non-linear, and noisy datasets. Their adaptability and capacity for representation make them a versatile tool in various fields, including image recognition, natural language processing, and financial forecasting.

The structure of a neural network typically includes three types of layers: an input layer, one or more hidden layers, and an output layer. The input layer represents the features of the dataset, while the hidden layers process these features through a series of transformations. Each neuron in a hidden layer receives inputs, computes a weighted sum, applies a bias, and passes the result through a non-linear activation function. This enables the network to learn complex relationships and patterns in the data. The output layer produces predictions, which can be in the form of probabilities for classification tasks or numerical values for regression tasks.

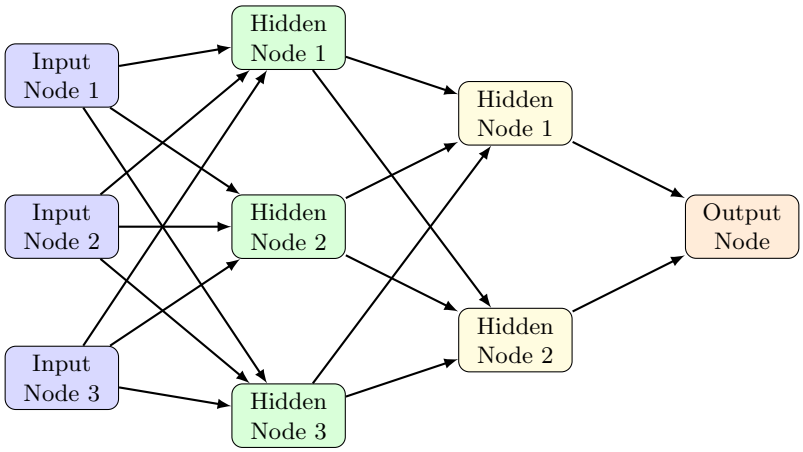


Figure 4.2: General Neural Network Structure

Training a neural network involves adjusting the weights and biases of the neurons to minimize a loss function, which quantifies the difference between the predicted outputs and the true values. This process is achieved using optimization algorithms such as Adam, combined with a technique called backpropagation. Backpropagation computes the gradient of the loss function with respect to each weight and bias in the network, allowing for iterative updates that improve the model's predictions.

The choice of hyperparameters, such as the number of hidden layers, the number of neurons per layer, the activation functions, and the learning rate, significantly impacts the performance of a neural network. Regularization techniques, such as dropout and weight decay, are often employed to prevent overfitting, especially when dealing with limited or noisy data.

Neural networks excel at capturing complex patterns in data, but they also have limitations. They require substantial computational resources and large amounts of labeled data for training. Additionally, their complexity can make them less interpretable compared to simpler models like decision trees. In financial contexts, this lack of interpretability can be a barrier, as stakeholders often require transparent models to justify predictions.

Despite these challenges, neural networks have proven to be highly effective for a wide range of applications. Their ability to model intricate, non-linear relationships makes them particularly well-suited for predicting rare and extreme events, such as EPMs. With advancements in hardware, algorithms, and training techniques, neural networks continue to play a pivotal role in modern machine learning.

4.4.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) designed to handle sequential data and capture long-term dependencies. Developed by Hochreiter and Schmidhuber (1997), LSTMs address the vanishing gradient problem commonly encountered in traditional RNNs. This makes them particularly effective for tasks involving time-series data, where understanding temporal relationships is crucial, such as financial forecasting or speech recognition.

LSTM networks consist of a series of interconnected memory cells, each containing three primary components: an input gate, a forget gate, and an output gate. These gates regulate the flow of information

within the cell, enabling it to retain or discard information as needed. The memory cell maintains a cell state, which acts as a long-term memory, and a hidden state, which serves as a short-term memory. The combination of these states allows LSTMs to capture both immediate and long-range dependencies in the data.

The key components of an LSTM memory cell are as follows:

- **Input Gate:** Determines which new information from the current input should be stored in the cell state.
- **Forget Gate:** Decides which information in the cell state should be discarded, allowing the network to forget irrelevant or outdated information.
- **Output Gate:** Controls how much of the information from the cell state should be passed to the next layer or output as the hidden state.

The mathematical operations for these gates are expressed as:

$$\begin{aligned}
&\textbf{Forget Gate:} \quad f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\
&\textbf{Input Gate:} \quad i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \\
&\textbf{Cell State Update:} \quad C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \\
&\textbf{Output Gate:} \quad o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad h_t = o_t \cdot \tanh(C_t),
\end{aligned}$$

where f_t, i_t, and o_t are the forget, input, and output gates, respectively; C_t and h_t represent the cell and hidden states; x_t is the input at time t; and σ and tanh denote the sigmoid and hyperbolic tangent activation functions.

LSTM Memory Cell Structure

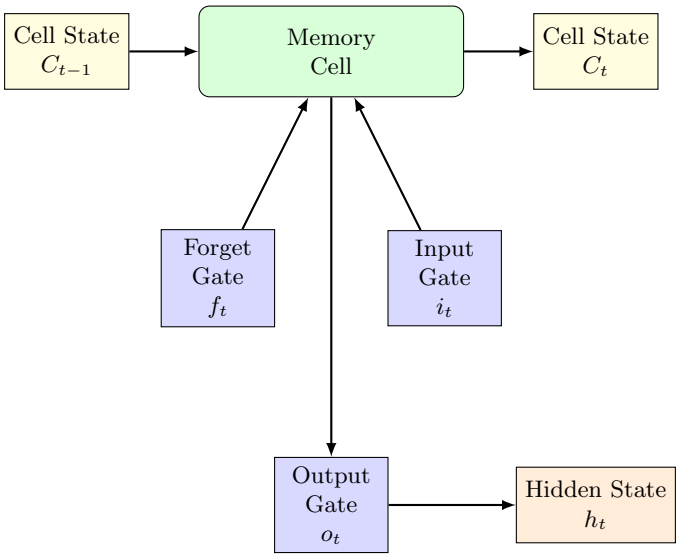


Figure 4.3: LSTM Memory Cell

LSTMs are particularly suited for sequential data because they can maintain and utilize information from earlier time steps. This makes them valuable for tasks like time-series prediction, where past observations significantly influence future outcomes. By stacking multiple LSTM layers or combining them with dense layers, more complex architectures can be built to capture hierarchical and temporal patterns.

However, LSTMs also have limitations. Training can be computationally intensive, and the models are sensitive to hyperparameters such as the number of units per layer, learning rate, and dropout rate. Additionally, while LSTMs can model long-term dependencies, they may struggle with extremely long sequences without architectural enhancements like bidirectional LSTMs or attention mechanisms.

Despite these challenges, LSTMs remain a cornerstone in time-series analysis and other sequential data tasks. Their ability to model temporal dependencies and adapt to non-linear patterns makes them a powerful tool for predicting rare and extreme events in complex domains.

Chapter 5

Empirical Results

This chapter presents and examines the results of the predictive models. For clarity and consistency, the analysis will focus primarily on the Facebook stock. This choice is made because the results for the five stocks are quite similar. However, any significant differences or unique findings for the other stocks will be highlighted and discussed when relevant. All confusion matrices for the five stocks, including those not covered in detail here, can be found in the appendix for reference 6.4.

5.1 Logistic Regression

In this phase of the analysis, a logistic regression model (GLM logit reg) is employed to predict the occurrence of flash crashes ('y') based on the provided covariates.
For this logistic regression, we use two separate datasets: one for training the model, and one for testing the model. The training dataset contains the first 19 weeks and the testing dataset contains the last 6 weeks. The model's summary is presented, and in case of convergence issues, adjustments to the 'maxit' parameter or consideration of correlated covariates are suggested.

Call:

```
glm(formula = "y .", family = binomial, data = dt_train, maxit = 50)
```

Coefficients:

Variable	Estimate	Std. Error	z value	Pr($\|z\|$)
(Intercept)	−2.462e−01	5.663e−01	−0.435	0.66380
Time	−1.077e−01	1.209e−02	−8.903	< 2e−16 ***
Size	−3.101e−05	9.559e−06	−3.244	0.00118 **
Price	−3.651e−02	2.161e−03	−16.893	< 2e−16 ***
ASKs1	1.367e−05	7.948e−06	1.720	0.08544 .
BIDs1	−7.155e−05	5.603e−05	−1.277	0.20157
Logreturn	−2.175e+02	3.657e+01	−5.948	2.71e−09 ***
spread	9.778e+00	1.229e+00	7.953	1.81e−15 ***
PES	3.209e+02	2.678e+02	1.198	0.23079
vol_short	3.482e+03	1.126e+02	30.923	< 2e−16 ***
kurt_short	1.443e−02	1.838e−02	0.785	0.43220
skew_short	1.162e−01	4.283e−02	2.714	0.00666 **
vpin	1.581e+00	3.707e−01	4.265	2.00e−05 ***

Signif. Codes: `0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1`

We can see that the covariates Time, Price, vol_short, spread, and Logreturn are the most relevant, followed by Size, skew_short, and vpin. The ROC curve is plotted (5.1), depicting the true positive rate against the false positive rate. Notably, the area under the curve (AUC) is calculated(4.2). The ROC curve analysis yielded an AUC of 0.798, indicating good discriminative ability. It is important to acknowledge that the AUC might be influenced by the imbalanced nature of the dataset, particularly when the majority class (y = 0) dominates.

Recognizing the dataset's imbalance, a precision-recall analysis is conducted to gain a nuanced understanding of the model's performance. A precision-recall curve is presented, highlighting the trade-off between precision and recall. Given the imbalanced nature, the analysis suggests focusing more on the minority class (y = 1), emphasizing precision over recall.

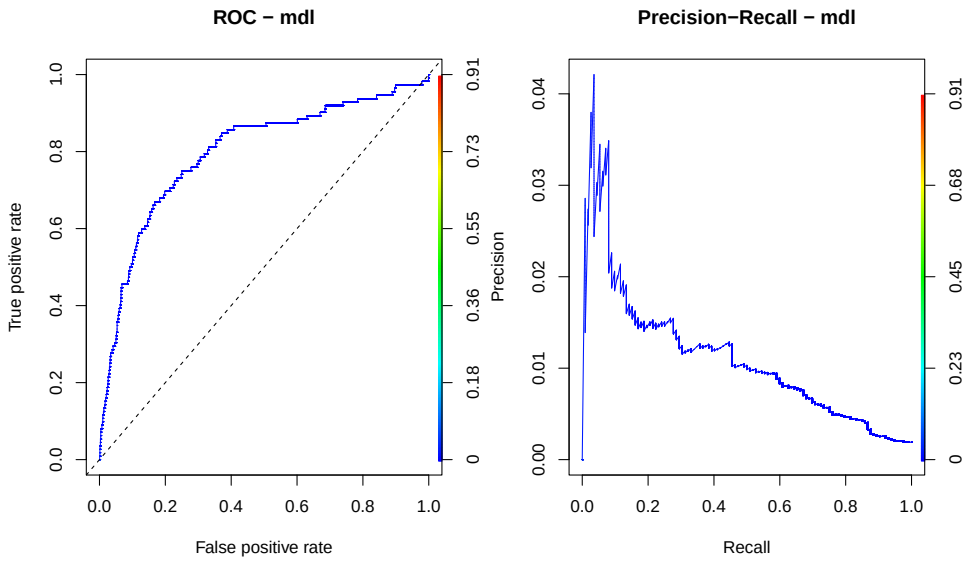


Figure 5.1: ROC and Precision-Recall for Logistic Regression

It is observed that, as anticipated, the precision-recall results are relatively poorer compared to the ROC curve (5.1) due to the dataset's imbalance. The choice of the cut-off point becomes crucial, representing a trade-off between precision and recall. Higher cut-off values enhance precision but may sacrifice recall, and this decision should be aligned with the specific cost considerations. A higher focus on precision is recommended when false positives carry significant costs, whereas prioritizing recall becomes imperative when it is crucial to identify all positive instances, even at the expense of more false positives.

The following evaluation metrics are calculated to assess the model's performance:

- **Akaike Information Criterion (AIC):** 21221.2
- **Bayesian Information Criterion (BIC):** 21312.5
- **Log-likelihood:** -10601.6

We use the G1 score and G2 score for helping us to find the right cutoff.

The G1 score, calculated as 0.002670691 (4.3), represents the square root of the product of the true

positive rate (TPR) and true negative rate (TNR) in the precision-recall curve. It indicates the model's ability to balance sensitivity and specificity, providing a comprehensive measure of its predictive accuracy.

The G2 score, calculated as 0.004737658 (4.4), is derived from the square root of the product of the true positive rate (TPR) and precision. This metric gauges the trade-off between true positive rate and precision, offering insights into the model's ability to correctly identify positive instances while minimizing false positives.

5.1.1 Confusion Matrix Analysis

To evaluate the performance of the logistic regression model in predicting Extreme Price Movements (EPM), confusion matrices are computed for three cut-off thresholds: the default threshold (0.5), the G1-optimal cut-off, and the G2-optimal cut-off. These matrices provide insights into the model's classification outcomes, including true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Confusion Matrix at Default Cut-off (0.5)

At the default cut-off of 0.5, a prediction of EPM ($y = 1$) is made if the predicted probability exceeds 50%. The confusion matrix is shown below:

Prediction	**Reference = 0**	**Reference = 1**
0 (No EPM)	58266	112
1 (EPM)	4	0

Table 5.1: Confusion matrix for the default cut-off (0.5) used for Facebook EPM prediction

Key observations:

- **Accuracy:** High overall accuracy driven by correct predictions for the majority class ($y = 0$).
- **Sensitivity:** The model fails to identify any EPM events ($y = 1$), resulting in zero true positives.
- **Specificity:** Nearly perfect specificity, with most non-EPM instances correctly identified.

Confusion Matrix at G1-optimal Cut-off

The G1-optimal cut-off, calculated as 0.00267, balances sensitivity (TPR) and specificity (TNR). The confusion matrix is as follows:

Prediction	**Reference = 0**	**Reference = 1**
0 (No EPM)	43700	29
1 (EPM)	14570	83

Table 5.2: Confusion matrix for the G1-optimal cut-off (0.00267) used for Facebook EPM prediction

Key observations:

- **Improved Sensitivity:** The model correctly identifies 83 EPM instances, significantly improving recall compared to the default cut-off.
- **Reduced Specificity:** A trade-off is observed, with 14570 false positives.
- **Balanced Classification:** This threshold provides a reasonable balance between identifying EPM events and avoiding false positives.

Confusion Matrix at G2-optimal Cut-off

The G2-optimal cut-off, calculated as 0.00474, prioritizes precision over sensitivity. The confusion matrix is as follows:

Prediction	Reference = 0	Reference = 1
0 (No EPM)	54295	62
1 (EPM)	3975	50

Table 5.3: Confusion matrix for the G2-optimal cut-off (0.00474) used for Facebook EPM prediction

Key observations:

- **Higher Precision:** False positives are significantly reduced, with only 3975 non-EPM instances misclassified as EPM.
- **Lower Sensitivity:** The model identifies 50 EPM events, slightly fewer than with the G1-optimal cut-off.
- **Precision Over Recall:** This threshold is ideal for applications where false positives carry significant costs.

Summary of Metrics

The following table summarizes the key performance metrics for each cut-off:

Metric	Default (0.5)	G1-optimal	G2-optimal
Accuracy	99.8%	74.9%	93.1%
Sensitivity (Recall)	0%	74.1%	44.6%
Specificity	100%	74.7%	93.2%
Precision	0%	0.6%	1.2%

Conclusion

Each cut-off serves distinct purposes:

- **Default (0.5):** Provides a baseline but performs poorly for the minority class ($y = 1$).
- **G1-optimal:** Balances sensitivity and specificity, making it suitable for general applications.
- **G2-optimal:** Emphasizes precision, ideal for scenarios where false positives carry significant costs.

5.2 Logistic Regression with Selected Covariates

A second logistic regression model is developed using only the covariates with a p-value ≤ 0.05 from the initial model. The selected covariates are `Price`, `Time`, `Logreturn`, `Size`, `spread`, `vol_short`, `skew_short`, and `vpin`.

It is important to note that the set of variables differs for other stocks:

- **Apple**: `Time`, `Price`, `Logreturn`, `spread`, `PES`, `vol_short`, `skew_short`.
- **Microsoft**: `Time`, `Size`, `Price`, `BIDs1`, `Logreturn`, `spread`, `PES`, `vol_short`.
- **NVIDIA**: `Time`, `Price`, `ASKs1`, `Logreturn`, `spread`, `vol_short`, `kurt_short`.
- **Google**: `Price`, `spread`, `PES`, `vol_short`.

This approach reduces model complexity while retaining key predictors, ensuring a balance between simplicity and predictive power. The reduction in degrees of freedom (df) underscores the emphasis on a parsimonious model that avoids overfitting.

Evaluation metrics, including the AIC, BIC, and log-likelihood, are computed to assess the model's goodness-of-fit:

- **AIC:** 21221.2
- **BIC:** 21312.5
- **Log-likelihood:** -10601.6

The ROC curve (5.2) shows an AUC of 0.797, comparable to the full-covariate model, suggesting similar discriminative power. A precision-recall curve (5.2) highlights the trade-off between precision and recall, with performance metrics optimized for the imbalanced dataset.

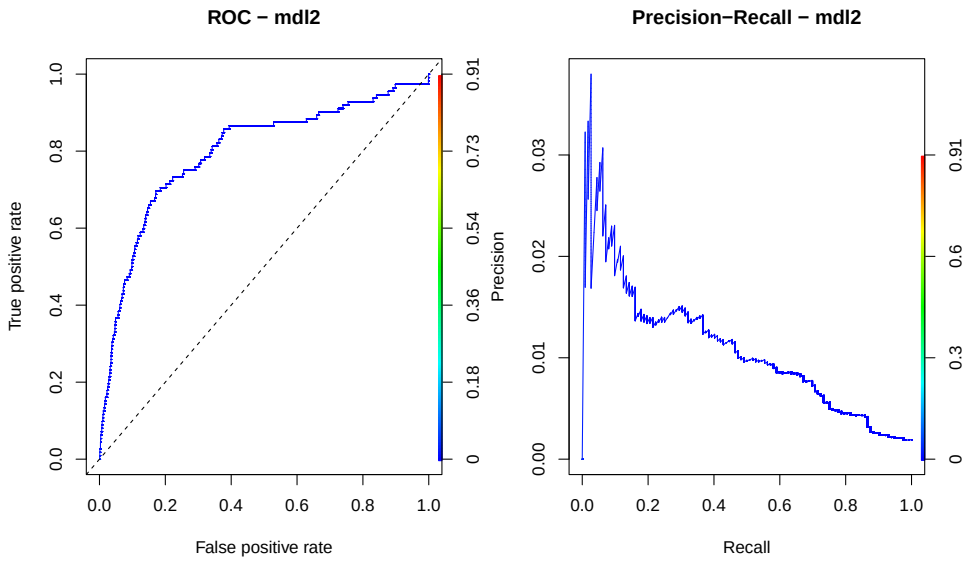


Figure 5.2: ROC and Precision-Recall for Logistic Regression with Selected Covariates

To address the imbalanced nature of the dataset, the G1 and G2 scores are used to identify optimal cut-offs:

- **G1-optimal cut-off:** 0.00319
- **G2-optimal cut-off:** 0.00339

The G1 score of 0.00319 reflects a balance between sensitivity and specificity, while the G2 score of 0.00339 emphasizes the trade-off between sensitivity and precision.

5.2.1 Confusion Matrix Analysis

The performance of the selected model is evaluated using confusion matrices for the G1-optimal and G2-optimal cut-offs.

Prediction	Reference = 0	Reference = 1
0 (No EPM)	48300	35
1 (EPM)	9970	77

Table 5.4: Confusion Matrix at G1-optimal Cut-off (0.00319) used for Facebook EPM prediction

Confusion Matrix at G1-optimal Threshold

Key observations:

- **Improved Sensitivity:** The model identifies 77 EPM events, demonstrating improved recall compared to the default model.
- **Reduced Specificity:** Higher false positives are observed, with 9970 non-EPM instances misclassified.

Confusion Matrix at G2-optimal Threshold

Prediction	Reference = 0	Reference = 1
0 (No EPM)	49562	39
1 (EPM)	8708	73

Table 5.5: Confusion Matrix at G2-optimal Cut-off (0.00339) used for Facebook EPM prediction

Key observations:

- **Higher Precision:** False positives decrease significantly compared to the G1-optimal cut-off.
- **Slightly Reduced Sensitivity:** The model identifies 73 EPM events, slightly fewer than with the G1-optimal cut-off.

Summary

The selected model performs comparably to the full-covariate model, achieving a similar AUC and demonstrating strong predictive capabilities. The G1-optimal cut-off is suited for applications requiring a balance between sensitivity and specificity, while the G2-optimal cut-off prioritizes precision, making it ideal for scenarios where false positives carry significant costs. The reduction in model complexity highlights the trade-off between predictive performance and interpretability.

5.3 Logistic Regression with Balanced Class Weights

In this phase, a logistic regression model is trained using balanced class weights to address the dataset's imbalance between the occurrence of Extreme Price Movements (EPM) ($y = 1$) and non-EPM events ($y = 0$). The balanced weights ensure that instances of the minority class ($y = 1$) are assigned greater importance, improving the model's ability to identify rare events.

5.3.1 Model Specification and Performance Metrics

To calculate class weights, the training dataset assigns weights inversely proportional to the frequency of each class:

$$w = \begin{cases} \frac{1}{P(y=1)}, & \text{if } y = 1, \\ \frac{1}{P(y=0)}, & \text{if } y = 0. \end{cases}$$

The logistic regression model is then trained with these weights, ensuring that minority class predictions are not overshadowed by the dominant majority class. The model's summary metrics, including the

Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and log-likelihood, are as follows:

- **AIC:** 20345.7
- **BIC:** 20467.9
- **Log-likelihood:** -10172.85

5.3.2 Threshold Optimization and G-Scores

To optimize the classification threshold, the G1 and G2 scores are calculated and smoothed to identify the thresholds that maximize performance:

- **G1-optimal threshold:** 0.047
- **G2-optimal threshold:** 0.0547
- **Maximum G1 score:** 0.7569
- **Maximum G2 score:** 0.0786

These thresholds emphasize the trade-offs between sensitivity, specificity, and precision. The G1 score balances sensitivity and specificity, while the G2 score prioritizes sensitivity and precision.

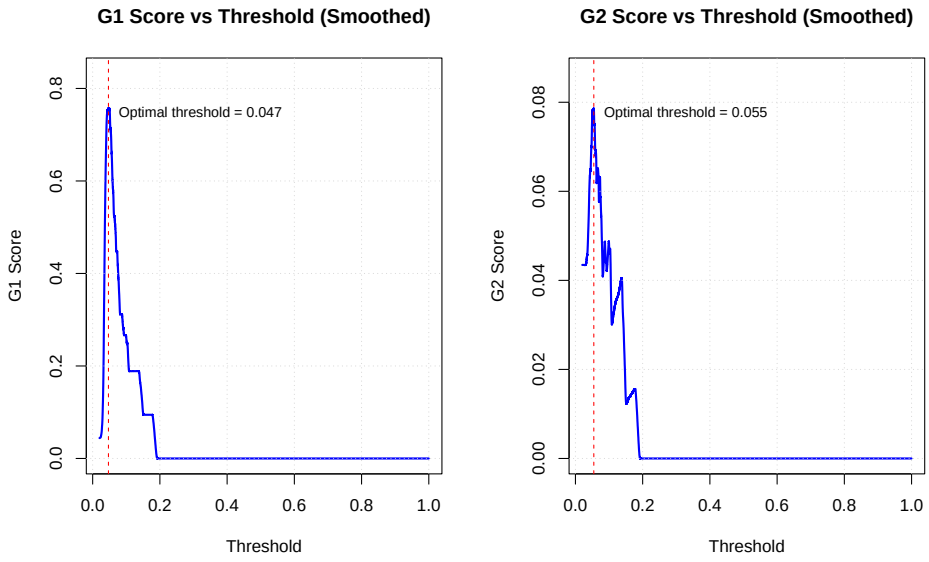


Figure 5.3: G1 and G2 Scores vs Threshold (Smoothed) for Logistic Regression with Balanced Class Weights

5.3.3 Confusion Matrix Analysis

The performance of the balanced logistic regression model is evaluated using confusion matrices for the G1-optimal and G2-optimal thresholds.

Prediction	Reference = 0	Reference = 1
0 (No EPM)	8091	4
1 (EPM)	50179	108

Table 5.6: Confusion Matrix at G1-optimal Threshold (0.047) used for Facebook EPM prediction

Confusion Matrix at G1-optimal Threshold

The G1-optimal threshold provides the following insights:

- **High Sensitivity:** The model identifies a larger proportion of true EPM events ($y = 1$).
- **Reduced Specificity:** Many non-EPM instances are misclassified, resulting in high false-positive rates.

Confusion Matrix at G2-optimal Threshold

Prediction	Reference = 0	Reference = 1
0 (No EPM)	13134	9
1 (EPM)	45136	103

Table 5.7: Confusion Matrix at G2-optimal Threshold (0.0547) used for Facebook EPM prediction

The G2-optimal threshold emphasizes precision, resulting in:

- **Higher Precision:** Fewer non-EPM instances are misclassified as EPM ($y = 1$).
- **Slightly Reduced Sensitivity:** The model captures slightly fewer true EPM events compared to the G1-optimal threshold.

5.3.4 Performance Summary

The following table summarizes the key metrics for each threshold:

Metric	G1-optimal	G2-optimal
Accuracy	83.6%	89.5%
Sensitivity (Recall)	96.4%	91.9%
Specificity	13.9%	22.5%
Precision	0.2%	0.3%

5.3.5 Conclusion

The logistic regression model with balanced class weights improves the identification of rare EPM events ($y = 1$) by addressing class imbalance through weighted adjustments. The G1-optimal threshold offers a balance between sensitivity and specificity, making it suitable for scenarios where identifying true EPMs is critical despite higher false positives. The G2-optimal threshold, with improved precision, is better suited for applications where false positives carry significant costs. These results demonstrate the effectiveness of balanced weights in improving the model's ability to detect EPM events in highly imbalanced datasets.

5.4 Logistic Regression with Balanced Class Weights and Significant Covariates

This section explores a logistic regression model that incorporates balanced class weights and uses only the covariates identified as statistically significant ($p \leq 0.05$) in the full model. The selected covariates

include: `Time`, `Price`, `Logreturn`, `spread`, `vol_short`, `skew_short`, and `vpin`. This approach aims to improve model efficiency by focusing on key predictors while mitigating the effect of the dataset's imbalance.

5.4.1 Model Specification and Performance Metrics

Balanced class weights were computed to address the dataset's imbalance:

$$w = \begin{cases} \frac{1}{P(y=1)}, & \text{if } y = 1, \\ \frac{1}{P(y=0)}, & \text{if } y = 0. \end{cases}$$

The logistic regression model was trained with these weights and the significant covariates. The evaluation metrics for this model are summarized below:

- **Akaike Information Criterion (AIC):** 355843
- **Bayesian Information Criterion (BIC):** 357974
- **Log-likelihood:** -177921.5

5.4.2 ROC and Precision-Recall Curves

Figure 5.4 illustrates the ROC curve and Precision-Recall curve for the model. These plots provide a visual assessment of the model's discriminative ability and its trade-off between precision and recall.

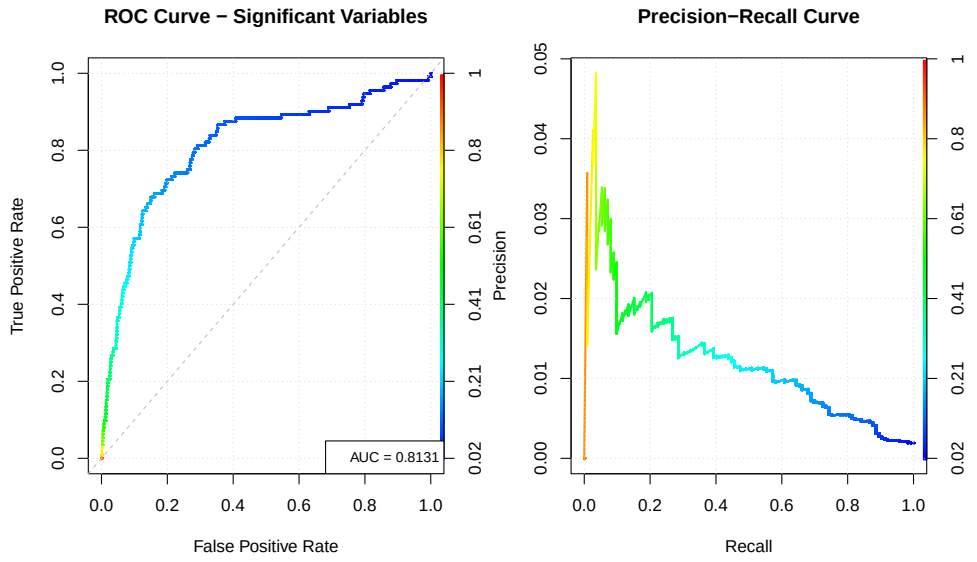


Figure 5.4: ROC Curve and Precision-Recall Curve for Logistic Regression with Balanced Class Weights and Significant Covariates

The ROC curve exhibits a smooth trade-off between true positive rate and false positive rate, with an AUC of 0.8131, indicating strong discriminative power. The Precision-Recall curve highlights the model's ability to handle the imbalanced dataset, where precision is emphasized over recall.

5.4.3 Threshold Optimization and G-Scores

The G1 and G2 scores were computed to identify the optimal thresholds for classification. These scores were smoothed for better visualization and interpretation. The results are:

- **G1-optimal threshold:** 0.0471
- **G2-optimal threshold:** 0.0538
- **Maximum G1 score:** 0.7575
- **Maximum G2 score:** 0.0784

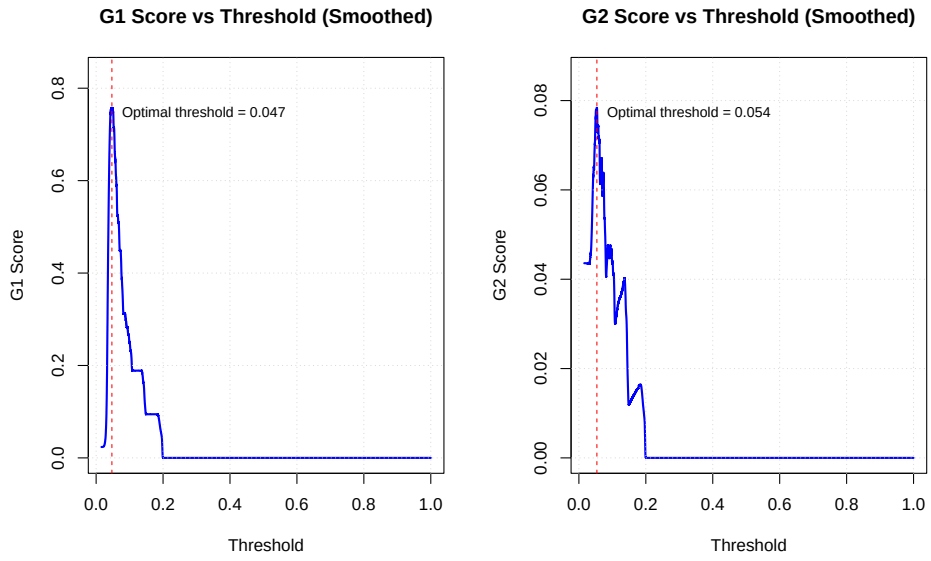


Figure 5.5: G1 and G2 Scores vs Threshold (Smoothed) for Logistic Regression with Balanced Class Weights and Significant Covariates

5.4.4 Confusion Matrix Analysis

The model's performance was evaluated using confusion matrices for the G1-optimal and G2-optimal thresholds.

Confusion Matrix at G1-optimal Threshold

Prediction	Reference = 0	Reference = 1
0 (No EPM)	8327	4
1 (EPM)	49943	108

Table 5.8: Confusion Matrix at G1-optimal Threshold (0.0471) used for Facebook EPM prediction

Key observations:

- **High Sensitivity:** The model captures most EPM events ($y = 1$).
- **Reduced Specificity:** A high number of false positives ($y = 0$) is observed.

Prediction	Reference = 0	Reference = 1
0 (No EPM)	12634	9
1 (EPM)	45636	103

Table 5.9: Confusion Matrix at G2-optimal Threshold (0.0538) used for Facebook EPM prediction

Confusion Matrix at G2-optimal Threshold

Key observations:

- **Higher Precision:** Fewer false positives are observed compared to the G1-optimal threshold.
- **Slightly Lower Sensitivity:** Fewer true positives are identified.

5.4.5 Performance Summary

The following table summarizes the key metrics for each threshold:

Metric	G1-optimal	G2-optimal
Accuracy	83.6%	89.5%
Sensitivity (Recall)	96.4%	91.9%
Specificity	13.9%	22.5%
Precision	0.2%	0.3%

5.4.6 Conclusion

The logistic regression model with significant covariates and balanced class weights demonstrates comparable performance to the full-covariate model while offering greater simplicity and interpretability. This trade-off between model complexity and performance highlights the value of feature selection in improving predictive efficiency for highly imbalanced datasets.

5.5 LASSO Regression Model

This section explores the application of the LASSO (Least Absolute Shrinkage and Selection Operator) regression model for predicting the occurrence of flash crashes ('y'). The LASSO approach effectively combines variable selection and regularization, enabling the identification of significant predictors while mitigating multicollinearity issues. Additionally, it allows for the inclusion of interaction terms, capturing complex relationships between covariates.

5.5.1 Model Specification and Feature Selection

The LASSO regression model was implemented using the `glmnet` package. The input matrix included both main effects and all pairwise interaction terms, ensuring a comprehensive exploration of covariate relationships. The penalty parameter (λ) was tuned using 10-fold cross-validation to minimize the mean squared error (MSE). The MSE curve is shown in Figure 5.6, with the optimal λ highlighted.

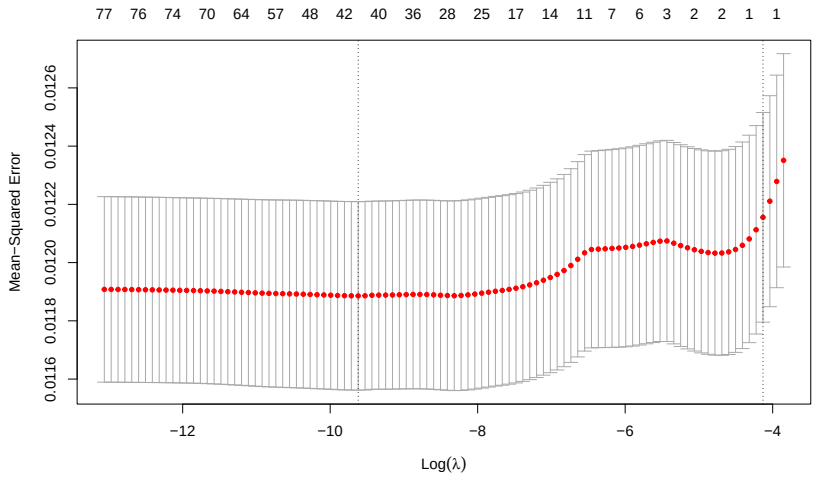


Figure 5.6: Cross-Validation for LASSO: Optimal λ Selection

At the optimal λ, LASSO selected covariates with non-zero coefficients, including both main effects and interaction terms. The main effects represent the individual influence of predictors, while interaction terms highlight combined effects of covariates. Notable selections include:

- **Main Effects:** `Price`, `BIDs1`, `spread`, `PES`, `vol_short`, `vpin`.
- **Interaction Terms:**
  - `Time:Size` — capturing the temporal variation in trade sizes.
  - `Size:spread` — linking trade size with market spread.
  - `PES:skew_short` — highlighting how Percent Effective Spread interacts with skewness.
  - `spread:vol_short` — connecting market spread and short-term volatility.

This combination of features underscores the importance of both individual and synergistic effects in predicting extreme price movements.

5.5.2 Model Evaluation and Metrics

The LASSO-selected features were used to train a logistic regression model. Key model fit metrics include:

- **Akaike Information Criterion (AIC):** 20664.68
- **Bayesian Information Criterion (BIC):** 20969
- **Log-likelihood:** -10302.34

These metrics indicate that the LASSO model achieves a balance between model complexity and predictive accuracy.

5.5.3 ROC and Precision-Recall Analysis

Figure 5.7 presents the ROC curve and Precision-Recall curve for the LASSO model. The AUC of 0.8126 suggests excellent discriminative power. The Precision-Recall curve highlights the model's robustness in handling imbalanced data, with a trade-off favoring precision at lower recall levels.

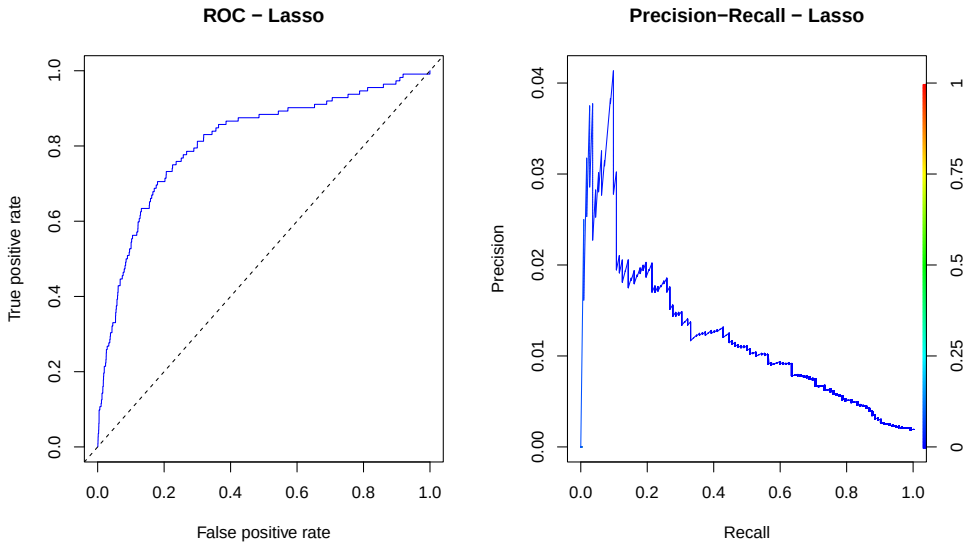


Figure 5.7: ROC and Precision-Recall Curves for LASSO Regression

5.5.4 Confusion Matrix Analysis

The LASSO model was evaluated using confusion matrices at three thresholds: default (0.5), G1-optimal (0.0021), and G2-optimal (0.003).

Confusion Matrix at Default Threshold (0.5)

Prediction	Reference = 0	Reference = 1
0 (No EPM)	58252	112
1 (EPM)	18	0

Table 5.10: Confusion Matrix at Default Threshold (0.5) used for Facebook EPM prediction

At the default threshold, the model demonstrates exceptionally high specificity (100%), correctly identifying nearly all non-EPM cases. However, this comes at the cost of missing nearly all EPM events, leading to a sensitivity of 0.2%. This threshold is ideal for scenarios where false positives must be minimized, such as resource-constrained environments where investigating false positives is costly.

Confusion Matrix at G1-optimal Threshold

Prediction	Reference = 0	Reference = 1
0 (No EPM)	46237	31
1 (EPM)	12033	81

Table 5.11: Confusion Matrix at G1-optimal Threshold (0.0131) used for Facebook EPM prediction

The G1-optimal threshold improves sensitivity dramatically to 72.3%, allowing the model to correctly identify most EPM events. However, this increase in sensitivity is accompanied by a substantial drop in specificity (26.1%), resulting in a high number of false positives. This threshold is suitable for applications where the primary objective is to maximize recall, such as in early-warning systems for potential flash crashes.

Confusion Matrix at G2-optimal Threshold

Prediction	Reference = 0	Reference = 1
0 (No EPM)	50609	42
1 (EPM)	7661	70

Table 5.12: Confusion Matrix at G2-optimal Threshold (0.0341) used for Facebook EPM prediction

The G2-optimal threshold provides a balanced approach, achieving a sensitivity of 62.5% and a specificity of 63.6%. This threshold significantly reduces false positives compared to the G1-optimal threshold while maintaining reasonable sensitivity. It is well-suited for scenarios where both precision and recall are critical, such as when the goal is to flag flash crashes for further analysis without overwhelming the system with false alarms.

5.5.5 Performance Summary

The key metrics for each threshold are summarized below:

Metric	Default (0.5)	G1-optimal (0.0021)	G2-optimal (0.003)
Accuracy	98.8%	78.9%	86.2%
Sensitivity (Recall)	0.2%	72.3%	62.5%
Specificity	100%	26.1%	63.2%
Precision	0.0%	0.7%	0.9%

5.5.6 Conclusion

The LASSO regression model effectively handles high-dimensional data by selecting a parsimonious set of predictors, including both main effects and interaction terms. The G1-optimal threshold is suited for applications requiring high recall, while the G2-optimal threshold is preferable for scenarios emphasizing precision. With an AUC of 0.8126, the model demonstrates strong discriminative ability, making it a valuable tool for identifying rare flash crash events.

5.6 Decision Tree

In this section, we analyze two decision tree models designed to predict EPMs using the methods and variables highlighted in the previous sections. Both models are trained on a class-imbalanced dataset, where $y = 1$ (flash crash or EPM) is the rare class and $y = 0$ is the majority class. To address this imbalance, a synthetic oversampling method is applied using `ROSE`, thus generating a balanced training set.

The first decision tree uses a higher complexity parameter (`cp`) and simpler splitting controls, whereas the second model employs a lower `cp` and additional parameters like `maxdepth` and cross-validation (`xval`). We examine the pruned trees, confusion matrices, and variable importance to gauge model performance and understand the impact of the parameters.

5.6.1 First Decision Tree Model (Simpler Configuration)

After balancing the training data via `ROSE`, we create a decision tree with `cp` $= 0.01$, `minsplit` $= 20$, and `minbucket` $= 7$. This initial choice of `cp` encourages the tree to stop growing sooner, aiming to reduce overfitting. The model includes predictors such as `Time`, `Size`, `Price`, `ASKs1`, `BIDs1`, `Logreturn`, `spread`, `PES`, `vol_short`, `kurt_short`, `skew_short`, and `vpin`. After training on the balanced set, we use the test data (unseen and still imbalanced) to evaluate performance.

Complexity Parameter (*cp*) and Tree Structure.
The complexity parameter controls the size of the tree by penalizing each split. A higher *cp* leads to earlier stopping and a simpler tree. In this first model, `cp` = 0.01 quickly prunes away less important splits, yielding a more compact structure. The resulting tree reveals the most influential splits around short-term volatility (`vol_short`), market spread (`spread`), and price-related features.

Confusion Matrix and Performance Metrics.
Upon predicting on the original (imbalanced) test set, the confusion matrix is:

Prediction	**Reference = 0**	**Reference = 1**
0 (No EPM)	56219	79
1 (EPM)	2051	33

Table 5.13: Confusion Matrix for the First Decision Tree Model ($cp = 0.01$)

In this table, the minority class ($y = 1$) is still overwhelmed by the large number of majority cases ($y = 0$). The overall accuracy is 0.9635, which is mainly driven by correctly identifying the majority class. By examining further metrics:

- The *Sensitivity* (recall for class $y = 0$) is ≈ 0.9648, indicating the model correctly classifies 96.48% of true non-EPM cases.
- The *Specificity* for detecting EPM events (class $y = 1$) is notably lower (≈ 0.2946), reflecting the challenge in identifying the rare positive class.
- Precision (for class $y = 0$) is extremely high (≈ 0.9986), owing to the imbalance.
- The F1 score (considering $y = 0$ as the positive class in this software output) is around 0.9814.

In most financial applications, the specificity (how well we detect the actual EPM instances) is of particular importance. Although the accuracy is high, the relatively low specificity underscores the difficulty in catching rare flash crashes.

Variable Importance.
The decision tree quantifies the relevance of each predictor in creating splits. Sorted in descending order of importance, the top contributors are:

$$\texttt{vol\_short} > \texttt{spread} > \texttt{PES} > \texttt{Price} > \texttt{Logreturn} > \texttt{Time}.$$

These features repeatedly appear in the upper splits, emphasizing how short-term volatility, the bid-ask spread, and certain price metrics govern the detection of abrupt price swings.

5.6.2 Second Decision Tree Model (More Complex Configuration)

To capture more nuanced interactions in the data, we develop a second model with `cp` = 0.001, `minsplit` = 5, `minbucket` = 3, `maxdepth` = 15, and `xval` = 10. This configuration allows deeper splits, smaller leaf nodes, and cross-validation to help generalize the complexity parameter. We again use the `ROSE` balanced training set to train the tree, but we now carefully prune it based on the cross-validation error estimates (the `printcp` and `plotcp` outputs).

Cross-Validation and Pruning

This model incorporates cross-validation to guide pruning, ensuring that the tree does not overfit while capturing relevant patterns. The pruning process relies on error metrics, with splits retained only if they meaningfully reduce classification error. Figure 5.8 illustrates the relationship between tree complexity and cross-validation error, highlighting the optimal complexity parameter.

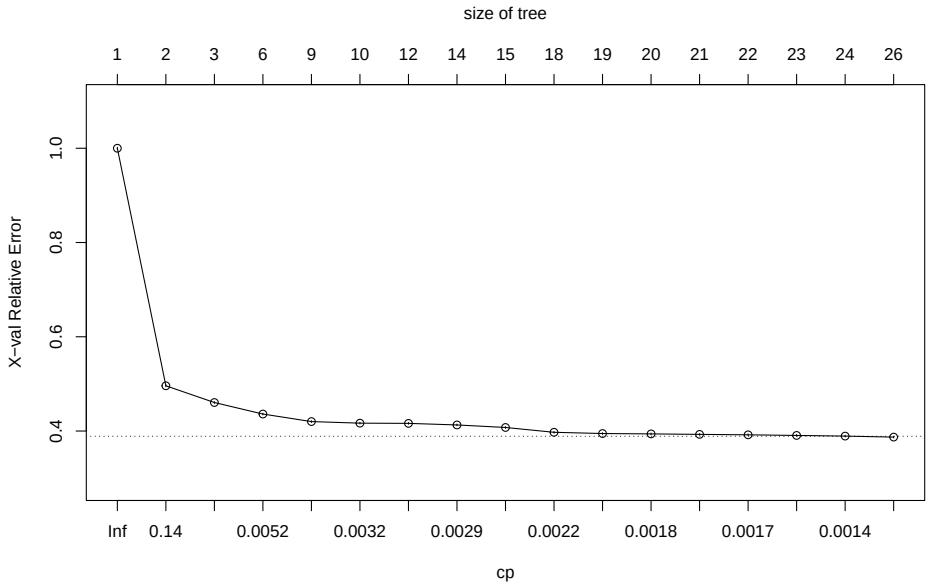


Figure 5.8: Cross-Validation Plot for Complexity Parameter Selection

Pruning at the optimal complexity parameter yields a balanced tree that captures important interactions while avoiding overfitting. This ensures that the model generalizes well to unseen data.

Refined Threshold and Confusion Matrix.
Following pruning, we generate probability predictions on the test set. Instead of defaulting to a 0.5 cutoff, we adjust the threshold to 0.4 to balance the trade-off between detecting EPMs and limiting false alarms. The confusion matrix with the custom threshold is:

Prediction	**Reference = 0**	**Reference = 1**
0 (No EPM)	56620	80
1 (EPM)	1650	32

Table 5.14: Confusion Matrix for the Second Decision Tree Model ($cp = 0.001$, threshold = 0.4)

The overall accuracy is around 0.9704. As before, the large majority of cases are correctly predicted as $y = 0$. Importantly, the specificity (≈ 0.2857) remains a challenge, illustrating how the imbalance plus the relatively small proportion of true EPM events affects detection. Nonetheless, the tree does slightly better in capturing positives than the first model at a comparable threshold. Notable metrics include:

$$\text{Sensitivity (class 0)} \approx 0.9717, \quad \text{Precision (class 0)} \approx 0.9986,$$
$$\text{F1 score (class 0)} \approx 0.9850, \quad \text{Balanced Accuracy} \approx 0.6287.$$

Here, we also see a slightly higher recall for the negative class ($y = 0$) and a marginally better detection of $y = 1$.

Variable Importance.
Once pruned, the most influential variables remain largely consistent with the simpler tree. The top contributors, in descending order, are:

$$\texttt{vol\_short} > \texttt{spread} > \texttt{PES} > \texttt{Price} > \texttt{Logreturn} > \texttt{Time} > \texttt{ASKs1} > \texttt{Size},$$

with minor roles for `vpin`, `kurt_short`, `skew_short`, and `BIDs1`. The elevated roles of short-term volatility and spread reaffirm the importance of liquidity and recent price dynamics in predicting extreme

price changes. Figure 2 displays the pruned tree, illustrating its complexity and the key splits driving predictions.

Discussion and Practical Considerations.
Comparing the two models underscores the interplay between tree depth, pruning, and class imbalance. The second, more complex approach tends to fit the data more closely, with `xval` controlling overfitting. Introducing a custom prediction threshold (e.g., 0.4) can improve the capture of $y = 1$ cases, although specificity for the minority class remains an obstacle. In practice, setting thresholds should hinge on real-world costs: false alarms might be preferable if missing a true crash is extremely costly.

Overall, both tree models demonstrate high accuracy on the majority class. However, the relatively low specificity for $y = 1$ suggests that decision trees, even when balanced and pruned, require careful threshold tuning or possibly ensemble methods (e.g., random forests) to robustly detect rare events. Regardless, insights from the variable importance confirm that short-term volatility, the bid-ask spread, and price signals are pivotal for anticipating EPMs.

5.7 Random Forest

In this section, we employ a Random Forest (RF) model to predict the occurrence of Extreme Price Movements (EPM). Random Forests, an ensemble technique built from multiple decision trees, can capture complex interactions among covariates and are less prone to overfitting compared to single decision trees. To address the strong class imbalance in the dataset, synthetic oversampling is performed via the `ROSE` method on the training split. The following covariates are included in the model:

`Time`, `Size`, `Price`, `ASKs1`, `BIDs1`, `Logreturn`, `spread`, `PES`, `vol_short`, `kurt_short`, `skew_short`, `vpin`.

5.7.1 Model Specification and Training

The Random Forest is trained with 500 trees (`ntree` = 500) and three variables tried at each split (`mtry` = 3). The `importance` parameter is set to `TRUE`, allowing us to measure the contribution of each covariate via the Mean Decrease in Gini. The training data are further balanced by sampling equally from both classes through `sampsize` = $\{5000, 5000\}$, thus reducing the bias toward the majority class. Once trained, the out-of-bag (OOB) estimate of the error rate is reported as approximately 15.29%, reflecting the model's ability to classify the balanced training data.

5.7.2 Prediction and Confusion Matrix

To evaluate the model, predictions are generated on the unseen (and imbalanced) test set. By default, the predicted class is assigned according to a 0.5 probability threshold ($\hat{p}(\text{EPM}) \geq 0.5$ implies a predicted EPM). Table 5.15 shows the resulting confusion matrix for the test set.

Prediction	**Reference = 0**	**Reference = 1**
0 (No EPM)	57027	84
1 (EPM)	1243	28

Table 5.15: Confusion Matrix for the Random Forest on the Test Set (Threshold = 0.5)

Most observations of the majority class (no EPM) are predicted correctly, but several true EPMs end up in the non-EPM category. The performance metrics from this confusion matrix are:

- *Accuracy*: 0.9773
- *Sensitivity (Recall, class = 0)*: 0.9787
- *Specificity (EPM detection rate, class = 1)*: 0.2500

- *Precision (class = 0)*: 0.9985
- *F1 Score (class = 0)*: 0.9885
- *Balanced Accuracy*: 0.6143

We note that, although the accuracy is high due to the strong class imbalance, the specificity (ability to detect the minority EPM class) remains relatively low. This underscores the challenge of identifying rare events in imbalanced scenarios.

5.7.3 Threshold Adjustment and ROC Analysis

The model outputs a probability for each class, enabling threshold tuning to control the trade-off between sensitivity and specificity. Table 5.16 illustrates the variation of specificity and sensitivity as the probability threshold changes from 0.1 to 0.9.

Threshold	**Specificity (class = 1)**	**Sensitivity (class = 0)**
0.1	0.8036	0.4481
0.2	0.5893	0.6500
0.3	0.4375	0.9059
0.4	0.3482	0.9575
0.5	0.2500	0.9788
0.6	0.1429	0.9898
0.7	0.0714	0.9963
0.8	0.0268	0.9991
0.9	0.0000	0.9999

Table 5.16: Sensitivity and specificity of the Random Forest at different probability thresholds

Lowering the threshold (below 0.5) increases the detection of rare EPM events but also produces more false positives. Conversely, a higher threshold reduces false positives at the cost of missing more actual EPMs. The receiver operating characteristic (ROC) curve yields an AUC of approximately 0.863, indicating that the model has a good overall discriminative capability, though threshold tuning is essential to balance practical needs in real trading environments.

5.7.4 Variable Importance and Interpretability

Random Forests compute various measures of variable importance. Table 5.17 shows a subset of the importance scores, namely the Mean Decrease in Accuracy and Mean Decrease in Gini, for the top predictors in the model. The Mean Decrease in Accuracy indicates how much accuracy the model would lose by excluding a given feature, while the Mean Decrease in Gini summarizes the total impurity reduction contributed by each variable. These metrics highlight which covariates the forest relies on most for splitting nodes.

Feature	Mean Decrease Accuracy	Mean Decrease Gini
vol_short	293.10	1190.98
spread	214.22	819.46
PES	142.37	576.90
Price	140.95	459.86
Size	130.47	372.55
Logreturn	123.82	414.60
ASKs1	116.50	291.82
Time	94.36	262.35
BIDs1	78.36	227.82
skew_short	42.14	193.12
kurt_short	29.57	190.54

Table 5.17: Variable importance for the Random Forest model

Short-term volatility (`vol_short`) emerges as the single most critical driver, followed by the bid-ask spread (`spread`) and the Percent Effective Spread (`PES`). These findings resonate with the notion that extreme price movements are typically preceded by heightened volatility, widening spreads, and abnormal flows of buy or sell pressure. In practical trading scenarios, these signals can serve as early indicators of instability. Variables such as `Price`, `Size`, and `Logreturn` also show notable influence, underscoring the importance of price levels and recent returns in capturing abrupt market shifts.

5.7.5 Discussion and Conclusion

The Random Forest model, combined with ROSE-based balancing in the training set, achieves a high overall accuracy ($\sim 97.7\%$) on the imbalanced test data. Yet, because the event of interest (EPM) is so rare, this accuracy is largely driven by correct classifications of the majority class. The specificity for detecting actual crashes remains modest at the standard 0.5 threshold, emphasizing the importance of considering metrics beyond raw accuracy.

By adjusting the classification threshold, practitioners can dial the trade-off between missing fewer true crashes (increasing EPM sensitivity) and reducing false alarms. When the real-world cost of missing a flash crash is prohibitively high, lowering the threshold may be justified, even at the expense of some false positives. Alternatively, cost-sensitive learning approaches or boosting methods can further refine crash detection while maintaining manageable false-alarm rates.

Overall, the Random Forest shows promise for capturing nonlinear interactions among time-series features. The variable importance analysis suggests that short-term volatility, market spread, and order flow pressure are significant drivers of abrupt and extreme price changes. In live trading scenarios, such signals could help highlight imminent price instability, provided the classification threshold is tuned to match organizational risk tolerance.

5.8 Neural Network

In this section, we explore the performance of a feed-forward neural network for predicting Extreme Price Movements (EPM). Neural networks are powerful universal function approximators capable of capturing complex, nonlinear relationships among features. Here, we employ a straightforward architecture with two hidden layers consisting of 8 and 4 neurons, respectively. Despite its relatively simple structure, this network demonstrates how deep learning techniques can be adapted to the EPM detection problem.

5.8.1 Model Specification and Training

To prepare the data for the neural network, all numeric predictors in the training and test sets are standardized (mean zero and unit variance) to improve convergence and stability. The target variable, `y`, is converted to numeric values as required by the `neuralnet` package.

- **Architecture:** The network has two hidden layers: the first layer has 8 neurons, and the second layer has 4 neurons.
- **Activation Function:** A logistic (sigmoid) function is used for all hidden neurons.
- **Output:** Since we are dealing with a binary classification (EPM vs. No EPM), a logistic output neuron is employed, enabling the network to output probabilities of each class.
- **Regularization:** Although not explicitly used here, the `threshold` and `stepmax` parameters help control training stability. We set `threshold` $= 0.1$ and `stepmax` $= 10^6$.
- **Optimization:** The training uses resilient backpropagation (by default in `neuralnet`) to minimize the error function.

The neural network is fit on the scaled training data. After convergence, we visualize the learned architecture in Figure 3, which shows the input, hidden, and output layers.

5.8.2 Prediction and Confusion Matrix

Once trained, the model is used to produce predicted probabilities on the test set. A default threshold of 0.5 is then applied: if $\hat{p}(\text{EPM}) \geq 0.5$, the instance is labeled as EPM (class = 1); otherwise, it is labeled as No EPM (class = 0). Table 5.18 presents the resulting confusion matrix.

Prediction	**Reference = 0**	**Reference = 1**
0 (No EPM)	58263	111
1 (EPM)	7	1

Table 5.18: Confusion Matrix for the Neural Network on the Test Set (Threshold = 0.5)

Notably, most instances of the majority class are predicted correctly. However, out of 112 true EPM cases (rows for Reference = 1), only one is predicted as EPM. This low detection rate highlights the challenge of modeling highly imbalanced data in a neural network framework without additional balancing or threshold tuning steps.

5.8.3 Performance Metrics

Table 5.19 details the primary performance metrics derived from the confusion matrix. We focus on accuracy, precision, recall (sensitivity), and the F1 score:

Metric	**Value**
Accuracy	0.9980
Precision (class = 1)	0.1250
Recall (class = 1)	0.0089
F1 Score (class = 1)	0.0167

Table 5.19: Key Performance Metrics for the Neural Network

Although the overall accuracy is impressive (99.80%), it is heavily influenced by the large volume of negative (No EPM) cases. The recall (or sensitivity) for true EPM events is merely 0.89%, meaning almost all EPM examples are misclassified. Consequently, the F1 score for the EPM class is also very low (1.67%). This pattern is symptomatic of severe class imbalance and highlights the importance of scrutinizing metrics beyond accuracy, especially for rare-event detection.

5.8.4 Interpretation and Discussion

The performance of the neural network, as indicated by the confusion matrix and associated metrics, reveals significant limitations in its ability to detect EPM events. While the model achieves a high overall accuracy (99.80%), this is largely driven by its ability to correctly classify the majority class (No EPM). The recall for the minority class (EPM) is extremely low (0.89%), indicating that the network fails to identify most true EPM events. This issue is symptomatic of imbalanced data and suggests that the current network architecture may lack the capacity to capture the nuanced patterns needed for rare event detection.

A promising direction to address these limitations is to develop more complex neural network architectures. Adding more neurons to the hidden layers and increasing the depth of the network could enable the model to learn richer representations of the data. Deeper architectures are better equipped to capture subtle, nonlinear relationships between features, which are likely critical for detecting the precursors of EPMs.

Furthermore, the introduction of Long Short-Term Memory (LSTM) networks in future analyses is a logical step. LSTMs are specifically designed to handle sequential data and temporal dependencies, which are inherent in financial time-series datasets. By leveraging the sequential nature of the covariates, LSTMs can uncover patterns that static feed-forward networks cannot. For instance, sudden changes in short-term volatility or widening spreads might signal an impending EPM, but these signals may only become apparent when viewed in the context of recent historical data.

Finally, incorporating techniques such as threshold tuning, class balancing (e.g., oversampling or weighted loss functions), and regularization in these more advanced architectures could further enhance the model's ability to generalize and detect rare events.

5.9 Enhanced Neural Network Analysis

Building on the initial neural network model, we now employ a more sophisticated architecture to address the limitations identified in the previous section. By increasing the complexity of the network and leveraging Python's `scikit-learn` library for its powerful neural network tools, we aim to improve the model's capacity to detect Extreme Price Movements (EPMs). This analysis marks the transition from simpler feed-forward architectures to progressively more advanced methods, ultimately paving the way for sequence modeling with Long Short-Term Memory (LSTM) networks.

5.9.1 Model Specification and Training

Data Preprocessing

The training and test datasets are preprocessed by scaling all numeric features to ensure zero mean and unit variance. This normalization step is essential for neural networks, as it improves convergence and helps avoid issues stemming from the magnitude differences among features. The target variable (`y`) is converted into integers to comply with the requirements of the `MLPClassifier` in Python's `scikit-learn` library.

Architecture and Parameters

The enhanced neural network employs the following architecture and training parameters:

- **Architecture:** Two hidden layers consisting of 128 and 64 neurons, respectively.
- **Activation Function:** Rectified Linear Unit (ReLU), chosen for its ability to model non-linear relationships while avoiding vanishing gradients.

- **Optimizer:** Adam optimizer, which combines the benefits of adaptive learning rates and momentum for efficient training.
- **Maximum Iterations:** The training process is capped at 5000 iterations to ensure convergence.

The training dataset is fed into the model, and the training process is conducted iteratively, minimizing the cross-entropy loss function.

5.9.2 Prediction and Confusion Matrix

Once trained, the model generates predictions for the test dataset. A default threshold of 0.5 is applied, classifying instances as EPM ($\hat{p}(\text{EPM}) \geq 0.5$) or No EPM ($\hat{p}(\text{EPM}) < 0.5$). The resulting confusion matrix is presented in Table 5.20.

Prediction	**Reference = 0**	**Reference = 1**
0 (No EPM)	57766	109
1 (EPM)	504	3

Table 5.20: Confusion Matrix for the Enhanced Neural Network on the Test Set

Despite the increased model complexity, the detection of EPM events remains extremely limited. Out of 112 true EPM cases, only 3 are correctly classified, while 504 false positives are generated.

5.9.3 Performance Metrics

The following performance metrics are computed based on the confusion matrix:

- **Accuracy:** 0.9895, indicating that 98.95% of the total test set is classified correctly.
- **Precision (class = 1):** 0.0059, reflecting the low reliability of EPM predictions.
- **Recall (class = 1):** 0.0268, highlighting the inability to detect most true EPMs.
- **F1 Score (class = 1):** 0.0097, representing the harmonic mean of precision and recall.

These metrics, summarized in Table 5.21, indicate that while the overall accuracy is high, the model struggles with the minority class, similar to the simpler neural network.

Metric	**Value**
Accuracy	0.9895
Precision (class = 1)	0.0059
Recall (class = 1)	0.0268
F1 Score (class = 1)	0.0097

Table 5.21: Key Performance Metrics for the Enhanced Neural Network

5.9.4 Interpretation and Discussion

This analysis reveals that while the enhanced neural network offers a slight improvement, it is insufficient to effectively detect rare events like EPMs. The challenges encountered underscore the need for more sophisticated architectures and targeted approaches to handle class imbalance and leverage temporal patterns.

The next logical step is to transition to LSTM networks, which are better suited for time-series data and can uncover dependencies that static feed-forward networks may miss. By combining the power of LSTM architectures with appropriate class balancing techniques, we anticipate significant improvements in detecting and predicting EPM events, setting the stage for a robust and reliable model.

5.10 First LSTM Neural Network

We now transition to a Long Short-Term Memory (LSTM) network, a specialized type of recurrent neural network (RNN) designed to capture temporal dependencies in sequential data. Unlike previous models, LSTMs are particularly well-suited for financial time series, as they can utilize historical context to make predictions. This section explores the performance of an initial LSTM architecture with one hidden layer and its ability to detect Extreme Price Movements (EPMs).

5.10.1 Model Specification and Training

Data Preparation

To adapt the data for sequential modeling, each feature set is transformed into overlapping sequences of 10 time steps. The target variable (`y`) is aligned with the last time step of each sequence. This preprocessing step results in the following shapes:

- **Training Set:** X_{train} with shape $(187981, 10, 11)$ and y_{train} with shape $(187981,)$.
- **Test Set:** X_{test} with shape $(58372, 10, 11)$ and y_{test} with shape $(58372,)$.

To address the class imbalance in the training set, class weights are computed based on the frequency of each class. These weights ensure that the model penalizes misclassifications of the minority class (EPM) more heavily during training:

$$\text{Class Weights:}\{0 : 0.506, 1 : 39.962\}.$$

Architecture and Training

The LSTM network is constructed using the `TensorFlow/Keras` library with the following architecture:

- **Input Layer:** Processes sequences of 10 time steps, each with 11 features.
- **LSTM Layer:** 50 units to capture temporal dependencies in the data.
- **Dense Layer:** A single output neuron with a sigmoid activation function for binary classification.

The model is compiled using the Adam optimizer, a binary cross-entropy loss function, and accuracy as the primary metric. Training is performed over 20 epochs with a batch size of 32.

5.10.2 Prediction and Confusion Matrix

After training, the model generates predictions for the test set. A threshold of 0.5 is applied to classify instances as EPM ($\hat{p}(\text{EPM}) \geq 0.5$) or No EPM ($\hat{p}(\text{EPM}) < 0.5$). The confusion matrix is presented in Table 5.22.

Prediction	Reference = 0	Reference = 1
0 (No EPM)	48694	47
1 (EPM)	9566	65

Table 5.22: Confusion Matrix for the LSTM on the Test Set

The results show that while the model correctly identifies a greater number of EPM events (65 out of 112) compared to previous neural networks, it also generates a large number of false positives (9566).

5.10.3 Performance Metrics

The following metrics are derived from the confusion matrix:

- **Accuracy:** 0.8399, reflecting the overall proportion of correctly classified instances.
- **Precision (class = 1):** 0.0067, indicating a low reliability of EPM predictions.
- **Recall (class = 1):** 0.5804, showing that the model captures 58.04% of actual EPM events.
- **F1 Score (class = 1):** 0.0132, highlighting the trade-off between precision and recall.

These metrics are summarized in Table 5.23.

Metric	Value
Accuracy	0.8399
Precision (class = 1)	0.0067
Recall (class = 1)	0.5804
F1 Score (class = 1)	0.0132

Table 5.23: Performance Metrics for the LSTM

5.10.4 Interpretation and Discussion

This first LSTM implementation demonstrates significant potential, particularly in its ability to recall a substantial proportion of EPM events. However, the low precision and high false positive rate highlight the need for further refinements. Future iterations will focus on increasing model complexity and exploring advanced techniques to balance the trade-off between precision and recall. These findings lay the groundwork for building a more sophisticated LSTM architecture to achieve reliable and robust EPM detection.

5.11 Advanced LSTM Neural Network Analysis

Having observed the efficacy of a basic LSTM model for capturing temporal dependencies, we now extend our approach with a more complex LSTM architecture. This section presents the performance of a deep LSTM network with two LSTM layers, each accompanied by dropout for regularization. By increasing model complexity, we seek to further enhance detection of Extreme Price Movements (EPMs).

5.11.1 Model Specification and Training

Data Preparation

As with the simpler LSTM model, data is preprocessed into overlapping sequences of 10 time steps. Class weights are computed to address the significant imbalance between the majority (No EPM) and minority (EPM) classes:

$$\text{Class Weights:}\{0 : 0.506, 1 : 39.962\}.$$

Architecture and Training

The advanced LSTM architecture incorporates multiple LSTM layers and dropout for regularization:

- **Input Layer:** Processes sequences of 10 time steps, each with 11 features.
- **First LSTM Layer:** 64 units, outputs a sequence for the next layer (`return_sequences=True`).
- **Dropout Layer:** A 30% dropout rate reduces overfitting.
- **Second LSTM Layer:** 32 units, outputs a single vector (`return_sequences=False`).

- **Dropout Layer:** A second 30% dropout rate for regularization.
- **Dense Layer:** A single output neuron with a sigmoid activation function for binary classification.

The model is trained using the Adam optimizer, binary cross-entropy loss, and accuracy as the primary metric. Training is performed over 20 epochs with a batch size of 32.

5.11.2 Prediction and Confusion Matrix

Predictions are generated for the test set, with a threshold of 0.5 applied to classify instances as EPM ($\hat{p}(\text{EPM}) \geq 0.5$) or No EPM ($\hat{p}(\text{EPM}) < 0.5$). The confusion matrix is presented in Table 5.24.

Prediction	**Reference = 0**	**Reference = 1**
0 (No EPM)	47130	37
1 (EPM)	11130	75

Table 5.24: Confusion Matrix for the Advanced LSTM on the Test Set

The advanced model identifies more EPM events (75 out of 112) than the simpler LSTM but at the cost of a significant increase in false positives (11130).

5.11.3 Performance Metrics

The following metrics are derived from the confusion matrix:

- **Accuracy:** 0.8069, reflecting the overall proportion of correctly classified instances.
- **Precision (class = 1):** 0.0067, indicating a low reliability of EPM predictions.
- **Recall (class = 1):** 0.6696, showing that the model captures 66.96% of actual EPM events.
- **F1 Score (class = 1):** 0.0133, highlighting the trade-off between precision and recall.

These metrics are summarized in Table 5.25.

Metric	**Value**
Accuracy	0.8069
Precision (class = 1)	0.0067
Recall (class = 1)	0.6696
F1 Score (class = 1)	0.0133

Table 5.25: Performance Metrics for the Advanced LSTM

5.11.4 Interpretation and Discussion

This more complex LSTM architecture confirms the potential for recurrent models to identify short-term patterns indicative of EPMs, achieving significantly higher recall than the earlier approaches. However, the sizeable number of false positives curtails the model's practical usability in high-stakes trading scenarios. Future work may focus on advanced regularization, customized loss functions, or ensemble methods to strike a more desirable balance between capturing critical events and managing false alarms. These outcomes demonstrate that while adding complexity can boost minority-class detection, it also emphasizes the importance of careful tuning and consideration of real-world costs. In practice, the decision threshold would likely be calibrated to match an organization's risk tolerance for missing a true event versus tolerating false alarms, ensuring the LSTM remains a valuable tool in the broader EPM detection pipeline.

Chapter 6

Conclusion

6.1 Logistic Regression Models

This section synthesizes the results from the five logistic regression models applied to detect Extreme Price Movements (EPMs). Each model was tailored to address specific challenges, such as class imbalance, model complexity, and feature selection. Through an array of approaches, logistic regression provided a robust foundation for analyzing predictors and assessing their influence on EPM detection.

6.1.1 Performance Overview

The five models explored in this study include:

1. **Full Logistic Regression Model:** Incorporates all covariates to provide a comprehensive baseline.
2. **Selected Covariate Model:** Focuses on predictors with statistically significant coefficients ($p \leq 0.05$).
3. **Balanced Class Weights Model:** Adjusts class weights to account for dataset imbalance.
4. **Balanced Weights with Selected Covariates:** Combines significant predictors with balanced weights.
5. **LASSO Regression Model:** Uses regularization to select predictors and interaction terms, reducing multicollinearity.

The performance of these models is summarized in Table 6.1.

Metric	Full Model	Selected Covariates	Balanced Weights	Balanced & Selected	LASSO
AIC	21221.2	21221.2	20345.7	20345.7	20664.7
BIC	21312.5	21312.5	20467.9	20467.9	20969.0
AUC	0.798	0.797	0.813	0.813	0.812
Accuracy (0.5)	99.8%	99.8%	83.6%	83.6%	98.8%
Sensitivity (0.5)	0.0%	0.0%	96.4%	96.4%	0.2%
Specificity (0.5)	100%	100%	13.9%	13.9%	100%

Table 6.1: Comparison of Logistic Regression Models for EPM Detection

6.1.2 Key Findings

Full Logistic Regression Model

The full model served as a baseline, incorporating all predictors. Despite its comprehensive nature, it struggled with imbalanced data, failing to identify EPMs ($y = 1$) at the default threshold of 0.5. However, the AUC of 0.798 indicated reasonable discriminative power, suggesting that better threshold tuning or balancing techniques could improve performance.

Selected Covariate Model

This model, which included only statistically significant covariates ($p \leq 0.05$), achieved performance comparable to the full model. By reducing complexity, it provided a more interpretable approach without sacrificing discriminative ability, maintaining an AUC of 0.797. Feature selection highlighted the importance of covariates like `Time`, `Price`, `Logreturn`, `spread`, and `vol_short`.

Balanced Class Weights Model

By addressing the class imbalance, this model improved sensitivity to 96.4% at the G1-optimal threshold. The trade-off was a significant drop in specificity (13.9%) due to an increase in false positives. The AUC of 0.813 underscored the benefits of balancing techniques for imbalanced datasets.

Balanced Weights with Selected Covariates

Combining balanced weights with significant predictors maintained the AUC of 0.813 while offering a streamlined model. This approach demonstrated the value of combining feature selection with class balancing, achieving similar performance to the full balanced model but with fewer predictors.

LASSO Regression Model

The LASSO model leveraged regularization to identify key predictors and interactions, providing a parsimonious and interpretable solution. With an AUC of 0.812, it achieved comparable performance to other models. The inclusion of interaction terms, such as `spread:vol_short`, highlighted the importance of covariate relationships in detecting EPMs. Threshold tuning allowed for customization based on precision-recall trade-offs.

6.1.3 General Observations

Across all logistic regression models:

- **Threshold Optimization:** Default thresholds (0.5) often failed to detect EPMs due to dataset imbalance. G1- and G2-optimal thresholds improved sensitivity and precision, demonstrating the importance of tuning for specific applications.
- **Trade-offs:** Models with balanced class weights prioritized recall, suitable for early warning systems. In contrast, models emphasizing precision, such as LASSO, were better suited for minimizing false positives in cost-sensitive environments.
- **Feature Importance:** Short-term volatility (`vol_short`), market spread (`spread`), and price metrics (`Price`, `Logreturn`) consistently emerged as key predictors, reflecting their critical role in EPM detection.
- **Interpretability vs. Complexity:** Simpler models (e.g., selected covariates) offered better interpretability without significant loss of performance, while more complex models (e.g., LASSO) captured nuanced relationships through interaction terms.

6.1.4 Discussion and Future Directions

The logistic regression models explored in this study provide a clear foundation for understanding the relationships between key predictors and the occurrence of Extreme Price Movements (EPMs). The results highlight the model's capacity to handle diverse challenges, including class imbalance, feature selection, and interpretability. By varying approaches from full models with all covariates to regularized models like LASSO important insights into the data and predictors were obtained.

One critical observation is that logistic regression's performance is heavily influenced by the imbalanced nature of the dataset. While metrics such as the Area Under the Curve (AUC) provide a broad measure

of performance, threshold optimization proved essential for tailoring models to specific applications. For instance, the G1-optimal threshold emphasized sensitivity, making it suitable for early-warning systems, while the G2-optimal threshold improved precision, reducing false positives in scenarios with high misclassification costs. These findings underscore the need for domain-specific considerations when deploying logistic regression in real-world scenarios.

Another notable finding is the consistent importance of features such as short-term volatility (`vol_short`), market spread (`spread`), and price-related variables (`Price`, `Logreturn`). These predictors emerged repeatedly as significant contributors across all models, reaffirming their role as key indicators of market instability. Moreover, the LASSO model's ability to identify interaction terms, such as the relationship between market spread and short-term volatility, suggests that nonlinear relationships may play a crucial role in predicting EPMs. These findings open the door to future research into advanced feature engineering and interaction modeling to further refine predictions.

Despite its strengths, logistic regression has limitations. The reliance on linear relationships between predictors and outcomes may restrict its ability to capture complex patterns inherent in financial markets. The results, particularly from the LASSO model, suggest that while logistic regression performs well in identifying significant predictors, it may not fully capture the nonlinear dynamics of flash crashes. Additionally, although class balancing techniques improved sensitivity, the trade-off between recall and precision highlights the need for more sophisticated methods to address class imbalance without inflating false positives.

Looking ahead, the integration of logistic regression with ensemble techniques or nonlinear models could provide a more robust approach to EPM detection. For example, hybrid models that combine logistic regression with machine learning methods like Random Forests or Neural Networks could leverage the strengths of each approach. Furthermore, time-series-specific extensions, such as logistic regression with autoregressive terms or state-space models, may enhance temporal prediction capabilities.

Future research should also explore the impact of external market factors, such as macroeconomic indicators or liquidity measures, to contextualize EPM events. Incorporating these external variables could provide a more holistic view of market behavior, potentially improving model performance and interpretability. Additionally, adaptive thresholding techniques that dynamically adjust based on market conditions could offer a more responsive approach to decision-making, especially in volatile trading environments.

In conclusion, while logistic regression models offer a solid baseline for EPM detection, their limitations highlight the need for continuous methodological advancement. By integrating insights from this analysis with more sophisticated techniques, future studies can build more robust and versatile models to address the complex challenges of predicting extreme financial events.

6.2 Decision Trees and Random Forest Models

This section synthesizes the findings from the decision tree and random forest models applied to detect EPMs. These methods highlight different facets of tree-based learning for imbalanced classification tasks, with decision trees providing simplicity and interpretability, while random forests offer enhanced predictive performance through ensemble methods.

6.2.1 Decision Trees: Simplicity and Interpretability

The two decision tree models explored in this analysis underscore the trade-offs between simplicity, complexity, and model performance. Both models were trained on a balanced dataset created via the `ROSE` method and evaluated on the original imbalanced test set.

Simpler Decision Tree (`cp` = 0.01)

The first decision tree was designed with a higher complexity parameter ($cp = 0.01$), limiting the tree depth to prevent overfitting. This approach resulted in a compact tree structure, emphasizing interpretability. Key findings include:

- **Performance Metrics:** The model achieved an overall accuracy of 96.35%, driven primarily by its ability to classify the majority class (No EPM). However, its specificity for detecting EPM events was only 29.46%, reflecting the difficulty in identifying the minority class.
- **Variable Importance:** Features such as `vol_short`, `spread`, and `PES` emerged as the most important predictors. These variables dominated the top splits in the tree, highlighting their critical role in explaining price volatility and liquidity imbalances that precede EPMs.

While the simpler decision tree provided clear insights into feature importance and decision pathways, its limited specificity suggested the need for a more nuanced approach to capture rare events.

More Complex Decision Tree (`cp` = 0.001)

To improve on the limitations of the simpler tree, a second model was developed with a lower complexity parameter ($cp = 0.001$) and additional hyperparameters, such as `maxdepth`, `minsplit`, and `xval`. Cross-validation was used to optimize pruning, ensuring that the model generalized well to unseen data. Key results include:

- **Performance Metrics:** The refined tree achieved an accuracy of 97.04% and a slightly higher balanced accuracy compared to the simpler tree. Specificity for detecting EPM events remained modest at 28.57%, though the model captured more nuanced interactions among features.
- **Threshold Tuning:** Adjusting the classification threshold to 0.4 resulted in marginal gains in detecting EPMs, illustrating the importance of balancing sensitivity and specificity in rare event detection.
- **Variable Importance:** The top contributors remained consistent with the simpler tree, with `vol_short`, `spread`, and `PES` leading in importance. Additional features, such as `Size` and `ASKs1`, also played minor roles.

The more complex decision tree demonstrated the benefits of deeper splits and optimized pruning, capturing additional patterns while maintaining interpretability. However, its performance on the minority class remained constrained, emphasizing the inherent challenges of decision trees in imbalanced datasets.

General Observations on Decision Trees

Both decision tree models exhibited strong accuracy on the majority class but struggled to detect rare EPM events. This reflects the limitations of single-tree approaches, which, even when balanced and optimized, are inherently prone to bias toward the majority class. Nonetheless, their interpretability and variable importance rankings provide valuable insights for understanding the drivers of extreme price movements.

6.2.2 Random Forest: Ensemble Learning for Enhanced Detection

The random forest model leveraged an ensemble of 500 decision trees to improve performance and capture nonlinear interactions among features. By aggregating predictions across multiple trees, this approach reduced overfitting and increased robustness. Key findings include:

Model Performance and Metrics

- **Accuracy:** The random forest achieved a high overall accuracy of 97.73%, largely driven by its correct classification of the majority class.
- **Specificity:** At the default threshold of 0.5, the model's specificity for detecting EPM events was 25.00%, similar to the decision trees. This indicates that the model, while powerful, still faces challenges in identifying rare events in imbalanced datasets.
- **Threshold Tuning:** Adjusting the classification threshold revealed significant flexibility in balancing sensitivity and specificity. For example, lowering the threshold below 0.5 increased EPM detection at the cost of more false positives, offering a customizable approach based on practical requirements.
- **AUC:** The model's area under the ROC curve (AUC) was 0.863, indicating good overall discriminative capability.

Variable Importance and Insights

The random forest provided robust measures of variable importance, reaffirming the critical role of specific features:

- `vol_short` (short-term volatility) emerged as the most significant predictor, emphasizing its central role in capturing market instability.
- `spread` and `PES` (Percent Effective Spread) highlighted the importance of liquidity and price impact measures in predicting EPMs.
- Additional variables, such as `Price`, `Size`, and `Logreturn`, contributed to the model's predictive power, underscoring the multifaceted nature of price dynamics and order flow.

The ensemble approach of random forests demonstrated greater resilience to noise and feature interactions, providing a comprehensive view of the predictors most relevant to EPM detection.

6.2.3 Discussion and Future Directions

The decision tree and random forest models underscore the potential and limitations of tree-based methods for detecting rare events in financial markets:

- **Strengths:** These models excel in interpretability, allowing for clear visualization of decision pathways and identification of key features. Random forests, in particular, mitigate overfitting and provide robust predictions across a range of thresholds.
- **Limitations:** Both methods struggled with the imbalanced nature of the dataset, with specificity for detecting EPM events remaining low. Single-tree approaches were particularly prone to bias toward the majority class, while random forests showed more flexibility but still faced challenges in detecting rare events.
- **Threshold Customization:** The ability to adjust thresholds in both random forests and decision trees offers practical flexibility. By lowering the threshold, models can prioritize sensitivity (EPM detection) at the expense of higher false positive rates, making them adaptable to different risk tolerances.

Future work could explore the integration of these tree-based methods with complementary techniques, such as boosting or cost-sensitive learning, to enhance their ability to detect rare events. Additionally, combining tree-based insights with more advanced models, such as neural networks, could yield a hybrid approach that leverages both interpretability and predictive power.

In conclusion, decision trees and random forests provide a strong foundation for EPM detection, offering valuable insights into feature importance and baseline performance. However, addressing the challenges posed by imbalanced datasets and rare events will require further advancements in methodology and model design.

6.3 Neural Network Models

In this section, we explored and analyzed four neural network architectures two feed-forward neural networks and two Long Short-Term Memory (LSTM) models with the goal of detecting EPMs in financial markets. Each model's architecture, training process, and evaluation were thoroughly documented, providing insights into their strengths and limitations. Below, we summarize and draw conclusions from the results of these experiments.

6.3.1 Performance Overview

The four neural network models demonstrated a clear progression in complexity and performance:

1. **Simple Feed-Forward Neural Network (8,4 Neurons):** This baseline model struggled with detecting EPMs, achieving a recall of just 0.89%. The overall accuracy was high (99.80%), but this was primarily driven by correct classifications of the majority class (No EPM). The F1 score for the minority class was only 0.0167, indicating the model's inability to handle the class imbalance effectively.

2. **Enhanced Feed-Forward Neural Network (128,64 Neurons):** Increasing the model complexity resulted in marginal improvements, with a recall of 2.68%. However, the precision remained low at 0.59%, and the F1 score for the minority class increased only slightly to 0.0097. The results highlighted that even a more complex feed-forward architecture struggled to capture the intricacies of EPM events.

3. **Simple LSTM Model:** Transitioning to a temporal architecture significantly improved recall, reaching 58.04%. This demonstrated the LSTM's ability to leverage temporal dependencies inherent in financial time-series data. However, the precision remained low (0.67%), and the false positive rate was high, resulting in an F1 score of 0.0132.

4. **Advanced LSTM Model:** Adding an additional LSTM layer and dropout for regularization further improved recall to 66.96%, the highest among all models. However, this improvement came at the cost of an increased false positive rate, and precision remained at 0.67%. The F1 score reached 0.0133, a slight improvement over the simpler LSTM model.

6.3.2 Key Insights and Challenges

Impact of Model Complexity

Increasing model complexity had mixed results. While the enhanced feed-forward network offered only slight improvements over the simple network, the transition to LSTM architectures led to significant gains in recall. The advanced LSTM model demonstrated the value of leveraging temporal data and incorporating regularization techniques, but it also underscored the difficulty of balancing recall and precision.

Class Imbalance Challenges

The severe class imbalance (with EPM events comprising less than 1% of the data) posed a persistent challenge. Despite employing class weighting during training, all models struggled to achieve a meaningful balance between recall and precision. While recall improved significantly with LSTM models, the precision remained low, reflecting a high number of false positives.

Suitability of Neural Networks for Rare Event Detection

The experiments highlight both the potential and the limitations of neural networks for rare event detection in financial data. Feed-forward networks, even when scaled up, were not effective for this task, as they lacked the ability to capture temporal dependencies. LSTMs showed promise in identifying

EPM events by leveraging sequential information, but the precision challenges raise questions about their practical applicability without further optimization.

6.3.3 Future Directions

The results suggest several avenues for future research and development:

- **Threshold Tuning:** Adjusting the classification threshold could help balance precision and recall to align with specific use cases.
- **Ensemble Methods:** Combining neural networks with other models, such as random forests or gradient boosting, may reduce false positives while maintaining high recall.
- **Data Augmentation:** Generating synthetic EPM examples or using techniques like SMOTE could address the class imbalance more effectively.
- **Architectural Enhancements:** Exploring deeper LSTM networks, attention mechanisms, or transformer-based architectures could improve the model's ability to focus on critical time-series patterns.
- **Cost-Sensitive Learning:** Introducing cost-sensitive loss functions that penalize false positives more heavily could improve precision without severely impacting recall.

This section highlights the evolution of neural network architectures for detecting EPMs. Feed-forward networks provided a foundational understanding but were insufficient for capturing the complexities of rare events in financial data. LSTM models, particularly the advanced architecture, demonstrated significant potential by leveraging temporal patterns. However, achieving an optimal balance between recall and precision remains an open challenge.

While neural networks are a promising tool for rare event detection, they require careful design, tuning, and integration with other methods to be effective in real-world financial applications. Future work should focus on overcoming the limitations identified in this study to build robust and reliable systems for detecting extreme price movements.

6.4 Final Insights

This study embarked on the challenging task of predicting Extreme Price Movements (EPMs) in financial markets, an area that remains an open problem due to the inherent unpredictability of market behavior. By exploring multiple modeling approaches including Logistic Regression, Decision Trees, Random Forests, and Neural Networks we highlighted both the opportunities and limitations of existing techniques in addressing this complex issue.

The difficulty of EPM prediction stems from the unpredictable nature of financial markets, where extreme events are influenced by myriad factors, many of which are difficult to quantify or anticipate. Unlike other domains where data patterns may follow predictable trends, financial markets are subject to dynamic changes driven by macroeconomic conditions, market sentiment, geopolitical events, and unforeseen shocks. This inherent complexity makes EPM detection an exceptionally challenging endeavor, underscoring the need for innovative and adaptive approaches.

Throughout the study, each model demonstrated distinct strengths. Logistic regression offered transparency and interpretability, helping to identify key covariates such as short-term volatility, market spread, and price metrics. Decision trees provided intuitive, hierarchical insights into variable interactions, while random forests addressed overfitting and improved robustness through ensemble learning. Neural networks, particularly LSTMs, excelled in capturing temporal dependencies and nonlinear relationships, showcasing their potential for time-series data. However, all models struggled with the

imbalanced nature of the dataset, a common issue in rare event prediction, and required careful optimization to detect EPMs effectively.

One critical finding was the importance of feature selection and engineering. However, the inclusion of additional covariates could further enhance model performance. For example, integrating external factors such as macroeconomic indicators, news sentiment analysis, or trading volumes across markets might provide a more comprehensive picture of the dynamics leading to EPMs. Incorporating real-time data streams, such as order book depth and high-frequency price movements, could also improve the timeliness and accuracy of predictions.

Besides improving the predictors, we should also rethink how we define and identify EPMs. Traditional ways of classifying EPMs, often relying on fixed limits for price changes, might not fully capture the complex nature of extreme market events. Using advanced methods could offer a more flexible and context-sensitive way to identify EPMs. For example, drift burst methods, which detect sudden and lasting shifts in the patterns of time-series data, closely match the traits of EPMs and could lead to better detection systems.

Another key insight was the trade-off between interpretability and predictive power. While simpler models like logistic regression and decision trees offer clear explanations, they are limited in capturing the complex interactions and nonlinearities present in financial markets. Conversely, neural networks provide powerful predictive capabilities but often operate as "black boxes," complicating their adoption in scenarios requiring transparency. Bridging this gap through hybrid models, explainable AI techniques, or ensemble approaches represents a promising direction for future research. Despite the progress made, the results reaffirm that EPM prediction is inherently difficult. The financial market's chaotic and adaptive nature implies that even the most sophisticated models can only approximate reality. As such, no model should be expected to provide perfect predictions. Instead, these tools should be viewed as aids to inform decision-making, highlight potential risks, and complement traditional market analysis.

Moving forward, researchers and practitioners should focus on several key areas. First, the development of dynamic models that adapt to evolving market conditions could improve performance. Second, exploring novel covariates, particularly those derived from external data sources, may uncover previously unrecognized patterns. Third, addressing class imbalance through advanced resampling techniques, cost-sensitive learning, or anomaly detection methods could further enhance rare event prediction. Fourth, integrating methodologies such as drift burst detection into EPM identification could redefine how these events are recognized, aligning detection mechanisms more closely with market realities. Finally, fostering interdisciplinary collaborations between finance, machine learning, and behavioral economics may yield innovative solutions to this persistent challenge.

In conclusion, while EPM prediction remains a formidable task, this study underscores the value of diverse modeling approaches and the potential for further innovation. By incorporating new covariates, adopting sophisticated identification methods, embracing adaptive techniques, and maintaining a focus on both interpretability and performance, we can advance toward more reliable tools for managing and mitigating extreme price movements in financial markets.

Appendix

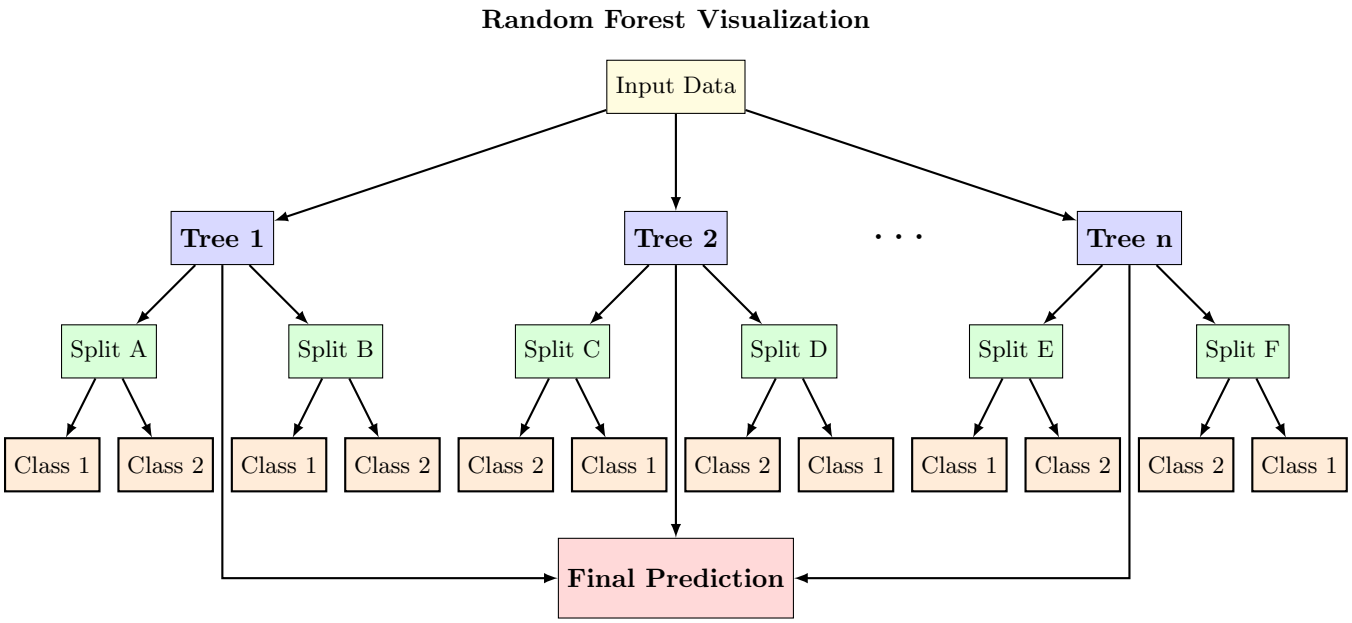


Figure 1: A Random Forest ensemble of decision trees

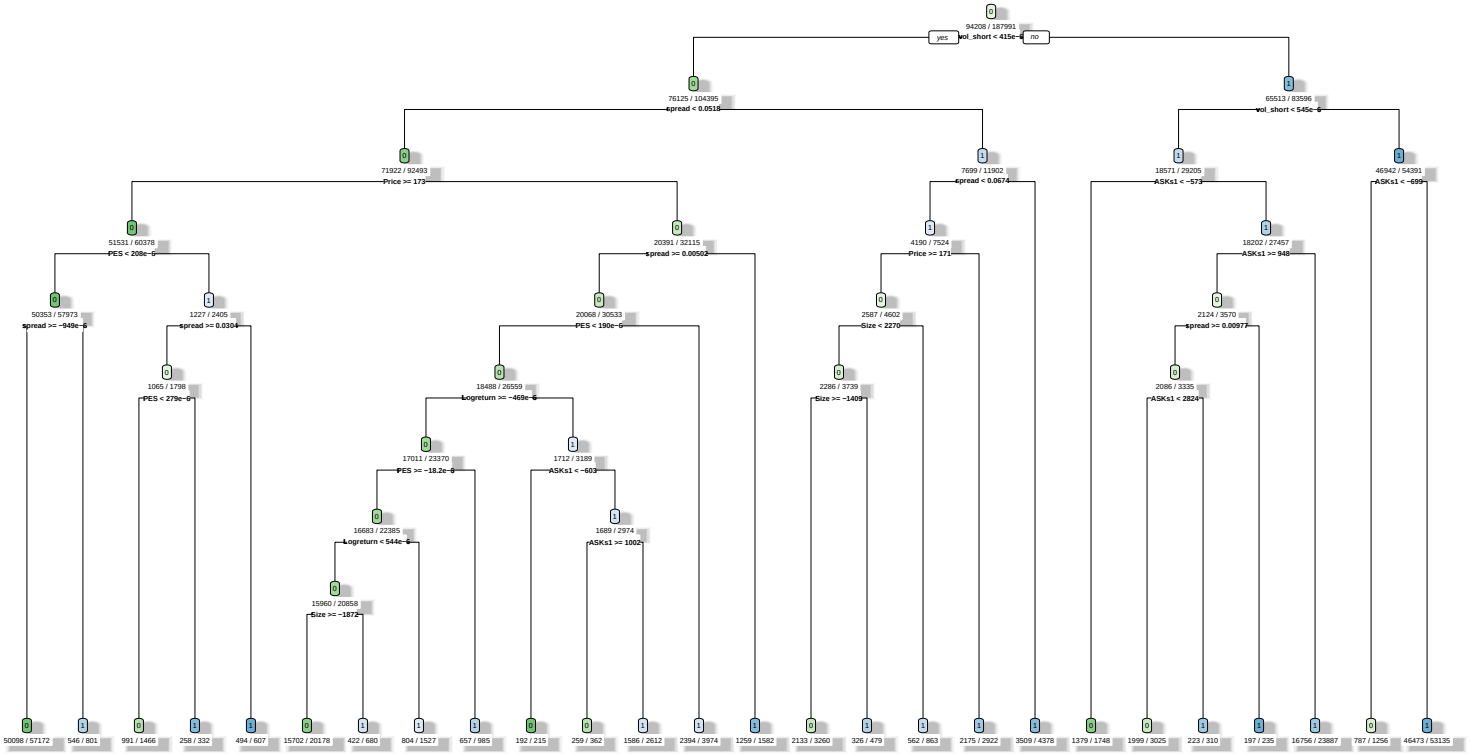


Figure 2: Decision Tree Structure for Complex Configuration used for Facebook EPM prediction

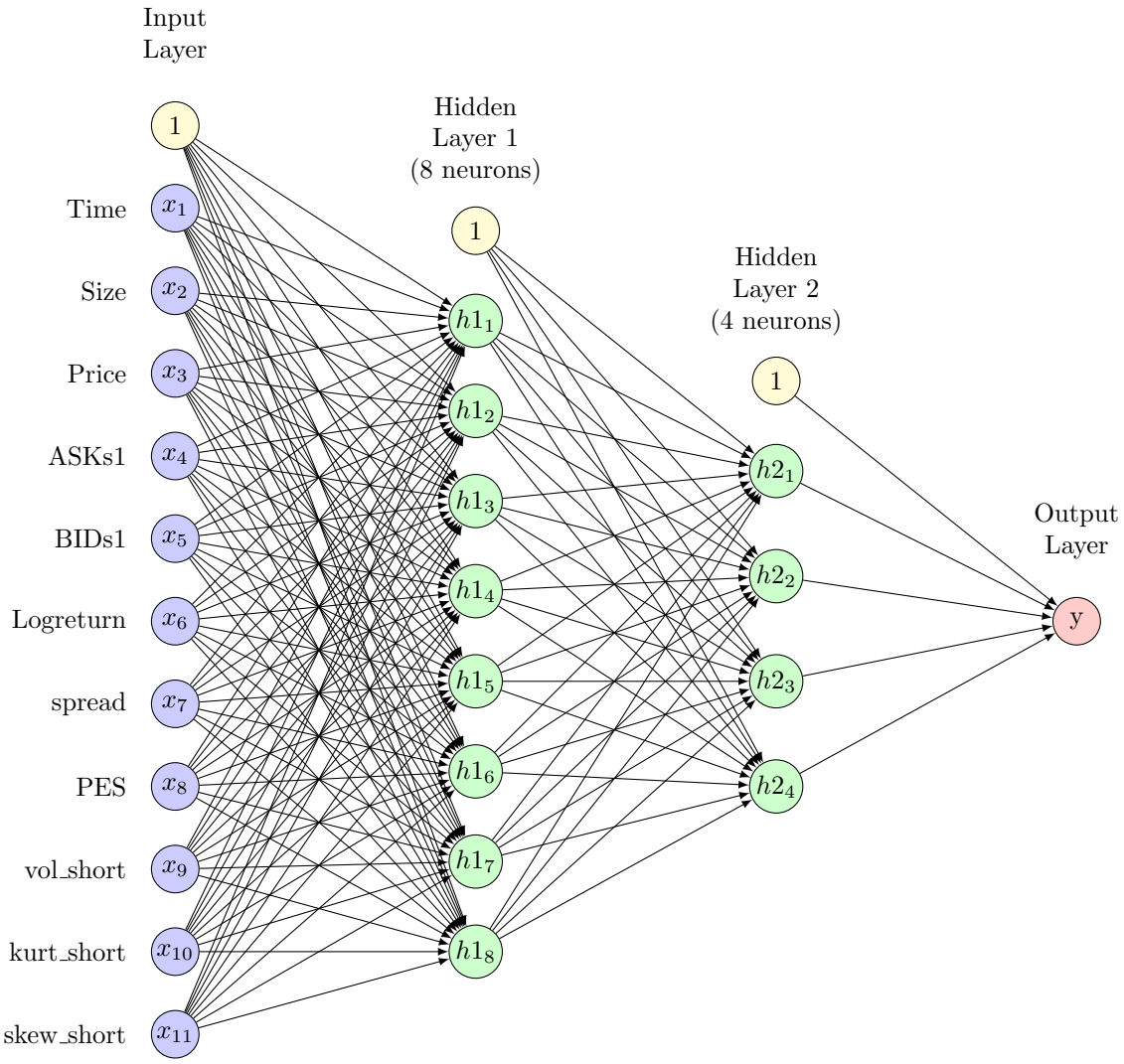


Figure 3: Neural network architecture with 11 input features, two hidden layers (8, 4 neurons), and one output neuron used for Facebook EPM prediction

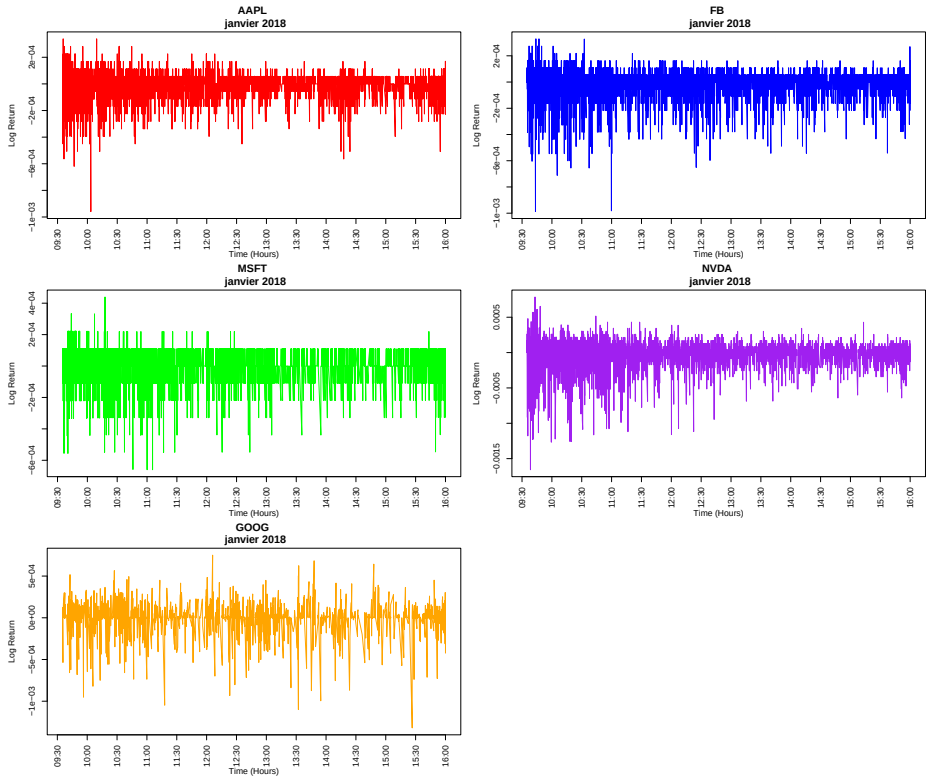


Figure 4: Log returns for the first day of January

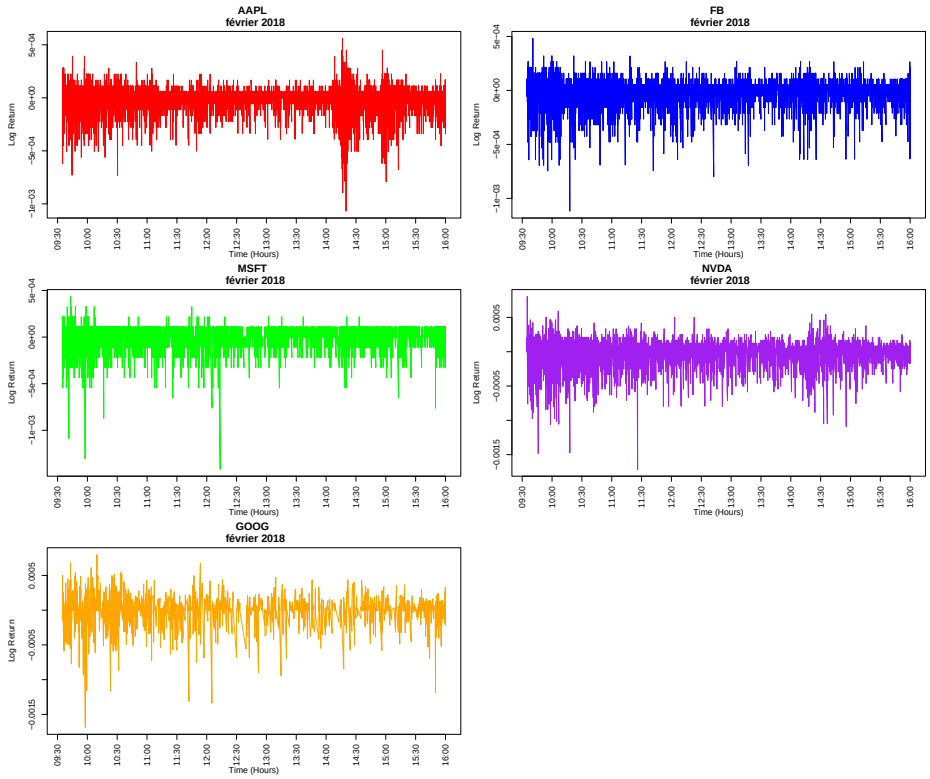


Figure 5: Log returns for the first day of February

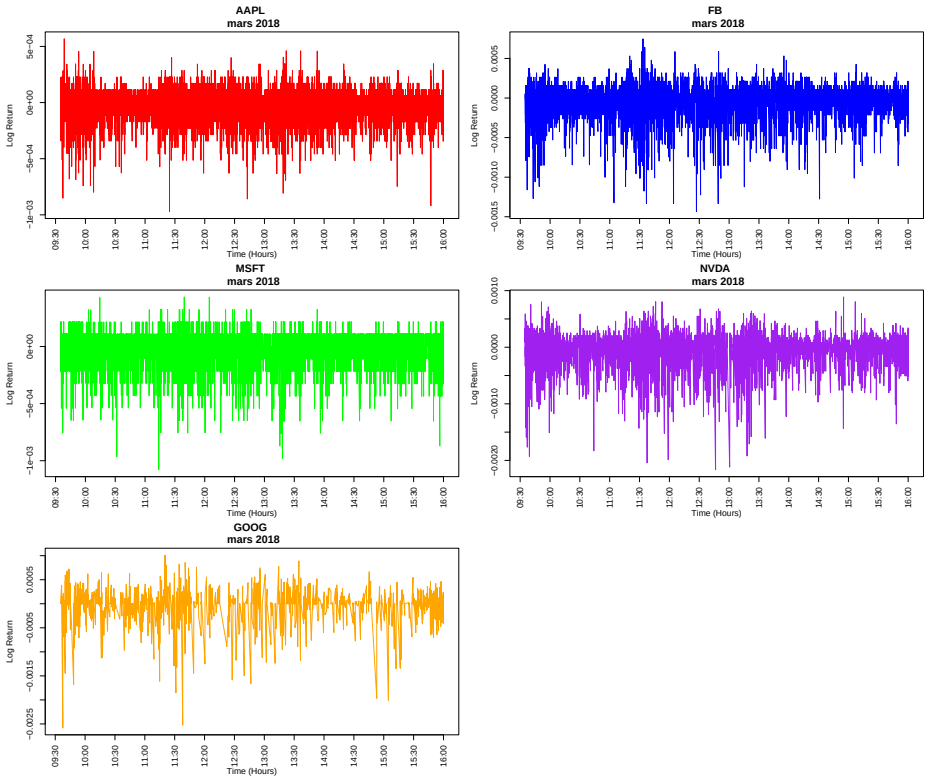


Figure 6: Log returns for the first day of March

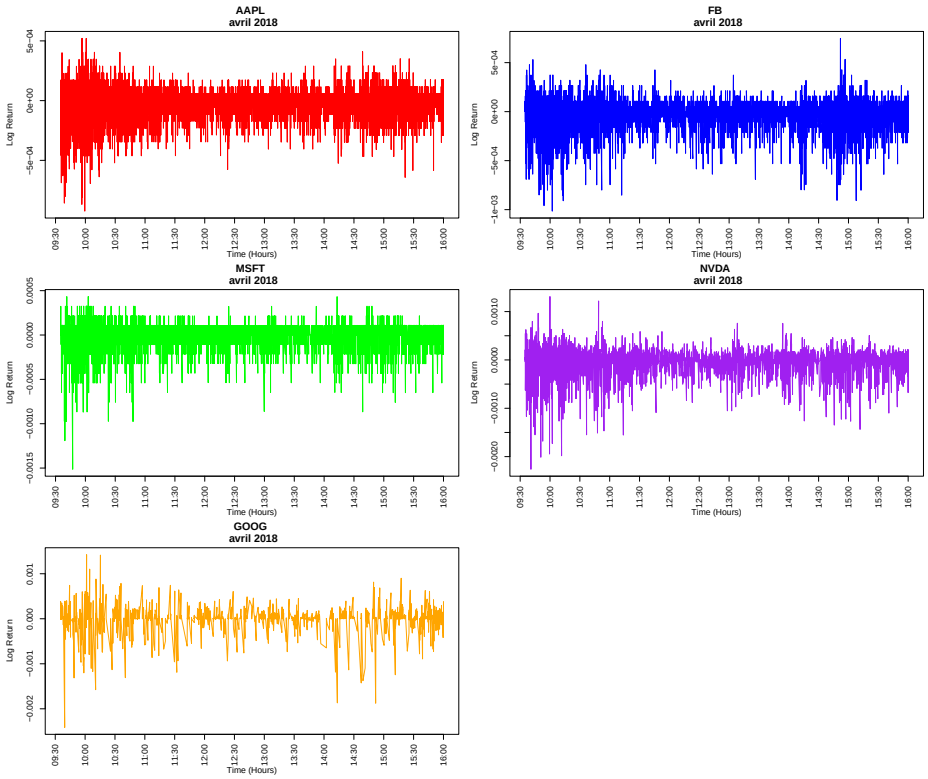


Figure 7: Log returns for the first day of April

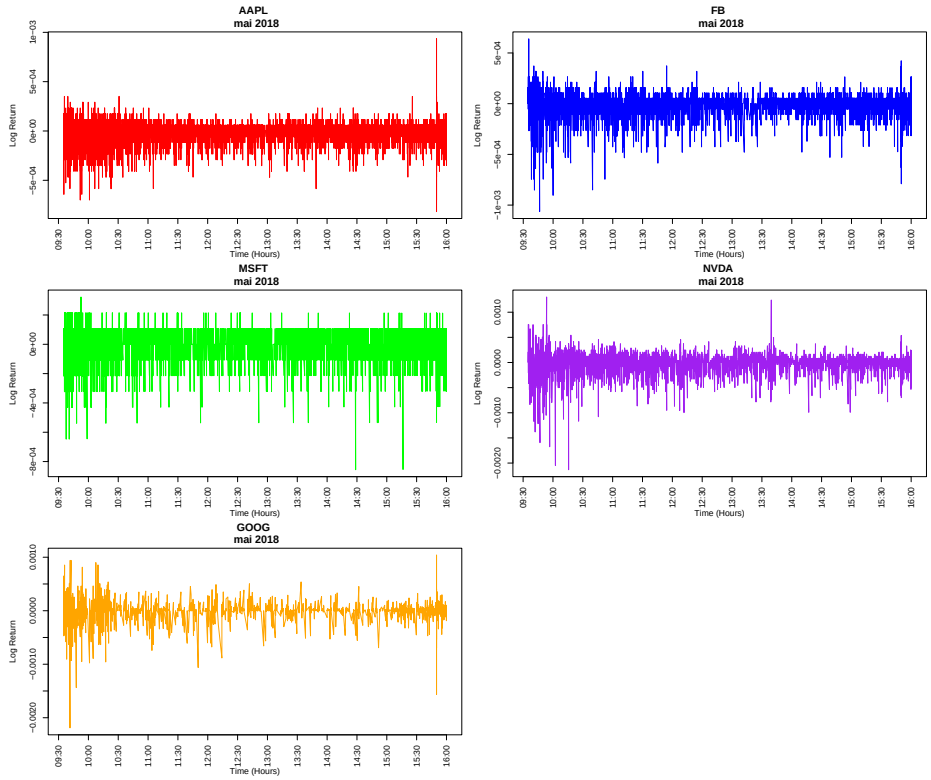


Figure 8: Log returns for the first day of May

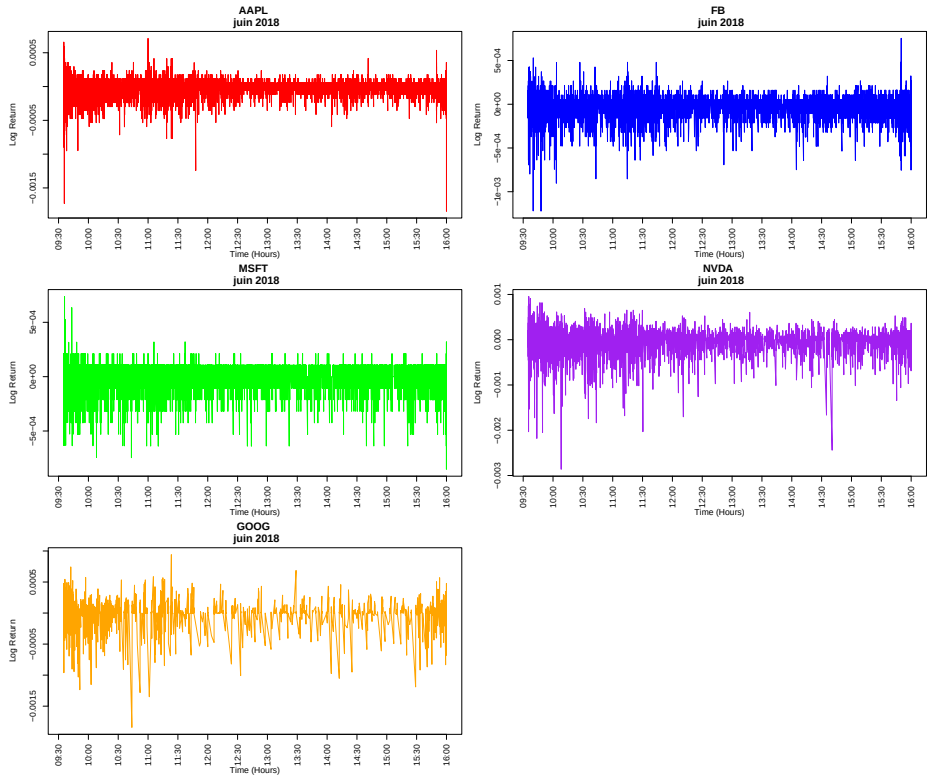


Figure 9: Log returns for the first day of June

Confusion Matrices by Model and Stock

Logistic Regression

Apple

Cut-off: 0.5

Prediction	0	1
0	60398	139
1	0	0

G1 (0.003519645)

Prediction	0	1
0	48280	49
1	12118	90

G2 (0.004360077)

Prediction	0	1
0	54158	68
1	6240	71

Microsoft

Cut-off: 0.5

Prediction	0	1
0	52832	142
1	1	0

G1 (0.001711655)

Prediction	0	1
0	36868	45
1	15965	97

G2 (0.005597668)

Prediction	0	1
0	52454	122
1	379	20

NVIDIA

Cut-off: 0.5

Prediction	0	1
0	51235	83
1	0	0

G1 (0.005045173)

Prediction	0	1
0	40529	24
1	10706	59

G2 (0.01564447)

Prediction	0	1
0	50794	67
1	441	16

Google

Cut-off: 0.5

Prediction	0	1
0	22262	58
1	1	0

G1 (0.004339626)

Prediction	0	1
0	14332	19
1	7931	39

G2 (0.005973074)

Prediction	0	1
0	18746	29
1	3517	29

Logistic Regression with Selected Variables

Apple

G1 (0.002970725)

Prediction	0	1
0	41297	35
1	19101	104

G2 (0.005039074)

Prediction	0	1
0	56803	86
1	3595	53

Microsoft

G1 (0.001657514)

Prediction	0	1
0	36571	45
1	16262	97

G2 (0.01332471)

Prediction	0	1
0	52810	137
1	23	5

NVIDIA

G1 (0.005298843)

Prediction	0	1
0	41127	24
1	10108	59

G2 (0.01218726)

Prediction	0	1
0	50167	57
1	1068	26

Google

G1 (0.003949181)		
Prediction	**0**	**1**
0	13577	18
1	8686	40

G2 (0.007312118)		
Prediction	**0**	**1**
0	20730	41
1	1533	17

Logistic Regression with Balanced Class Weights

Apple

G1 (0.0785)		
Prediction	**0**	**1**
0	10137	3
1	50261	136

G2 (0.1187)		
Prediction	**0**	**1**
0	31731	19
1	28667	120

Microsoft

G1 (0.0621)		
Prediction	**0**	**1**
0	7437	7
1	45396	135

G2 (0.0988)		
Prediction	**0**	**1**
0	24116	21
1	28717	121

NVIDIA

G1 (0.0748)		
Prediction	**0**	**1**
0	10606	1
1	40629	82

G2 (0.0797)		
Prediction	**0**	**1**
0	30543	8
1	20692	75

Google

G1 (0.1469)		
Prediction	**0**	**1**
0	4927	4
1	17336	54

G2 (0.1665)		
Prediction	**0**	**1**
0	7320	5
1	14943	53

Logistic Regression with Balanced Class Weights and Selected Covariates

Apple

G1 (0.0817)		
Prediction	**0**	**1**
0	8273	3
1	52125	136

G2 (0.1278)		
Prediction	**0**	**1**
0	32192	20
1	28206	119

Microsoft

G1 (0.0617)		
Prediction	**0**	**1**
0	8439	9
1	44394	133

G2 (0.0913)		
Prediction	**0**	**1**
0	22032	20
1	30801	122

NVIDIA

G1 (0.0774)

Prediction	0	1
0	10982	2
1	40253	81

G2 (0.1591)

Prediction	0	1
0	32802	9
1	18433	74

Google

G1 (0.1372)

Prediction	0	1
0	4927	4
1	17336	54

G2 (0.1665)

Prediction	0	1
0	7810	4
1	14453	54

LASSO Regression Models

Apple

CUT OFF 0.5

Prediction	0	1
0	60385	138
1	13	1

G1 (0.0027)

Prediction	0	1
0	44612	38
1	15786	101

G2 (0.0166)

Prediction	0	1
0	59958	121
1	440	18

Microsoft

CUT OFF 0.5

Prediction	0	1
0	52750	140
1	83	2

G1 (0.0027)

Prediction	0	1
0	39858	45
1	12975	97

G2 (0.0152)

Prediction	0	1
0	52514	123
1	319	19

NVIDIA

CUT OFF 0.5

Prediction	0	1
0	51230	82
1	5	1

G1 (0.0036)

Prediction	0	1
0	39196	14
1	12039	69

G2 (0.0149)

Prediction	0	1
0	50337	56
1	898	27

Google

CUT OFF 0.5

Prediction	0	1
0	22251	56
1	12	2

G1 (0.0038)

Prediction	0	1
0	16139	17
1	6124	41

G2 (0.0124)

Prediction	0	1
0	22007	38
1	256	20

First Decision Tree Models

Apple

Prediction	0	1
0	58970	115
1	1428	24

Microsoft

Prediction	0	1
0	51677	132
1	1156	10

NVIDIA

Prediction	0	1
0	50152	73
1	1083	10

Google

Prediction	0	1
0	21585	51
1	678	7

Second Decision Tree Models (More Complex Configuration)

Apple

Prediction	0	1
0	59181	116
1	1217	23

Microsoft

Prediction	0	1
0	51740	128
1	1093	14

NVIDIA

Prediction	0	1
0	50184	60
1	1051	23

Google

Prediction	0	1
0	21674	50
1	589	8

Random Forest Models

Apple

Prediction	0	1
0	59783	124
1	615	15

Microsoft

Prediction	0	1
0	52314	120
1	519	22

NVIDIA

Prediction	0	1
0	50893	69
1	342	14

Google

Prediction	0	1
0	22135	55
1	128	3

First Neural Network Model

Apple

Prediction	0	1
0	60391	138
1	7	1

Microsoft

Prediction	0	1
0	52828	140
1	5	2

NVIDIA

Prediction	0	1
0	51226	81
1	9	2

Google

Prediction	0	1
0	22255	56
1	8	2

Second Neural Network Model

Apple

Prediction	0	1
0	59867	136
1	531	3

Microsoft

Prediction	0	1
0	52420	138
1	413	4

NVIDIA

Prediction	0	1
0	50790	80
1	445	3

Google

Prediction	0	1
0	22070	54
1	193	4

First LSTM Model

Apple

Prediction	0	1
0	48694	52
1	9566	77

Microsoft

Prediction	0	1
0	44191	79
1	8642	63

NVIDIA

Prediction	0	1
0	43178	29
1	8057	54

Google

Prediction	0	1
0	17222	19
1	5041	39

Second LSTM Model

Apple

Prediction	0	1
0	45785	42
1	14613	97

Microsoft

Prediction	0	1
0	41512	41
1	11321	101

NVIDIA

Prediction	0	1
0)	40338	19
1	10897	64

Google

Prediction	0	1
0	17260	11
1	5003	47

Bibliography

David Abad and José Yagüe. From pin to vpin: An introduction to order flow toxicity. *The Spanish Review of Financial Economics*, 10(2):74–83, 2012. ISSN 2173-1268. URL https://doi.org/10.1016/j.srfe.2012.10.002.

Michael Aitken and Carole Comerton-Forde. How should liquidity be measured? *Pacific-Basin Finance Journal*, 11(1):45–59, 2003. ISSN 0927-538X. URL https://doi.org/10.1016/S0927-538X(02)00093-8.

Torben G. Andersen and Oleg Bondarenko. Reflecting on the vpin dispute. *Journal of Financial Markets*, 17:53–64, 2014. ISSN 1386-4181. URL https://doi.org/10.1016/j.finmar.2013.08.002.

ASIC. Australian securities and investments commission. australian equity market structure. *Victoria*, 2010.

Markus Baldauf and Joshua Mollner. High-frequency trading and market performance. *The Journal of Finance*, 75(3):1495–1526, 2020. URL https://doi.org/10.1111/jofi.12882.

Flavio Bazzana and Andrea Collini. How does hft activity impact market volatility and the bid-ask spread after an exogenous shock? an empirical analysis on s&p 500 etf. *The North American Journal of Economics and Finance*, 54:101240, 2020. URL https://doi.org/10.1016/j.najef.2020.101240.

E. Wes Bethel, David Leinweber, Oliver Rübel, and Kesheng Wu. Federal market information technology in the post flash crash era: roles for supercomputing. In *Proceedings of the Fourth Workshop on High Performance Computational Finance*, WHPCF '11, page 23–30, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450311083. doi: 10.1145/2088256.2088267. URL https://doi.org/10.1145/2088256.2088267.

Ekkehart Boehmer, Joachim Grammig, and Erik Theissen. Estimating the probability of informed trading—does trade misclassification matter? *Journal of Financial Markets*, 10(1):26–47, 2007. ISSN 1386-4181. URL https://doi.org/10.1016/j.finmar.2006.07.002.

Ekkehart Boehmer, Dan Li, and Gideon Saar. The competitive landscape of high-frequency trading firms. *The Review of Financial Studies*, 31(6):2227–2276, 01 2018. ISSN 0893-9454. doi: 10.1093/rfs/hhx144. URL https://doi.org/10.1093/rfs/hhx144.

Johannes Breckenfelder. How does competition among high-frequency traders affect market liquidity? *Research bulletin*, 2020.

Johannes Breckenfelder. Competition among high-frequency traders and market quality. *Journal of Economic Dynamics and Control*, 166:104922, 2024. ISSN 0165-1889. URL https://doi.org/10.1016/j.jedc.2024.104922.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001. URL https://doi.org/10.1023/A:1010933404324.

Jonathan Brogaard, Allen Carrion, Thibaut Moyaert, Ryan Riordan, Andriy Shkilko, and Konstantin Sokolov. High frequency trading and extreme price movements. *Journal of Financial Economics*, 128 (2):253–265, 2018. ISSN 0304-405X. URL https://doi.org/10.1016/j.jfineco.2018.02.002.

Jonathan Brogaard et al. High frequency trading and its impact on market quality. *Northwestern University Kellogg School of Management Working Paper*, 66:10, 2010.

Chris Brummer. Stock exchanges and the new markets for securities laws. *The University of Chicago Law Review*, 75(4):1435–1491, 2008.

Giovanni Cespa and Thierry Foucault. Illiquidity contagion and liquidity crashes. *The Review of Financial Studies*, 27(6):1615–1660, 03 2014. ISSN 0893-9454. doi: 10.1093/rfs/hhu016. URL https://doi.org/10.1093/rfs/hhu016.

CFTC & SEC. U.s. commodity futures trading commission (cft) u.s. securities and exchange commission (sec). preliminary findings regarding the market events of may 6, 2010. washington, d.c. 2010a.

CFTC & SEC. U.s. commodity futures trading commission (cft) u.s. securities and exchange commission (sec). findings regarding the market events of may 6, 2010. washington, d.c. 2010b.

Harold Demsetz. The cost of transacting*. *The Quarterly Journal of Economics*, 82(1):33–53, 02 1968. ISSN 0033-5533. doi: 10.2307/1882244. URL https://doi.org/10.2307/1882244.

David Easley and Maureen O'hara. Time and the process of security price adjustment. *The Journal of finance*, 47(2):577–605, 1992. URL https://doi.org/10.1111/j.1540-6261.1992.tb04402.x.

David Easley, Nicholas M Kiefer, Maureen O'hara, and Joseph B Paperman. Liquidity, information, and infrequently traded stocks. *The Journal of Finance*, 51(4):1405–1436, 1996. URL .http://dx.doi.org/10.1111/j.1540-6261.1996.tb04074.x.

David Easley, Nicholas M. Kiefer, and Maureen O'Hara. One day in the life of a very common stock. *The Review of Financial Studies*, 10(3):805–835, 04 1997. ISSN 0893-9454. doi: 10.1093/rfs/10.3.805. URL https://doi.org/10.1093/rfs/10.3.805.

David Easley, Soeren Hvidkjaer, and Maureen O'hara. Is information risk a determinant of asset returns? *The journal of finance*, 57(5):2185–2221, 2002. URL https://doi.org/10.1111/1540-6261.00493.

David Easley, Robert F. Engle, Maureen O'Hara, and Liuren Wu. Time-varying arrival rates of informed and uninformed trades. *Journal of Financial Econometrics*, 6(2):171–207, 02 2008. ISSN 1479-8409. doi: 10.1093/jjfinec/nbn003. URL https://doi.org/10.1093/jjfinec/nbn003.

David Easley, Marcos M. López de Prado, and Maureen O'Hara. The microstructure of the" flash crash": Flow toxicity, liquidity crashes, and the probability of informed trading. *Journal of Portfolio Management*, 37(2), 2011. ISSN 0095-4918. URL https://datascienceassn.org/sites/default/files/The%20Microstructure%20of%20the%20Flash%20Crash%20-%20Flow%20Toxicity%2C%20Liquidity%20Crashes%20and%20the%20Probability%20of%20Informed%20Trading.pdf.

David Easley, Marcos Lopez de Prado, and Maureen O'Hara. Vpin and the flash crash: a comment. *Journal of Financial Markets, Forthcoming, Johnson School Research Paper Series*, (25-2012), 2012a. URL https://dx.doi.org/10.2139/ssrn.2062450.

David Easley, Marcos M. López de Prado, and Maureen O'Hara. Flow toxicity and liquidity in a high-frequency world. *The Review of Financial Studies*, 25(5):1457–1493, 03 2012b. ISSN 0893-9454. doi: 10.1093/rfs/hhs053. URL https://doi.org/10.1093/rfs/hhs053.

David Easley, Marcos López de Prado, Maureen O'Hara, and Zhibai Zhang. Microstructure in the machine age. *The Review of Financial Studies*, 34(7):3316–3363, 07 2019. ISSN 0893-9454. doi: 10.1093/rfs/hhaa078. URL https://doi.org/10.1093/rfs/hhaa078.

Oğuz Ersan, Nihan Dalgıç, Cumhur Enis Ekinci, and Mehmet Bodur. High-frequency trading and its impact on market liquidity: A review of literature. *Alanya Akademik Bakış*, 5(1):345–368, 2021. URL https://doi.org/10.29023/alanyaakademik.799039.

Thierry Foucault, Ohad Kadan, and Eugene Kandel. Limit order book as a market for liquidity. *The Review of Financial Studies*, 18(4):1171–1217, 08 2005. ISSN 0893-9454. doi: 10.1093/rfs/hhi029. URL https://doi.org/10.1093/rfs/hhi029.

Kenneth A Froot, David S Scharfstein, and Jeremy C Stein. Herd on the street: Informational inefficiencies in a market with short-term speculation. *The Journal of finance*, 47(4):1461–1484, 1992. URL https://doi.org/10.1111/j.1540-6261.1992.tb04665.x.

Dean Furbush. Program trading and price movement: Evidence from the october 1987 market crash. *Financial Management*, 18(3):68–83, 1989. ISSN 00463892, 1755053X. URL http://www.jstor.org/stable/3665650.

Anton Golub, John Keane, and Ser-Huang Poon. High frequency trading and mini flash crashes, 2012. URL https://arxiv.org/abs/1211.6667.

Björn Hagströmer and Lars Nordén. The diversity of high-frequency traders. *Journal of Financial Markets*, 16(4):741–770, 2013.

Björn Hagströmer and Lars Nordén. The diversity of high-frequency traders. *Journal of Financial Markets*, 16(4):741–770, 2013. ISSN 1386-4181. URL https://doi.org/10.1016/j.finmar.2013.05.009. High-Frequency Trading.

Andrew Haldane. Patience and finance. In *Speech to Oxford China Business Forum, Beijing, Bank of England*, volume 2, 2010.

Larry Harris. *Trading and exchanges: Market microstructure for practitioners*. Oxford university press, 2002.

Larry Harris. What to do about high-frequency trading, 2013.

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

Joel Hasbrouck and Gideon Saar. Low-latency trading. *Journal of Financial Markets*, 16(4):646–679, 2013. ISSN 1386-4181. URL https://doi.org/10.1016/j.finmar.2013.05.003. High-Frequency Trading.

Joel Hasbrouck and Duane J. Seppi. Common factors in prices, order flows, and liquidity. *Journal of Financial Economics*, 59(3):383–411, 2001. ISSN 0304-405X. URL https://doi.org/10.1016/S0304-405X(00)00091-X.

Nikolaus Hautsch, Michael Noé, and S Sarah Zhang. The ambivalent role of high-frequency trading in turbulent market periods. 2017. URL http://dx.doi.org/10.2139/ssrn.3020142.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Roger D. Huang and Hans R. Stoll. The components of the bid-ask spread: A general approach. *The Review of Financial Studies*, 10(4):995–1034, 06 1997. ISSN 0893-9454. doi: 10.1093/rfs/10.4.995. URL https://doi.org/10.1093/rfs/10.4.995.

Ruihong Huang and Tomas Polak. Lobster: Limit order book reconstruction system. *Available at SSRN 1977207*, 2011.

John S. Hughes, Jing Liu, and Jun Liu. Information asymmetry, diversification, and cost of capital. *The Accounting Review*, 82(3):705–729, 05 2007. ISSN 0001-4826. doi: 10.2308/accr.2007.82.3.705. URL https://doi.org/10.2308/accr.2007.82.3.705.

Sandrine Jacob Leal, Mauro Napoletano, Andrea Roventini, and Giorgio Fagiolo. Rock around the clock: An agent-based model of low-and high-frequency trading. *Journal of Evolutionary Economics*, 26: 49–76, 2016. URL https://doi.org/10.1007/s00191-015-0418-4.

Pawan Jain. J-reit market quality: impact of high frequency trading and the financial crisis. *Available at SSRN 2972092*, 2017. URL https://dx.doi.org/10.2139/ssrn.2972092.

Elvis Jarnecic and Mark Snape. The provision of liquidity by high-frequency participants. *Financial Review*, 49(2):371–394, 2014. URL https://doi.org/10.1111/fire.12040.

ROBERT A. Jarrow and PHILIP Protter. A dysfunctional role of high frequency trading in electronic markets. *International Journal of Theoretical and Applied Finance*, 15(03):1250022, 2012. doi: 10.1142/S0219024912500227. URL https://doi.org/10.1142/S0219024912500227.

Neil Johnson, Guannan Zhao, Eric Hunsader, Hong Qi, Nicholas Johnson, Jing Meng, and Brian Tivnan. Abrupt rise of new machine ecology beyond human response time. *Scientific reports*, 3(1):2627, 2013. URL https://doi.org/10.1038/srep02627.

Jangkoo Kang, Kyung Yoon Kwon, and Wooyeon Kim. Flow toxicity of high-frequency trading and its impact on price volatility: Evidence from the kospi 200 futures market. *Journal of Futures Markets*, 40(2):164–191, 2020. URL https://doi.org/10.1002/fut.22062.

Andrei Kirilenko, Albert S Kyle, Mehrdad Samadi, and Tugkan Tuzun. The flash crash: High-frequency trading in an electronic market. *The Journal of Finance*, 72(3):967–998, 2017. URL https://doi.org/10.1111/jofi.12498.

Albert S. Kyle. Continuous auctions and insider trading. *Econometrica*, 53(6):1315–1335, 1985. ISSN 00129682, 14680262. URL https://doi.org/10.2307/1913210.

H Lash and J Spicer. Mystery trader in selloff was waddell: Kansas-based manager sold large order of e-mini contracts during crucial time period, document shows. *The Globe and Mail, May*, 14, 2010.

Sandrine Jacob Leal and Mauro Napoletano. Market stability vs. market resilience: Regulatory policies experiments in an agent-based model with low- and high-frequency trading. *Journal of Economic Behavior Organization*, 157:15–41, 2019. ISSN 0167-2681. URL https://doi.org/10.1016/j.jebo.2017.04.013.

Charles M. C. Lee, Belinda Mucklow, and Mark J. Ready. Spreads, depths, and the impact of earnings information: An intraday analysis. *The Review of Financial Studies*, 6(2):345–374, 04 1993. ISSN 0893-9454. doi: 10.1093/rfs/6.2.345. URL https://doi.org/10.1093/rfs/6.2.345.

Suzanne S. Lee and Per A. Mykland. Jumps in equilibrium prices and market microstructure noise. *Journal of Econometrics*, 168(2):396–406, 2012. ISSN 0304-4076. URL https://doi.org/10.1016/j.jeconom.2012.03.001.

Qin Lei and Guojun Wu. Time-varying informed and uninformed trading activities. *Journal of Financial Markets*, 8(2):153–181, 2005. ISSN 1386-4181. URL https://doi.org/10.1016/j.finmar.2004.09.002.

Lauren Liebenberg. *A Virtual Tour of the e-Markets of Today*, pages 51–86. Palgrave Macmillan UK, London, 2002. ISBN 978-1-4039-4606-5. doi: 10.1057/9781403946065_6. URL https://doi.org/10.1057/9781403946065_6.

Francois M Longin. Optimal margin level in futures markets: Extreme price movements. *Journal of Futures Markets: Futures, Options, and Other Derivative Products*, 19(2):127–152, 1999. URL https://doi.org/10.1002/(SICI)1096-9934(199904)19:2%3C127::AID-FUT1%3E3.0.CO;2-M.

François M Longin. From value at risk to stress testing: The extreme value approach. *Journal of Banking Finance*, 24(7):1097–1130, 2000. ISSN 0378-4266. URL https://doi.org/10.1016/S0378-4266(99)00077-1.

LSE. London stock exchange group response to cesr call for evidence on micro-structural issues of the european equity markets (cesr/10-142). 2010.

Laura Malceniece, Kārlis Malcenieks, and Tālis J. Putniņš. High frequency trading and comovement in financial markets. *Journal of Financial Economics*, 134(2):381–399, 2019. ISSN 0304-405X. URL https://doi.org/10.1016/j.jfineco.2018.02.015.

Viktor Manahov. A note on the relationship between high-frequency trading and latency arbitrage. *International review of financial analysis*, 47:281–296, 2016. URL https://doi.org/10.1016/j.irfa.2016.06.014.

Steven V Mann and Pradipkumar Ramanlal. The dealers'price/size quote and market liquidity. *Journal of Financial Research*, 19(2):243–271, 1996. URL https://doi.org/10.1111/j.1475-6803.1996.tb00596.x.

Jerry W Markham and Daniel J Harty. For whom the bell tolls: the demise of exchange trading floors and the growth of ecns. *J. Corp. L.*, 33:865, 2007.

Albert J. Menkveld. High frequency trading and the new market makers. *Journal of Financial Markets*, 16 (4):712–740, 2013. ISSN 1386-4181. URL https://doi.org/10.1016/j.finmar.2013.06.006. High-Frequency Trading.

MiFID II. European parliament and council. markets in financial instruments directive 2014/65/eu: Mifid ii. 2014.

Rena S Miller and Gary Shorter. *High frequency trading: Overview of recent developments*, volume 4. Congressional Research Service Washington, DC, 2016.

Imad Moosa and Vikash Ramiah. The profitability of high-frequency trading: is it for real. In *The handbook of high frequency trading*, number 2, pages 25–45. Elsevier, 2015.

NAFM. Netherlands authority of financial markets, 2010. high-frequency trading: The application of advanced trading technology in the european marketplace. 2010.

Nathaniel Popper. High-speed trading no longer hurtling forward. *The New York Times*, 14, 2012.

Angelo Ranaldo. Intraday market dynamics around public information arrivals. In *EFMA 2002 London Meeting, Forthcoming*, 2005. URL http://dx.doi.org/10.2139/ssrn.301823.

Richard Roll. A simple implicit measure of the effective bid-ask spread in an efficient market. *The Journal of finance*, 39(4):1127–1139, 1984. URL https://doi.org/10.1111/j.1540-6261.1984.tb03897.x.

Abdourahmane Sarr and Tonny Lybek. Measuring liquidity in financial markets. 2002. URL https://ssrn.com/abstract=880932.

SEC. U.s. securities and exchange commission. part iii: Concept release on equity market structure; proposed rule, 17 cfr part 242. *Federal Register*, 75(13):3594–3614, 2010.

Andriy Shkilko and Konstantin Sokolov. Every cloud has a silver lining: Fast trading, microwave connectivity, and trading costs. *The Journal of Finance*, 75(6):2899–2927, 2020. URL https://doi.org/10.1111/jofi.12969.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 12 1996. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1996.tb02080.x. URL https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Clara Vega. Stock price reaction to public and private information. *Journal of Financial Economics*, 82 (1):103–133, 2006. ISSN 0304-405X. URL https://doi.org/10.1016/j.jfineco.2005.07.011.

Hsiou-Wei William Lin and Wen-Chyan Ke. A computing bias in estimating the probability of informed trading. *Journal of Financial Markets*, 14(4):625–640, 2011. ISSN 1386-4181. URL https://doi.org/10.1016/j.finmar.2011.03.001.

Frank Zhang. High-frequency trading, stock volatility, and price discovery. *Available at SSRN 1691679*, 2010. URL https://dx.doi.org/10.2139/ssrn.1691679.

Executive Summary

The financial markets operate as complex systems where prices are influenced by factors ranging from macroeconomic indicators to trading dynamics. Among these, high-frequency trading (HFT) has emerged as a transformative force, enhancing liquidity and efficiency while introducing challenges like market instability during extreme price movements (EPMs). This thesis addresses the critical need to predict and mitigate risks associated with EPMs under normal market conditions, leveraging machine learning models and high-frequency data.

The study focuses on technology stocks, specifically those of Facebook, Nvidia, Google, Microsoft, and Apple, over a six-month period in 2018. Using data from LOBSTER, a tool for reconstructing limit order books, novel liquidity covariates were developed to enhance prediction granularity. The research employs a comprehensive methodological framework, comparing logistic regression, decision trees, random forests, and advanced neural networks, including Long Short-Term Memory (LSTM) models, to identify and forecast EPMs.

Key findings reveal that logistic regression offers interpretability, while random forests and LSTM models provide superior predictive performance. The study also addresses challenges like class imbalance and model transparency, crucial for practical financial applications. By identifying predictive patterns in EPMs, this thesis contributes to improving market resilience and informing regulatory frameworks.

The results underscore the importance of integrating advanced modeling techniques with high-frequency data for early detection of EPMs, offering actionable insights for market practitioners and regulators. Future research directions include expanding datasets and exploring hybrid models to further enhance predictive accuracy and robustness.

Keywords: High-Frequency Trading, Extreme Price Movements, Machine Learning, Financial Markets, Risk Management