Essays in High-Frequency Trading: Insights of trading speed, systematic risk and market sentiment

by

Bo Liu
B.Sc., University of Oregon, 2014
M.Sc., New York University, 2016
M.Sc., Georgia Institute of Technology, 2019

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Economics

We acknowledge with respect the Lekwungen peoples on whose traditional territory the university stands and the Songhees, Esquimalt and W̱SÁNEĆ peoples whose historical relationships with the land continue to this day.

Essays in High-Frequency Trading: Insights of trading speed, systematic risk and market sentiment

by

Bo Liu
B.Sc., University of Oregon, 2014
M.Sc., New York University, 2016
M.Sc., Georgia Institute of Technology, 2019

Supervisory Committee

Dr. Ke Xu. Supervisor Main, Supervisor
(Department of Economics)

Dr. M. Vasco Gabriel, Departmental Member
(Department of Economics)

Dr. Xuekui Zhang, Outside Member
(Department of Mathematics and Statistics)

ABSTRACT

The landscape of financial markets has undergone a profound transformation with the advent of high-frequency trading (HFT), fundamentally altering traditional notions of market dynamics. The dissertation explores the profound transformation of financial markets by HFT, examining its impact on trading speed, systematic risk, and market sentiment. Traditional methodologies are questioned, leading to an exploration of market microstructure and the integration of technologies like machine learning. Chapter 1 sets the stage by discussing the evolving nature of financial markets and the necessity of adapting research methodologies. Chapter 2 analyzes the regulation of trading speed, revealing trade-offs in market liquidity and price discovery. Chapter 3 focuses on detecting and mitigating mini flash crashes, leveraging machine learning to develop an Early Warning System. Chapter 4 examines the efficacy of speed bump mechanisms in reducing mini flash crashes, highlighting both benefits and unintended consequences. Overall, the dissertation enhances understanding of HFT's effects on market dynamics, risk management, and regulation. This dissertation contributes to a deeper understanding of the ramifications of high-frequency trading on market dynamics, risk management strategies, and regulatory paradigms.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

Writing the acknowledgments section of my dissertation has been a daunting task. Despite my best efforts to recognize all those who supported me, I worry that I may inadvertently overlook someone given the multitude of individuals who contributed to my journey.

First and foremost, I extend my sincere gratitude to my supervisor, Dr. Ke Xu, for her unwavering support, invaluable guidance, and continuous encouragement throughout my Ph.D. research.

I am grateful to Dr. Vasco Gabriel and Dr. Xuekui Zhang for their insightful suggestions and guidance, as well as for serving on my Doctoral committee. Additionally, I extend my genuine appreciation to Dr. Xinwei Zheng and Dr. Yu-Lun Chen, my research co-authors, for their valuable support and guidance in providing research data.

I would like to thank the faculty and staff of the Department of Economics, particularly Dr. Paul Schure, Dr. Tao Wang, and Dr. Justin Wiltshire, for their support during my final year of Ph.D. studies. The financial support from the University of Victoria and the Department of Economics is highly appreciated.

Special thanks go to my cohort and lifelong friends from the New York University(NYU) MFE program, including Yu Yang and Kai Yan, whose companionship made my student life memorable. I am also grateful to Dr. Hao Wang at the University of Oregon(UO) and Dr. Ron Slivka at NYU for their profound impact on my outlook on life and my interest in quantitative finance research.

I extend my thanks to my former colleagues Christopher Koch, Sia Hosseinian, and Gigy Alexander at PricewaterhouseCoopers (PWC) in Jersey City, whose warmth and encouragement sustained me during challenging times.

Finally, I owe an immense debt of gratitude to my family, especially my parents, Guohui Liu and Yumei Zhan, for their support. I also want to express my heartfelt appreciation to my grandfather, Shiqing Zhan, and my grandmother, Fengying Liu, who passed away while I was studying abroad. In addition, I am deeply grateful to my great grandfather, Hongsheng Zhan, who survived the war, carried on our family legacy, and fostered my beautiful childhood, laying the foundation for my journey from a small village in China to the global stage. Their sacrifices and boundless love have guided me throughout this endeavor.

DEDICATION

To the indelible decade of my life, in New York, Hong Kong, and Victoria, 2014-2024.

Chapter 1

Introduction

The dynamics of markets have undergone fundamental shifts. Markets are different now in fundamental ways. The advent of high-frequency trading (HFT) in financial markets has catalyzed a profound transformation in market dynamics, presenting multifaceted challenges for policymakers and regulators operating at the microstructure level (O'Hara (2015)). While velocity is indeed a hallmark of HFT, its impact transcends mere transaction speed. The ascendance of HFT has fundamentally altered various aspects of markets, encompassing trading patterns, market architecture, and mechanisms governing liquidity provision and price discovery. Notably, microstructure has assumed paramount importance in this swiftly evolving landscape, exerting a pronounced influence on market outcomes.

Currently, high-frequency traders not only dominate the influx of order messages received by stock exchanges but also wield substantial influence over trading volumes. In the US stock exchanges, HFTs are acknowledged to generate upwards of 50 percent of the total traded volume (Banerjee and Roy (2023)).

Beyond velocity, the pervasive integration of artificial intelligence, including machine learning algorithms, into a plethora of market activities and trading strategies, underscores the potential contagion of algorithmic behavior and strategies, thus posing a formidable challenge to market stability and investor asset integrity. The ripple effects of high-frequency trading algorithms are widely recognized as a plausible catalyst for the Flash Crash event that transpired on May 6, 2010.

In light of the widespread adoption of artificial intelligence (AI) trading algorithms, financial markets have witnessed a notable surge in efficiency and speed, albeit accompanied by a corresponding escalation in systemic risks. Despite the lack of definitive causative factors for the May 6, 2010 Flash Crash event, recent years

have seen a proliferation of perplexing mini flash crashes in individual stocks, akin to persistent minor afflictions that continually threaten the stability of financial markets. Addressing the regulation and mitigation of systemic risks arising from the rapid proliferation of AI-driven trading behaviors has become an urgent priority for policymakers. This dissertation embarks on a comprehensive analysis of high-frequency trading behaviors and regulatory frameworks, examining market quality and systemic risk from multiple perspectives. Additionally, it addresses research questions, including market fairness and the microstructure information time frame proposed by O'Hara (2015) in the area of high-frequency microstructure.

In Chapter 2, my research delves into an empirical investigation of the widely debated speed bump—a mechanism designed to curb high-frequency trading (HFT) activities—and its effects on market liquidity and price discovery across diverse stock datasets. Focusing on the New York Stock Exchange (NYSE) American, notable as the second exchange to implement a speed bump, I scrutinize the impact of trading speed on market liquidity and price discovery. Leveraging historical data, I conduct a differences-in-differences analysis to comprehensively assess market quality, revealing nuanced effects of speed bumps on market dynamics.

My research findings shed light on the contentious issue of regulating the trading speed of high-frequency traders. Utilizing NYSE American data, I examine the impact of the speed bump policy on various market liquidity and price discovery metrics. The results of my analysis indicate that while the implementation of a speed bump can mitigate the costs associated with adverse selection and reduce the prevalence of informed trading, it also has a detrimental effect on price discovery as trading activity slows down. Thus, while the speed bump enhances liquidity by deterring certain trading behaviors, it simultaneously diminishes the informativeness of market prices, illustrating its dual nature as a regulatory tool.

Moving beyond conventional market quality metrics, in Chapter 3, my research transcends traditional approaches to investigate the detection and mitigation of mini flash crashes. Leveraging advanced machine learning methodologies, I analyze patterns preceding these sudden market disruptions, demonstrating the transformative potential of machine learning in bolstering market resilience and refining risk management practices. This investigation lays the groundwork for the development of an Early Warning System, providing actionable insights for market participants.

My findings underscore the substantial risk posed by mini flash crashes in contemporary electronic trading markets. Utilizing NYSE Trade and Quote data (TAQ), our

study employs sophisticated machine learning techniques to evaluate the probability of these events. While predicting mini flash crashes is akin to forecasting earthquakes—impossible to prevent but crucial for preparedness—our research uncovers discernible patterns preceding such occurrences, underscoring the efficacy of machine learning in this predictive endeavor.

Furthermore, in Chapter 4, I conduct a comprehensive causal effect analysis of the speed bump policy, aiming to evaluate its effectiveness as a protective measure against mini flash crashes. Drawing upon empirical evidence from the TAQ, our study represents a pioneering effort in assessing the impact of a speed bump on the likelihood of mini flash crashes.

My research introduces an innovative empirical approach that leverages machine learning techniques applied to real-time limit order book data for estimating the probability of mini flash crashes. The analysis reveals compelling insights, demonstrating that the implementation of a speed bump can indeed decrease the probability of identifiable patterns leading to mini flash crashes.

Through my empirical investigation, we categorize mini flash crashes as intricate market microstructure events characterized by nonlinear patterns within the real-time limit order book. Identifying the specific trading behaviors responsible for triggering these crashes presents a significant challenge, but our findings suggest that discernible patterns leading to market turmoil do exist, and the introduction of an order delaying policy proves effective in significantly mitigating the probability of these disruptive patterns emerging. Additionally, I addressed questions posed by O'Hara (2015) regarding the fairness among different categories of traders in high-frequency settings, as well as the adoption of a new time frame, replacing the traditional five-minute "lifetime" for evaluating trading activities.

Moreover, my results, contextualized within existing literature, highlight the speed bump's pronounced impact on short-term market dynamics, indicating its effectiveness in attracting more uninformed, noise traders into the market.

In summary, while the "speed bump" policy may initially appear somewhat nebulous or enigmatic, our study demonstrates its remarkable effectiveness in mitigating the probability of mini flash crashes while reshaping market dynamics. Collectively, these chapters provide a comprehensive exploration of the ramifications of high-frequency trading on market dynamics, risk mitigation strategies, and regulatory paradigms, offering valuable insights for navigating the complexities of contemporary financial markets in the age of HFT and AI.

Chapter 2

Speed bump and stock market quality: evidence from NYSE American

2.1 Introduction

Since the integration of information and communication technologies into financial markets during the late 20th century, markets have undergone significant transformations, becoming faster, more liquid, and more efficient. However, the advent of high-frequency trading (HFT) has introduced new challenges and risks into these markets, particularly in the fragmented trading landscape of the US and Canada, where even microseconds can make a difference. High-frequency traders leverage their speed advantage to engage in practices such as frontrunning order anticipation and latency arbitrage, as illustrated in Lewis (2014), where they exploit slower traders for unfair profits.

Another concerning phenomenon in the era of high-frequency trading is the occurrence of flash crashes, characterized by sudden and steep sell-offs triggered by trading algorithms interacting in rapid succession, leading to the swift withdrawal of orders from the market. The speed at which HFT operates can sometimes overwhelm the trading system, making it difficult to diagnose and address issues in real-time.

In response to these negative side effects, some market participants have proposed implementing a "speed bump" mechanism to slow down market transactions uniformly. The aim is to create a more measured trading environment, reducing the

likelihood of market crashes. When orders that deplete market liquidity cause contagion among trading algorithms, driving the market towards a crash, a speed bump can delay this contagion from reaching NYSE American. Given the rapid pace of high-frequency trading, this delay could potentially avert the crash. Reports from Osipovich (2019) indicate that several markets across stocks, futures, and currencies, spanning from Toronto to New York to Moscow, either have already implemented or are planning to introduce speed bumps or similar measures. Advocates argue that speed bumps can provide protection to slower investors against ultrafast trading strategies.

However, the proposal for speed bumps is not without controversy. Critics, including electronic-trading firms, argue that such measures would unnecessarily complicate markets and could unfairly advantage certain participants Osipovich (2019). The debate surrounding the implementation of speed bumps underscores the ongoing tension between the need for market efficiency and the desire for fairness and stability in financial markets.

The theoretical framework presented in Baldauf and Mollner (2020) offers insights into the consequences of the rapid speed of high-frequency trading (HFT), highlighting a tradeoff between liquidity and information production. These findings have sparked interest among regulators and investors regarding the impact of speed bumps on market liquidity and price discovery. Du and Zhu (2017) develops a theoretical model and finds that for small stocks, slower trading speeds at the minute and second levels improve market efficiency. This paper aims to empirically investigate the influence of trading speed on market liquidity and price discovery within NYSE American, particularly notable as the second stock exchange to implement a speed bump, in contrast to the founding principle of IEX, which incorporated a speed bump from its inception. Leveraging historical data from NYSE American, prior to the implementation of a speed bump, enables us to conduct a differences-in-differences analysis. Utilizing high-frequency data, our study evaluates market quality across key dimensions: price discovery and market liquidity. We find evidence can confirm Du and Zhu (2017)'s findings that slower down the trading speed will benefit the market efficiency.

Through a comparative analysis with Nasdaq, a stock exchange without a speed bump, we seek to elucidate the role of speed in fostering a fair and efficient market. Our focus lies particularly on market liquidity and price discovery, pivotal determinants of fair market value. However, we acknowledge the multifaceted nature of speed bumps and the need to consider additional factors, such as the informative-

ness of prices, to comprehensively assess their impact on market quality. Thus, our analysis endeavors to provide a holistic evaluation of speed bumps' effect on market quality, emphasizing the critical importance of market liquidity and price discovery in nurturing a "fair" and efficient market environment.

Our findings reveal differential impacts of speed bumps on price discovery and market liquidity. Market liquidity undergoes significant changes post-speed bump implementation, as evidenced by evaluations of liquidity metrics such as the cost of immediacy and adverse selection. Specifically, the effective spread, serving as a measure of the cost of immediacy, contracts following the introduction of the speed bump. In conjunction with analyses of price impact and impulse response, our findings indicate a decrease in adverse selection. This further bolsters the findings of Goldstein et al. (2023), who investigated how the accelerated speed of the ASX exchange engine facilitates more anticipatory trading, thereby exacerbating adverse selection for slower traders. However, in terms of price discovery, we observe a decrease attributable to the speed bump. Our empirical examination underscores the presence of a trade-off inherent in speed bump policies, highlighting the nuanced effects on market dynamics.

Numerous scholars have delved into the examination of speed bumps through theoretical models, shedding light on both their negative and positive implications for the market. For instance, Aldrich and Friedman (2023) constructs a theoretical model of IEX, illustrating that the speed bump configuration of IEX has the potential to enhance price efficiency and mitigate transaction costs. Similarly, Menkveld and Zoican (2017) demonstrate that faster exchanges with improved market liquidity may not necessarily translate into genuine acceleration of exchanges, as they might engender heightened zero-sum "duels" among fast traders. Conversely, Aoyagi (2022) posits that speed bumps could widen the bid-ask spread and exacerbate adverse selection. In our empirical investigation, we observe that the implementation of speed bumps decreases the effective spread by mitigating adverse selection, corroborating the findings of Harris (2013), who likens certain high-frequency trading (HFT) strategies to poker players anticipating opponents' moves and proposes order processing delays as a solution.

Moreover, Brolley and Cimon (2020) underscore that informed investors tend to migrate from exchanges equipped with speed bumps to those without. This observation resonates with our own research findings on price discovery, as assessed through the permanent-transitory (PT) decomposition method by Gonzalo and Granger (1995). Specifically, we discern a decline in price discovery on NYSE American subsequent

to the introduction of the speed bump. Furthermore, our analysis corroborates prior studies by confirming a shift in trading volume from NYSE American to Nasdaq following the implementation of the speed bump. Our investigation further reveals a partial loss of effective price discovery alongside a notable enhancement in market quality, indicating a migration of speed-dependent informed traders from NYSE American, while liquidity-providing traders, less reliant on speed, have favored trading on NYSE American.

In addition to theoretical projections, empirical studies offer valuable insights into the impact of nanosecond-latency trading on market liquidity. For instance, Peng et al. (2019) reveal that IEX has emerged as the exchange with the lowest adverse selection for small- and medium-sized stocks. Similarly, Ye et al. (2013) observe a reduction in market depth as exchange latency transitions from the microsecond to the nanosecond level. Leveraging data from IEX, Hu (2019) demonstrate that the speed bump implemented by the exchange enhances market liquidity by mitigating price impact. Moreover, Chakrabarty et al. (2020) evaluate the effects of the IEX speed bump on market quality across various dimensions, including quoted spread, effective spread, and realized spread, concluding that the speed bump significantly bolsters market liquidity.

In complement to existing literature, our study contributes further evidence indicating that speed bumps enhance market liquidity while potentially impeding price discovery on information sensitive to speed. Notably, another distinction between our research and Chakrabarty et al. (2020) lies in the pre-existence of NYSE American before the implementation of the speed bump, enabling us to conduct a robust difference-in-difference analysis to compare market quality pre- and post-speed bump. The availability of trade and quote data from NYSE American prior to the speed bump facilitates an examination of changes in price discovery subsequent to the speed bump's implementation.

Our analysis primarily focuses on market liquidity and price discovery, building upon the framework proposed by Hendershott and Moulton (2011). We confirm that the speed bump indeed augments realized spread and mitigates price impact. Additionally, our impulse response analysis reveals a notable decrease in adverse selection, aligning with the findings of Hasbrouck (1991a) and indicating a reduction in arbitrage opportunities post-speed bump. However, we also observe an increase in the cost of immediacy NYSE American following the speed bump, consistent with the findings of Chen et al. (2017). Drawing upon these insights, we conclude that

while the speed bump enhances market liquidity, it entails a trade-off with respect to price discovery and immediacy costs.

The subsequent sections of this paper are organized as follows: Section 2.2 offers an overview of the related literature, while Section 2.3 outlines the institutional background and describes the dataset utilized in our analysis. In Section 2.4, we delve into the empirical analysis, which encompasses discussions on price discovery, market liquidity, and simulation results. Section 2.5 evaluates price discovery through simulation modeling, comparing exchanges with and without speed bumps. Finally, Section 2.6 concludes our findings, while Section 2.7 presents robustness checks and further evidence through an extended analysis of relative price discovery and market liquidity between NYSE American and Nasdaq.

2.2 Literature

The literature exploring the dynamics of speed bumps and trading velocity has extensively investigated their impacts on market quality, adverse selection, and the cost of immediacy. In addition to these conventional metrics, our study contributes to the burgeoning literature by examining relative price discovery as an additional dimension of market efficiency.

Foucault et al. (2016) offer theoretical insights grounded in optimal trading strategies of fast-informed speculators, affirming that the behavior of fast speculators aligns well with observed patterns in high-frequency trading (HFT). Their analysis underscores the significance of speed, indicating that traders with a speed advantage can effectively forecast short-term price changes, exerting a substantial influence on a significant portion of trading volume in the market. Conversely, Menkveld and Zoican (2017) caution against the assumption that faster exchanges inherently boast superior liquidity, as heightened speed may engender a proliferation of zero-sum duels among rapid traders.

From an economic perspective, Biais et al. (2015) highlight the possibility that investments in fast trading technology might surpass the social optimum, suggesting potential inefficiencies in the allocation of resources within the market ecosystem. These theoretical propositions find empirical support in the work of Baron et al. (2019), who observe that market orders executed by high-frequency traders tend to yield a more pronounced price impact, further accentuating the complexities associated with rapid trading environments.

Moreover, Shkilko and Sokolov (2020) shed light on a unique scenario wherein market liquidity experiences enhancement due to environmental factors, such as disruptions in microwave networks caused by adverse weather conditions. This empirical observation underscores the intricate interplay between technological infrastructure and environmental dynamics in shaping market behavior.

Zooming in on the phenomenon of speed bumps, numerous scholars have leveraged this mechanism to delve into the evolving role of trading speed within the stock market landscape. Aoyagi (2022) undertakes a meticulous analysis, constructing a theoretical market model featuring competitive slow uninformed market makers, risk-neutral high-frequency traders, and liquidity traders subject to a liquidity shock in a transparent market setting. By delineating the optimal behavior of market makers and forecasting the strategic adjustments of high-frequency traders in response to speed considerations, Aoyagi (2022) predicts that the implementation of a speed bump may intensify adverse selection issues and expand the bid-ask spread. Our empirical inquiry, centered on NYSE American, corroborates Aoyagi (2022)'s observation of an increased bid-ask spread subsequent to a 350-microsecond speed bump. However, diverging from Aoyagi (2022)'s projections, our findings suggest a potential reduction in adverse selection opportunities attributable to optimized trading speed. It's noteworthy that our empirical analysis pertains specifically to NYSE American's structured speed bump, deviating from Aoyagi (2022)'s premise of randomness, thus introducing a notable distinction between our respective methodologies.

Similarly, Brolley and Cimon (2020) employ a model to anticipate the ramifications of a speed bump on exchanges' competitive dynamics within a multi-exchange framework. Their theoretical framework assumes the coexistence of two initially identical markets trading the same risky security with stochastic payoffs. Upon the implementation of a speed bump in one exchange, Brolley and Cimon (2020) predict the migration of informed traders to the unaffected exchange lacking the speed impediment. Consistent with their forecast, our analysis of price discovery across NYSE American and NASDAQ supports the notion of informed trader migration away from speed-bumped exchanges. However, while Brolley and Cimon (2020) anticipate heightened total trading volume and narrower quoted spreads in the "slow" exchange, our examination of NYSE American and NASDAQ post-speed bump yields contrary outcomes.

In contrast to the framework of two-exchange competition, Pagnotta and Philippon (2018) develop a theoretical model of multi-exchange competition welfare, aimed

at dissecting the influence of speed and speed-related costs on exchange welfare. Their premise challenges the notion that constraining trading speed is the optimal regulatory approach, arguing that the fastest traders in the market often operate at speeds slower than regulators deem optimal. Their findings advocate for incentivizing slower traders to upgrade their technologies rather than imposing speed restrictions. While our study diverges from assessing exchange welfare, opting instead to scrutinize relative adverse selection and price discovery, our objective remains aligned in assessing the efficacy of speed bumps as a means to enhance market liquidity, albeit with a nuanced trade-off.

Beyond the theoretical constructs put forth by scholars such as Brolley and Cimon (2020) and Pagnotta and Philippon (2018), a growing body of empirical research offers insights into the operational dynamics of speed bumps across exchanges. Much of this empirical evidence suggests a positive impact of speed bumps on market quality, albeit often accompanied by trade-offs. For instance, Chen et al. (2017) scrutinizes TSX Alpha's performance post-speed bump, revealing an increase in market quoted spread—an indicator of immediacy cost—suggesting that speed bumps might not universally attract all investors due to associated declines in trade volume. Our findings resonate with this observation, yet we extend the analysis by incorporating additional adverse selection measures, revealing potential reductions in adverse selection costs due to speed bumps.

Similarly, Chakrabarty et al. (2020) delve into IEX's data to unveil the multifaceted impacts of speed bumps. Their investigation unveils reductions in reaction time, order detection, and back-running strategies following the implementation of the speed bump. Furthermore, they observe enhancements in IEX's market quality, evidenced by decreases in quoted spread, quote-to-trade ratio, and order imbalance, alongside increases in quoted depth and quote life. Echoing these sentiments, Hu (2019) leverages IEX data to underscore the speed bump's role in augmenting market liquidity, particularly through mitigating price impact. By evaluating market quality across various metrics—quoted, effective, and realized spreads—Hu (2019) concludes that speed bumps bolster market liquidity, underscoring their potential to enhance market efficiency.

In our examination of NYSE American, we undertake a similar calculation of market liquidity using the three measures mentioned. However, our empirical findings deviate somewhat from those presented by Hu (2019); notably, we observe an increase in both quoted spread and realized spread following the implementation of the

speed bump. Expanding upon the analysis of these three measures—quoted, effective, and realized spreads—we introduce two additional metrics, namely price impact and impulse response (Hasbrouck (1991a)). Subsequently, we delve deeper into the evaluation of market liquidity, focusing on two key facets: the cost of immediacy and adverse selection. Our study aligns with Hu (2019) in terms of the consistency of results across these market measures.

Following the footsteps of IEX and NYSE American, NASDAQ introduced its own speed bump known as the Midpoint Extended Life Order (MELO). Gonçalves et al. (2019) undertake an empirical examination of MELO's impact on NASDAQ. Employing metrics such as cancel-to-trade, trade-to-order, order-to-trade ratio, hidden rate, and volume coefficient of variation, they conclude that market liquidity improves post-MELO implementation—a finding that corroborates despite the utilization of different metrics than those employed in our study.

We adhere to established methodologies in the assessment of market liquidity and price discovery. As a principal contribution of this paper, we employ relative price discovery as a metric to evaluate stock market quality, elucidating the process by which information is assimilated into security prices. Hasbrouck (1995) introduces the concept of cointegration within microstructure models, offering a well-established approach to discerning the respective contributions to price discovery across different markets. Moreover, Gonzalo and Granger (1995) utilizes cointegrated vector autoregressive (CVAR) models as a framework to estimate error correction terms, which serve to quantify the speed of price adjustment towards a long-run equilibrium among prices in various markets. By transforming these error correction coefficients into relative price discoveries that sum to one, we can discern how changes in one market impact the equilibrium in another. While this approach may encounter challenges in interpreting the effects of policies that influence total information incorporation into prices across both markets, our study mitigates this issue by employing NASDAQ as a control group, which lacks a speed bump. Consequently, any alterations in absolute price discovery can be attributed to the presence of the speed bump. Nielsen and Popiel (2018) offers a Matlab package facilitating the computation of estimators and test statistics, along with tools for calculating P values and critical values for cointegration rank tests. This body of literature underscores the suitability of the CVAR model framework for comprehensive price discovery analysis.

More than price discovery, this paper also studies other market liquidity measures followed by Hendershott and Moulton (2011). We examine market liquidity by looking

into cost of immediacy and adverse selection. Besides some papers we have mentioned above, many research papers refer to these measures including Xu et al. (2020) and Wang et al. (2010). In the calculation of these adverse selections rely on models suggested by Hasbrouck (1991a), which is briefly reviewed in Section 2.4.3, Section 2.4.4 and Appendix 2.8.

2.3 Institutional Background and Data Description

With trading speeds experiencing rapid escalation and the US and European stock markets becoming increasingly fragmented, it has become customary for the same stock to undergo trading across multiple exchanges. This scenario presents an opportunity for High-Frequency Trading (HFT) firms to engage in arbitrage and assume the role of "fast informed traders," a notion perceived by many as conferring an unfair advantage. Scholarly literature, exemplified by Yang and Zhu (2019), highlights the potential for front-runner or back-runner trading opportunities in today's exceedingly fast-paced market environment. Consequently, numerous market participants have raised concerns about this issue, prompting a fundamental question: should trading speeds be slowed down?

IEX stands out as the pioneer in implementing the speed bump, introducing a 350-microsecond delay before executing orders, thus becoming the inaugural stock market to adopt such a measure. Subsequently, IEX garners interest from investors wary of HFTs and manages to capture a market share exceeding 2%. In response, NYSE, as its primary competitor, experiences a substantial decline in market share to IEX by May 2017, prompting the submission of an application for a speed bump to the SEC. However, the SEC rejects their application, granting approval solely to NYSE American, which boasts a listing of approximately 370 small and mid-sized companies. On July 24, 2017, NYSE implements the speed bump. Nevertheless, in November 2019, NYSE issues an announcement indicating dissatisfaction with the performance of their speed bump. Notably, several market quality metrics, including quoted spread and daily volume, register declines. Observing a continuous decline in market share, NYSE decides to discontinue the speed bump. Despite NYSE American's eventual decision to retract the measure, the episode provides a valuable opportunity for examination.

As stated in the official announcement from NYSE American, the speed bump leads to an increase in the quoted spread of stocks listed on the exchange, a finding consistent with our own results regarding quoted spread. However, in addition to the quoted spread, we employ difference-in-difference methods to evaluate other market metrics, offering a more nuanced understanding of market liquidity. Our findings indicate that although the quoted spread widens following the implementation of the speed bump, the effective spread experiences a narrowing. Furthermore, we observe a reduction in adverse selection costs on NYSE American following the introduction of the speed bump. This evidence suggests that while slower trading may widen the bid-ask spread, potentially creating the impression of deteriorating market liquidity, the implementation of a speed bump can effectively mitigate opportunities for front-running and back-running trades, thereby enhancing overall market fairness and efficiency. Thus, while not unequivocally detrimental as portrayed by NYSE American, the speed bump represents a double-edged sword.

According to Osipovich (2019), as of 2019, more than a dozen markets spanning stocks, futures, and currencies had either implemented or expressed interest in implementing speed bumps to mitigate rapid trading. The earliest instances of speed bump implementation trace back to ParFX and EBS in 2013, both operating within the domain of Foreign Exchange (FX). Subsequently, the Securities and Exchange Commission (SEC) granted approval for IEX to transition into an exchange on June 17, 2016, sparking a wave of competition within the stock exchange industry, with other notable exchanges following suit. Among these early adopters, TSX Alpha, a Canadian exchange, introduced a randomized speed bump ranging from 1 to 3 milliseconds in 2015. Following TSX Alpha's initiative, Refinitive, an FX exchange, embraced the trend in April 2016, followed by futures exchange Eurex in 2017. While not all exchanges have experienced favorable outcomes with speed bumps, this mechanism has nonetheless emerged as one of the most widely discussed and adopted trading protocols across exchanges.

Following the implementation of the speed bump by NYSE American, Nasdaq introduced MELO as a complementary measure. Moreover, Intelligent Cross incorporated a speed bump in 2018, with the Moscow Exchange joining the trend in 2019. Several other exchanges are also contemplating joining the speed bump trend, pending approvals from the SEC.

Exchanges vary in their approaches to designing speed bumps. Refinitiv, for instance, offers protections against aggressive latency arbitrage strategies. TSX im-

plements a randomized speed bump solely for market orders, while excluding limit orders, potentially safeguarding liquidity. Meanwhile, both IEX and NYSE American opt to delay all incoming orders by 350 microseconds, striving for fairness across all order types.

Table 2.1 provides an overview of the reality and historical implementation of various iterations of speed bumps in FX and stock exchanges.

Table 2.1

This paper leverages trade and quote data from the NYSE Daily TAQ dataset for NYSE American and Nasdaq, spanning from June to August 2017, encompassing the date of NYSE American's implementation of the speed bump on July 24, 2017. Nasdaq, boasting the largest number of commonly traded stocks with NYSE American among all stock exchanges, serves as an apt control group. Prior to the speed bump's implementation, NYSE American facilitated trading for approximately 240 stocks. However, post-implementation, this figure surged dramatically to over 4,500, as depicted in Figure 2.1.

Our empirical findings offer partial insights into this phenomenon. Subsequent to the enforcement of the speed bump policy, NYSE American witnessed a decline in informed traders, manifested by a reduction in price discovery. Conversely, relative market quality exhibited enhancements. While information share and component share demonstrated relative declines, a notable upswing in information leadership share was observed. This suggests that post-implementation, with relative market noise mitigated, price discovery experienced enhancements. The presence of the speed bump deterred certain informed traders contributing to market noise, fostering increased confidence among investors and stock financiers gravitating towards the speed bump. Consequently, a surge in participants transacting on NYSE American was noted following the speed bump's enforcement.

To scrutinize stocks highly susceptible to fast trading technologies such as front running and back running, our focus lies on common stocks traded on NYSE American, IEX, and Nasdaq. Given the limited activity of most stocks on NYSE American, we select the most actively traded stocks, those transacted daily over the past three months on both NYSE American and Nasdaq, totaling 50 stocks. In our robustness check, we extend our analysis to encompass 103 stocks commonly traded on both NYSE American and Nasdaq, yielding consistent albeit slightly less significant results. Hence, we juxtapose the trading activities of these 50 stocks traded on both

NYSE American and Nasdaq concurrently over the entire three-month duration.[1]

Figure 2.1

Figure 2.2 illustrates the impact of the speed bump removal on the number of stocks (market share) traded on NYSE American. Contrary to expectations, the removal does not yield a significant effect on the number of stocks being traded, diverging from the marked impact observed during its implementation phase. This outcome contrasts with the anticipated positive contribution to NYSE American's market share, as highlighted in NYSE American's official announcement. Instead, the market share of NYSE American exhibits a high degree of volatility.

Figure 2.2

Furthermore, given that NYSE American specializes in trading small and mid-size equities, all 50 selected stocks fall within this category. Figure 2.3 presents the exchange average bid-ask spread for both NYSE American and Nasdaq. Analysis of this figure reveals a consistent pattern where Nasdaq typically exhibits a wider bid-ask spread compared to NYSE American. Notably, following the implementation of the speed bump, the bid-ask spread of NYSE American registers an increase. This observation suggests a rise in the cost of immediacy attributable to the speed bump.

Figure 2.3

The descriptive statistics for the 50 stocks traded in NYSE American and Nadasq during the periods before and after the speed bump are shown in Table 2.2. From the summary statistics below, a significant observation is evident: the implementation of the speed bump has led to a notable decrease in market depth and trading volume on NYSE American. However, it is also observed that there has been a substantial increase in the number of stocks traded on NYSE American. This indicates that the speed bump, by slowing down trading speeds and curbing latency arbitrage, has instilled strong confidence among ordinary, low-frequency, and slow-paced traders and

[1]As we want to escape from any other events that may hit the market, we restrict our time range to 3 months. More than that, because we use nanosecond timestamp data, 3 months has covered a big bunch of data. We expect they are enough. Because of data outliers, we remove one of them (ticker: SEB). When we do the data filtering, we restricted the trading data in regular trading hours from 9:30 am to 4 pm. We use only trades for which TAQ's CORR filed is zero, one, or two and for COND field is either blank or equal to @, E, F, I, J, or K. Obviously, we eliminate trades with nonpositive prices or quantities. We also remove trades with prices more than(less than) 150%(50%) of the previous trade price. After that, we restrict quotes for which TAQ's MODE field is equal to 1, 2, 6, 10, 12, 21, 22, 23, 24, 25, or 26. Then we eliminate quotes with nonpositive prices or sizes or with bid prices greater than the asking price. We also exclude quotes when the quoted is greater than 25% of the quote midpoint or when the asking price is more than 150% of the bid price. There are 51 stocks commonly traded on NYSE American, Nasdaq and IEX in every trading day through 3 months, one of them has a significant missing data problem, we use the left 50 stocks.

stockholders. Consequently, they have opted to trade stocks on NYSE American, despite their smaller trading volumes compared to the large number of fast-paced high-frequency traders.

Table 2.2

2.4 Empirical Analysis

The empirical analysis section is subdivided into four distinct subsections, each addressing a particular aspect. The initial subsection delineates the methodology employed for assessing price discovery. Subsequent to this, three subsequent subsections undertake a difference-in-difference analysis to gauge the influence of the speed bump on various facets of market quality measures. Each subsection is tailored to scrutinize a specific category of market quality measures, elucidating the ramifications of the speed bump on the stock market.

2.4.1 Price Discovery

This sub-section examines the impact of the speed bump on price discovery. Firstly, we will show the price discovery which is calculated by the model of cointegration model (CVAR). Cointegration is a classical model in the time series modeling area, which models two-time series moving together in a long run. The cointegration in microstructure models from Hasbrouck (1995) shows when security is traded in two separate markets and that the evolution and linkage of the prices in the markets are given as:

$$p_{Nasdaq,t} = p_{Nasdaq,t-1} + \omega_t \tag{2.1}$$

$$p_{NYSEAmerican,t} = p_{Nasdaq,t-2} + \epsilon_t \tag{2.2}$$

where the $p_{Nasdaq,t}$, and $p_{NYSEAmerican,t}$ are price variables such as the midpoint of the bid and ask quotes (mid-quote) of stock in this paper. The ω_t and ϵ_t are zero-mean disturbances that are also independently and identically distributed and uncorrelated with each other. The $p_{Nasdaq,t}$ is a stock mid-quote in the Nasdaq. It follows a random walk with an increment ω_t. The $p_{NYSEAmerican,t}$ tracks $p_{Nasdaq,t}$ lagged two periods as it is being traded at NYSE American. $p_{NYSEAmerican,t}$ also

reflects a random error. As they are mid-quote of the same stock, with the same fundamental value, although both prices are integrated, they will never be far away from each other. The difference between the two prices is:

$$p_{Nasdaq,t} - p_{NYSEAmerican,t} = p_{Nasdaq,t} - (p_{Nasdaq,t-2} + \varepsilon_t) = \omega_t + \omega_{t-1} - \varepsilon_t \tag{2.3}$$

which is a stationary random variable. When a linear combination of integrated variables is stationary, the variables are said to be cointegrated. Then the price changes relation could be represented through the vector moving average (VMA) in terms of current and lagged innovations:

$$p_{Nasdaq,t} = \omega_t. \tag{2.4}$$

$$p_{NYSEAmerican,t} = (p_{NYSEAmerican,t-1} - p_{Nasdaq,t-1}) - p_{Nasdaq,t-1} + \varepsilon_t. \tag{2.5}$$

As both $p_{Nasdaq,t}$ and $p_{NYSEAmerican,t}$ are random walks that have the same order ($I(1)$), we can assert that (4) and (5) represent a system of time series cointegrated by the Engle-Granger representation theorem. We call this model the "error correction model", traders in the slow market (NYSE American) are responding to the price discrepancy Hasbrouck (1995). Based on this error correction model, Figuerola-Ferretti and Gonzalo (2010) factor model offers a way to calculate permanent and transitory composition (P-T decomposition) of X_t=$(p_{Nasdaq,t}, p_{NYSEAmerican,t})$ which is a difference-stationary sequence. The P-T decomposition for X_t is a pair of stochastic processes (P_t, T_t) such that P_t is difference stationary and T_t is covariance stationary. Figuerola-Ferretti and Gonzalo (2010) have theoretically proved that the only shocks that can affect the long-run forecast of X_t are coming from P_t. Considering this theoretical proof, we represent the X_t in the form of cointegration:

$$X_t =' X_{t-1} + \sum_{i=1}^{\infty} \Gamma_i \Delta X_{t-i} + \varepsilon_t, \tag{2.6}$$

where $\Delta = I - L$, with L the lag operator. As the X_t is a 2×1 vector of $I(1)$, there exists $\beta = (1, -1)'$ such that $\beta' X_t$ is $I(0)$. The columns of β' show the cointegrating vectors such that $\beta' X_t$ are the stationary linear combinations of the variables in the

system, which show the long-run equilibrium relations. α shows the adjustment or loading coefficients that represent the speed of adjustment towards equilibrium for each of the variables. According to the P-T decomposition, the common permanent component of X_t is $W_t = \alpha'_\perp$, where $\alpha'_\perp \alpha = \alpha' \alpha_\perp = 0$. This common permanent component constitutes the dominant price or the long-run market price, in the sense that information that does not affect W_t will not have a permanent effect on X_t; therefore $\alpha_\perp$ is a measure of the price discovery contribution of the NYSE American and Nasdaq. Figure 2.4 shows GSAT stock, which is one of the highest active stocks traded by both NYSE American and Nasdaq. The mid-quotes of the two exchanges are very close, which could be a hint of CVAR model fitting.

Figure 2.4

Table 2.3 shows the average relative price discovery of the 50 commonly traded stocks in NYSE American and Nasdaq. They are calculated using a cointegration vector autoregressive model implemented by Nielsen and Popiel (2018) through an accompanying Matlab package. As we can observe through the summary data, the relative price discovery of NYSE American, which is a much smaller exchange, is significantly lower than Nasdaq, then its price discovery declines after the speed bump.

Table 2.3

2.4.2 Price Discovery Linear Regression

Based on the relative price discovery observed during all trading days from June 2017 to August 2017, we conducted the following regression analysis using a panel data approach with fixed effects of stock:

$$PriceDis_{i,t} = \alpha_i + \beta \text{SpeedBump}_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_j \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.7)$$

The regression model we used includes variables to capture the relative price discovery and other factors affecting the stock prices. The relative price discovery is represented by the variable $PriceDis$, which is calculated as the ratio of the price discovery on NYSE American to that on Nasdaq, estimated using the CVAR method.

To account for the potential impact of individual stocks, we introduce the stock fixed effect α. Additionally, we use an indicator variable called *speedbump*, which

takes the value of zero before the implementation of the speed bump and one after it is implemented, to capture the effect of the speed bump on the stock prices.

We also consider the opening value of the CBOE's VIX index on day t as a measure of volatility. To control for the potential influence of other factors that may affect the stock prices, we include two stock-level control variables, daily turnover difference and daily stock volatility difference, represented by $ControlVariable_{i,t,q}$. These control variables are calculated using the method proposed by Alizadeh et al. (2002).

Finally, we also consider the market capitalization of the stock as an additional variable for the linear regression model without fixed effect. By including all these variables, we aim to capture the various factors that may influence stock prices and provide a more comprehensive analysis.

Table 2.4

Table 2.4 displays the outcomes of the linear regression analysis conducted both with and without stock fixed effects. The regression coefficients associated with the dummy variable *speedbump* reveal a significant impact of implementing the speed bump on the relative price discovery of NYSE American. Specifically, the coefficient estimate indicates a decrease in the relative price discovery of NYSE American following the adoption of the speed bump, suggesting a reduction in price discovery post-implementation. This finding implies that the speed bump decelerates the assimilation of information into stock prices, thereby curtailing opportunities for informed trading on NYSE American. Consequently, informed investors may opt to migrate to other exchanges lacking a speed bump, such as Nasdaq.

This implies that NYSE American, similar to IEX, transitions into an exchange aimed at curbing latency arbitrage. However, it's important to acknowledge that this pursuit of fairness comes with associated costs. The introduction of the speed bump impedes the swiftness of information assimilation, indicating a potential slowdown in the responsiveness of stock prices to new information. However, it's essential to recognize that fairness initiatives entail inherent trade-offs.

In addition, we use the information share and component share (Hasbrouck (1995)) and information leadership share (Patel et al. (2020) and Yan and Zivot (2010)) to represent the price discovery and conduct the following regression analysis with same panel data with fixed effects of stock.

$$IShare_{i,t} = \alpha_i + \beta \text{SpeedBump}_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.8)$$

$$CShare_{i,t} = \alpha_i + \beta \text{SpeedBump}_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.9)$$

$$ILShare_{i,t} = \alpha_i + \beta \text{SpeedBump}_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.10)$$

Table 2.5 presents the results of the linear regression analysis conducted with and without accounting for stock fixed effects. The regression coefficients for the dummy variable *speedbump* illustrate the substantial impact of its implementation on the relative price discovery of NYSE American, measured by information share, component share, and information leadership share. Notably, we observed inconsistencies among these metrics. Both information share and component share coefficients decrease with the introduction of the speed bump, indicating a reduction in their contributions to price discovery, boostering our results in Table 2.4. While the result for information share lacks statistical significance, the coefficient for component share is highly significant, underscoring the speed bump's pronounced effect on the component share of NYSE American. Conversely, the coefficient for information leadership share increases, suggesting an amplified role in reflecting new information, as discussed by Putniņš (2013). According to Putniņš (2013), information leadership share isolates relative price discovery, emphasizing new information while attenuating the influence of relative noise. This reaffirms our earlier observations that the speed bump on NYSE American decelerates the assimilation of information into stock prices, resulting in diminished price discovery. However, when we alleviate concerns regarding fast relative noise in the markets in second level, we observe an increase in price discovery when fundamental value changes. This result is further bolstered by our subsequent empirical findings on adverse selection in the following subsections.

2.4.3 Cost of immediacy

In this specific sub-section, we will delve into a comprehensive analysis of the cost of immediacy in trading. Utilizing Nasdaq as the control group, we will employ a differences-in-differences analysis. Following the methodology outlined by Hendershott and Moulton (2011), we will gauge the cost of immediacy through two key metrics: the percentage quoted spread and the effective spread.

The percentage quoted spread $QSpread_{i,k,t}$ is defined as the difference between the best ask quote $ask_{i,k,t}$ and the best bid quote $bid_{i,k,t}$ at the time of a trade, divided by the prevailing midpoint $m_{i,k,t}$ of the bid and ask quotes. Mathematically, we can represent this as:

$$QSpread_{i,k,t} = \frac{ask_{i,k,t} - bid_{i,k,t}}{m_{i,k,t}} \tag{2.11}$$

In addition to the quoted spread, we also calculate the effective spread for each trade. The effective spread is defined as the disparity between the estimated midpoint of the quote and the actual transaction price. This metric aids us in quantifying the supplementary cost incurred by a trader to execute a trade promptly.

By incorporating these metrics, we aim to furnish a comprehensive analysis of the cost associated with immediacy, thereby empowering traders to make judicious decisions regarding their trading strategies. Mathematically, this can be represented as:

$$ESpread_{i,k,t} = \frac{2q_{i,k,t}\left(p_{i,k,t} - m_{i,k,t}\right)}{m_{i,k,t}} \tag{2.12}$$

with i, k, t, as stock i at time k on day t. The trading indicator $q_{i,k,t}$ equals one for buyer-initiated trades and a negative one for seller-initiated trades.The $p_{i,k,t}$ is the transaction trade price, and $m_{i,k,t}$ is the mid-quote which is the latest mid-quote initiated before the transaction trade. The trading indicator is inferred by Lee and Ready (1991). From our descriptive statistics, we note a notable increase in both the quoted spread and effective spread subsequent to the introduction of the speed bump, evident across both NYSE American and Nasdaq. However, the magnitude of this increase is significantly more pronounced for NYSE American compared to Nasdaq. To delve deeper into the impact of the speed bump on these spread variables, we employ a differences-in-differences analysis utilizing panel data regression, incorporating fixed effects for stocks and dates. The regression equation takes the following form:

$$\text{Sprd}_{i,t} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.13)$$

where $Sprd_{i,t}$ is the log difference of percentage quoted spread or effective spread for stock i on day t between NYSE American and Nasdaq, α_i is the stock and date fixed effect, and $SpeedBump_t$ is an indicator variable taking the value of 1 after speed bump implementation, 0 otherwise. Volatility is the opening value of CBOE's VIX index on day t. The other independent variables are control variables. The $ControlVariable_{i,t,q}$ represents two stock-level control variables: the daily turnover difference and the daily stock volatility difference between NYSE American and control group Nasdaq(calculated as Alizadeh et al. (2002)). For the linear regression without fixed effect, we also add difference of market capitalization of the stock.

Table 2.6

Table 2.6 presents the results of a regression analysis conducted with and without fixed effects. In the linear regression, we estimated the coefficients of the dummy variable speed bump, which allowed us to observe the causal effect of the speed bump on the quoted spread, which is an indicator of the cost of immediacy. We observed that the speed bump had a positive effect on the quoted spread, suggesting an increase in the cost of immediacy. Given that the speed bump delays various trading activities in NYSE American, including the cancellation of standing limit orders, it is plausible that high-frequency trading activities could be diminished. This likely contributes to the observed higher quoted spread.

However, when we examined the effective spread, which is the difference between the executed price and the midpoint of the bid-ask spread, we found that the cost of immediacy actually decreased due to the presence of the speed bump. Another interesting observation is that only inside-quote trades cause a difference between the effective spread and the quoted spread. Inside-quote trades are more closely related to market makers' behaviors and conditional trades, demonstrating that the speed bump impacts market makers' reactions. We further decomposed the effective spread into permanent and transitory components to evaluate changes in adverse selection and liquidity provider profits. This analysis helped us to understand why the coefficient of the dummy variable's effective spread was negative, and we will discuss this finding in the adverse selection section.

Interestingly, it is noteworthy that the effective spread is a measure of market liquidity that is directly linked to transactions. In light of this, the results also suggest that although the speed bump worsened market liquidity, it successfully resolved the front running and back running problems, thereby increasing "real market liquidity." This result confirm Goldstein et al. (2023)'s finding that the probability of limit order execution increases for when slower traders when market access speeds decrease.

2.4.4 Adverse Selection

In this subsection, we delve into the examination of the impact of the speed bump on the adverse selection costs incurred by investors. Adverse selection delineates a circumstance wherein either buyers or sellers possess information that is not available to their counterparties, thereby engendering informational asymmetry within the stock market. The adverse selection cost serves as a metric to quantify the extent of this asymmetry. To scrutinize this cost, we employ a decomposition of the effective quoted spreads into their permanent and transitory components. By focusing on the permanent component, we aim to assess the enduring effects of individual trades, a methodology widely employed for evaluating adverse selection costs within the literature.

To calculate the permanent effect, we use the percentage price impact for each trade in stock j at time k on day t, as follows:

$$PImpact_{i,k,t} = \frac{2q_{i,k,t}\left(m_{i,k+5,t} - m_{i,k,t}\right)}{m_{i,k,t}} \tag{2.14}$$

with i, k, t, as stock i at time k on day t. The trading indicator $q_{i,k,t}$ equals one for buyer-initiated trades and a negative one for seller-initiated trades. In this equation, we rely on Lee and Ready (1991) to determine our trading indicator. The value of the indicator is one for buyer-initiated trades and negative one for seller-initiated trades. The midpoint of the matching quote, $m_{i,k,t}$, represents the latest midquote that initiated before the transaction trade. The realized spread reflects the temporary effect, which estimates the profit that could be earned by the liquidity provider. The effective quoted spread can be decomposed into two components, the permanent effect, and the temporary effect. Therefore, the realized spread is the difference between the percentage effective spread and the price impact mentioned earlier:

$$RSpread_{i,k,t} = \frac{2q_{i,k,t}(p_{i,k,t} - m_{i,k+5,t})}{m_{i,k,t}} \tag{2.15}$$

Here, $RSpread_{i,k,t}$ denotes the realized spread of trade i at time k on day t, $q_{i,k,t}$ is the trading indicator, $p_{i,k,t}$ is the trade price, and other variables are defined as above. We use the same stock i at time k on day t notation for consistency. This equation utilizes the trading indicator to differentiate between buyer-initiated and seller-initiated trades. The midpoint of the matching quote represents the latest midquote that occurred before the trade, and the realized spread estimates the temporary effect. Following Hendershott and Moulton (2011), we calculate both realized spread and price impact by five-minute.

Another measure we calculate is the impulse response measure introduced in Hasbrouck (1991a) (See Appendix). As Hasbrouck (1991a) demonstrates, impulse response measures adverse selection as it estimates potential lagged adjustments to the information in trades and quotes. Panel A, B, C of Table 2.7 show Price Impact, Realized spread, and Impulse response of NYSE American and Nasdaq.

Table 2.7

For estimating the effect of a speed bump on adverse selection measures, we also run the following regression for each adverse selection measure by the panel data with fixed effects of stock. The linear regression without fixed effect is also implemented:

$$adv selection_{i,t} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{2.16}$$

In this equation, the $adv selection_{i,t}$ is the average impulse response, price impact, or realized spread measure for stock i on day t. On the independent variables side, α_i are stock fixed effects. Then we use $SpeedBump_t$ as an indicator variable equals one for trading days after speed bump implementation, and equals zero otherwise. $Volatility_t$ is the opening value of CBOE's VIX index on day t. $ControlVariable_{i,t,q}$ represents two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as Alizadeh et al. (2002)). For the linear regression without fixed effect, we also add market capitalization of the stock. We give empirical results in Table 2.8.

Table 2.8

Table 2.8 displays the results of our regression analysis conducted over a three-

month period spanning 64 trading days. This table focuses on the adverse selection feature of market liquidity. Our findings indicate that the implementation of the speed bump has a negative impact on price impact, which demonstrates a decline in the permanent effect of trades due to the presence of the speed bump. [2]

In contrast, the profit of liquidity providers increases, as evidenced by the observed increase in realized spread. Our analysis further reveals that the effective spread increases due to the increase in realized spread. Additionally, the impulse response measure, which is robust to price discreteness and lagged price adjustment, shows a significant decrease after the implementation of the speed bump.

Collectively, our examination of the impulse response, price impact, and realized spread yields insights into the efficacy of the speed bump in mitigating the costs associated with adverse selection. This discovery substantiates assertions advanced by proponents of the speed bump, advocating its capacity to curtail opportunities for latency arbitrage and foster equitable exchanges. Our findings partially support the research by Aquilina et al. (2024), suggesting that the implementation of a speed bump can reduce latency arbitrage, consequently lowering adverse selection. In addition, our investigation also reaffirms the conclusions drawn by Goldstein et al. (2023), as evidenced by the observation that a slower pace of trading correlates with a heightened likelihood of favorable executions for slower traders. This phenomenon aligns with a reduction in adverse selection. However, concomitantly, our analysis reveals a diminution in profits for second-level short-term liquidity providers, underscoring the potential trade-off inherent in speed bump implementation, wherein rapid arbitrage opportunities are constrained, potentially disadvantaging fast liquidity providers.

[2]In addition, we extend our analysis of price impact and realized spread into different time frame from shorter to longer as: 1s/15s/1min/2min/3min/4min, until closed to the well used time range 5 min in literature like Hendershott and Moulton (2011). Our analysis reveals a distinct trend: employing shorter time intervals such as 1 second and 15 seconds shows a significant reduction in realized spread due to the presence of speed bumps. This stands in contrast to our observations with a 5-minute realized spread, indicating a decrease in short-term profitability for liquidity providers. However, as we extend the time intervals, we note different outcomes. For intervals of 15 seconds and 30 seconds, the coefficients of speed bump dummy variables are positive but not statistically significant. With further increases in time intervals, such as 90 seconds, 2 minutes, 3 minutes, and 4 minutes, we observe a significant increase in realized spread attributable to speed bumps, leading to higher profitability for liquidity providers. We also undertake a comparable examination of price impact. However, the analysis reveals that the price impacts across these intervals lack discernible patterns, with no statistically significant findings emerging. Given that price impact reflects the lasting repercussions of trades influenced by the presence of the speed bump, it is reasonable to anticipate challenges in identifying distinct trends within very short time intervals. This finding strength our conclusion, the speed bump damage short term market liquidity in short term (in seconds), at the same time of reducing short term price discovery.

Significantly, our findings also elucidate that the reduction in adverse selection, evidenced by the decline in price impact, underpins the decrease in effective spread. Nevertheless, we observe a concurrent increase in immediacy costs, manifest in the augmented percentage of quoted spread. This observation aligns with the decision by the New York Stock Exchange (NYSE) to eliminate the speed bump on November 1, 2019.

In summary, our findings furnish compelling evidence supporting the contention that the speed bump effectively diminishes adverse selection costs for the majority of traders, thereby endorsing its adoption as a mechanism for thwarting insider trading and fostering equity within the stock market. Analogously to the promotion of the Investors Exchange (IEX) by Katsuyama, the NYSE American emerges as a fairer exchange through the implementation of similar measures.

2.5 Simulation Price discovery of exchange has a speed bump

In this section, we adhere to the model proposed by Hasbrouck (2002) to formulate the time series for the price discovery process and employ a CVAR model to estimate the price discovery dynamics. We establish two distinct models during the simulation process. The first model, referred to as the original role model, embodies a scenario where only public information is considered. Subsequently, the second model is devised, wherein two exchanges possess public information, while private information is exclusive to one exchange.

The original Roll model fosters a markedly equitable market environment, characterized by the exclusive incorporation of public information:

$$\begin{gathered}\text{efficient price: } m_l = m_{t-1} + u_t, \quad u_t \sim \mathrm{N}\left(0, \sigma_u^2\right) \\ \text{trae direction: } q_{it} = \pm 1, \text{ each with prob } = 1/2 \text{ for } i = 1, 2 \\ \text{transaction price: } p_{it} = m_t + c q_{it} \quad \text{for } i = 1, 2\end{gathered}$$

Model 2 exactly fits our situation: one exchange has a speed bump (slow market) and the other one has no speed bump (fast market). Then we construct the simulation process as following:

$$\begin{gathered}\text{efficient price: } m_t = m_{t-1} + \lambda q_{1it} + u_t, \text{where } u_t \sim \mathrm{N}\left(0, \sigma_u^2\right) \\ \text{trae direction: } q_{it} = \pm 1, \text{ each with prob } = 1/2 \text{ for } i = 1, 2 \\ \text{transaction price: } p_{1t} = m_t + c_1 q_{1t}, \quad p_{2t} = m_{t-1} + c_2 q_{2t}\end{gathered}$$

Through this simulation, we construct 100,000 pairs of prices and calculate price discovery. The relative price discovery of the slow market (NYSE American) is getting lower and lower, which is the same as the actual data from NYSE American and Nasdaq. We simulate the stock price time series from $lag = 1$ to $lag = 6$ and calculate their price discovery through Nielsen and Popiel (2018) Matlab package. The result is shown in Table 2.9:

Table 2.9

The estimation results for the two models are presented in Panel A of Table 2.9. For Model 1 (Roll Model), which only considers public information, the price discoveries of both markets are very close to 50%, indicating that the market environment satisfies the efficient-market hypothesis.

On the other hand, Model 2 (Fast and Slow Exchanges Model) considers two exchanges, where Nasdaq is the fast market with access to private information and NYSE American has access only to public information. The situation with $lag = 0$ indicates that both markets have the same comprehensive market environment. However, when we increase the number of lags, the Nasdaq exchange is much faster than the NYSE American exchange, and the relative price discovery of Nasdaq significantly increases. This implies that the stock price is dominated by the price in Nasdaq, reducing the opportunities for arbitrage in NYSE American. This simulation result aligns with our expectations of relative price discovery.

In summary, the results from the simulations indicate that the fast exchange (Nasdaq) has a significant advantage over the slow exchange (NYSE American), as it can update prices much faster due to access to private information. In this situation, the stock price is dominated by the price in Nasdaq, which shows that the opportunities of arbitrage will diminish in the NYSE American.

2.6 Conclusion

Based on our empirical findings, we have observed that the implementation of a speed bump as a mechanism to reduce trading speed within an exchange results in a decrease in trading speed among all investors operating on the NYSE American exchange, regardless of whether these trades involve latency arbitrage exploiting speed advantages. On the day of its implementation, a considerable number of stocks were listed on the NYSE American exchange. However, concomitantly, there was a significant decrease in trading volume on NYSE American, and we also observed a reduction in price

discovery. This provides evidence that informed traders, who facilitate a substantial portion of trading activity, migrated away from NYSE American.

In contrast, ordinary investors seemed to welcome this change, as evidenced by their increased participation. This exodus of some informed traders possessing greater speed advantages and information from NYSE American results in the erosion of price information or price discovery at higher update frequencies, measured in seconds. Concurrently, by curbing the speed advantage of high-frequency traders, the speed bump effectively reduces adverse selection, thereby enhancing market quality over longer time intervals. We provide evidence that the speed bump represents a trade-off for the exchange.

2.7 Robustness check

In this section, we conduct a robustness check on all 103 stocks that were actively traded on both NYSE American and Nasdaq between June 1st, 2017 and August 31st, 2017. We examine all same measures include market liquidity and price discovery. We calculate the relative price discovery through information and component share, as described by Hasbrouck (1995) and information leadership share as Patel et al. (2020) and Yan and Zivot (2010). The results of this analysis are comprehensively consistent with our initial findings; however, some of the significant levels have slightly decreased. We attribute this to the fact that by expanding the range of the sampling data set, many stocks that are not strongly sensitive to front running and back running arbitraging strategies have also been included.

2.7.1 Price Discovery Linear Regression by information and component share

Based on the relative price discovery of 103 common stocks calculated during all trading days from June 2017 to August 2017, we compute the information share and component share proposed by Hasbrouck (1995) using the pdshare function of the R package ifrogs, as implemented by Aggarwal (Aggarwal). Following the same procedure, we perform a regression analysis using a panel data approach with stock fixed effects.

$$IShare_{i,t} = \alpha_i + \beta \text{SpeedBump}_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.17)$$

$$CShare_{i,t} = \alpha_i + \beta \text{SpeedBump}_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.18)$$

$$ILShare_{i,t} = \alpha_i + \beta \text{SpeedBump}_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.19)$$

The regression model we used includes several variables to capture the relative

price discovery and other factors affecting the stock prices. Following the same procedure, the relative price discovery is represented by the variable $IShare$ and $CShare$, which are calculated as the ratio of the information share or component share on NYSE American to that on Nasdaq for stock i on day t, α_i is the stock fixed effect, and $SpeedBump_{i,t}$ is an indicator variable taking the value of 1 after speed bump implementation, 0 otherwise. Volatility is the opening value of CBOE's VIX index on day t. The other independent variables are control variables. The $ControlVariable_{i,t,q}$ represents two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as Alizadeh et al. (2002)). For the linear regression without fixed effect, we also add market capitalization of the stock.

Table 2.10

Table 2.10 presents the outcomes of the linear regression analysis conducted with and without stock fixed effects. The regression coefficients for the dummy variable *speedbump* reveal the significant impact of its implementation on the relative price discovery metrics of NYSE American, as measured by information share, component share, and information leadership share.

Remarkably, both information share and component share coefficients exhibit a decline following the introduction of the speed bump, signifying a reduction in their contributions to price discovery. This trend aligns with our earlier observations in Table 2.5. While the result for information share lacks statistical significance, the coefficient for component share demonstrates high statistical significance, underscoring the pronounced effect of the speed bump on the component share of NYSE American.

Conversely, the coefficient for information leadership share registers an increase, implying an augmented role in reflecting new information, as previously noted. This reaffirms our earlier findings that the speed bump on NYSE American decelerates the assimilation of information into stock prices, resulting in diminished price discovery. However, price discovery reflecting fundamental value is expected to increase due to the speed bump.

2.7.2 Cost of immediacy

In this sub-section, we will conduct a similar analysis of the cost of immediacy in trading as the empirical analysis section through percentage quoted spread and effective spread. The regression equation is as follows:

$$\text{Sprd}_{i,t} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (2.20)$$

where $Sprd_{i,t}$ is the percentage quoted spread or effective spread for stock i on day t, α_i is the stock fixed effect, and $SpeedBump_{i,t}$ is an indicator variable taking the value of 1 after speed bump implementation, 0 otherwise. Volatility is the opening value of CBOE's VIX index on day t. The other independent variables are control variables. The $ControlVariable_{i,t,q}$ represents two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as Alizadeh et al. (2002)). For the linear regression without fixed effect, we also add market capitalization of the stock.

Table 2.11

Table 2.11 displays the results of a regression analysis that we conducted with and without fixed effects. In the linear regression, we calculated the coefficients of the dummy variable "speed bump," which allowed us to observe the causal effect of the speed bump on both the quoted spread and effective spread. The results were consistent with our findings in previous sections.

We noticed a decrease in the significant level of both the percentage quoted spread and effective spread when compared to the 50 stocks commonly traded on NYSE American, Nasdaq, and IEX. This outcome was in line with our expectations, as more stocks became less sensitive to fast trading strategies such as front running and back running.

The reduced sensitivity could be attributed to the implementation of the speed bump, which has a decelerating effect on trading activities. As a result, stocks that were initially sensitive to fast trading strategies are now less prone to these practices, leading to a decrease in the significant level of quoted spread and effective spread. Overall, these results suggest that the speed bump has a positive impact on market efficiency by reducing the prevalence of predatory trading strategies.

2.7.3 Adverse Selection

Following the procedure outlined earlier, this subsection aims to analyze the impact of speed bumps on investors' adverse selection costs. Adverse selection arises when buyers or sellers possess information that is unavailable to their counterparts, resulting

in informational asymmetry within the stock market. Consequently, the adverse selection cost quantifies the extent of this asymmetry. To investigate this cost, we employ a decomposition approach, separating the effective quoted spreads into their permanent and transitory components.

To conduct our analysis, we use a linear regression model without fixed effects. This model allows us to explore the relationship between the presence of speed bumps and changes in the adverse selection cost. By running this regression, we can identify any significant differences in the adverse selection cost before and after the implementation of speed bumps.

$$advselection_{i,t} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{2.21}$$

In this equation, the $advselection_{i,t}$ is the average impulse response, price impact, or realized spread measure for stock i on day t. On the independent variables side, α_i are stock and date fixed effects. Then we use $SpeedBump_i$ as an indicator variable equals one for trading days after speed bump implementation, and equals zero otherwise. $Volatility_t$ is the opening value of CBOE's VIX index on day t. $ControlVariable_{i,t,q}$ represents two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as Alizadeh et al. (2002)). For the linear regression without fixed effect, we also add market capitalization of the stock. We give empirical results in Table 2.12.

Table 2.12

Table 2.12 presents the results of a regression analysis conducted over a three-month period spanning 64 trading days. We analyzed 103 stocks that were commonly traded every day on NYSE American and Nasdaq. The findings are consistent with our original results. Specifically, the significant level of price impact has decreased. However, the realized spread and impulse response remain highly significant. This demonstrates our findings are quite robust.

2.8 Appendix 1. Hasbrouck decomposition.

Hasbrouck (1991a, 1991b) demonstrates a $VectorAutoregressive(VAR)$ based model that makes almost no structural assumptions about how to detect the information of order flow. This model offers us a framework to understand trades and quotes. Hasbrouk's model infers the information flow from quotes and trading indicators from the observed sequence of prices and orders. In the Hasbrouck (1991a, 1991b) framework, all stock prices changes are assigned either associated or unassociated with a recent trade. The model does not require any structural assumptions about the information flow; however, in practice, stock prices changes are usually referred to as private-information-based if they are associated with a recent trade. Price moves that are orthogonal to recent trade arrivals are recognized as public information. Hasbrouck (1991a, 1991b) constructs a VAR model with two equations to separate price moves into trade-related and trade-unrelated components. The first equation describes the trade-by-trade evolution of the quote midpoint, while the second equation describes the persistence of order flow. In this paper, we use Lee and Ready's approach to infer the trade direction:+ 1 for buys:-1 for sells. The indicator is $q_{j,t}$ for trade t in stock j. We define $r_{j,t}$ to be the log return based on the mid-quote of stock j from trade $t-1$ to trade t. The VAR picks up order flow dependence out to lags estimated by the price cointegration $vectorautoregressivemodel(CVAR)$. Here, we assume $lag = 10$:

$$r_t = \sum_{i=1}^{10} \alpha_i r_{t-i} + \sum_{i=0}^{10} \beta_i q_{t-i} + \varepsilon_{rt},$$
$$q_t = \sum_{i=1}^{10} \gamma_i r_{t-i} + \sum_{i=1}^{10} \phi_i q_{t-i} + \varepsilon_{qt},$$

where the stock subscripts j are suppressed from here on. The VAR is inverted to get the $VectorMovingAverage(VMA)$ representation:

$$y_t = \begin{bmatrix} r_t \\ q_t \end{bmatrix} = \theta(L)\varepsilon_t = \begin{bmatrix} a(L) & b(L) \\ c(L) & d(L) \end{bmatrix} \begin{bmatrix} \varepsilon_{rt} \\ \varepsilon_{qt} \end{bmatrix},$$

where $a[L]$, $b[L]$, $c[L]$, and $d[L]$ are lag polynomial operators. The permanent effect on the price of an innovation ε_t is given by $a[L]\ \varepsilon_{rt} + b[L]\ \varepsilon_{qt} = 0$ and the variance of this random-walk component can be written as:

$$\sigma_w^2 = \left(\sum_{i=0}^{\infty} a_i\right)^2 \sigma_r^2 + \left(\sum_{i=0}^{\infty} b_i\right)^2 \sigma_q^2$$

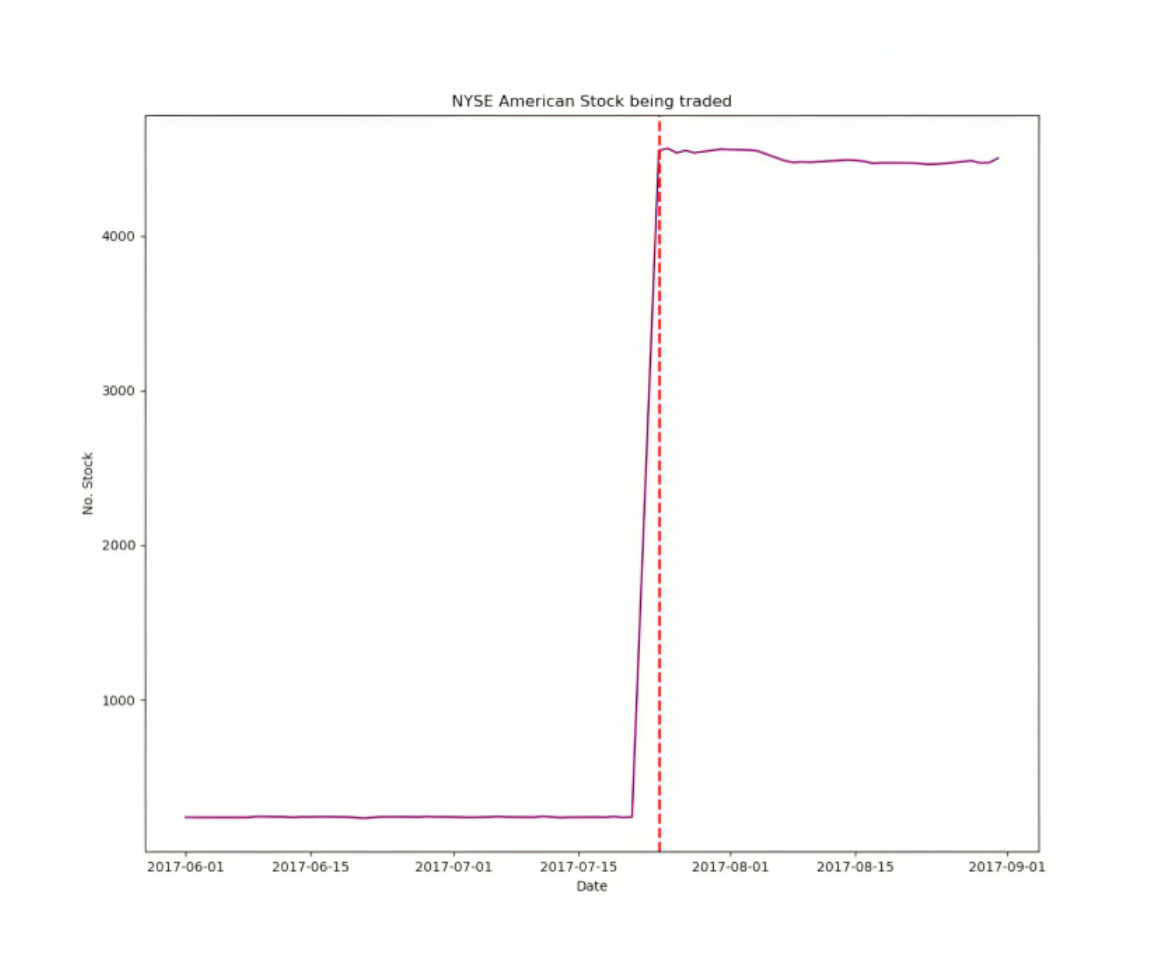
From this equation, we summarize the first term as the component of price changes that are related to public information and the second term that is related to private information. As Hasbrouck (1991a, 1991b) demonstrates, this method is robust to price discreteness, lagged adjustment to information, and lagged adjustment to trades. The

VAR is estimated for every stock each day. The impulse response is the permanent impact of a trade innovation:

$$\sum_{i=0}^{\infty} b_i$$

Using a similar VAR, Hasbrouck (1993) decomposes price changes into their permanent price changes (the random walk) and transitory (pricing error) price changes and calculates a lower bound on the pricing error we estimate in this paper. By using mid-quote prices and return in our VAR, we remove the effects of the increase in spread and focus on the efficiency of quotes. Through this VAR, we have the power to estimate a very informative market quality measures: impulse response.

2.9 Appendix 2. Figures and Tables of Chapter 2

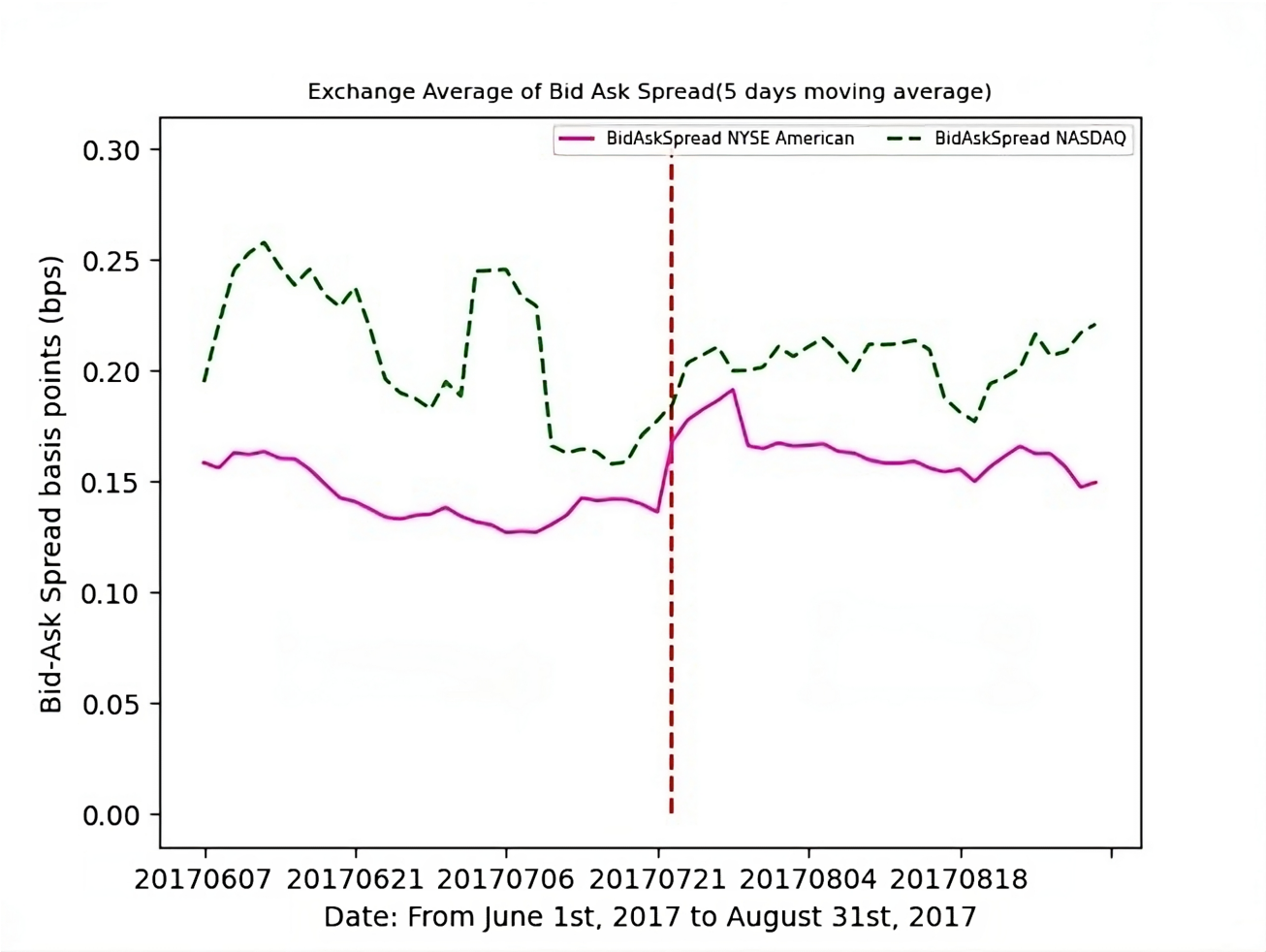


Notes: The Number of stocks being trades on NYSE American is 242, it became 4550 on and after the implementation of the speed bump, the red dotted line is the date of speed bump implementation

Figure 2.1: Number of stocks being traded on NYSE American around the speed bump implementation

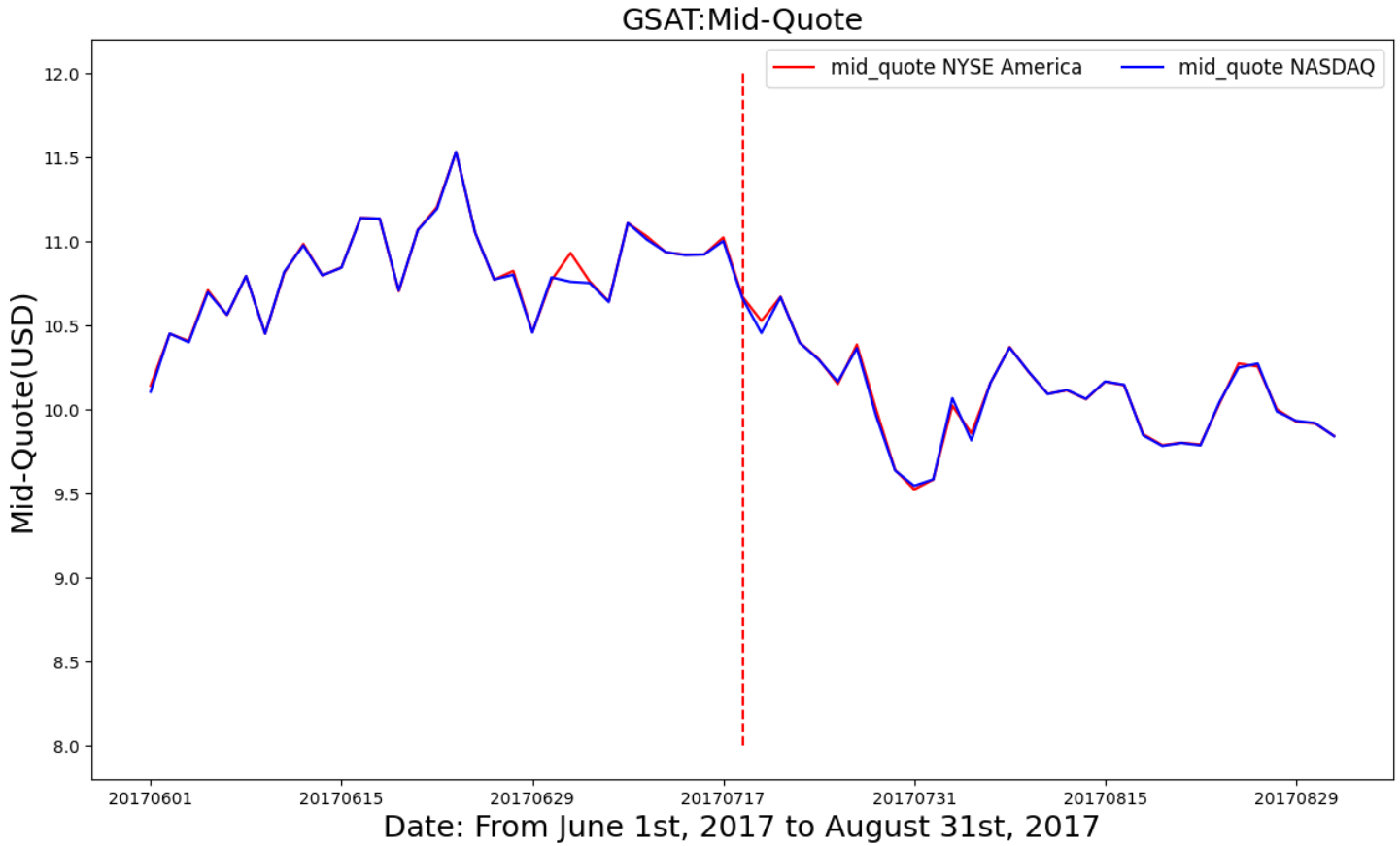
Notes: The Number of stocks being trades on NYSE American is not significantly changed after the speed bump is removed, the red dotted line is the date of speed bump removal

Figure 2.2: Number of stocks being traded on NYSE American around the speed bump removal



Notes: There are 51 stocks commonly traded in NYSE American and Nasdaq, we remove one of them because of its data quality. We calculate the 5 days moving average of each stocks in every trading day, the plot shows the average of all 50 stocks Bid-Ask spread

Figure 2.3: Exchange Average Bid Ask spread of all 50 commonly traded stocks



Notes: GSAT is the most active stock commonly traded in both NYSE American and NASDAQ

Figure 2.4: Bid Ask mid-quote of stock: GSAT

Exchange	Description of Speed Bump	Implementation Date
ParFX	Randomly slows down orders by 20-80 milliseconds.	April, 2013
Refinitive Matching	Emphasize order cancellation messages in batch releases to the order book, providing enhanced protection against aggressive latency arbitrage strategies. [3]	2014
Aequitas NEO	3 to 9 milliseconds random delay on high-frequency traders. [4]	March 27, 2015
TSX Alpha	Randomized delays on market orders up to 3 milliseconds. [5]	September 21, 2015
IEX	Delay all incoming orders by 350 microseconds. [6]	June 17, 2016
NYSE American	Delay all incoming orders by 350 microseconds. [7]	July 17th, 2017
Nasdaq	Hold 10 millisecond period before becoming executable. (Midpoint Extended Life order)[8]	March 12, 2018
EBS Market	Batches orders before matching every 1, 2 or 3 milliseconds. [9]	2018
IntelligentCross	Matches mid-point orders every millisecond and limit orders every few hundred microseconds. (Alternative Trading System (ATS)) [10]	2018
Euronext	Orders triggering immediate transactions undergo 1-3 millisecond delay before integration.	June 3, 2019
ICE Futures US	Delays non-modification or passive orders by 3 milliseconds.	May 2019
LME	Delays all new incoming orders by 8 milliseconds.	March 2020
CBOE EDGA	Asymmetric speed bump by4 milliseconds. [11]	April, 2020

Table 2.1: Description of Speed Bumps in Different Exchanges

	NYSE American							Nasdaq						
	MktCap ($million)	Turnover (10^{-3})	DailyReturn ($)	MidQuote ($)	Qspread (bps)	Espread (bps)	Depth	MktCap ($million)	Turnover (10^{-3})	DailyReturn ($)	MidQuote ($)	Qspread (bps)	Espread (bps)	Depth
Before speed bump														
Mean	275.661	5.005e-4	6.37e-5	63.953	67.93	45.94	751.737	275.672	7.117e-4	-2.53e-5	63.941	63.078	36.558	658.635
Std.Dev	464.001	6.719e-4	3.238e-2	139.976	49.181	32.753	1421.629	463.982	7.497e-4	0.033	139.919	52.278	29.663	1297.812
Max	2881.158	7.757e-3	0.336	922.188	454.001	341.134	15012.81	2881.158	6.986e-3	0.321	922.344	420.583	278.175	10057.25
Min	4.196	3.78e-6	-0.170	5.224	4.222	3.198	10.866	4.197	2.59e-6	-0.157	5.229	4.333	1.931	10.378
After speed bump														
Mean	290.418	3.371e-4	-3.6139e-3	63.893	79.708	51.207	430.452	290.416	5.57e-4	2.976e-3	63.917	66.442	40.517	578.043
Std.Dev	480.353	6.16e-4	3.280e-2	142.101	64.642	42.323	716.615	480.362	8.446e-3	3.201e-2	142.148	58.31	36.843	1097.527
Max	2874.622	1.224e-2	0.228	917.216	767.159	396.65	6489.444	2874.622	1.196e-2	0.197	917.238	583.37	562.108	7902.668
Min	0.635	2.00e-6	-0.154	5.132	5.345	2.935	10.924	0.635	4.39e-6	-0.174	5.136	3.863	2.053	10.506

Notes: MktCap is market capitalization, Turnover is stock turn over rate, DailyReturn is the daily return, MidQuote is stock midpoint of bid and ask quote. Qspread and Espread is by the number of basis point (bps). Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. All of them are calculated through 50 stocks for each day in our sample period, from June 1st 2017 to August 31st 2017, which are around NYSE American speed bump implementation.The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after the speed bump.

Table 2.2: Sample descriptive statistics

	$\hat{\beta}$	$AverageP_{NYSEAmerican}$	$AverageP_{Nasdaq}$
Before speed bump	(1,-1)	0.431	0.569
After speed bump	(1,-1)	0.416	0.584
Diff= Before-After		0.015	-0.015

Notes: Price discovery estimation results through cointegration vector autoregressive model(CVAR) implemented by Nielsen and Popiel (2018).We used 64 trading days within 3 months to calculate all 50 stocks' price discovery on a daily basis, then calculate the average of all stocks. All data are in nano-second time stamp.

Table 2.3: Relative price discovery estimation results

	Price Discovery	Price Discovery	Price Discovery
Speedbump	-0.129***	-0.133***	-0.196***
	(4.176e-2)	(4.422e-2)	(4.422e-2)
VIX	5.679e-2***	5.914e-2***	8.187e-2***
	(1.555e-2)	(1.655e-2)	(1.643e-2)
Turnover	0.139***	0.214***	
	(3.622e-2)	(3.470e-2)	
StockVolatility	0.119***	9.048e-2***	
	(3.246e-2)	(3.234e-2)	
MarketCap		0.411	
		(1.225)	
Stock F.E.	Yes	No	No
N	2,507	2,507	2,638
Adjust R-square	17.67%	4.2%	1.12%

Notes: This table reports the linear regression result of relative price discovery through the observations from June 1st 2017 to August 31st 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump was implemented on July 24th, 2017. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 2.4: Panel regression of relative price discovery

Regression results

	Information Share			Component Share			Information Leadership Share		
Speedbump	-0.058	-0.036	-0.170**	-0.130***	-0.120**	-0.210***	0.158**	0.175**	0.087
	(0.070)	(0.076)	(0.077)	(0.045)	(0.049)	(0.049)	(0.645)	(0.075)	(0.076)
VIX	0.012	0.019	0.047	0.031*	0.037**	0.050***	-0.418*	-0.040	-0.009
	(0.026)	(0.029)	(0.029)	(0.017)	(0.018)	(0.018)	(0.241)	(0.028)	(0.028)
Turnover	0.498***	0.784***		0.233***	0.346***		0.558***	0.902***	
	(0.060)	(0.060)		(0.039)	(0.038)		(0.056)	(0.059)	
StockVolatility	0.135**	0.023		0.094***	0.066*		0.087*	-0.089*	
	(0.054)	(0.68)		(0.035)	(0.035)		(0.050)	(0.054)	
MarketCap		-1.198			-0.921			-0.615	
		(2.073)			(1.327)			(2.036)	
Stock F.E.	Yes	No	No	Yes	No	No	Yes	No	No
N	2,574	2,574	2,741	2,574	2,574	2,741	2,574	2,574	2,741
Adjust R-square	8.7%	8.8%	0.13%	5.7%	5.68%	0.65%	9.87%	10.24%	-0.03%

Notes: This table reports the linear regression result of relative price discovery by information share and component share through the observations from 50 stocks trades on NYSE American, Nasdaq and IEX June 1st 2017 to August 31st 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump was implemented on July 24th, 2017. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 2.5: Panel regression of price discovery by information and component share

	Quoted Spread	Quoted Spread	Quoted Spread	Effective Spread	Effective Spread	Effective Spread
Speedbump	0.110***	0.105***	0.128***	-0.028*	-0.030*	-0.004
	(0.011)	(0.013)	(0.014)	(0.015)	(0.017)	(0.017)
VIX	-0.013***	-0.011**	-0.018***	-0.003	-0.002	-0.009
	(0.004)	(0.005)	(0.005)	(0.005)	(0.006)	(0.006)
Turnover	-0.147***	-0.221***		-0.134***	-0.196***	
	(0.009)	(0.010)		(0.013)	(0.013)	
StockVolatility	0.040***	0.092***		0.010	0.050***	
	(0.008)	(0.010)		(0.011)	(0.012)	
MarketCap		0.135			-0.242	
		(0.432)			(0.543)	
Stock F.E.	Yes	No	No	Yes	No	No
N	2,552	2,552	2,706	2,552	2,552	2,707
Adjust R-square	46.94%	18.03%	2.8%	30.59%	8.85%	0.03%

Notes: This table reports the linear regression result of percentage quoted spread and effective spread for the sample period from June 1st, 2017 to August 31st, 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump was implemented on July 24th, 2017. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 2.6: Panel regression of cost of immediacy

Panel A: Price impact

	NYSE American			Nasdaq		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	35.088	36.039	-0.951	35.665	38.34	-2.675
std	31.193	39.239	-8.046	32.148	39.447	-7.299
N	1,515	1,192	323	1,515	1,192	323

Panel B: Realized spread

	NYSE American			Nasdaq		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	10.852	15.169	-4.317	0.893	2.177	-1.284
std	26.645	37.559	-10.914	22.553	27.951	-5.398
N	1,515	1,192	323	1,515	1,192	323

Panel C: Impulse Response

	NYSE American			Nasdaq		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	96.415	77.486	18.929	76.216	64.879	11.337
std	106.604	99.504	7.1	99.53	77.911	21.619
N	1,370	1,009	361	1,370	1,009	361

Notes: This table reports the average values of daily price impact,realized spread and impulse response, their standard deviation and number of observations from June 1st, 2017 to August 31st, 2017, which are around NYSE American speed bump implementation. All of them are in basis points. The speed bump was implemented on July 24th, 2017.

Table 2.7: Adverse selection

Regression results

	Price Impact			Realized Spread			Impulse Response		
Speedbump	-0.132***	-0.113***	-0.087**	0.131*	0.122*	0.114*	-0.251***	-0.213***	-0.207***
	(0.040)	(0.0410)	(0.041)	(0.068)	(0.068)	(0.066)	(0.077)	(0.076)	(0.075)
VIX	0.002	0.004	0.001	-0.018	-0.020	-0.022	-0.031	-0.030	-0.034
	(0.015)	(0.015)	(0.015)	(0.025)	(0.026)	(0.025)	(0.029)	(0.029)	(0.028)
	(1.178e-4)	(1.359e-4)		(2.024e-4)	(2.27e-4)		(8.768e-4)	(6.204e-4)	
Turnover	-6.004e-3	-0.012		-0.146**	-0.132**		-0.092	-0.098	
	(0.034)	(0.031)		(0.058)	(0.053)		(0.071)	(0.063)	
StockVolatility	-0.034	-7.595e-3		0.028	-6.446e-3		-0.077	0.027	
	(0.031)	(0.030)		(0.053)	(0.050)		(0.678)	(0.627)	
MarketCap	1.359	2.265			-3.608			-0.517	
		(1.341)			(2.239)			(1.998)	
Stock F.E.	Yes	No	No	Yes	No	No	Yes	No	No
N	2,552	2,552	2,700	2,552	2,552	2,707	2,337	2,336	2,368
Adjust R-square	8.33%	0.15%	0.12%	3.14%	0.42%	0.04%	1.56%	0.71%	0.49%

Notes: This table reports the linear regression result of price impact, realized spread and impulse response for the sample period from June 1st, 2017 to August 31st, 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump was implemented on July 24t,h 2017. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 2.8: Panel regression of adverse selection

Panel A: Roll Model

lag	*d*	*b*	*beta*	Price Discovery NYSE American	Price Discovery Nasdaq
1	1	1	(1,-1)	0.492	0.508

Panel B: Fast and slow exchanges Model

lag	*d*	*b*	*beta*	Price Discovery NYSE American	Price Discovery Nasdaq
1	1	1	(1,-1)	0.360	0.640
2	1	1	(1,-1)	0.198	0.802
3	1	1	(1,-1)	0.087	0.913
4	1	1	(1,-1)	0.113	0.887
5	1	1	(1,-1)	0.074	0.926
6	1	1	(1,-1)	0.012	0.988

Notes: Panel A reports CVAR estimation results for data simulated in fair Roll model. We generated 100,000 pairs of price data for two exchanges. The price discovery contributions of NYSE American and Nasdaq (P_A and P_T) are reported. The efficient price is $m_t = m_{t-1} + u_t, u_t \sim N(0, \sigma_u^2)$. The model is

$$p_{At} = m_t + cq_{At},$$
$$p_{Tt} = m_t + cq_{Tt},$$

where $q_{it} = \pm 1$ with equal probability.
Panel B reports CVAR estimation results for data simulated in a model of two exchanges, one of them has speed bump, which is the slow exchange NYSE American, the other exchange has no speed bump, which is the fast exchange Nasdaq. We generated 100,000 pairs of price data. The price discovery contributions of NYSE American and Nasdaq (P_A and P_T) are reported. The efficient price is $m_t = m_{t-1} + \lambda q_{Tit} + u_t, u_t \sim N(0, \sigma_u^2)$. The model is

$$p_{At} = m_{t-1} + cq_{At},$$
$$p_{Tt} = m_t + cq_{Tt},$$

where $q_{it} = \pm 1$ with equal probability.

Table 2.9: Price efficiency estimation results for Simulation

Regression results

	Information Share			Component Share			Information Leadership Share		
Speedbump	-0.053	-0.066	-0.187***	-0.162***	-0.204***	-0.283***	0.236***	0.292***	0.202***
	(0.067)	(0.070)	(0.069)	(0.049)	(0.056)	(0.054)	(0.636)	(0.070)	(0.069)
VIX	0.042*	0.031	0.041	0.037**	0.027	0.034*	3.928e-3	1.234e-3	0.011
	(0.024)	(0.026)	(0.025)	(0.018)	(0.021)	(0.020)	(0.023)	(0.026)	(0.026)
Turnover	0.321***	0.461***		0.158***	0.224***		0.366***	0.516***	
	(0.049)	(0.046)		(0.0372)	(0.037)		(0.047)	(0.046)	
StockVolatility	0.049	-0.080***		-0.0184	-0.153***		-0.0140	0.1427***	
	(0.031)	(0.029)		(0.023)	(0.023)		(0.030)	(0.030)	
MarketCap		-0.235			-0.117			-0.287	
		(0.316)			(0.253)			(0.321)	
Stock F.E.	Yes	No	No	Yes	No	No	Yes	No	No
N	4,021	4,021	4,395	4,021	4,021	4,395	4,150	4,150	4,526
Adjust R-square	2.57%	2.58%	0.13%	0.89%	1.67%	0.58%	5.23%	5.92%	0.2%

Notes: This table reports the linear regression result of relative price discovery by information share and component share for 103 stocks commonly traded on both NYSE American and Nasdaq from June 1st, 2017 to August 31st, 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump was implemented on July 24th 2017. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 2.10: Panel regression of price discovery by information share, component share and information leadership share

	Quoted Spread	Quoted Spread	Quoted Spread	Effective Spread	Effective Spread	Effective Spread
Speedbump	0.015	0.010	0.019*	-0.032	-0.038	-0.020
	(9.9738e-3)	(0.011)	(0.011)	(0.036)	(0.037)	(0.038)
VIX	-2.730e-3	-1.906e-3	-7.066e-3*	0.169	0.015	0.025*
	(3.670e-3)	(3.927e-3)	(3.987e-3)	(0.132)	(0.014)	(0.014)
Turnover	-0.052***	-0.076***		-0.048*	-0.087***	
	(7.2834e-3)	(6.83e-3)		(0.026)	(0.024)	
StockVolatility	0.024***	0.030***		0.109***	0.157***	
	(4.649e-3)	(4.4288e-3)		(0.017)	(0.015)	
MarketCap		6.809e-3			0.786***	
		(0.049)			(0.169)	
Stock F.E.	Yes	No	No	Yes	No	No
N	4,209	4,209	4,658	4,209	4,209	4,658
Adjust R-square	2.99%	2.92%	0.05%	2.68%	3.1%	0.02%

Notes: This table reports the linear regression result of percentage quoted spread and effective spread for 103 commonly traded stocks on NYSE American and Nasdaq from June 1st, 2017 to August 31st, 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump was implemented on July 24th, 2017. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 2.11: Panel regression of cost of immediacy

Regression results

	Price Impact			Realized Spread			Impulse Response		
Speedbump	-0.013	1.566e-4	0.026	0.106*	0.132**	0.106*	-0.045***	-0.405***	-0.4139***
	(0.054)	(0.054)	(0.057)	(0.056)	(0.058)	(0.056)	(0.064)	(0.068)	(0.067)
VIX	6.189e-3	5.768e-3	-6.112e-3	0.0571***	0.062***	0.058***	0.029	0.017	0.0195
	(0.020)	(0.020)	(0.021)	(0.021)	(0.022)	(0.021)	(0.024)	(0.025)	(0.025)
Turnover	0.039	0.027		-0.148***	-0.161***		-0.095*	-0.081	
	(0.039)	(0.035)		(0.041)	(0.038)		(0.056)	(0.053)	
StockVolatility	-0.016	0.057**		0.139***	0.244***		-0.072*	-0.139***	
	(0.025)	(0.023)		(0.026)	(0.024)		(0.043)	(0.040)	
MarketCap		-0.148			0.505*			-0.743	
		(0.025)			(0.269)			(0.400)	
Stock F.E.	Yes	No	No	Yes	No	No	Yes	No	No
N	4,209	4,209	4,647	4,209	4,209	4,657	2,512	2,512	2,552
Adjust R-square	0%	0.15%	-0.04%	2.62%	2.86%	0.34%	2.34%	2.44%	1.47%

Notes: This table reports the linear regression result of price impact, realized spread and impulse response for 103 commonly traded stocks on NYSE American and Nasdaq from June 1st, 2017 to August 31st, 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump was implemented on July 24th, 2017. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 2.12: Panel regression of adverse selection

Chapter 3

Machine Learning-Based Prediction of Mini Flash Crashes

3.1 Introduction

The correlation between high-frequency trading (HFT) and flash crashes has been a prominent topic of discussion in the academic literature. A notable instance of such a crash took place on May 6, 2010, when the Dow Jones Industrial Average recorded its largest intraday decline in history, which is named as "Flash Crash". The screenshot of Bloomberg is shown in Figure 3.1. While the Securities and Exchange Commission (SEC) and the Commodity Futures Trading Commission (CFTC) have acknowledged the challenge of pinpointing the precise cause of this flash crash, there is a prevalent belief in media and public discourse that one of the primary suspects is the cancellation of existing buy orders by high-speed trading algorithms when they detect market imbalances (Nolte (2010)).

Figure 3.1

Fosset et al. (2020) highlights the vulnerability of market liquidity to potential death spirals when both human and machine-driven market makers react excessively to unforeseen events. High-frequency traders (HFTs), acting as ultra-fast intermediaries, face susceptibility to significant overexposure in either long or short positions, necessitating swift adjustments. While HFTs themselves may not be the primary

instigators, their algorithms can exacerbate market conditions, potentially leading to flash crashes Bellia, Mario; Christensen, Kim; Kolokolov, Aleksey; Pelizzon, Loriana; Reno, Roberto (2020).

Kyle and Obizhaeva (2023) presents a structural framework shedding light on market crashes, leveraging insights from market microstructure invariance. Incorporating variations in business time across markets, the framework yields predictions consistent with observed price dynamics during significant events such as the 1987 market crash and the 2008 sales by Société Générale. The larger-than-expected price drops observed during the 1987 and 2010 flash crashes could potentially have been exacerbated by overly rapid selling.

The study cited in Baron et al. (2019) highlights the pronounced sensitivity of High-Frequency Traders (HFTs) to speed, driven by their adeptness at swiftly assimilating fleeting information. This underscores the focus of HFT strategies on exploiting short-term events, including market making and cross-market arbitrage. Furthermore, Fishe et al. (2019) underscores the critical role of speed for traders anticipating local price movements, which consistently outpace broader market reactions. Although manual and algorithmic methods are employed, their pace typically does not match that of HFTs.

In parallel, research on flash crashes, such as Easley et al. (2012), Andersen and Bondarenko (2014b), and Brogaard et al. (2018), yields divergent conclusions. While some studies refute HFT as the primary cause, others, like Leal et al. (2014), suggest HFTs can exacerbate flash crashes through unique trading strategies.

Scholarly discourse, as in Foucault (2016), underscores concerns about HFT's market impact, citing heightened adverse selection costs and reduced market informativeness. The paper suggests that recent market dislocations are more likely due to trading automation and market structural shifts than solely attributable to HFT.

According to Keller (2012)), regulators are being asked to strike a careful balance, in order to restore market confidence without compromising the efficiency brought about by HFT, achieved through greater transparency and more robust reporting

requirements. Moreover, Keller (2012) also advocates the implementation of internal risk management measures within HFT firms.

Due to the sporadic and unpredictable occurrences of events similar to flash crash, as exemplified by the incident on 6 May 2010, researchers have redirected their attention to analyzing smaller-scale flash crashes to glean valuable insights (Nanex (2010)). Various studies have emerged with the objective of estimating mini flash crashes, employing the widely accepted rule-of-thumb definition initially introduced by Nanex (2010).[1].

In particular, Golub et al. (2012) have discovered evidence that underscoring the detrimental effect of these mini flash crashes on market liquidity. In line with their discoveries, our research highlights that information related to market liquidity, encompassing trade volume and limit order book data, serves as a particularly robust signal for predicting the occurrence of mini flash crashes.

In fact, Golub et al. (2012) have highlighted the detrimental impact of mini flash crashes on market liquidity. Our research further emphasizes the importance of market liquidity indicators, such as trade volume and limit order book data, in predicting these events.

Moreover, Kirilenko et al. (2017a) investigated the trading behaviors of High-Frequency Traders (HFTs) during the Flash Crash, revealing distinctive patterns. In our study, we used machine learning techniques to identify clear patterns in market conditions, providing evidence of specific participant behaviors during mini flash crash events.

Following the May 6, 2010 flash crash, regulatory authorities took measures to prevent future occurrences. The SEC's approval of Stock-by-Stock Circuit Breaker rules mandates a temporary trading pause for stocks experiencing a 10 percent price

[1]Nanex (2010)'s definition of mini flash crashes stipulates a minimum requirement of 10 consecutive upward or downward ticks occurring within a 1.5-second interval. This criterion differentiates mini flash crashes from classical events termed as "Price Jumps", which are often initiated by information shocks. For an event to be classified as a mini flash crash, it requires significant price fluctuations that exceed 10 occurrences. Failing to meet this threshold would make the event outside the scope of the defined criteria

change within five minutes. This development has sparked interest in predicting stock-specific flash crashes, prompting the exploration of early warning systems before circuit breakers are triggered.

The flash crash phenomenon has represented a notable escalation of systemic risk in financial markets (Min and Borch (2022)). Panic-driven sales can lead to substantial financial losses for investors. According to a report by the International Organization of Securities Commissions (IOSCO) in July 2011, over 20,000 trades spanning 300 securities occurred at prices deviating by as much as 60% from their recent values, with trade values ranging from mere pennies to substantial sums (OICU-IOSCO (2011)).

Peter Chung and Thomas Kim (2017) discovers that stocks exhibiting the highest levels of herding with aggregate-level trades tend to experience the most negative (positive) returns during market crashes (booms). While herding behavior leads to extreme returns on both sides, investors seem to seek compensation for the risk of experiencing exceptionally low returns.

Our research significantly advances the ongoing discourse by revealing the crucial role of informed trading dynamics in the emergence of mini flash crashes. Using microstructure metrics, particularly Kyle's lambda and the probability of informed trading, we identify robust predictive signals for these events.

Given the substantial financial implications of flash crashes, accurate prediction becomes imperative for effective risk management. The ability to forecast such events empowers investors to make well-informed decisions about their holdings during crises and offers a promising avenue for risk management. In addition, our research introduces the potential for real-time prediction of mini flash crashes, enabling investors and regulators to establish an early warning system before the activation of circuit breakers. This underscores the practical importance of our contribution in improving the understanding and proactive management of market risks.

Figure 3.2

In this paper, we adopt the framework established by Aït-Sahalia et al. (2022) and

draw upon the insights of Easley et al. (2021) to evaluate the predictability of efficient information and microstructure measures across short- and long-time horizons. Our empirical findings demonstrate that machine learning effectively anticipates mini flash crashes by leveraging real-time market data. Notably, short-term information related to trade volume and transaction counts exhibits robust predictive power, reflecting market fluctuations through trading behaviors.

In addition to the domains identified by Kearns and Nevmyvaka (2013), we highlight another promising application area for machine learning: the detection of anomalous events, including mini flash crashes. Building upon their insights, our research underscores machine learning's proficiency in identifying and predicting these events. As emphasized by Easley et al. (2021), market microstructure information remains a valuable source of predictability even in the era of machine learning. Our prediction model is heavily based on the microstructure data highlighted by Easley et al. (2021).

Furthermore, we identify the status of the limit order book as another valuable source of signals, providing insights into the increased likelihood of impending mini flash crashes. Additionally, we uncover significant predictive signals in microstructure measures, particularly exemplified by Kyle's lambda and the probability of informed trading. These metrics highlight the influential role of informed trading in flash crash events.

This study adds valuable insights to the literature by exploring novel perspectives on market data, building on the groundwork laid by O'Hara (2015). Furthermore, we consider this paper as an initial step towards the application of machine learning for the prediction of mini flash crashes.

To explore our findings comprehensively, we have structured the paper into six sections. After the introduction, the paper unfolds as follows: Section 3.2 reviews the related literature. Section 3.3 details our methodology, covering problem formulation, machine learning models, accuracy assessment, and data imbalance mitigation. Section 3.4 outlines our dataset and our pre-processing approach. In Section 3.5, we present our empirical results and their implications. Finally, Section 3.6 offers

concluding remarks on our findings.

3.2 Literature

This paper combines two strands of literature: one focusing on flash crashes, extensively researched with particular attention to their connection with high-frequency trading (HFT), exemplified by Kirilenko et al. (2017a), who identified consistent HFT trading patterns during crashes and developed a framework for analyzing market dynamics. While our study doesn't precisely analyze HFT trading, we do detect significant shifts in market behavior before mini flash crashes, suggesting predictive signals for such events and supporting the notion that crashes often have precursor indicators.

Golub et al. (2012) delve into the adverse impacts of mini flash crashes on market liquidity. Our research aligns with this perspective, highlighting the strong correlation between market liquidity information and the potential occurrence of a mini flash crash, these indicators functioning as warning signals for impending crashes..

An alternative approach to understanding flash crashes involves the concept of Probability of Informed Trading (VPIN). Easley et al. (2012) introduce a solution in the form of the "VPIN contract", allowing liquidity providers to remain active in the marketplace while helping to avert flash crashes. In contrast, Andersen and Bondarenko (2014b) explore the effectiveness of VPIN as a predictor of short-term volatility and conclude that it exhibited limited predictive power due to its mechanical relationship with the underlying trading intensity.

The role of High-Frequency Trading (HFT) in flash crashes has been a captivating subject of exploration within the existing literature. Brogaard et al. (2018) observed that HFTs contribute liquidity by offsetting imbalances created by non-HFT participants, particularly in individual stocks. Benos et al. (2017) reveals that HFT order flow, net positions, and total volume display notably greater commonality compared to a control group of investment banks. This observation suggests that strategies

employed by HFTs may have a tendency to be contagious. However, when multiple stocks experience flash crashes simultaneously, the liquidity demand of HFTs exceeds its supply, leading Brogaard et al. (2018) to conclude that HFTs are not the primary cause of flash crashes.

In contrast, Leal et al. (2014), employing an agent-based model, argue that HFTs have a notably positive impact on flash crashes. They suggest that HFTs can significantly widen bid-ask spreads and synchronize their activities on the sell side of the limit order book, contributing to flash crash occurrences. Furthermore, Leal et al. (2014) posit that higher rates of order cancelations by HFTs increase the likelihood of flash crashes while reducing their duration.

Bellia, Mario; Christensen, Kim; Kolokolov, Aleksey; Pelizzon, Loriana; Reno, Roberto (2020) evaluated HFT behaviors during flash crashes and concluded that HFTs exacerbate transient price impacts, with slower traders providing liquidity. Viewed from a different perspective, Breckenfelder (2019) examined the competitive dynamics among HFTs and found that increased competition among them results in deteriorating market liquidity, potentially contributing to flash crashes.

In our research, we provide additional evidence that the state of market liquidity, characterized by significant fluctuations, exhibits a strong association with mini flash crashes. This connection underscores the significance of market liquidity as a reliable signal for predicting mini flash crashes well in advance, enhancing our understanding of flash crash dynamics.

The second strand of literature pertains to the domain of machine learning in which there has been a significant proliferation of research in recent years. Kearns and Nevmyvaka (2013) conducted three case studies, highlighting the diverse applications of machine learning in the realm of HFT. These studies encompassed optimized trade execution, the prediction of price movements based on order book states, and optimized execution within dark pools through censored exploration. In our research, we consider ourselves as advancing this field by employing machine learning techniques to address the specific challenge of order flow anomaly detection, representing a novel

step in the application of machine learning within this domain.

The prediction of short horizon stock returns consistently draws substantial attention for a long time (Mase (1999) and Chuang and Ho (2013)) from both the financial industry and the academic machine learning community. A wealth of literature has demonstrated that machine learning techniques excel at elucidating non-linear relationships within data. In the financial research domain, there is a growing emphasis on identifying informative features and refining feature engineering approaches to enhance stock prediction performance.

In alignment with this focus, Easley et al. (2021) employed a random forest algorithm to explore the predictability of classical microstructure variables, encompassing metrics such as Roll measures, Roll impact, Kyle lambda, Amihud, VPIN, and UX (VIX). In our study, we also incorporate all these microstructure measures into our predictive models, aligning with the overarching research theme dedicated to enhancing stock prediction through feature selection and engineering.

In addition to feature engineering for time series data within the limit order book, machine learning models have found diverse applications in predicting financial asset behavior. Ke et al. (2019) pioneered an interpretable machine learning model that leverages text data to predict stock returns. Sirignano (2016) introduced a novel neural network architecture tailored to modeling spatial distributions within the limit order book. Khan (2020) conducted a study utilizing text data from Twitter and the Indian stock market to explore the predictive power of public sentiment and political factors on stock market trends. Patel et al. (2015) conducted a comparative analysis of four prediction models - Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest, and Naïve-Bayes - assessing their performance in predicting stock prices and indices.

Furthermore, Sirignano and Cont (2018) utilized deep learning as a framework and unearthed nonparametric evidence indicating the presence of a universal and stationary price formation mechanism associated with supply and demand dynamics for stocks. Notably, they successfully trained deep learning models with the ability to

predict stocks not even included in the training dataset, underscoring the universality of the relationships captured by the model Sirignano and Cont (2018).

In our study, we make a significant contribution by demonstrating that, unlike long-term stock returns, short-term events such as mini flash crashes are universally predictable, and specific asset information is not imperative for their prediction.

We can summarize this paper's contributions from three key perspectives. First, we provide compelling evidence that machine learning can be effectively harnessed for order flow anomaly detection, expanding its application beyond areas such as optimized trade execution, price movement prediction, and optimized execution in dark pools. This underscores the versatility of machine learning in financial analysis.

Second, we establish that even in the case of mini flash crashes, which are events that defy interpretation by linear models, ensemble machine learning models can predict them. This underscores the ability of machine learning to excel in capturing complex, non-linear relationships within financial data.

Third, we empirically demonstrate that certain crucial features, notably those related to market liquidity and informed trading measures, exhibit strong real-time predictive capabilities concerning flash crashes. These findings serve as compelling evidence suggesting the potential involvement of market makers and informed traders in flash crash events, shedding light on the dynamics of these occurrences.

3.3 Methodology

In this section, we provide a detailed explanation of our methodologies, which encompass the predictor variables employed, the machine learning models utilized, the approach used to measure prediction accuracy, and our strategies for handling imbalanced data.

3.3.1 Predictor Variables

In forecasting the foregoing response variables, a diverse array of predictor features is integrated into the model's framework. The construction of most of these predictor variables closely adheres to the methodology originally proposed by Aït-Sahalia et al. (2022),), encompassing numerous derived variables characterized by nonlinear transformations of historical data, particularly when using fine-grained time intervals. Building on the insights gleaned from established works in the field, such as those by Cont et al. (2013) and Kercheval and Zhang (2015), we anticipate that the pivotal determinants for predicting forthcoming short-term events, such as mini flash crashes, will pivot around the characteristics of the prevailing Limit Order Book (LOB). Mase (1999) also points out liquidity effects can provide prediction ability on short horizon return. These attributes encompass potential imbalances and historical trade returns at the point of prediction. Additionally, in consonance with the approach advanced by Aït-Sahalia et al. (2022), we incorporate microstructure measures as expounded upon by Easley et al. (2021). This integration captures information related to market microstructure noise, which may harbor significant signals of unforeseen events. Notably, three supplementary predictors are introduced, with a specific focus on microstructure noise. Comprehensive descriptions of all these features will be explained in subsequent sections.

In parallel with the forward-looking intervals that are considered in relation to the response variable, namely, mini flash crashes, an analogous approach is adopted to establish look-back intervals based on calendar time. This entails employing the current timestamp, denoted as T, and defining lookback spans as (Δ_1, Δ_2). For the derivation of predictor variables, a set of lookback windows is designated as $I = Int(T - \Delta_2, T - \Delta_1)$. More specifically, the values assigned to (Δ_1, Δ_2) span a spectrum from $(0s, 0.1s)$ to $(102.4s, 204.8s)$. The most extensive span, which is $204.8s$, effectively encompasses a slightly extended 3-minute horizon beyond each timestamp T. This strategic choice ensures that the prediction model avoids re-

liance on transient information for forecasting longer-term outcomes. In total, our framework encompasses 11 distinct look-back windows, where features are computed directly upon the conclusion of each respective interval.

Let D^{txn} represent the set of all timestamps t within the dataset D^{txn}, corresponding to trade transactions, and D^{qt} representing its counterpart for quote data. The combined set is denoted as $D = D^{txn} \cup D^{qt}$. Within this dataset, the National Best Bid and Offer (NBBO) prices are indexed by $t \in D$ and expressed as $\left(P_t^b, P_t^a\right)$, where P_t^b represents the best bid price, and P_t^a designates the best ask price. The mid-price is computed as the simple average of these two, denoted as $P_t = \frac{P_t^b + P_t^a}{2}$.

Further, if t belongs to the set D^{txn}, we denote the transacted price as P_t^{txn}. The best bid and ask sizes are also represented as S_t^b and S_t^a respectively, pertaining to the record indexed by t.

Volume and duration: Predictors in this category are linked to a stock's recent trading activity. For instance, block trades or frequent transactions may indicate increased short-term trading activity. While this heightened activity doesn't inherently reveal the trend's direction, it can interact nonlinearly with other predictors, potentially strengthening a trend's emergence.

Breadth measures the number of transactions in the interval:

$$Breath(T, \Delta_1, \Delta_2) = |D^{txn} \cap Int^{back}(T, \Delta_1, \Delta_2)| \tag{3.1}$$

Immediacy measures the average time between successive transactions in the interval:

$$Immediacy(T, \Delta_1, \Delta_2) = \frac{\Delta_1 - \Delta_2}{Breath(T, \Delta_1, \Delta_2)} \tag{3.2}$$

VolumeAll measures the total number of shares transacted in the interval:

$$VolumeAll(T, \Delta_1, \Delta_2) = \sum_{t \in Int^{back}(T, \Delta_1, \Delta_2)} V_t \tag{3.3}$$

VolumeAvg measures the average number of shares transacted for each transaction in the interval:

$$VolumeAvg(T, \Delta_1, \Delta_2) = \frac{VolumeAll(T, \Delta_1)}{Breath(T, \Delta_1, \Delta_2)} \tag{3.4}$$

VolumeMax measures the maximum number of shares transacted in one transaction in the interval:

$$VolumeMax(T, \Delta_1, \Delta_2) = max\{V_t : t \in Int^{back}(T, \Delta_1, \Delta_2)\} \tag{3.5}$$

Return and imbalances: These predictors relate to recent trading imbalances within the stock, providing insights into short-term trends. These trends are discerned through the analysis of trade transactions and quotes data. For instance, a significant volume of buying leading to triggering of limit sell orders, or a consistently higher bid compared to the ask in Level I quotes, indicates a potential upward price influence. To capture these dynamics, the following variables are outlined.

Lambda measures the price change in the interval proportional to total volume.

Let $I = D^{txn} \cap Int^{back}(T, \Delta_1, \Delta_2)$, then:

$$Lambda(T, \Delta_1, \Delta_2) = \frac{P_{max(I)} - P_{min(I)}}{VolumeAll(T, \Delta_1, \Delta_2)} \tag{3.6}$$

LobImbalance is the average imbalance in the depth of the limit order book over the lookback interval:

$$LobImbalance(T, \Delta_1, \Delta_2) = Average[\frac{S_t^a - S_t^b}{S_t^a + S_t^b}] : t \in Int^{back}(T, \Delta_1, \Delta_2) \tag{3.7}$$

TxnImbalance measures the asymmetry of buy and sells volumes in recent transactions. Denote by Dir_t^{LR} the binary transaction direction at time t signed using the algorithm of Lee and Ready (1991). Then transaction imbalance is calculated as

$$TxnImbalance(T,\Delta_1,\Delta_2) = Average[\frac{\sum_{t\in D^{txn}\cap Int^{back}(T,\Delta_1,\Delta_2)}(V_t Dir_t^{LR})}{VolumeAll(T,\Delta_1,\Delta_2)}] \quad (3.8)$$

PastReturn is the past return in the lookback window. Let $I = D^{txn}\cap Int^{back}(T,\Delta_1,\Delta_2)$:

$$PastReturn(T,\Delta_1,\Delta_2) = 1 - Average\frac{[P_t^{txn} : t\in I]}{P_{max(I)}} \quad (3.9)$$

Speed and cost: This set of predictors we employ measure the speed and cost inherent in the stock's trading.

Turnover is the speed of transactions to the stock's total number of shares outstanding.

$$Turnover(T,\Delta_1,\Delta_2) = \frac{VolumeAll(T,\Delta_1,\Delta_2)}{S} \quad (3.10)$$

AutoCov is the autocovariance of transaction returns in the interval. For any $t \in D^{txn}$, denote by $L_t = \text{argmax}_s\{s : s < t, s \in D^{txn}\}$ the timestamp of the transaction right before time t. Then the autocovariance is:

$$\text{AutoCov}\,(T,\Delta_1,\Delta_2) = \text{ Average }\left[\log\left(\frac{P_t^{txn}}{P_{Lt}^{txn}}\right)\log\left(\frac{P_t^{txn}}{P_{L(Lt)}^{txn}}\right) : \text{t}\in D^{txn}\cap \text{Int}\,t^{\text{back}}\ (T,\Delta_1,\Delta_2)\right] \quad (3.11)$$

QuotedSpread is the average proportional nominal spread in the quotes over the lookback interval:

$$\text{QuotedSpread }\ (T,\Delta_1,\Delta_2) = \text{Average}\left[\frac{P_t^a - P_t^b}{P_t} : \text{t}\in \text{Int}^{\text{back}}\ (T,\Delta_1,\Delta_2)\right] \quad (3.12)$$

EffectiveSpread is the dollar-weighted percent effective spread over the interval:

$$\text{EffectiveSpread}\left(T,\Delta_1,\Delta_2\right)=\frac{\sum_{t\in D^{txn}nInt^{\text{back}}\ (T,\Delta_1,\Delta_2)}\left[\log\left(\frac{P_t^{txn}}{P_t}\right)\text{Dir}_t^{LR}V_tP_t^{txn}\right)}{\sum_{t\in D^{txn}\cap\ln t^{\text{back}}\ (T,\Delta_1,\Delta_2)}\left(V_tP_t^{txn}\right)} \tag{3.13}$$

Microstructure Measures As outlined by Easley et al. (2021), this set of measures comprises widely recognized market microstructure variables, which have gained prominence, especially after the May 6, 2010 "flash crash". With a rising need for warning signals indicating impending market stress, the use of measures to predict real-time market turbulence has become increasingly appealing, though some are also subject to scrutiny Andersen and Bondarenko (2014a). We'll provide additional evidence to assess their suitability. As mentioned earlier, the effectiveness of forecasting mini flash crashes relies on the model's real-time predictive capability. Hence, we calculate all microstructure measures using look-back windows. Specifically, we have:

Roll measure:

$$\begin{aligned}R_t&=2\sqrt{\left|\text{cov}\left(\Delta\boldsymbol{P}_t,\Delta\boldsymbol{P}_{t-1}\right)\right|}\\\Delta\boldsymbol{P}_t&=\left[\Delta P_{t-w},\Delta P_{t-w-1},\ldots\Delta P_t\right],\\\Delta\boldsymbol{P}_{t-1}&=\left[\Delta P_{t-w-1},\Delta P_{t-w},\ldots\Delta P_{t-1}\right],\end{aligned} \tag{3.14}$$

Where ΔP_t is the change in close price between bars $t-1$ and t and W is the lookback window size.

Roll impact, which is the Roll measure divided by the value traded over the look-back window, is:

$$\tilde{R}_t=\frac{2\sqrt{\left|\text{cov}\left(\Delta\boldsymbol{P_t},\Delta\boldsymbol{P}_{t-1}\right)\right|}}{P_tV_t} \tag{3.15}$$

Kyle's lambda is given by:

$$\lambda_t=\frac{P_t-P_{t-w}}{\sum_{i=t}^{t}b_iV_t} \tag{3.16}$$

Where b_i is the trade indicator inferred by Lee and Ready (1991), which is com-

puted through one lookback window.

Amihud's measure:

$$\lambda_t^A = \frac{1}{W} \sum_{i=t-W+1}^{t} \frac{|r_i|}{p_i V_i} \tag{3.17}$$

Where r_i, p_i, V_i are the return, price, and volume at look back window i and W is the lookback window size in terms of the number of trades.

Volume-synchronized probability of informed trading is estimated as:

$$\text{VPIN}_t = \frac{1}{W} \sum_{i=\tau-W+1}^{\tau} \frac{\left|P_t^a - P_t^b\right|}{V_i} \tag{3.18}$$

P_t^a and P_t^b are bid and ask quotes.

Microstructure Realized Volatility and Noise The final set of measures draws upon a body of literature concerning microstructure noise, aligning with theoretical insights provided by Zhang et al. (2005). Works such as Bossaerts et al. (2013) demonstrate that price volatility escalates as information is incorporated into prices. Short-term volatility can be highly informative for predicting real-time prices. We will introduce three measures:

Realized Volatility in each lookback window:

$$[P, P]_w = \sum_{t \in W} \left(P_{t_{i+1}} - P_{t_i}\right)^2 \tag{3.19}$$

Two-Scales Realized Volatility (TSRV) in each lookback window:

$$\widehat{\langle P, P \rangle}_T = [Y, Y]_T^{\text{avg}} \ - \frac{\bar{n}}{n} [Y, Y]_T^{\text{all}} \tag{3.20}$$

The combination is of two time scales, "all" and "average" sampling. More details could be found in Zhang et al. (2005).

The third one is $RealizedmoMentsofDisjointIncrements(ReMeDI)$ which is a new-developed measure to estimate microstructure noise, see Li and Linton (2022).

The last one is the daily Volatility Index (**VIX**). **VIX** has no look-back window; it is the CBOE Volatility Index, a popular measure of the stock market's expectation of volatility. It is based on S&P 500 index options and is provided daily by the Chicago Board Options Exchange.

3.3.2 Machine Learning Methods

Models

This paper employs four primary machine learning models for the prediction of mini flash crashes. These models encompass a regularized logistic regression (LASSO) as a representative of a linear parametric approach, alongside a penalized support vector machine (penalized-SVM) offering a nonparametric alternative. Additionally, two ensemble models, namely Random Forest and Extreme Gradient Boosting (XGBoost), are utilized. Detailed theoretical expositions of these models can be found in the works of Hastie et al. (2009) and Murphy (2013).

Consider the challenge of imbalanced classification, where the goal is to predict a response variable Y. Here, Y takes the value 1 when a mini flash crash is anticipated within the designated look-forward window, and 0 when no mini flash crash is expected. This prediction is based on a predictor vector X, utilizing a random sample (X_i, Y_i). We can represent the response vector as $\mathbf{Y} = (y_1, ...y_n)^T$. In our feature vectors, X_i, 232 dimensions, encompassing 11 time spans for each of the 21 predictor variables, with the final dimension representing the volatility index (VIX) from the previous trading day. Machine learning algorithms inherently have the capability to generate additional combinations of these predictors or select the most informative subsets from them.

Penalized Logistic Regression (LR)

Within the realm of well-established machine learning models, linear models stand out for their unique appeal stemming from their simplicity and high interpretability.

Logistic regression, a classification model, leverages the sigmoid function to map the value range of linear regression into the interval $[0, 1]$. In this context, the response variable is formulated as follows:

$$y = \frac{1}{1 + e^{-z}} \tag{3.21}$$

Through linear regression model:

$$Z = \beta^T \mathbf{X} + \epsilon \tag{3.22}$$

$\mathbf{X}$ represents the predictor variables. In the absence of some form of regularization, standard OLS in a large dimensional setting is likely to have poor out-of-sample predictive power due to in-sample overfitting. A standard method to address this issue consists in regularizing the model using a penalty function applied to normalized variables. Penalized least squares with an L_1 penalty is recognized as the Least Absolute Shrinkage and Selection Operator (LASSO). Specifically, consider $\bar{X} = \frac{1}{n}\sum_i X_i$ and $s_i = \sqrt{\frac{1}{n}\sum_i (x_i - \bar{x}_\imath)^2}$, representing the mean vector and standard deviations of predictor variables, respectively. Let $\bar{Z} = \frac{1}{n}\sum_i Z_i$ denote the mean of the linear regression response variable. Define the centered regression response as $\tilde{Z}_i = Z_i - \bar{Z}$ and standardized predictors as $\widetilde{\boldsymbol{X}}_l = \text{diag}\left(s_1^{-1}, s_2^{-1} \ldots s_p^{-1}\right)\left(\boldsymbol{X}_i - \overline{\boldsymbol{X}}\right)$ (for $i = 1, \ldots, n$). LASSO proceeds to fit the centered response onto the standardized predictors by solving the ensuing optimization problem:

$$\hat{\beta} = \text{argmin}_{\beta \in R^p} \left\{ \frac{1}{n} \sum_i \left(\widetilde{Z}_l - \beta^T \widetilde{\boldsymbol{X}}_l \right)^2 + \lambda \|\beta\| \right\} \tag{3.23}$$

This optimization problem could be solved by convex optimization. In this paper, we use the coordinate descent algorithm implemented in the Scikit-learn software by Python.

After I solve the coefficient $\hat{\beta}$, I can predict each new data $\boldsymbol{X}_{new}$ as:

$$\widehat{Y_{\text{new}}} = \frac{1}{1+e^{-\left(\overline{\boldsymbol{Z}}+\widehat{\boldsymbol{\beta}}^T \boldsymbol{X}_{\text{new}}\right)}}, \text{ with } \widetilde{\boldsymbol{X}_{\text{new}}} = \text{diag}\left(s_1^{-1}, s_2^{-1}, \ldots, s_p^{-1}\right)\left(\boldsymbol{X}_{\text{new}} - \overline{\boldsymbol{X}}\right) \tag{3.24}$$

LASSO is a simple and very easily explained model that shrinks the coefficients of less useful predictors towards zero. This can help us to rank the relevance of different predictors in the prediction problem. Through this process, we can summarize the feature that can show significant power to become the signal of mini flash crashes.

Support Vector Machine (SVM)

Support vector machine (SVM) is one of the most popular non-parametric algorithms for its outstanding performance and clear mathematical details. We use it as the representative of the non-parametric forecasting model. In this section, we briefly introduce SVM; more details may be found in Hastie et al. (2009) and Murphy (2013).

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite-dimensional space, which can be used for both classification and regression tasks.

Giving training vectors $x_i \in R^p, i = 1, \ldots n$ in two classes, and the response variable as $y \in \{1, -1\}^n$, we want to find a $\omega \in R^p$ and $b \in R$ such that $\omega^T \phi(x) + b$ can predict the sign of most x_{new} .

Having this ω, SVM solves the following primal optimization problem:

$$\min_{\omega, b, \zeta 2} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\zeta_i \tag{3.25}$$

$$\text{Subject to } y_i\left(\omega^T\phi(x) + b\right) \geq 1 - \zeta_i, \zeta_i \geq 0, i = 1, \ldots, n$$

Through the SVM, we are trying to maximize the margin between two classes by minimizing $|\omega|^2 = \omega^T\omega$. The perfect situation is the hyperplane that can separate all samples, which are $y_i\left(\omega^T\phi\left(x_i\right) + b\right) \geq 1$ for all samples. However, in the real world, samples are usually not perfectly separable by a hyperplane; we must allow some

samples for the generalizing ability of the model. Therefore, some samples need to be at a distance denoted as ζ_i, from their correct margin boundary. The penalty term C controls the strength of this penalty. The dual problem to the primal problem is:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$

Subject to $y^T\alpha = 0$ and $0 \leq \alpha_i \leq C$ for $i = 1, \ldots, n$, where e is the vector of all ones, and Q is the matrix: $Q_{ij} \equiv y_i y_j K\left(x_i x_j\right)$, with $K\left(x_i x_j\right) = \phi\left(x_i\right)^T \phi\left(x_j\right)$ is the kernel.

The kernel function can map the samples into higher dimensional or infinite-dimensional space; for more details see Hastie et al. (2009).

In this paper's mini flash crash forecasting problem, we have conducted many iterations to tune the hyper-parameters, determining that using a linear kernel (simple inner product $< x, x >$) and hinge loss to construct the optimization problem, can significantly outperform all other kernels. In the empirical results, we will simply show the linear kernel support vector machine results.

When the optimization problem is solved, we can use a support vector to predict a new sample x_new by:

$$\sum_{i \in SV}^{n} y_i \alpha_i K(x_i, x_{new}) + b \tag{3.26}$$

Then the predicted class corresponds to its sign.

Random forests (RF)

While lacking the interpretability of linear models, ensemble learning tree-style models exhibit impressive forecasting abilities in various practical scenarios. In this paper, we employ two ensemble models, starting with a random forest. Random forest, a scalable nonparametric learning method, is built upon individual decision trees. Given the instability and limited predictive power of a single decision tree, constructing a forest of many trees serves as a direct approach to enhance predictive accuracy. By averaging outcomes from numerous sampled decision trees, the variance of predictions

decreases, resulting in more stable and reliable forecasts.

According to Hastie et al. (2009)), random forests are fitted via iteratively growing regression trees through the bootstrap sampling process from the given data set. Hastie et al. (2009) describe the algorithm as follows:

- For $b = 1$ to B
  - Draw a bootstrap sample Z* of size N from the training data.
  - Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{\min}$ is reached.
    - Select m variables at random from the p variables.
    - Pick the best variable/split-point among the m.
    - Split the node into two daughter nodes.
  - Output the ensemble of trees $\{T_b\}_1^B$.

For new data, we want to predict x: Let $\hat{C}_b(x)$ be the class prediction of the b th random-forest tree. Then $\hat{C}_{rf}^B$(x) = majority vote $\left\{\hat{C}_b(x)\right\}_1^B$. (Hastie et al. (2009)).

Predictions from bagging are often highly correlated as the samples are from the same data set. Through the bagging of many independently trained decision trees from bootstrapped samples, but no selection of variables at the decision tree's node, the random forest can achieve better performance as it increases the independence of resulting trees and reduces the dependence of the prediction. Through the random forest, we could obtain a smaller variance than a single decision tree, thus obtaining more reliable predictions.

Extreme Gradient Boosting (XGB)

The second ensemble model we use is the extreme gradient boosting (XGBoost), which is from Friedman (2001). Out of the world of deep learning, XGBoost has been the most popular algorithm to impress the machine learning community in the past

few years. As an ensemble model, XGBoost relies on many weak-based tree learners. Those weak learners' biases are high, and their predictive power is just slightly better than random guessing.

In contrast to bagging techniques such as Random Forest, in which trees are grown to their maximum extent, boosting attempts to create small trees, which are not deep and easy to interpret. XGBoost starts from initial model F_0 to predict the target variable y, and the residual $y - F_0$. The new model h_1 is fit to the residual, through the combination of h_1 and F_0, and the mean squared error will be reduced. Then update $F_1(x) \leftarrow F_0(x) + h_1(x)$, until the residuals are minimized as much as possible.

The algorithm proceeds as follows: Given training set $\{(x_i, y_i)\}_{i=1}^{N}$, and a well-defined differentiable loss function $L(y, F(x))$, several weak tree learners M and a learning rate $\propto$.

- Initialize model with a constant:

$$f_0(x) = \widehat{\operatorname{argmin}} \sum_{l=1}^{N} L\left(y_l, \theta\right). \tag{3.27}$$

- For m = 1 to M :
  - Compute the gradients and hessians:

$$\begin{aligned} \hat{g}_m\left(x_i\right) &= \left[\frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)}\right]_{f(x)=f_{(m-1)}(x)} \\ \hat{h}_m\left(x_i\right) &= \left[\frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)^2}\right]_{f(x)=f_{(m-1)}(x)} \end{aligned} \tag{3.28}$$

  - Fit a weak tree leaner using the training set $\left\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right\}_{i=1}^{N}$ by solve the optimization problem:

$$\circ\widehat{\emptyset_m} = \text{argmin}_{\emptyset} \sum_{i=1}^{N} \hat{h}_m(x_i) \left[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \emptyset(x_i) \right]^2 \tag{3.29}$$
$$\hat{f}_{(m)}(x) = \propto \widehat{\emptyset_m}(x)$$

– Update the model:

$$\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) + \hat{f}_{(m)}(x)$$

- Output $\hat{f}(x) = \sum_{m=0}^{M} \hat{f}_{(m)}(x)$

XGBoostContributors (2023)

More details could be found in Friedman (2001) and XGBoost documents (XGBoostContributors (2023)).

Measuring Prediction Accuracy

We use Receiver Operating Characteristic (ROC) and Area under the ROC Curve (AUC) metrics to evaluate our model's predictive accuracy. Predicting mini flash crashes resembles a supervised anomaly detection problem, where such occurrences are rare, leading to significant dataset imbalance. In a hypothetical scenario where only 0.1% of look-forward windows are labeled as mini flash crashes, predicting no crashes would yield a misleadingly high accuracy score of 99.9%.

In such cases, ROC provides a more suitable measure for evaluating prediction accuracy in imbalanced datasets. Originating from signal analysis technology developed during World War II, ROC has wide application, including medical issue detection. It comprehensively depicts a classification model's performance across all thresholds, graphing two key parameters.

- True Positive Rate (TPR): $TPR = \frac{TP}{TP+FN}$
- False Positive Rate (FPR): $FPR = \frac{FP}{TP+TN}$

TP, TN, FN, and TN are all from the confusion matrix, see Figure 3.3:

Figure 3.3

An ROC curve visually depicts the relationship between True Positive Rate (TPR) and False Positive Rate (FPR) across various classification thresholds. Lowering the threshold increases both False Positives and True Positives. The curve organizes test samples by predicted probability of being True, enabling sequential prediction and calculation of TPR and FPR. Well-performing models exhibit an initially steep ascent followed by a gradual rise. AUC quantifies the entire area beneath the ROC curve, ranging from (0,0) to (1,1), akin to calculating the integral of a function. It offers a comprehensive performance measure across all thresholds, representing the probability of ranking a randomly chosen positive example higher than a randomly chosen negative one. Figure 3.4 shows the ROC curve of our best-tuned model for 15s mini flash crashes in Window 7, covering July to September 2018.

Figure 3.4

Imbalanced Data Processing Strategies

As previously mentioned, our problem entails dealing with imbalanced data. In addition to utilizing the raw data as-is, we will also implement five distinct strategies to mitigate the data imbalance issue.

Undersampling (UD): One method to tackle data imbalance involves removing excess samples from the majority class, specifically those where the look-forward window doesn't result in a mini flash crash, to establish a balanced sample set relative to the minority class. While straightforward, this approach results in significant data loss from the majority class, potentially weakening its impact, especially for instances offering crucial information between classes. For example, in support vector machine algorithms, removed samples may include support vectors near the margin hyperplane on the majority class side, potentially further biasing the model.

Oversampling (OV): Another strategy involves randomly duplicating synthetic samples within the minority class, particularly when the look-forward window results

in a mini flash crash. This approach aims to balance the class distribution by creating additional synthetic samples. However, it's important to note that this strategy carries potential risks. Introducing noise samples into the minority class may amplify noise presence and lead to overfitting the model.

Synthetic Minority Oversampling Technique (SMOTE): SMOTE is a sophisticated oversampling technique that utilizes the K-Nearest Neighbors (KNN) algorithm to generate new samples for the minority class. These new samples mimic the originals while introducing some variation. Compared to basic oversampling, SMOTE functions as an ensemble learning method, potentially reducing variance and overfitting risk. However, it relies on the minority class to create synthetic samples, which might amplify noise due to shifts in data distribution. Moreover, SMOTE is computationally more complex than both oversampling and undersampling techniques.

Threshold Moving (TM): This strategy involves adjusting the classification threshold to increase the model's sensitivity to the minority class. In balanced data scenarios, the threshold typically remains at 0.5. However, in imbalanced data situations, the threshold can be modified to align with the ratio of minority to majority class samples within the training dataset.

Ensemble Undersampling (EN): As previously discussed, undersampling removes numerous majority class samples to balance the dataset, leading to substantial data loss. An alternative involves multiple rounds of undersampling. For instance, with 10,000 majority class samples and 50 minority class samples, 200 new sets can be created by selecting all 50 minority samples and randomly choosing an additional 50 for each set. These 200 models are trained independently, and their results are combined for the final prediction.

This strategy has limitations, including increased computational complexity and potential overfitting due to repeated use of minority class samples. However, in anomaly detection scenarios like ours, where successfully detecting mini flash crashes is paramount, this strategy proves invaluable with high-quality minority class data.

3.4 Data description and pre-processing

In this paper, we use all 95 stocks in S&P 100 from the NYSE Tick and Quote (TAQ) data set. The time range of data covers two years, 2017 and 2018. Following Hendershott and Moulton (2011), we use the same standard to clean the quote and transaction data. [2]

To enhance the predictive robustness of our model, we conducted a meticulous segmentation of the two-year dataset, partitioning it into eight discrete windows, each spanning a three-month period. Within each window, a structured approach was employed: the first month was allocated for training the model, the second month for the fine-tuning of parameters aimed at identifying the most optimal configuration, and, subsequently, the third month was reserved for assessing the model's performance.

This study utilizes short-term, high-frequency transaction and quote data to predict mini flash crashes. Unlike some existing research, which often aggregates data using event windows or information bars, exemplified by Easley et al. (2021), we adopt a temporal framework, relying on a Time clock approach.

In practice, portfolio managers facing an imminent mini flash crash require real-time predictions rather than waiting for additional events to accumulate. Therefore, to ensure real-time predictive capability, we exclusively employ a temporal framework for detecting relevant variables.

Mini flash crashes have been extensively investigated in the literature, with the most accepted definition originating from Nanex (2010). According to Nanex (2010), mini flash crashes are succinctly defined as follows:

- The time window does not exceed 1.5 seconds;

[2]When we do the data filtering, we restricted the trading data in regular trading hours from 9:30 am to 4 pm. We use only trades for which TAQ's CORR filed is zero, one, or two and for COND field is either blank or equal to @, E, F, I, J, or K. Obviously, we eliminate trades with nonpositive prices or quantities. We also remove trades with prices more than(less than) 150%(50%) of the previous trade price. After that, we restrict quotes for which TAQ's MODE field is equal to 1, 2, 6, 10, 12, 21, 22, 23, 24, 25, or 26. Then we eliminate quotes with nonpositive prices or sizes or with bid prices greater than the asking price. We also exclude quotes when the quoted is greater than 25% of the quote midpoint or when the asking price is more than 150% of the bid price.

- Price change exceeds 0.8%;
- At least 10 times tick down before ticking up OR at least 10 times tick up before ticking down;

In this paper, three additional, more lenient time window definitions are introduced: 180 seconds, 90 seconds, and 15 seconds, in conjunction with the Nanex (2010)'s established 1.5-second definition. These four discrete time intervals function as distinct criteria for the analysis and categorization of mini flash crashes, accounting for a broad spectrum of time frames and accommodating the diverse operational velocities of traders.

For individual investors and slower-paced traders, the 180-second time window provides ample time for trading activities to unfold. Mini flash crashes within this interval encompass a range of trader speeds and behaviors. In contrast, the 90-second timeframe represents a medium-term window where slower traders may still influence market dynamics to some extent. However, mini flash crashes within this timeframe are mainly influenced by fast-paced traders, with possible participation from slower traders.

The third temporal definition used is 15 seconds, a brief timeframe where individual traders are unlikely to significantly impact market dynamics. However, it's crucial to note that machine-executed trading algorithms vary in speed. While some quantitative trading firms use automated systems that respond within seconds, high-frequency traders (HFTs) operate at a much faster pace, executing trades at the nanosecond level. This study aligns with Nanex (2010)'s 1.5-second definition, focusing exclusively on mini flash crashes involving extremely rapid trading algorithms.

In addition to Nanex (2010)'s definition, Dugast and Foucault (2018) define mini flash crashes as large sudden price drops or spikes followed by quick price reversals, i.e. "V-shape" or "inverted V-shape" price movements. Inspired by this idea, a reverse version of mini flash crashes is implemented:

- The time window does not exceed 180 seconds/90 seconds/15 seconds/1.5 sec-

onds;

- Price change exceeds 0.8%;
- At least 10 times tick down before ticking up AND at least 10 times tick up before ticking down;

In pursuing the capture of the entire spectrum of mini flash crashes and in the endeavor to construct a real-time forecasting model, a comprehensive approach is undertaken. This approach entails considering all possible time intervals within each trading day. To achieve this, a forward-looking window is employed for the detection of mini flash crashes:

$\text{Int}^{\text{forward}}(T, T+\Delta) = \{t \in R : T < t \leq T+\Delta\}$, a span Δ equals 180s/90s/15s/1.5s.

Table 3.1 shows the descriptive statistics of all the windows, covering the time range from January 2017 to December 2018, including four different definitions of mini flash crashes without reversal.

Table 3.1

Table 3.2 shows the descriptive statistics of all of our windows, cover time range from January 2017 to December 2018 includes four different definitions of mini flash crashes with reversal.

Table 3.2

3.5 Empirical Analysis

In addition to using the original imbalanced data, all previously discussed strategies were applied alongside four models: logistic regression with an l_1 penalty (LASSO), random forest, support vector machine, and XGBoost. As outlined in Section 3, we investigate eight distinct scenarios concerning mini flash crashes based on different look-forward windows (180s/90s/15s/1.5s) and reversal criteria, as defined in Section 3.4.

Inspired by Nanex (2010), this paper adopts a concise 1.5-second time window to define mini flash crashes. This duration is chosen to account for the swift pace of trading activity, where even slower traders cannot react quickly enough to significantly impact stock prices. To provide further insights, we introduce three additional time windows: 180 seconds (long-term), 90 seconds (medium-term), and 15 seconds (short-term).

Aligned with Nanex (2010), the extremely short-term category is the strictest classification of mini flash crashes. Within the 15-second timeframe, the short-term perspective primarily reflects fast traders' influence, offering insights into overall market conditions. Moving to the medium term, a robust check on the long-term viewpoint is established, revealing the involvement of fast-reactive slow traders in these events. Finally, mini flash crashes unfold over an extended 180-second window in the long term, allowing both slow and fast traders ample time to react to market dynamics.

In my emprical analysis, ensemble learning model XGBoost, has become dominant in predictive analytics due to their superior performance over traditional linear models. Techniques like bagging (e.g., random forests) and boosting (e.g., gradient boosting machines) combine multiple models to enhance predictive accuracy. These ensemble methods reduce variance and bias, leading to significantly improved performance compared to single models like linear regression. This approach leverages the strengths of various models, resulting in robust and reliable predictions.

As outlined in section 3.3.2, this paper employs ROC and AUC as the evaluation metrics to assess the performance of machine learning predictions, primarily due to the challenges posed by imbalanced data.

In this section, we also summarize and analyze the features' importance in mini flash crash predictions; the histograms of all windows' features importance are provided in Internet Appendix 3.8.

3.5.1 Nanex (2010) Mini Flash Crashes Prediction

Tuning and Testing

In this section, our focus is directed towards the real-time prediction of mini flash crashes that adhere to the strictest criteria outlined by Nanex (2010), requiring a time range not exceeding 1.5 seconds. As in the preceding sections, we explore scenarios both with and without reversals, employing various strategies for processing imbalanced data in conjunction with eight distinct machine learning models. In accord with our previous approach, we aggregate the counts of each feature selected by the reduced model when implementing reduced-feature models to evaluate the importance of individual features. Given the substantial data processing workload, we perform undersampling of the data without mini flash crashes to a scale 100 times the total number of mini flash crashes in each month within each rolling window, under the condition that the ratio of time intervals in all intervals is lower than 1% for each stock.

Table 3.3 and Table 3.4 present the AUC tuning results for all 1.5s Nanex (2010) mini flash crashes without reversals across all eight windows. As can be observed, ensemble undersampling achieves the highest performance across all eight windows for both versions, with and without reversals, of 1.5s mini flash crashes. For the versions without reversals, XGBoost combined with ensemble undersampling demonstrates the best performance in four windows, while reduced-feature XGBoost achieves the highest AUC in three of the eight windows. The remaining window sees the best performance from reduced-feature random forest, which is also an ensemble-style model. In comparison to the earlier sections, the tuning of the AUC results is not as successful, with five AUC values exceeding 0.8, including two exceeding 0.9. In the remaining three windows, the best AUC values are 0.774, 0.778, and 0.736. While not perfect, these values still indicate a reasonable level of predictability.

The tuning AUC values for Nanex (2010) mini flash crashes with reversals are presented in Table 3.5 and Table 3.6. Our results are generally consistent with the

versions without reversals. XGBoost combined with ensemble undersampling exhibits the best performance in four windows, while reduced-feature XGBoost achieves the highest AUC in three of the eight windows. The only window where random forest achieves the highest performance is Window 5, mirroring the results from the version without reversals. All the results, whether with or without reversals, are consistent with the expectation presented in Section 3.4, which highlights the prevalence of reversals in most mini flash crashes.

Utilizing the best parameter sets, models, and the optimal imbalanced data processing strategy of ensemble undersampling, the model performance was assessed. The Nanex (2010) definition necessitates a time range not exceeding 1.5 seconds, categorizing these mini flash crashes as extreme low-probability events that are more challenging to predict than any of our previous versions. Consequently, it is essential to set realistic expectations for the test results. Nevertheless, a commendable performance is still achieved.

For both 1.5s Nanex (2010) mini flash crashes with, and without, reversals, the AUC values from the out-of-sample tests indicate that five of eight windows exceeded 0.8. In the other windows (Windows 1, 7 and 8), the AUC falls within the range of 0.7 to 0.8. The results for 1.5s Nanex (2010) mini flash crashes without reversals are presented in the last row of Panel A in Table 3.9, while the reversal test results are displayed in the last row of Panel B.

Our findings provide evidence that mini flash crashes, as defined by Nanex (2010), are universally predictable by machine learning. This discovery sheds light on the development of an early warning system for mini flash crashes, offering investor protection and enhancing market stability ahead of circuit breakers.

Feature Importance

Relying on the successful prediction, we also evaluated the feature importance in 1.5s mini flash crashes. While the critical role played by similar features in prediction is observed, including those consistent with the following three versions of mini flash

crash definitions, such as $LobImbalance$, $Lambda$, $QuotedSpread$, and $ESpread$, it is significant that the substantial advantages of specific features like $LobImbalance$ and $Lambda$ have diminished. In certain windows of prediction, volume-related features such as $VolumeAll$, $VolumeAvg$, and $VolumeMax$ have become equally, if not more, important. Moreover, in some windows, the role of real-time microstructure noise, calculated according to Li and Linton (2022), seems to have become more vital than in any of the previous versions.

Due to the stringent 1.5-second definition of mini flash crashes by Nanex (2010), predicting these extremely rare events poses a significant challenge. Feature importance statistics reveal that relying on individual or a few features as signals for mini flash crashes becomes less feasible in machine learning models. The impact of limit order book imbalance on driving mini flash crashes decreases, making it harder to rank the importance of multiple features. This suggests that these rigorously defined mini flash crashes, characterized by their low-probability nature, may stem from a complex interplay of real-time market conditions and various trading algorithms. While somewhat predictable, they defy easy summarization in terms of identifying regular patterns, akin to human recognition of images or scenes. Machine learning algorithms, leveraging extensive data processing, can identify these patterns to some extent, offering a degree of predictability for mini flash crashes.

We've observed that volume and trade-related features, such as Volume Average and Breadth, strongly influence prediction success. This is interpretable: as a mini flash crash nears, informed traders swiftly impact the market, with algorithms and HFTs detecting anomalies within seconds. This information manifests in quotes and transactions just before the crash. The significance of $ReMeDI$ performance, assessing short-term microstructure noise, further confirms this observation.

Our analysis emphasizes the significant impact of microstructure measures like Kyle's lambda and the Probability of Informed Trading (VPIN) on prediction accuracy. Kyle's lambda represents the cost of acquiring liquidity within a timeframe, while VPIN estimates the presence of informed traders with superior information.

These findings suggest that informed traders likely play a crucial role in triggering mini flash crashes. They use private information to influence market liquidity, raising costs for other investors seeking liquidity and driving the market towards a crash.

3.5.2 Short Term Mini Flash Crashes Prediction

Tuning and Testing

In this section, we shift our focus to the real-time prediction of short-term (15s) mini flash crashes, both with and without reversals. We apply a combination of data imbalanced processing strategies and eight distinct machine learning models to determine the most effective approach. Given the brief 15-second duration of these mini flash crashes, it is reasonable to assume that they are predominantly driven by trading algorithms, even if these algorithms are not necessarily as fast as High-Frequency Trading (HFT) algorithms. Utilizing a methodology akin to that outlined in Section 3.5.1, we examine a dataset encompassing the period from January 2017 to December 2018. Across this 24-month span, the data is segmented into eight discrete windows, each spanning a three-month duration. Within each window, the initial training dataset comprises the first month, while the second month is dedicated to tasks such as hyper-parameter optimization, model selection, and identification of optimal data imbalanced processing strategies.

This section, along with subsequent sections, exclusively showcases the best-selected tuning model and its corresponding results. In Panel A of Table 3.7 and Table 3.8, we present the optimal AUC tuning results for all 15s mini flash crashes without reversals across the eight defined windows. The combination of ensemble undersampling and reduced-feature XGBoost consistently outperforms alternative approaches in six out of eight windows. Notably, ensemble undersampling in conjunction with reduced-feature random forest and reduced-feature XGBoost represents two other effective strategies. All tuning AUC scores either meet or surpass 0.8, with some even exceeding the 0.9 threshold. The consistent superiority of ensemble undersam-

pling underscores its effectiveness in addressing data imbalance. This performance provides further evidence that the time interval preceding mini flash crashes often exhibits robust patterns compared to normal intervals. While these patterns may pose challenges in terms of causal interpretation, machine learning proves adept at detecting them.

The tuning AUC values for 15s mini flash crashes with reversals are available in Panel B of Table 3.7 and Table 3.8. The combination of ensemble undersampling and XGBoost consistently excels in seven out of eight windows and in five out of nine windows. In the remaining three windows, reduced-feature random forest and full-feature XGBoost achieve outstanding performance, both being ensemble-style models akin to reduced-feature XGBoost. All AUC values exceed 0.87, with some exceeding 0.9, demonstrating strong predictive performance.

Adhering to the identical process, we proceed to retrain the models utilizing the finest parameter sets, selected models, and the optimal data imbalanced processing strategy iden tified during the tuning procedure. The training dataset aligns with the second month of each window, and we subsequently perform out-of-sample tests deploying the test data from the third month within each window. The outcomes for short-term (15s) mini flash crashes without reversals are delineated in the third row of Panel A in Table 3.9, whereas the results for reversal tests are exhibited in the third row of Panel B.

In both short-term mini flash crashes, both with and without reversals, the AUC values derived from out-of-sample tests showcase remarkably robust performance. Across 14 of 16 windows, the AUC exceeds 0.9, with the lowest recorded value standing at 0.88.

These findings emphasize the persistent strength of patterns in short-term (15s) mini flash crashes, primarily influenced by trading algorithms. This substantiates our anticipation of leveraging machine learning-based anomaly detection as an advanced model for establishing an early warning system to detect mini flash crashes before circuit breakers are triggered.

Feature Importance

The feature importance in short-term (15s) mini flash crashes displays variations comparable to the subsequent two versions. Although the limit order book imbalance stands out as one of the strongest predictors, in specific windows variables such as *Lambda*, *QuotedSpread*, and $Turnover$ outshine $LobImbalance$. This divergence is notable and contrasts with the patterns observed in subsequent sections covering long and medium-term analyses.

As defined in Section 3.3.1, *Lambda* measures the price change in the interval relative to the total volume, and $Turnover$ represents the speed of transactions. Both variables provide insights into actual transactions, in addition to bid and ask quotes. This discovery may imply that in the case of short-term (15s) mini flash crashes primarily driven by trading algorithms, as these events approach, the triggering of specific stop-loss programs likely results in a significantly larger number of market orders being submitted by trading algorithms. This is done to rapidly liquidate or reposition assets. These actions, which impact trading volume within the market, may further contribute to the onset of mini flash crashes. These behaviors can be captured by turnover rate and lambda, making them important signals in predicting 15s mini flash crashes.

Furthermore, volume-related features such as Volume Average and Breadth continue to exhibit their strength. Additionally these, microstructure measures such as Kyle's lambda, the Probability of Informed Trading (VPIN), and $RollMeasure$ demonstrate stronger performance than in the long and medium-term versions of the empirical results. This discovery may suggest that in even shorter and faster mini flash crash events, informed trading might play an even more pivotal role. Importantly, all our findings for mini flash crashes, whether with or without reversals, are consistent

3.5.3 Medium Term Mini flash crashes Prediction

Tuning and Testing

In this section, we shift our focus to the real-time prediction of medium-term (90s) mini flash crashes, both with and without reversals. We utilize the same combination of imbalanced data processing strategies as employed in the previous section on Nanex (2010) version (1.5s) and medium-term (15s) mini flash crashes. The primary distinction from the prior two versions lies in the 90s timeframe, where slow traders may contribute to mini flash crashes, not just fast traders as observed in algorithmic scenarios.

As with the earlier sections covering 1.5s and 15s, we analyze a dataset spanning two years, from January 2017 to December 2018. Over this 24-month period, we segment the data into eight distinct windows, each spanning a three-month period. To capture the most recent information within each window, the first month serves as the initial training dataset, and the second month is dedicated to hyper-parameter optimization, model selection, and the identification of optimal imbalanced data processing strategies.

The Panel A in Table 3.7 and Table 3.8 presents the best AUC tuning results for all 90s mini flash crashes without reversals across all eight windows. The combination of ensemble undersampling and reduced-feature XGBoost consistently outperforms other approaches in six of all eight windows. All tuning AUC scores reach or exceed 0.8, with some approaching 0.9. For the other two windows, direct undersampling shows the best performance through reduced-features random forest and XGBoost in Windows 7 and 8, respectively. These findings provide evidence that mini flash crashes in the medium term also have strong patterns that could be detected in real time by machine learning.

The tuning AUC values for 90s mini flash crashes with reversals are available in Panel B of Table 3.7 and Table 3.8.The combination of ensemble undersampling and reduced-feature XGBoost consistently excels in seven of eight windows. The

only exception is Window 1, covering data from July 2017 to September 2017, where XGBoost, in conjunction with ensemble undersampling, achieves the highest performance with an AUC of 0.96. As with mini flash crashes without strict requirements on reversals, machine learning models demonstrate strong predictive performance.

Having utilized the best parameter sets, models, and the optimal imbalanced data processing strategy of ensemble undersampling, we retrain the models using the tuning data, which corresponds to the second month of each window. Subsequently, we conduct out-of-sample tests using the test data, representing the third month in each window. The results for medium-term (90s) mini flash crashes without reversals are presented in the second row of Panel A in Table 3.9, while the reversal test results are displayed in the second row of Panel B.

For both medium-term mini flash crashes with and without reversals, the AUC values from the out-of-sample tests indicate remarkably strong performance. In 14 of 16 windows, the AUC exceeds 0.9, with the lowest value being 0.892. These results demonstrate the robust predictability of mini flash crashes at the minute level.

Feature Importance

The feature importance in medium-term mini flash crashes exhibits a striking consistency with the following long-term version. The feature with the highest importance is the limit order book imbalance, reinforcing our finding that, on a minute-to-minute level, mini flash crashes are heavily influenced by information asymmetry.

Additionally, variables such as *Lambda*, *QuotedSpread*, and *ESpread* continue to demonstrate their significant importance in predicting mini flash crashes. These variables reflect short-term market liquidity and stability, maintaining a robust correlation with the incidence of mini flash crashes across all eight window configurations.

Furthermore, the strong performance of Volume Average and Breadth provides further evidence that informed traders with privileged information can swiftly influence the market. Algorithms and HFTs can rapidly detect anomalies, resulting in increased trading activity and liquidity imbalances just moments before an actual

crash occurs. This observation is further supported by the noteworthy performance of $ReMeDI$ as a feature, which assesses short-term microstructure noise.

Our analysis also emphasizes the substantial influence of microstructure measures, such as Kyle's lambda and the Probability of Informed Trading (VPIN), on prediction accuracy

3.5.4 Long Term Mini flash crashes Prediction

Tuning and Testing

In this section, we explore real-time predictions of long-term (180s) mini flash crashes, covering scenarios with and without reversals. The key distinction between these two definitions lies in the assumption that within a 180s timeframe, it is reasonable to presume that even slow traders, such as humans, can exert influence. The market has the capacity to assimilate information from both fast traders, such as HFTs, and slow traders. However, in a 90s timeframe, slow traders may only contribute "a portion of their information" to the market.

We employ all five strategies for processing imbalanced data alongside the original dataset, which serves as our baseline. The objective is to assess eight distinct machine learning models to identify the best-performing model. These models encompass Logistic regression with l_1 regularization, support vector machines, random forests, and XGBoost, each with its corresponding reduced-feature model. Reduced-feature models are generated by excluding features that do not significantly contribute to predictability. Furthermore, when implementing reduced-feature models, we aggregate the counts of each selected feature to evaluate the importance of individual features.

To ensure a comprehensive understanding of long-term trends and the robustness of our findings, we scrutinize a dataset spanning two years, from January 2017 to December 2018. Over this 24-month duration, we partition the data into eight distinct windows, each encompassing a three-month period. To capture the most recent information, within each window, the first month is allocated for the initial train-

ing dataset, the second month is dedicated to hyper-parameter optimization, model selection, and the identification of the optimal imbalanced data processing strategies.

The Panel A in Table 3.7 and Table 3.8 present the best AUC tuning results for all 180s mini flash crashes without reversals across all eight windows. It is evident that the combination of ensemble undersampling and reduced-feature XGBoost consistently delivers the best performance in seven out of eight windows. Window 1, is very close to being the best, with the direct undersampling approach taking the top spot. All our tuning AUC scores achieve values of 0.8 or above, with some even approaching 0.9.

Furthermore, it is worth noting that while direct undersampling may lead to a substantial loss of data, it still manages to achieve reasonably good performance. This observation suggests that the signals associated with "mini flashes" can be quite robust when compared to the time intervals preceding mini flash crashes. Through the ensemble undersampling approach, which involves comparing all time intervals marked with mini flash crashes and systematically reconstructing a balanced dataset by integrating them with other datasets containing no mini flash crashes, we have successfully achieved the best performance.

The tuning AUC values for 180s mini flash crashes with reversals are available in Panel B of Table 3.7 and Table 3.8. The combination of ensemble undersampling and reduced-feature XGBoost consistently delivers the best performance across seven of the eight windows. The sole exception is Window 3, covering data from July 2017 to September 2017, where XGBoost outperforms and achieves an AUC of 0.86. As with mini flash crashes without strict requirements on reversal, the machine learning models exhibit strong predictive performance. This observation clearly highlights the distinct patterns in the limit order book when a mini flash crash is impending.

By employing the best parameter sets, models, and the optimal imbalanced data processing strategy of ensemble undersampling, we retrain the models using the tuning data, which corresponds to the second month of each window. Subsequently, we conduct out-of-sample tests using the test data, which represents the third month in

each window. The results for long-term (180s) mini flash crashes without reversal are presented in the first row of Panel A in Table 3.9, while the reversal test results are displayed in the first row of Panel B.

For both long-term mini flash crashes with and without reversals, the AUC values from the out-of-sample tests indicate remarkably strong performance. In 14 of 16 windows, the AUC exceeds 0.9, with the lowest value being 0.899.

Furthermore, as evident from the data description and pre-processing section, most mini flash crashes exhibit reversals. This observation leads to a reasonable inference that mini flash crashes are events intricately linked to market conditions and are significantly influenced by the stability of the financial market.

Feature Importance

It is noteworthy that through all eight windows and for both long term mini flash crashes with and without reversal, the limit order book imbalance takes the first place on feature importance. The limit order book imbalance encompasses the differences in stock market information that investors have regarding price changes at a given moment, as reflected by disparities in the quotes provided by buyers and sellers. This market condition is highly likely to be strongly correlated with the occurrence of mini flash crashes. This finding is highly reasonable as it highlights how short-term market information asymmetry could be a contributing factor to mini flash crashes. High-speed trading algorithms may indeed exacerbate this phenomenon within the trading ecosystem.

In addition to the limit order book imbalance, variables such as *Lambda*, *QuotedSpread*, and *ESpread* also exhibit significant importance in predicting mini flash crashes. They all reflect the short-term market liquidity and stability and show a strong correlation with the occurrence of mini flash crashes. This consistent result holds across all eight window configurations, both with and without reversal, in our predictive analysis.

3.6 Conclusion

In this study, we pioneer the application of machine learning techniques to predict real-time mini flash crashes in individual stocks. Our research provides empirical evidence and methodologies for forecasting mini flash crashes using data from the limit order book across four distinct time frames: 180 seconds, 90 seconds, 15 seconds, and 1.5 seconds. Notably, all these intervals fall within the 5-minute circuit breaker system set by the SEC, suggesting the viability of real-time early warning systems for mini flash crashes. Our analysis spans mini flash crashes in S&P100 stocks from 2017 to 2018, revealing that most of these events swiftly reverse within a short time frame. This observation underscores the systemic issues in current market mechanisms, particularly the prevalence of high-frequency algorithmic trading, as the primary drivers of mini flash crashes.

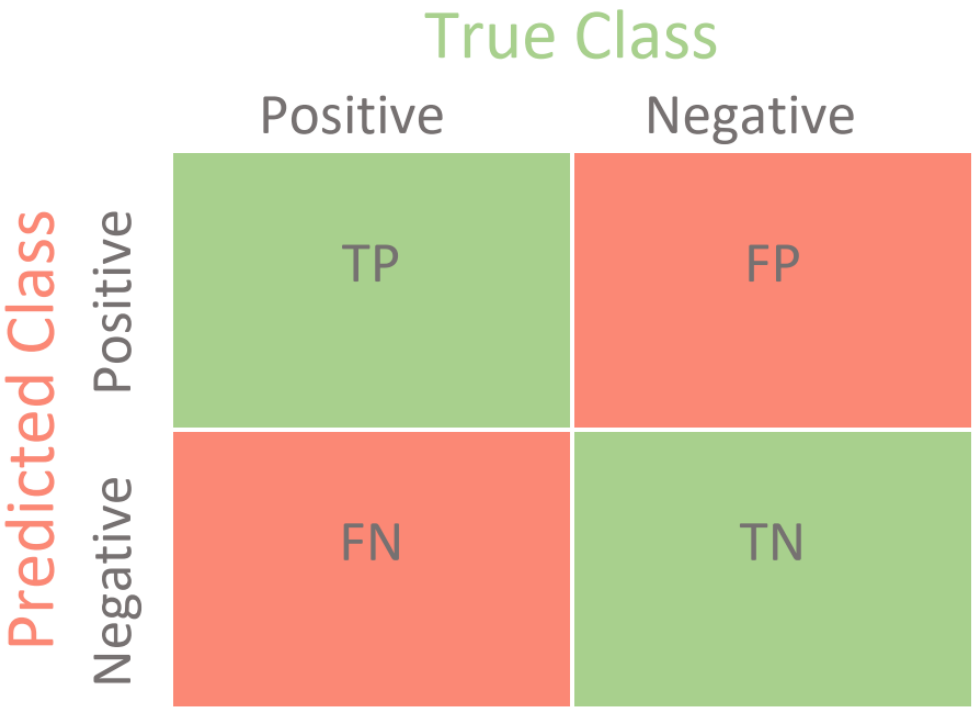
We introduce different time frames based on the involvement of slow traders and fast, automated trading algorithms. The 180-second and 90-second time frames accommodate the participation of slow traders, contributing to enhanced predictability, while the 15-second and 1.5-second time frames focus more on algorithmic trading behaviors. Our findings indicate varying levels of predictability across these time frames, with machine learning models showing efficacy in forecasting mini flash crashes, albeit with decreasing predictive accuracy in ultra-short durations. Overall, our research underscores the importance of monitoring real-time market dynamics and offers insights for investors and regulatory authorities in effectively managing and mitigating the impact of mini flash crashes.

3.7 Appendix 3. Figure and Table of Chapter 3

Source: Levine (2015)

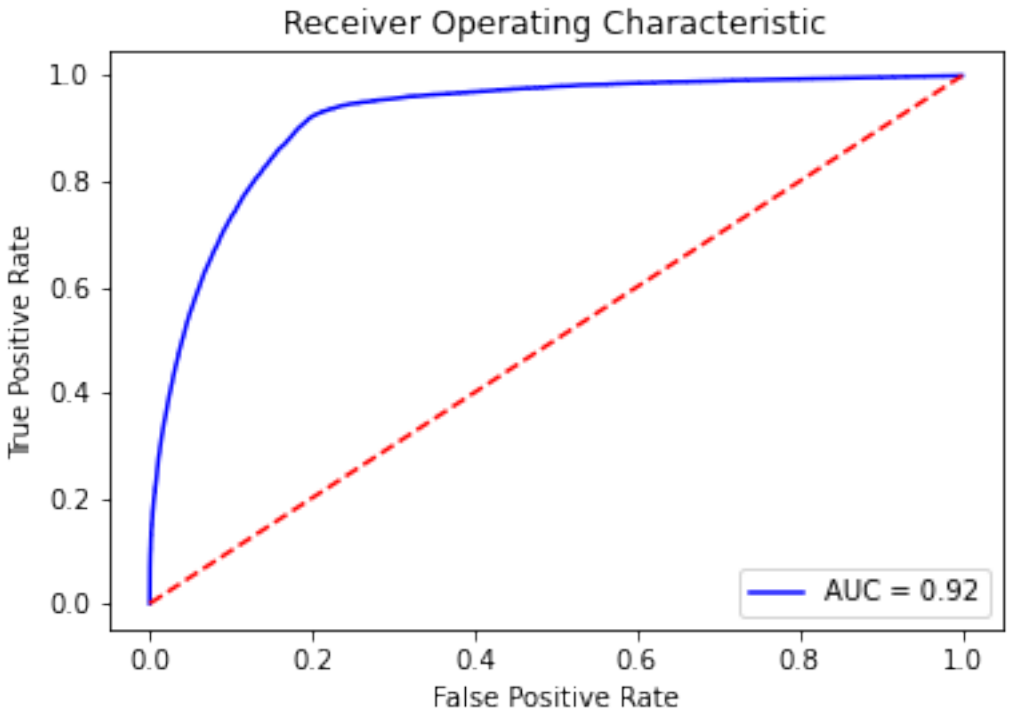
Figure 3.1: Flash Crash, 06 May 2010

Figure 3.2: Mini flash crash, Nanex (2010) definition, AAPL, 01/09/2017



Source: Mohajon (2020) Confusion Matrix is used to evaluate the performance of a classification model. It displays the counts of true positive (TP), false positive (FP), false negative (FN), and true negative (TN) predictions. The rows represent the predicted class, while the columns represent the true class.

Figure 3.3: Confusion Matrix



ROC of our best tuning result of 15s Mini Flash Crash in Window 7, The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

Figure 3.4: ROC

	Window1				Window2				Window3				Window4			
	180s	90s	15s	1.5s	180s	90s	15s	1.5s	180s	90s	15s	1.5s	180s	90s	15s	1.5s
Train																
No. Stocks:	92	91	54	19	81	77	33	14	82	70	28	8	89	82	41	22
0:	222,953	462,021	2,993,543	30,314,733	215,398	444,430	2,849,597	28,769,886	244,541	497,331	3,031,001	30,323,834	255,366	521,831	3,243,399	32,601,218
1:	29,747	43,379	38,857	9,267	24,401	35,168	27,991	5,994	8,159	8,069	1,399	166	16,353	21,607	17,229	5,062
Ratio:	13.34%	9.39%	1.30%	0.03%	11.33%	7.91%	0.98%	0.02%	3.37%	1.62%	0.05%	5×10^{-4}%	6.40%	4.14%	0.53%	0.02%
Tune																
0:	213,814	442,553	2,850,253	28,801,256	251,140	519,132	3,302,603	33,316,820	282,267	573,961	3,486,235	34,872,484	239,902	495,788	3,084,512	30,971,492
1:	26,251	37,577	30,527	6,544	26,564	36,276	29,845	7,660	8,338	7,249	1,025	116	18,251	20,518	13,324	6,868
Ratio:	12.27%	8.49%	1.07%	0.02%	10.58%	6.99%	0.90%	0.02%	2.95%	1.26%	0.03%	3.3×10^{-4}%	7.61%	4.14%	0.43%	0.02%
Test																
0:	259,790	537,320	3,449,582	34,863,892	267,022	544,736	3,333,017	33,356,129	224,450	463,547	2,951,403	29,772,659	227,719	472,972	2,966,048	29,806,386
1:	30,815	43,890	37,678	8,708	10,948	11,204	2,623	271	23,728	32,809	26,733	8,701	20,725	23,916	15,280	6,894
Ratio:	11.86%	8.17%	1.09%	0.02%	4.10%	2.06%	0.08%	8.1×10^{-4}%	10.57%	7.08%	0.91%	0.03%	9.10%	5.06%	0.51%	0.02%

	Window5				Window6				Window7				Window8			
	180s	90s	15s	1.5s	180s	90s	15s	1.5s	180s	90s	15s	1.5s	180s	90s	15s	1.5s
Train																
No. Stocks:	93	93	59	34	90	88	46	21	88	84	52	19	92	92	84	45
0:	240,882	499,205	3,135,355	31,483,995	243,635	503,211	3,138,680	31,455,558	241,451	499,866	3,126,088	31,475,602	257,666	536,657	3,400,811	34,308,989
1:	21,527	25,613	13,553	5,085	18,508	21,075	7,036	1,602	20,958	24,952	22,820	13,478	28,417	35,509	32,185	20,971
Ratio:	8.94%	5.13%	0.43%	0.02%	7.60%	4.19%	0.22%	5×10^{-3}%	8.68%	4.99%	0.73%	0.04%	11.03%	6.62%	0.95%	0.06%
Tune																
0:	215,448	449,302	2,836,701	28,500,849	258,590	532,319	3,291,900	32,986,388	256,010	528,623	3,269,617	32,896,930	237,957	494,377	3,100,068	31,176,774
1:	22,090	25,774	13,755	3,711	16,321	17,503	7,032	2,932	18,236	19,869	21,335	12,590	21,925	25,387	18,516	9,066
Ratio:	10.25%	5.74%	0.48%	0.01%	6.31%	3.29%	0.21%	8.89×10^{-3}%	7.12%	3.76%	0.65%	0.04%	9.21%	5.14%	0.60%	0.03%
Test																
0:	244,624	505,052	3,145,115	31,504,509	243,618	502,663	3,125,179	31,442,651	218,135	451,570	2,805,565	28,177,196	207,544	432,827	2,785,092	28,186,570
1:	17,918	20,032	5,389	531	18,525	21,623	20,537	14,509	16,743	18,186	12,971	8,164	27,467	37,195	35,040	14,750
Ratio:	7.32%	3.97%	0.17%	1.6×10^{-3}%	7.60%	4.30%	0.66%	0.05%	7.68%	4.03%	0.46%	0.03%	13.23%	8.59%	1.26%	0.05%

Table 3.1: Sample descriptive statistics of mini flash crashes without reversal

	Window1				Window2				Window3				Window4			
	180s	90s	15s	1.5s	180s	90s	15s	1.5s	180s	90s	15s	1.5s	180s	90s	15s	1.5s
Train																
No. Stocks:	92	89	44	18	81	75	30	11	81	64	25	7	88	81	39	20
0:	223,189	464,536	3,003,712	30,318,614	215,626	446,586	2,857,338	28,772,622	244,937	499,122	3,031,453	30,323,885	255,640	523,291	3,248,093	32,603,512
1:	29,511	40,864	28,688	5,386	24,173	33,012	20,250	3,258	7,763	6,278	947	115	16,079	20,147	12,535	2,768
Ratio:	13.22%	8.80%	0.96%	0.02%	11.21%	7.39%	0.71%	0.01%	3.17%	1.26%	0.03%	3.79×10^{-4}	6.29%	3.85%	0.39%	8.49×10^{-3}%
Tune																
0:	214,074	444,661	2,858,674	28,804,205	251,541	521,712	3,310,816	33,320,420	282,812	575,715	3,486,595	34,872,520	241,053	499,114	3,088,258	30,974,456
1:	25,991	35,469	22,106	3,595	26,163	33,696	21,632	4,060	7,793	5,495	665	80	17,100	17,192	9,578	3,904
Ratio:	12.14%	7.98%	0.77%	0.01%	10.40%	6.46%	0.65%	0.01%	2.76%	0.95%	0.02%	2.29×10^{-4}	7.09%	3.44%	0.31%	0.01%
Test																
0:	260,140	539,818	3,459,361	34,867,804	267,333	546,502	3,333,796	33,356,199	224,712	465,722	2,958,328	29,776,581	228,885	476,292	2,970,256	29,809,151
1:	30,465	41,392	27,899	4,796	10,637	9,438	1,844	201	23,466	30,634	19,808	4,779	19,559	20,596	11,072	4,129
Ratio:	11.71%	7.67%	0.81%	0.014%	3.98%	1.73%	0.06%	6.03×10^{-4}%	10.44%	6.58%	0.67%	0.02%	8.55%	4.32%	0.37%	0.014%

	Window5				Window6				Window7				Window8			
	180s	90s	15s	1.5s	180s	90s	15s	1.5s	180s	90s	15s	1.5s	180s	90s	15s	1.5s
Train																
No. Stocks:	93	93	52	26	90	84	42	17	86	79	48	13	92	92	79	39
0:	241,647	502,257	3,139,262	31,486,054	244,155	505,747	3,141,160	31,456,084	242,546	503,207	3,131,021	31,480,601	258,217	539,712	3,406,986	34,316,774
1:	20,762	22,561	9,646	3,026	17,988	18,539	4,556	1,076	19,863	21,611	17,887	8,479	27,866	32,454	26,010	13,186
Ratio:	8.59%	4.49%	0.31%	9.61×10^{-3}	7.37%	3.67%	0.15%	3.42×10^{-3}%	8.19%	4.29%	0.57%	0.03%	10.79%	6.01%	0.76%	0.04%
Tune																
0:	215,964	451,659	2,840,512	28,501,890	259,403	535,035	3,293,622	32,987,443	257,264	531,410	3,274,296	32,901,994	238,458	497,202	3,104,191	31,180,534
1:	21,574	23,417	9,944	2,670	15,508	14,787	5,310	1,877	16,982	17,082	16,656	7,526	21,424	22,562	14,393	5,306
Ratio:	9.99%	5.18%	0.35%	9.37×10^{-3}%	5.98%	2.76%	0.16%	5.69×10^{-3}	6.60%	3.21%	0.51%	0.02%	8.98%	4.54%	0.46%	0.02%
Test																
0:	245,235	507,645	3,147,302	31,504,664	244,508	505,569	3,129,221	31,447,848	219,248	454,343	2,808,154	28,180,138	207,676	434,685	2,793,725	28,193,173
1:	17,307	17,439	3,202	376	17,635	18,717	16,495	9,312	15,630	15,413	10,382	5,222	27,335	35,337	26,407	8,147
Ratio:	7.06%	3.44%	0.10%	1.2×10^{-3}	7.21%	3.70%	0.53%	0.03%	7.13%	3.39%	0.37%	0.02%	13.16%	8.13%	0.95%	0.03%

Table 3.2: Sample descriptive statistics of mini flash crashes with reversal

Panel A

	Window 1						Window 2					
	TM	OR	UD	OV	EN	SM	TM	OR	UD	OV	EN	SM
LOG	0.739	0.724	0.737	0.739	0.672	0.728	0.742	0.729	0.740	0.742	0.675	0.733
SVM	0.664	0.505	0.664	0.664	0.669	0.662	0.668	0.504	0.664	0.667	0.668	0.658
RF	0.690	0.690	0.763	0.695	0.748	0.679	0.595	0.657	0.757	0.624	0.691	0.713
XGB	0.727	0.744	0.748	0.724	0.774	0.662	0.660	0.715	0.747	0.661	0.777	0.660
RD&LOG	0.736	0.716	0.734	0.736	0.670	0.668	0.740	0.727	0.739	0.740	0.675	0.663
RD&SVM	0.665	0.505	0.665	0.665	0.667	0.663	0.666	0.504	0.663	0.664	0.669	0.655
RD&RF	0.688	0.683	0.766	0.700	0.769	0.676	0.595	0.673	0.754	0.610	0.765	0.691
RD&XGB	0.723	0.748	0.749	0.734	0.772	0.628	0.684	0.718	0.753	0.684	0.778	0.610

Panel B

	Window 3						Window 4					
	TM	OR	UD	OV	EN	SM	TM	OR	UD	OV	EN	SM
LOG	0.466	0.539	0.717	0.471	0.731	0.482	0.733	0.699	0.769	0.747	0.687	0.749
SVM	0.637	0.572	0.732	0.637	0.784	0.627	0.688	0.507	0.690	0.687	0.689	0.689
RF	0.803	0.798	0.836	0.791	0.821	0.804	0.721	0.728	0.810	0.736	0.795	0.757
XGB	0.791	0.824	0.827	0.787	0.842	0.739	0.750	0.789	0.800	0.761	0.820	0.651
RD&LOG	0.467	0.534	0.752	0.472	0.743	0.628	0.725	0.684	0.760	0.728	0.674	0.672
RD&SVM	0.637	0.551	0.730	0.637	0.721	0.627	0.682	0.507	0.687	0.681	0.682	0.684
RD&RF	0.760	0.787	0.829	0.782	0.847	0.847	0.718	0.729	0.808	0.740	0.816	0.740
RD&XGB	0.770	0.794	0.838	0.811	0.850	0.796	0.766	0.796	0.801	0.748	0.820	0.619

Notes: This table presents the tuning results for mini flash crashes without reversal occurring within 1.5 seconds, spanning from window 1 to window 4 in the year 2017.

Table 3.3: Mini Flash Crash in 1.5s without reversal predication modeling tuning AUC, 2017

Panel A

	Window 5						Window 6					
	TM	OR	UD	OV	EN	SM	TM	OR	UD	OV	EN	SM
LOG	0.888	0.799	0.914	0.890	0.825	0.871	0.878	0.733	0.887	0.879	0.791	0.873
SVM	0.570	0.579	0.740	0.573	0.764	0.560	0.705	0.503	0.791	0.705	0.788	0.688
RF	0.877	0.927	0.958	0.915	0.948	0.869	0.861	0.925	0.931	0.886	0.912	0.909
XGB	0.935	0.957	0.950	0.930	0.964	0.917	0.870	0.926	0.927	0.889	0.940	0.893
RD&LOG	0.888	0.804	0.915	0.889	0.818	0.703	0.879	0.729	0.887	0.880	0.789	0.770
RD&SVM	0.569	0.585	0.732	0.571	0.768	0.564	0.706	0.502	0.791	0.706	0.788	0.690
RD&RF	0.903	0.941	0.956	0.916	0.965	0.814	0.866	0.922	0.925	0.884	0.932	0.886
RD&XGB	0.929	0.956	0.946	0.926	0.963	0.907	0.897	0.926	0.928	0.882	0.938	0.886

Panel B

	Window 7						Window 8					
	TM	OR	UD	OV	EN	SM	TM	OR	UD	OV	EN	SM
LOG	0.700	0.685	0.699	0.700	0.700	0.697	0.777	0.734	0.767	0.777	0.772	0.774
SVM	0.642	0.501	0.640	0.642	0.641	0.639	0.670	0.508	0.688	0.669	0.691	0.670
RF	0.686	0.684	0.724	0.698	0.723	0.691	0.814	0.817	0.846	0.825	0.845	0.715
XGB	0.703	0.721	0.712	0.705	0.736	0.637	0.838	0.844	0.840	0.837	0.855	0.703
RD&LOG	0.700	0.685	0.698	0.699	0.700	0.697	0.777	0.731	0.768	0.776	0.770	0.774
RD&SVM	0.636	0.501	0.638	0.637	0.635	0.633	0.670	0.511	0.688	0.669	0.690	0.669
RD&RF	0.688	0.683	0.722	0.700	0.730	0.686	0.814	0.817	0.845	0.823	0.851	0.591
RD&XGB	0.716	0.726	0.714	0.713	0.736	0.607	0.841	0.846	0.840	0.840	0.854	0.699

Notes: This table presents the tuning results for mini flash crashes without reversal occurring within 1.5 seconds, spanning from window 5 to window 8 in the year 2018.

Table 3.4: Mini Flash Crash in 1.5s without reversal predication modeling tuning AUC, 2018

Panel A

	Window 1						Window 2					
	TM	OR	UD	OV	EN	SM	TM	OR	UD	OV	EN	SM
LOG	0.739	0.724	0.737	0.739	0.744	0.728	0.742	0.728	0.739	0.741	0.744	0.733
SVM	0.665	0.505	0.666	0.664	0.665	0.662	0.668	0.505	0.66	0.667	0.669	0.659
RF	0.685	0.69	0.762	0.695	0.748	0.679	0.605	0.669	0.752	0.608	0.687	0.714
XGB	0.725	0.746	0.748	0.725	0.774	0.651	0.663	0.721	0.749	0.671	0.777	0.638
RD&LOG	0.736	0.716	0.734	0.736	0.739	0.727	0.74	0.727	0.738	0.741	0.741	0.733
RD&SVM	0.665	0.505	0.668	0.665	0.665	0.663	0.665	0.504	0.658	0.664	0.669	0.656
RD&RF	0.688	0.684	0.761	0.7	0.771	0.675	0.597	0.671	0.756	0.621	0.768	0.691
RD&XGB	0.733	0.746	0.743	0.739	0.774	0.639	0.689	0.724	0.752	0.678	0.777	0.611

Panel B

	Window 3						Window 4					
	TM	OR	UD	OV	EN	SM	TM	OR	UD	OV	EN	SM
LOG	0.518	0.484	0.689	0.527	0.733	0.548	0.729	0.698	0.756	0.744	0.75	0.749
SVM	0.634	0.614	0.659	0.613	0.745	0.647	0.685	0.507	0.684	0.686	0.688	0.688
RF	0.787	0.807	0.828	0.77	0.811	0.804	0.712	0.729	0.808	0.734	0.795	0.751
XGB	0.738	0.823	0.83	0.75	0.842	0.779	0.747	0.786	0.796	0.737	0.817	0.656
RD&LOG	0.518	0.485	0.711	0.526	0.731	0.545	0.721	0.681	0.744	0.728	0.744	0.728
RD&SVM	0.634	0.582	0.658	0.603	0.757	0.647	0.68	0.507	0.679	0.679	0.681	0.682
RD&RF	0.794	0.807	0.834	0.748	0.836	0.803	0.72	0.731	0.81	0.731	0.815	0.724
RD&XGB	0.734	0.804	0.816	0.779	0.83	0.768	0.758	0.789	0.797	0.771	0.820	0.639

This table presents the tuning results for mini flash crashes with reversal occurring within 1.5 seconds, spanning from window 1 to window 4 in the year 2017.

Table 3.5: Mini Flash Crash in 1.5s with reversal predication modeling tuning AUC, 2017

Panel A

	Window 5						Window 6					
	TM	OR	UD	OV	EN	SM	TM	OR	UD	OV	EN	SM
LOG	0.881	0.786	0.912	0.881	0.918	0.867	0.877	0.731	0.848	0.878	0.893	0.872
SVM	0.564	0.573	0.762	0.565	0.751	0.555	0.695	0.503	0.764	0.696	0.798	0.678
RF	0.897	0.931	0.963	0.914	0.945	0.869	0.863	0.924	0.927	0.885	0.913	0.905
XGB	0.928	0.951	0.952	0.928	0.961	0.912	0.877	0.926	0.922	0.874	0.939	0.902
RD&LOG	0.881	0.786	0.916	0.888	0.906	0.879	0.878	0.731	0.851	0.879	0.892	0.874
RD&SVM	0.563	0.582	0.766	0.564	0.737	0.559	0.701	0.502	0.767	0.806	0.684	0.683
RD&RF	0.902	0.934	0.96	0.906	0.963	0.774	0.879	0.925	0.919	0.886	0.933	0.886
RD&XGB	0.933	0.957	0.956	0.932	0.962	0.899	0.875	0.925	0.916	0.889	0.936	0.891

Panel B

	Window 7						Window 8					
	TM	OR	UD	OV	EN	SM	TM	OR	UD	OV	EN	SM
LOG	0.721	0.701	0.716	0.721	0.721	0.718	0.798	0.753	0.796	0.798	0.794	0.793
SVM	0.66	0.501	0.655	0.66	0.662	0.658	0.691	0.511	0.71	0.691	0.704	0.69
RF	0.695	0.695	0.742	0.705	0.742	0.712	0.808	0.824	0.858	0.822	0.856	0.741
XGB	0.718	0.737	0.728	0.722	0.757	0.646	0.843	0.855	0.848	0.839	0.867	0.733
RD&LOG	0.721	0.7	0.716	0.72	0.721	0.718	0.797	0.754	0.798	0.797	0.794	0.792
RD&SVM	0.655	0.501	0.652	0.654	0.655	0.652	0.692	0.513	0.711	0.692	0.707	0.691
RD&RF	0.697	0.699	0.744	0.708	0.751	0.704	0.809	0.826	0.859	0.824	0.865	0.627
RD&XGB	0.723	0.738	0.73	0.724	0.757	0.614	0.849	0.855	0.848	0.85	0.866	0.716

This table presents the tuning results for mini flash crashes with reversal occurring within 1.5 seconds, spanning from window 5 to window 8 in the year 2018.

Table 3.6: Mini Flash Crash in 1.5s with reversal predication modeling tuning AUC, 2018

Panel A: Mini Flash Crashes without reversal

	Window1	Window2	Window3	Window4	Window5	Window6	Window7	Window8
180s:	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	UD+RD&XGB	UD+RD&XGB
90s:	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&RF	UD+RD&XGB
15s:	EN+XGB	EN+XGB	EN+XGB	EN+XGB	EN+RD&RF	EN+RD&XGB	EN+XGB	EN+XGB
1.5s:	EN+XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&RF	EN+XGB	EN+XGB	EN+XGB

Panel B: Mini Flash Crashes with reversal

	Window1	Window2	Window3	Window4	Window5	Window6	Window7	Window8
180s:	EN+RD&XGB	EN+RD&XGB	EN+XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	UD+RD&XGB	UD+RD&XGB
90s:	EN+XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&XGB	EN+RD&RF	UD+RD&XGB
15s:	EN+XGB	EN+XGB	EN+RD&RF	EN+RD&RF	EN+EGB	EN+RD&XGB	EN+XGB	EN+XGB
1.5s:	EN+RD&XGB	EN+RD&XGB	EN+XGB	EN+RD&XGB	EN+RD&RF	EN+XGB	EN+XGB	EN+XGB

Notes: This table presents the model with the best performance achieved by tuning its parameters and selecting an imbalanced data processing strategy. These choices were made using the training dataset, and the model's performance was subsequently assessed on the tuning data. "RD" denotes the reduced dimension model, in which only crucial features are selected.

Table 3.7: Best model combined imbalanced data strategy of Tuning

Panel A: Mini Flash Crashes without reversal

	Window1	Window2	Window3	Window4	Window5	Window6	Window7	Window8
180s:	0.895	0.895	0.864	0.829	0.850	0.877	0.872	0.812
90s:	0.896	0.897	0.856	0.853	0.849	0.885	0.878	0.829
15s:	0.856	0.877	0.812	0.934	0.878	0.918	0.919	0.925
1.5s:	0.774	0.778	0.850	0.820	0.965	0.940	0.736	0.855

Panel B: Mini Flash Crashes with reversal

	Window1	Window2	Window3	Window4	Window5	Window6	Window7	Window8
180s:	0.894	0.893	0.860	0.832	0.849	0.876	0.868	0.812
90s:	0.960	0.898	0.835	0.864	0.852	0.890	0.953	0.847
15s:	0.874	0.884	0.875	0.913	0.904	0.936	0.918	0.937
1.5s:	0.774	0.777	0.842	0.820	0.963	0.939	0.757	0.867

Notes: This table displays the Area Under the ROC Curve (AUC) values generated by the model chosen after optimizing its parameters and selecting an imbalanced data processing strategy. These selections were made using a tuning dataset, and the model's performance was subsequently evaluated on the test data.

Table 3.8: Best AUC of Tuning

Panel A: Mini Flash Crashes without reversal

	Window1	Window2	Window3	Window4	Window5	Window6	Window7	Window8
180s:	0.967	0.899	0.904	0.951	0.937	0.958	0.940	0.909
90s:	0.967	0.898	0.905	0.951	0.937	0.959	0.941	0.909
15s:	0.969	0.880	0.933	0.976	0.946	0.988	0.984	0.932
1.5s:	0.761	0.843	0.803	0.821	0.885	0.984	0.741	0.725

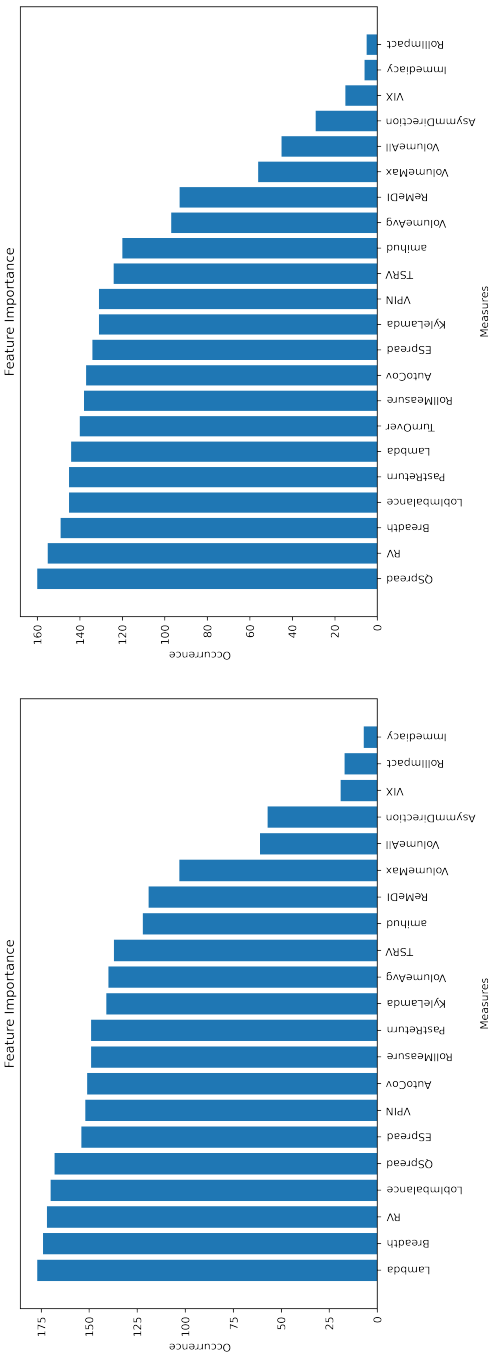
Panel B: Mini Flash Crashes with reversal

	Window1	Window2	Window3	Window4	Window5	Window6	Window7	Window8
180s:	0.964	0.904	0.915	0.943	0.926	0.956	0.938	0.900
90s:	0.967	0.892	0.903	0.954	0.942	0.958	0.944	0.908
15s:	0.967	0.895	0.945	0.979	0.940	0.990	0.987	0.932
1.5s:	0.769	0.868	0.817	0.816	0.877	0.984	0.740	0.714

Notes: This table displays the Area Under the ROC Curve (AUC) values generated by the model chosen after optimizing its parameters and selecting an imbalanced data processing strategy. These selections were made using a tuning dataset, and the model's performance was subsequently evaluated on the test data.

Table 3.9: AUC of Test data set

3.8 Appendix 4. Feature Importance



(a) 1.5s

(b) 15s

(c) 90s

(d) 180s

Figure 3.5: Features importance of mini flash crashes without reversal in Window 1

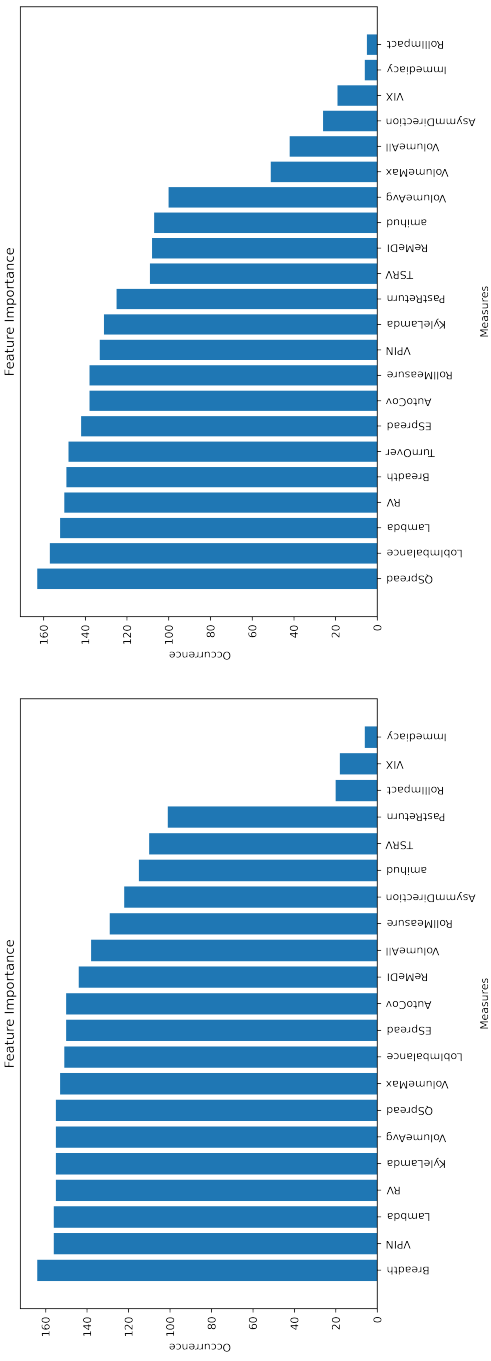


Figure 3.6: Features importance of mini flash crashes without reversal in Window 2

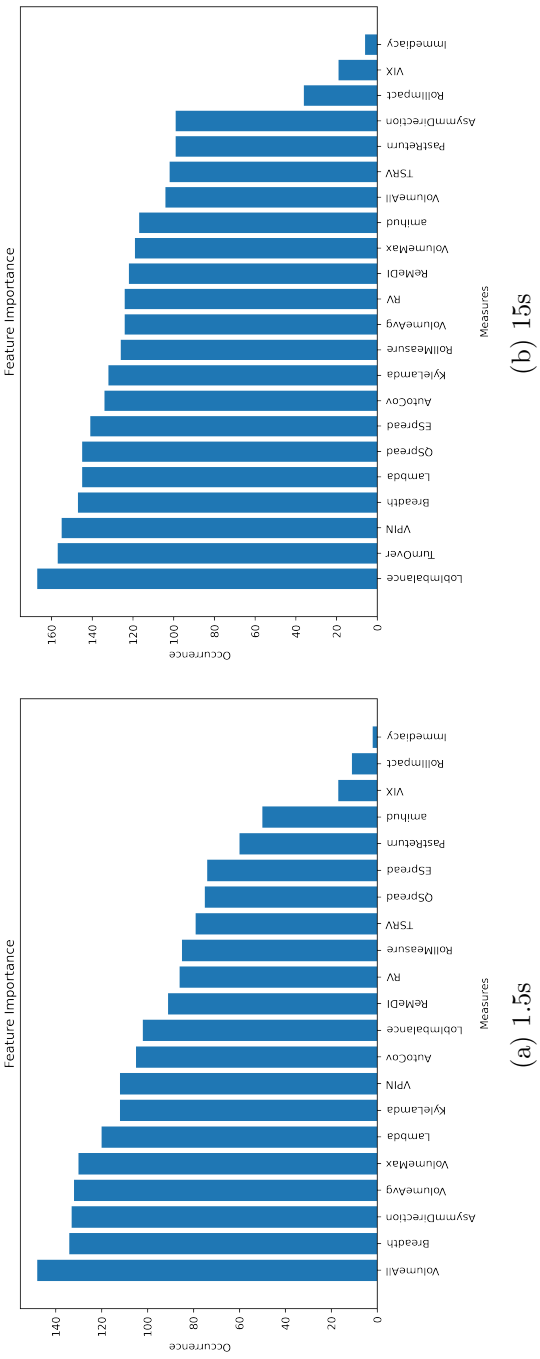
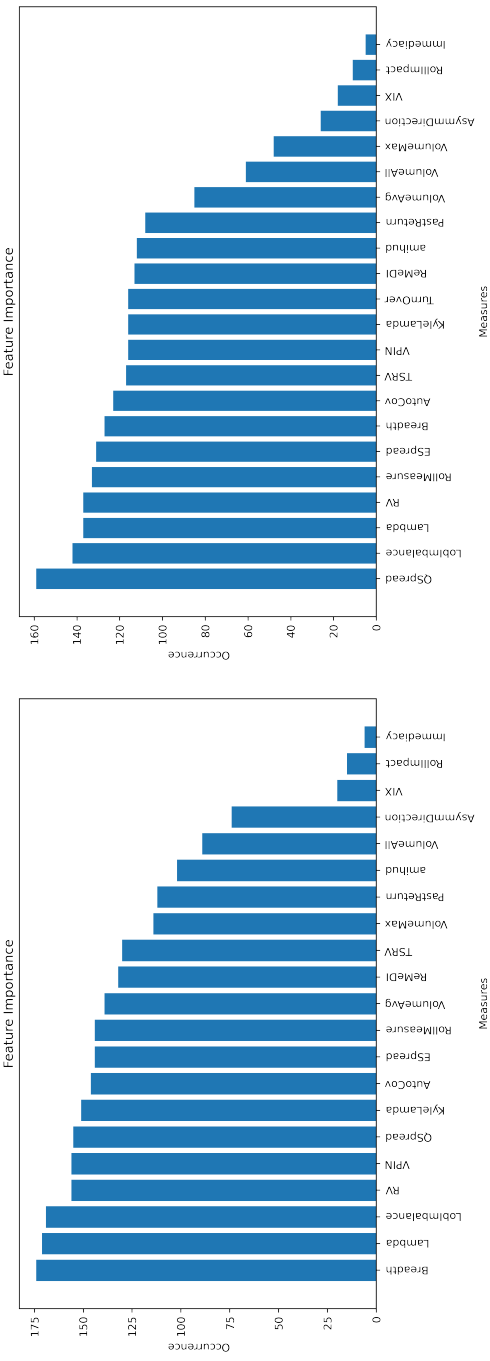
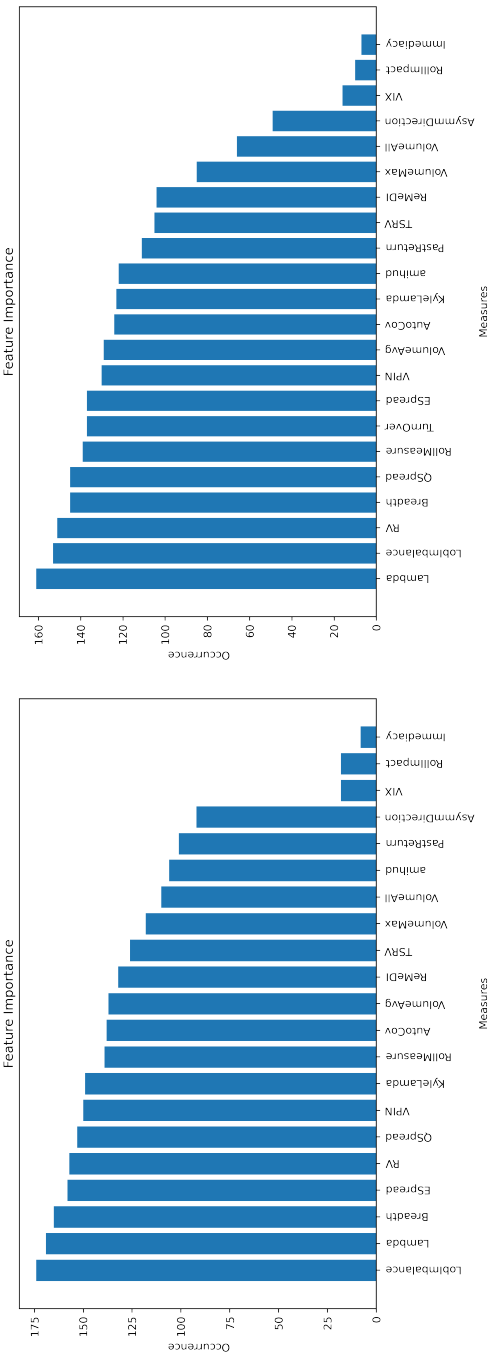


Figure 3.7: Features importance of mini flash crashes without reversal in Window 3



(a) 1.5s (b) 15s (c) 90s (d) 180s

Figure 3.8: Features importance of mini flash crashes without reversal in Window 4



(a) 1.5s (b) 15s (c) 90s (d) 180s

Figure 3.9: Features importance of mini flash crashes without reversal in Window 5

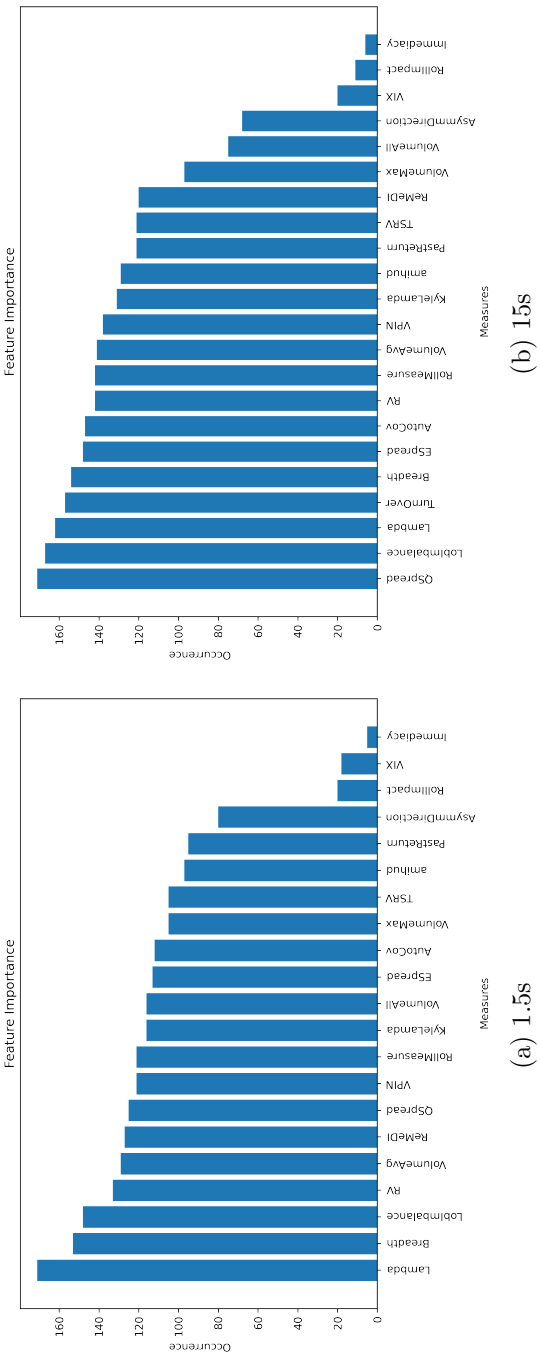
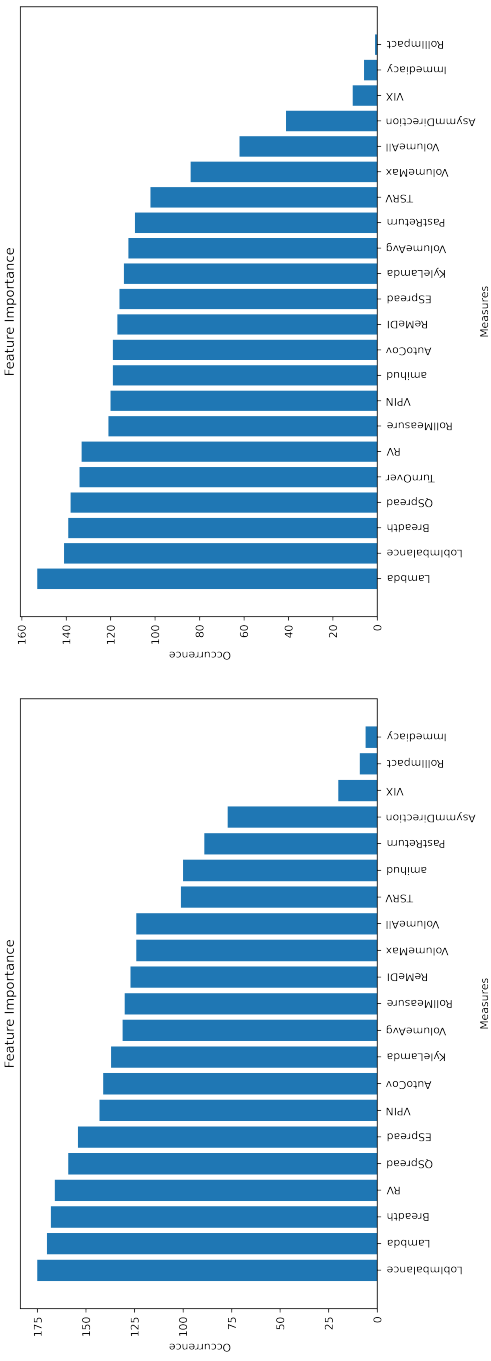
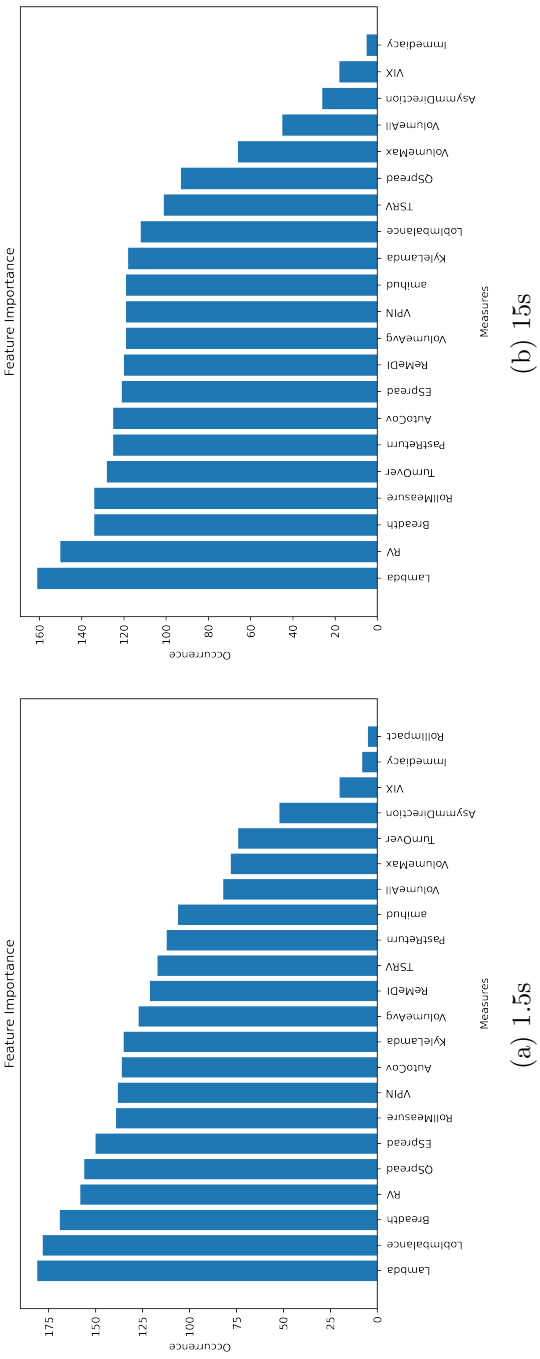


Figure 3.10: Features importance of mini flash crashes without reversal in Window 6



(a) 1.5s (b) 15s (c) 90s (d) 180s

Figure 3.11: Features importance of mini flash crashes without reversal in Window 7



(a) 1.5s (b) 15s (c) 90s (d) 180s

Figure 3.12: Features importance of mini flash crashes without reversal in Window 8

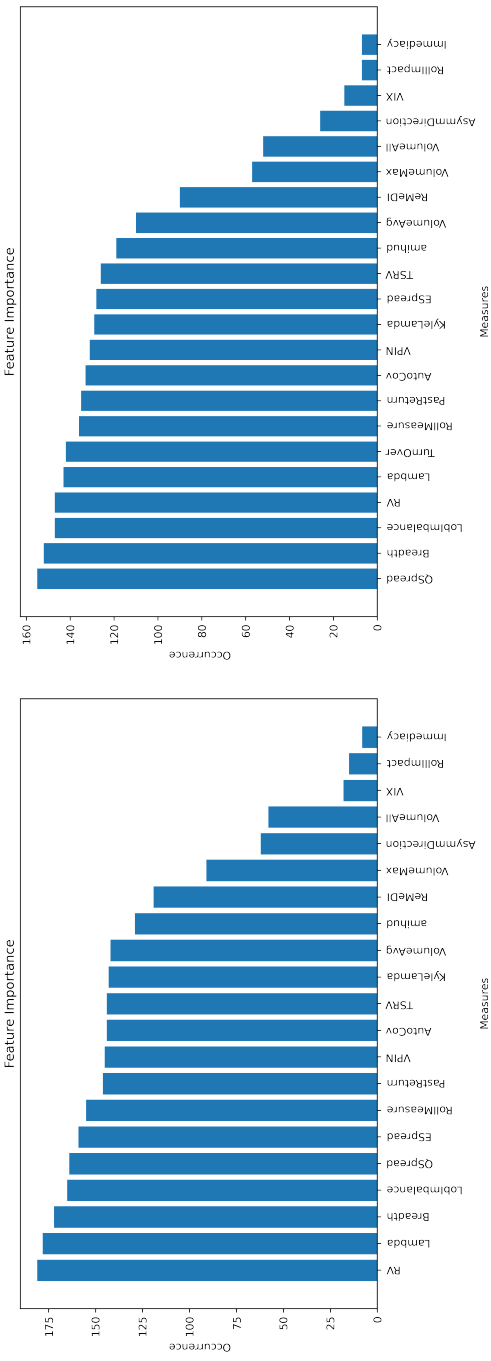


Figure 3.13: Features importance of mini flash crashes with reversal in Window 1

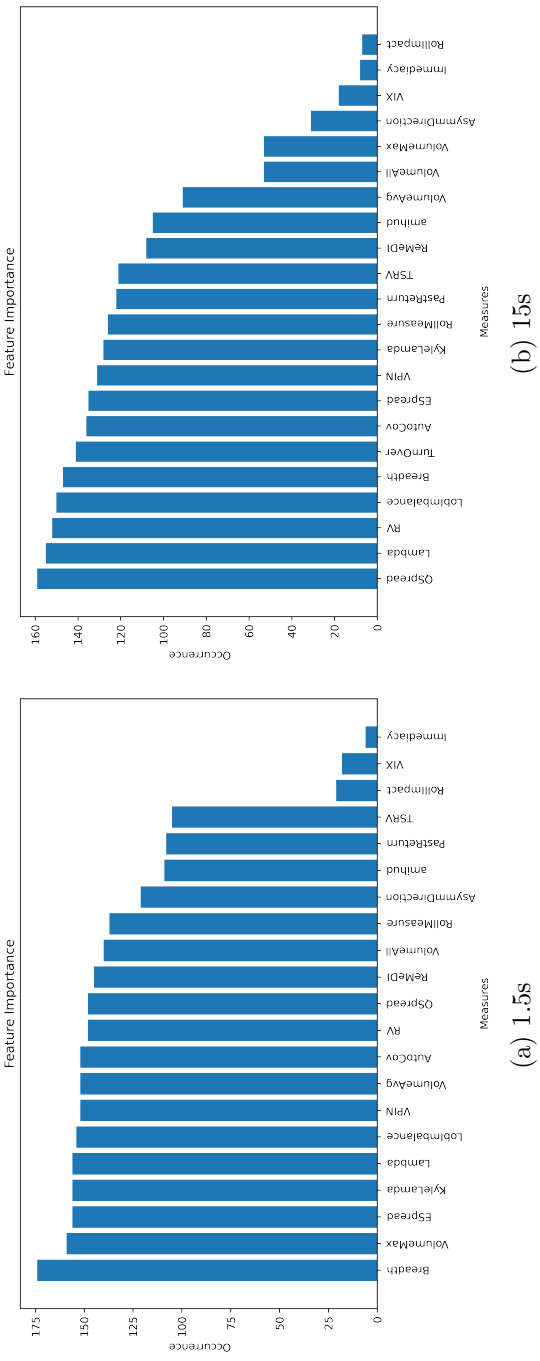


Figure 3.14: Features importance of mini flash crashes with reversal in Window 2

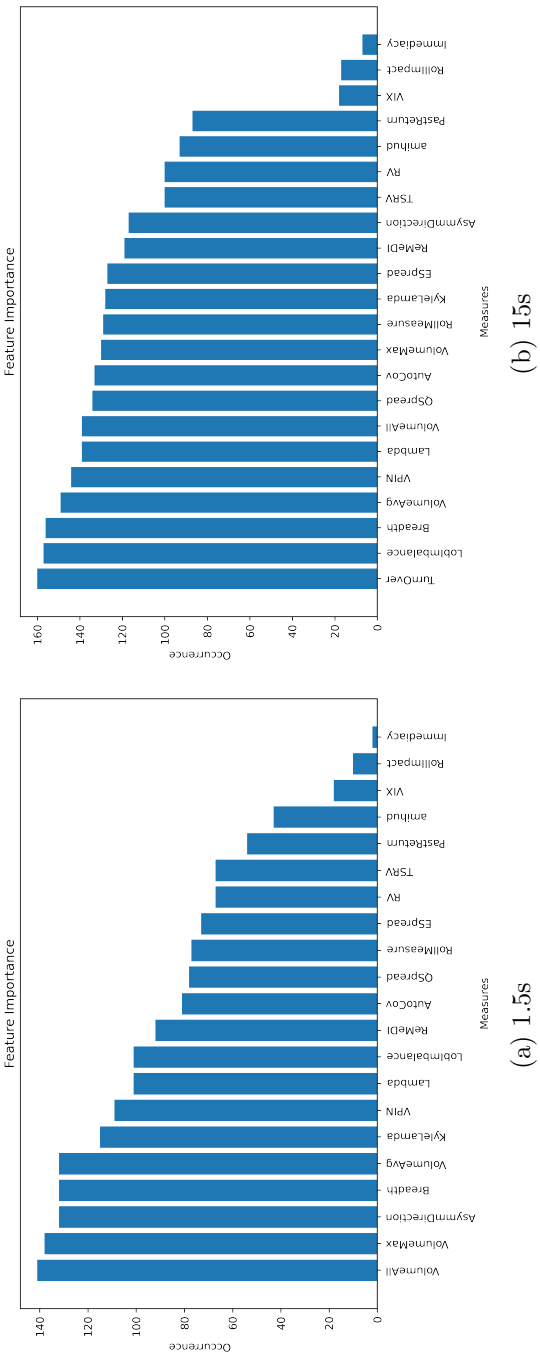
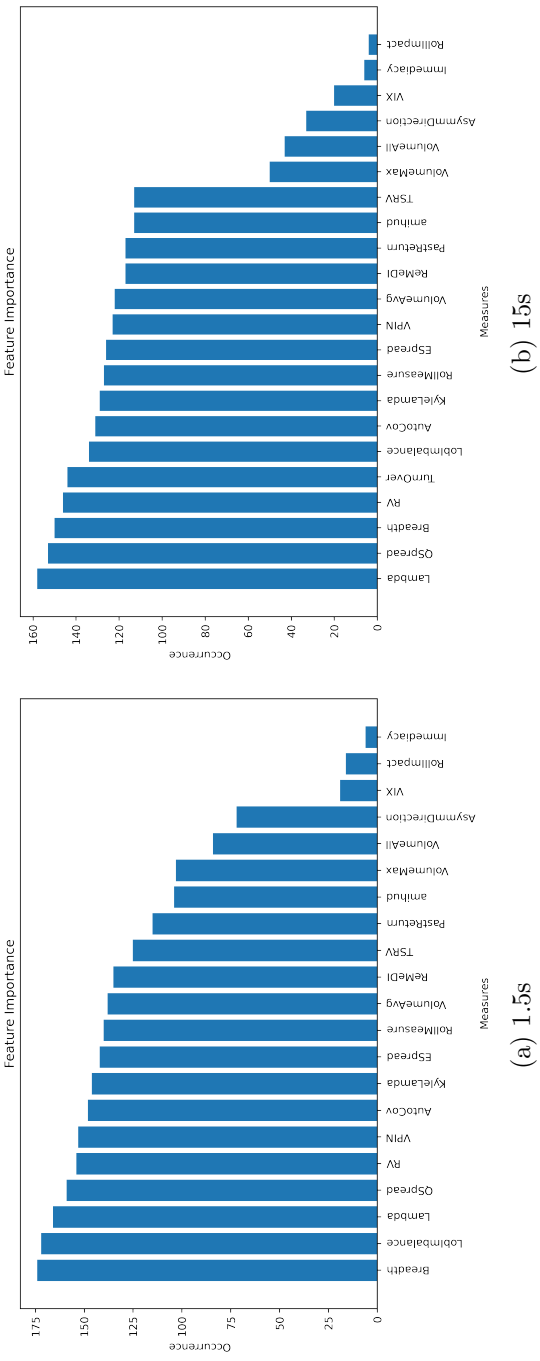
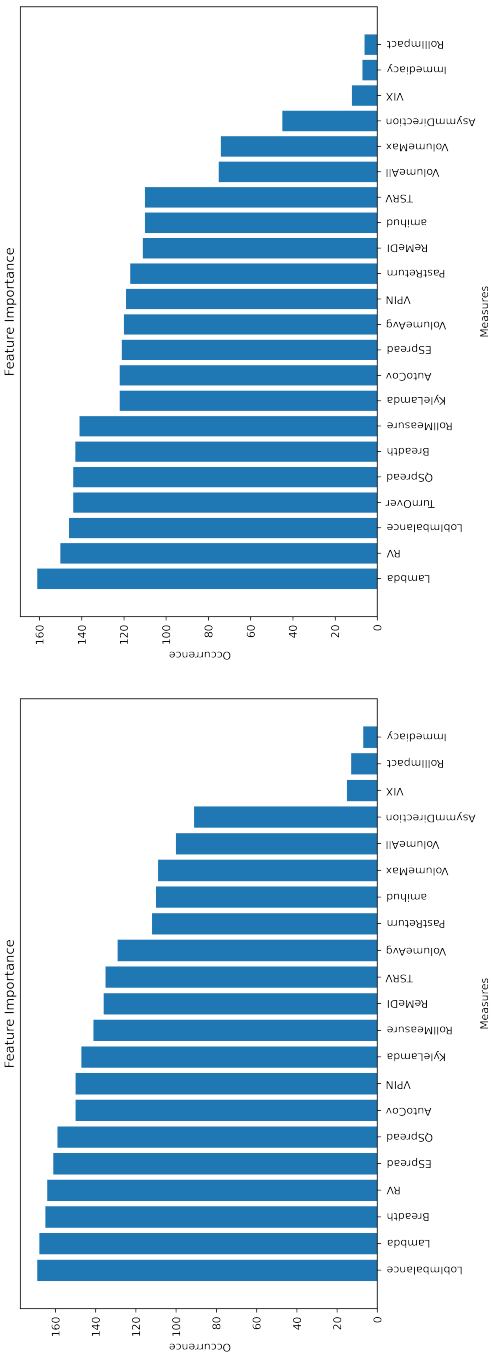


Figure 3.15: Features importance of mini flash crashes with reversal in Window 3



(a) 1.5s (b) 15s (c) 90s (d) 180s

Figure 3.16: Features importance of mini flash crashes with reversal in Window 4



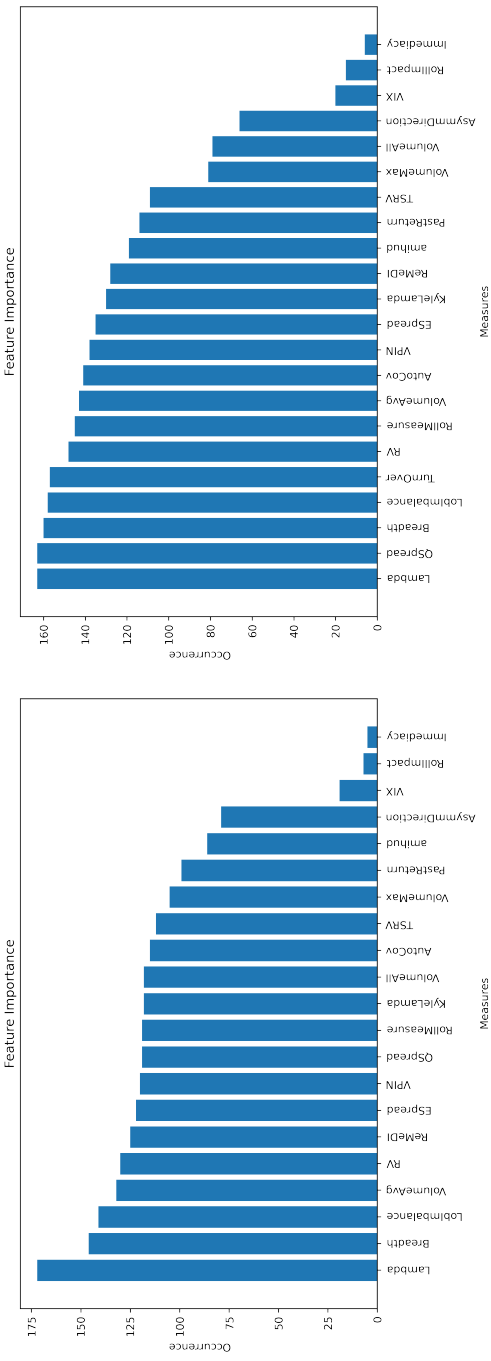
(a) 1.5s

(b) 15s

(c) 90s

(d) 180s

Figure 3.17: Features importance of mini flash crashes with reversal in Window 5



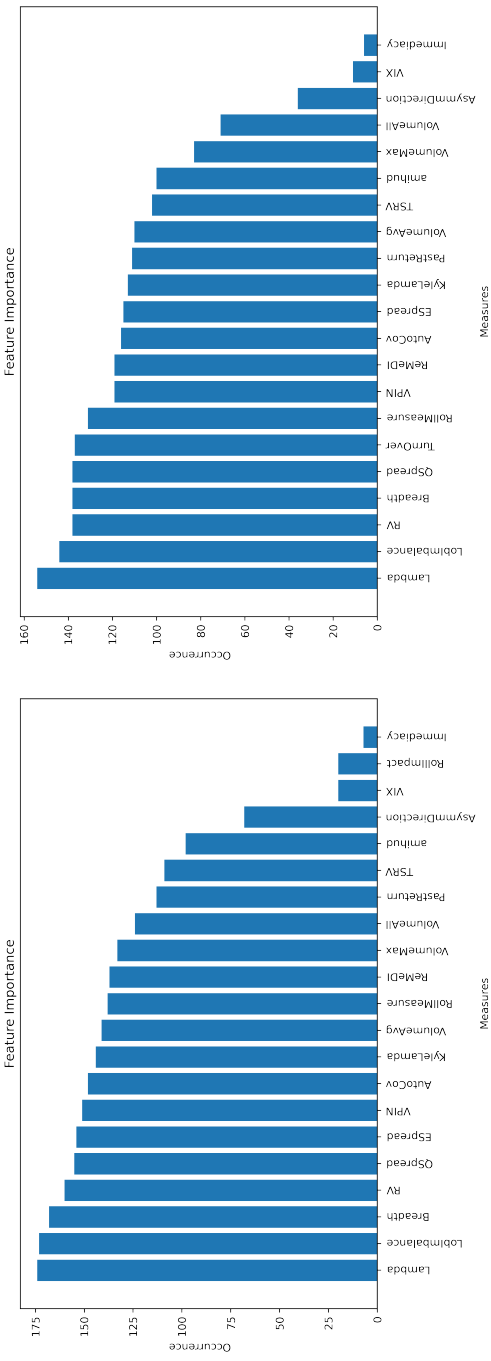
(a) 1.5s

(b) 15s

(c) 90s

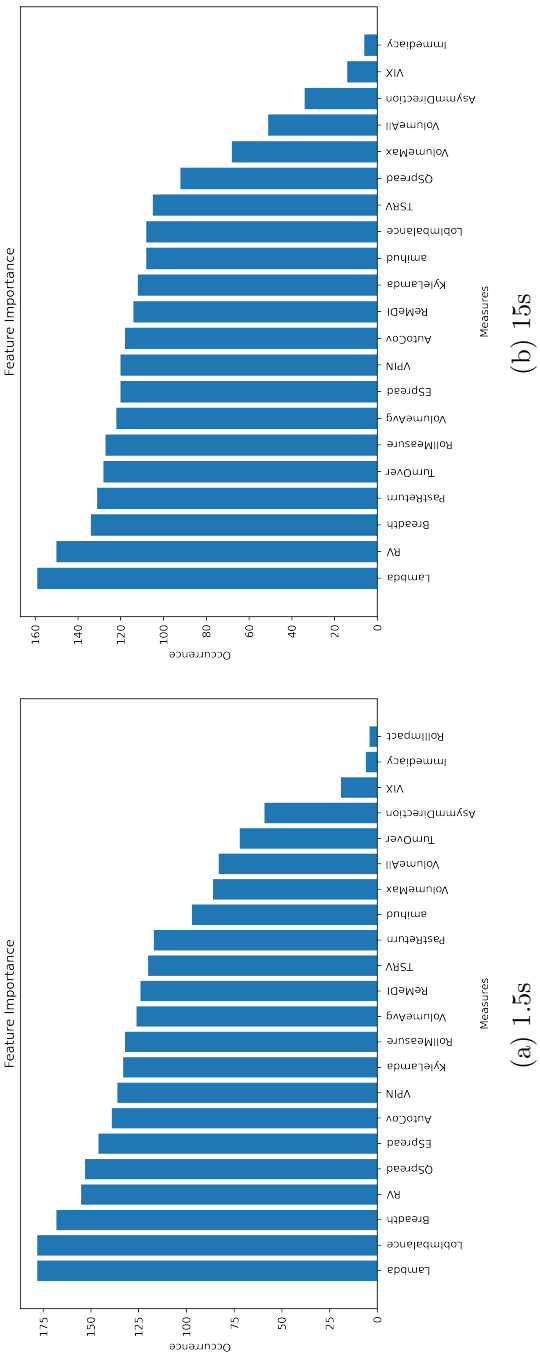
(d) 180s

Figure 3.18: Features importance of mini flash crashes with reversal in Window 6



(a) 1.5s (b) 15s (c) 90s (d) 180s

Figure 3.19: Features importance of mini flash crashes with reversal in Window 7



(a) 1.5s (b) 15s (c) 90s (d) 180s

Figure 3.20: Features importance of mini flash crashes with reversal in Window 8

Chapter 4

Does speed bump resolve the problems of mini flash crashes? Evidence from NYSE American

4.1 Introduction

The impact of high-frequency trading (HFT) on flash crashes has been extensively discussed in the literature, notably the historic "flash crash" that took place on May 6, 2010, marking the largest intraday decline in the Dow Jones Industrial Average's history. Despite being investigated by both the Securities and Exchange Commission (SEC) and the Commodity Futures Trading Commission (CFTC), the precise cause of this event remains elusive. However, many market participants suspect that the cancellation of existing buy orders by high-speed trading algorithms when detecting market imbalances is a primary contributor (Nolte (2010)). These cancellations set off a chain reaction of rapid sell-offs as trading algorithms interact, resulting in a steep price plummet. Consequently, this incident has garnered significant attention, sparking concerns among regulators, academics, and investors regarding the practice of high-frequency trading, which is based on computer-based algorithms.

'Liquidity is a coward; it is never there when it is needed.' (Christopher (2008)). Bouchaud (2021) believes there is a destabilizing feedback loop in HFT market-making activity:

$$\text{Volatility} \Longrightarrow \text{Higher spreads and lower liquidity} \Longrightarrow \text{More volatility}$$

Fosset et al. (2020) suggests that excessive reactions by market makers, whether human or machine-based, to unexpected events can set off a detrimental decline in market liquidity. In such a scenario, High-Frequency Traders (HFTs), serving as ultra-fast market intermediaries, may find themselves vulnerable to significant short or long positions, necessitating swift maneuvers to exit these high-risk positions. While HFT is not the primary instigator, there is a notable potential for this algorithmic trading approach to exacerbate market conditions, potentially culminating in a flash crash. The findings from a study by Bellia, Mario; Christensen, Kim; Kolokolov, Aleksey; Pelizzon, Loriana; Reno, Roberto (2020) support this notion, revealing that HFTs often consume liquidity precisely when it is most crucial, even when offered incentives by exchanges to provide immediate liquidity.

Figure 4.1

In the aftermath of the May 6th, 2010 flash crash, regulators responded with heightened scrutiny and proactive measures to avert any future reoccurrence. The Securities and Exchange Commission (SEC) introduced a proposal mandating exchanges to establish an audit system, providing regulators with access to data regarding received and executed orders. As a follow-up, the SEC ratified new regulations designed to mitigate stock volatility by instituting temporary trading halts in the event of substantial price fluctuations (Nolte (2010)).

Academic literature also underscores the importance of cautious regulation regarding high-frequency trading (HFT) Keller (2012). It suggests that regulators should strive to restore market confidence without undermining the efficiencies that HFT brings. This can be achieved through enhanced transparency and reporting requirements. Additionally, it recommends that regulators mandate internal risk manage-

ment practices within HFT firms themselves Keller (2012).

In summary, the flash crash compelled regulators to respond proactively by suggesting audit systems, putting trading halts into effect, and exploring strategies to both reinstate market trust and preserve the advantages of HFT. Moreover, academia underscores the significance of transparency, reporting prerequisites, and the enforcement of internal risk management practices within HFT firms.

Beyond the United States, the Investment Industry Regulatory Organization of Canada (IIROC) also expressed its concerns on May 6th, 2010, a day when Toronto's primary index experienced a relatively milder decline of approximately 3.8 percent, as reported by Reuters Reuters (2010). IIROC highlights that Canadian markets responded swiftly to the decline in the United States, with no specific event such as erroneous orders driving this downturn. Furthermore, IIROC's review recommends that regulators should assess the trigger levels of their existing market-wide circuit breaker mechanisms Langton (2010).

Given that events akin to the flash crash of May 6th, 2010, are exceedingly rare and nearly impossible to predict, researchers have turned their attention to gleaning insights from miniaturized versions of such crashes Nanex (2010). Supporting this perspective, Golub et al. (2012) provide evidence indicating that mini flash crashes have adverse effects on market liquidity. In addition to their findings, my research suggests that market liquidity indicators, including trade volume and limit order book information, serve as potent signals for predicting mini flash crashes. This is supported by the empirical investigation conducted by Kirilenko et al. (2017b), who delved into the trading patterns of high-frequency traders during the Flash Crash.

Various studies have examined the flash crash through the lens of the probability of informed trading Easley et al. (2012); Andersen and Bondarenko (2014b). Contrarily, Jonathan et al. (2018) ascertain that HFT is not the primary cause of flash crashes, based on empirical analysis of HFT data from NASDAQ Brogaard et al. (2018). In contrast, Leal et al. (2014), utilizing an agent-based model, illustrate that HFTs can indeed have a substantial positive impact on flash crashes due to their specific trading

strategies Leal et al. (2014).

The flash crash has significantly heightened systemic market risk Min and Borch (2022). In moments of panic, investors may incur losses if they hastily sell their positions. As highlighted in the International Organization of Securities Commissions (IOSCO) report from July 2011, more than 20,000 trades in 300 securities occurred at prices deviating by as much as 60% from their recent values, ranging from mere pennies to a staggering $100,000 (Technical Committee Of The International Oganization Of Securities Commissions (2011)). Addressing the issue of mini flash crashes has become a prominent subject in financial research. This paper focuses on evaluating the effectiveness of the speed bump policy, a market mechanism specifically designed to target High-Frequency Trading (HFT) and mitigate mini-flash crashes.

Numerous studies have adopted a widely recognized rule-of-thumb definition for assessing flash crashes, originally proposed by Nanex, a trading technology company and data service provider (Nanex). This method often provides access to extensive datasets, which facilitates precise statistical analysis. Furthermore, recent advancements in data analysis techniques have ushered in a new era. Beyond the realm of finance, the application of machine learning techniques has demonstrated remarkable effectiveness in analyzing vast datasets across diverse domains, shedding light on the potential for predicting real-world mini flash crashes using market data.

Figure 4.2 depicts the mini flash crash of stock PLX on March 6th, 2017, according to Nanex's definition.

Figure 4.2

Mini flash crashes of a single stock can occur multiple times within a single trading day, as illustrated by the example of April 2nd, 2017, with stock PLX, as shown in Figure 4.3.

Figure 4.3

Chapter 3 introduces both the evidence and methodology for utilizing machine learning to demonstrate the real-time predictability of mini flash crashes, achieved by analyzing market microstructure features extracted from the limit order book. Build-

ing upon this methodology, it becomes straightforward to compute the probability of potential mini flash crashes in real-time, a task that can be effectively detected using machine learning techniques.

Michael Lewis's book, "Flash Boys" Lewis (2014), shone a spotlight on the issue of unfairness within the financial market, particularly by showcasing the story of Katsuyama and IEX. Katsuyama vividly portrayed a landscape where high-frequency traders could exploit their speed advantage to front-run orders before they even reached the exchanges. This was made possible due to stocks being traded across multiple exchanges, each with its own order arrival times. If the claims presented by Lewis and Katsuyama held true, speed had transformed into a potent advantage, endowing fast traders with inherent informational superiority. The subsequent ascent of IEX as a competitive exchange further underscored the urgency for investors and regulators to critically scrutinize these assertions concerning market fairness.

In conversations about the unique nature of High-Frequency Trading (HFT), various voices from investors and academics have emerged, and one of the most prevalent topics in recent years has been the concept of the "Speed Bump." This practice, often highlighted in Michael Lewis's popular book "Flash Boys" Lewis (2014), illustrates the phenomenon of frontrunning order anticipation. Lewis narrates a story that underscores how high-frequency traders leverage speed as a competitive advantage to unfairly profit from slower traders.

The Investor Exchange (IEX), founded by Katsuyama, the central figure in "Flash Boys" made history as the first stock exchange to implement a speed bump in June 2016. The U.S. Securities and Exchange Commission (SEC) officially sanctioned this pioneering approach. Katsuyama, a former trader at the Royal Bank of Canada, shared his personal journey of sending orders to other exchanges, where his orders often received limited attention and encountered low execution rates. He attributed the liquidity issues to the delayed arrival of substantial orders, stemming from the intricate layout of the fiber-optic network. This complexity created opportunities for high-frequency traders to intercept slower orders, giving rise to what Katsuyama

identified as a new form of informed trading.

In response to the problem of unfair competition in the stock market, Katsuyama established IEX and secured regulatory approval from the SEC in October 2016. IEX implemented a speed bump ahead of its matching engine, accomplished through a 38-mile optical fiber coil. This deliberate delay, averaging around 350 microseconds for all quotes, aimed to offer slower traders a brief opportunity to route orders to other exchanges before faster traders could potentially identify and capitalize on them (Aoyagi (2022)).

A recent study has reported a significant surge in short-selling volume on the Chicago Board Options Exchange (CBOE) subsequent to the introduction of IEX's speed bump Chakrabarty et al. (2020). These findings pose an intriguing question: if a speed bump effectively curtails the capacity of high-frequency traders to exploit speed for adverse market consequences, could it also decrease the likelihood of mini flash crashes?

In summary, the implementation of a speed bump by IEX has sparked substantial interest and debate concerning market fairness. The repercussions on market dynamics and the potential influence on mini flash crashes merit deeper examination and thorough analysis.

In a study conducted by chapter 3, the NYSE American, the second exchange to implement a speed bump following IEX, was examined. The design of NYSE's speed bump closely mirrors that of their competitor IEX, with the exception of a 350-microsecond delay. Utilizing high-frequency data with nanosecond timestamps, chapter 3 assessed the impact of the speed bump on market qualities and evaluated traded stocks on both NYSE American (with a speed bump) and Nasdaq (without a speed bump). Through their analysis of price discovery and market liquidity, Liu (2023) discovered that the speed bump policy is a double-edged sword, as it enhances liquidity but diminishes the informativeness of prices.

In this research, I adopt the methodological framework established by Aït-Sahalia et al. (2022), Easley et al. (2021), and the chapter 3. I leverage machine learning

models that utilize microstructure measures, encompassing both short- and long-term data extracted from the limit order book, to estimate the probability of mini flash crashes. The rationale behind choosing machine learning as the methodology stems from its capability to identify intricate non-linear patterns effectively.

Mini flash crashes, being relatively infrequent events driven by a variety of factors that can occur randomly at any moment, pose a significant challenge. When considering policies aimed at preventing such events, the objective is to target patterns within the market information found in the limit order book. Machine learning's ability to detect these events through their patterns transforms them from random occurrences into identifiable events characterized by market information.

To gain a comprehensive understanding, I categorize mini flash crashes into four distinct time intervals, drawing from existing literature: long-term (180 seconds), medium-term (90 seconds), short-term (15 seconds), and extremely short mini flash crashes (1.5 seconds). By examining these varying time frames, I can discern diverse patterns within these infrequent events, uncovering various market insights. I will employ machine learning techniques to calculate the probabilities of detecting these distinct patterns associated with mini flash crashes across the four different time intervals.

Having successfully estimated the probability of mini flash crashes, I proceed to conduct a difference-in-difference analysis to examine how the introduction of the speed bump influences mini flash crashes using panel data linear regression. Across the four different time intervals, I have observed varying outcomes. Notably, I found that following the establishment of market equilibrium, shaped by both slow traders and fast traders, the speed bump tends to increase the probability of long-term mini flash crashes. This outcome aligns with my expectations, as Liu (2023) indicates that NYSE American's speed bump can reduce price discovery and diminish adverse selection, leading to the belief that more noise traders migrate to the exchange with the speed bump, thereby introducing greater volatility, in line with the findings of Liu (2023).

Similarly, this trend is reflected in the medium-term version, which encompasses an adaptable time frame allowing both slow and fast traders to take actions. However, my empirical results for the short and strictly short versions show a significant reduction in the probability of detectable mini flash crashes associated with the presence of the speed bump. This result provides compelling evidence that the speed bump indeed represents an ideal policy to safeguard the market against mini flash crashes stemming from exceptionally fast trading algorithms.

This work contributes to the existing literature on the topic discussed in O'Hara (2015), providing fresh insights into market data. It also serves as a valuable addition to the ongoing discussion surrounding speed bump-related research on market quality. To delve deeper into the aforementioned findings, I have structured the paper into six distinct sections. In addition to the introductory section presented above, the remainder of the paper is organized as follows: Section 4.2 provides an overview of the relevant literature. Section 4.3 outlines the dataset employed and the data pre-processing techniques employed. Section 4.4 details the methodology employed, encompassing the problem setup, machine learning models utilized, accuracy evaluation methods, and data imbalance processing strategies. Section 4.5 presents the empirical results and their implications. Section 4.6 concludes the findings and summarizes the key takeaways from the study.

4.2 Literature

This paper draws upon three distinct bodies of literature. The initial segment of literature delves into the phenomenon of Flash crashes, a subject that has garnered substantial attention from various perspectives within academic discourse. Notably, the literature has extensively explored its connection with high-frequency trading (HFT). A study conducted by Kirilenko et al. (2017a) reveals that the trading behavior of high-frequency traders remained consistent even as prices plummeted during flash crashes. Furthermore, they offer a comprehensive framework for analyzing in-

traday market dynamics both prior to and during these systemic events.

While I cannot precisely discern the trading patterns of high-frequency traders, my analysis reveals a noteworthy alteration in the behavior patterns of market participants just prior to mini flash crashes. This suggests that market patterns may serve as indicators of impending mini flash crashes, dispelling the notion that market crashes occur abruptly without any precursor, as demonstrated by the chapter 3. Building upon these findings, this paper leverages machine learning techniques to identify mini flash crashes based on detectable market patterns. Given the intricacies of market conditions and the potentially stochastic nature of mini flash crashes, the ability of machine learning to detect such events suggests that they may be managed effectively.

The examination of High-Frequency Trading (HFT) characteristics during flash crashes has been a focal point in the literature. According to Brogaard et al. (2018), HFTs play a crucial role in enhancing market liquidity by offsetting imbalances arising from non-HFT activity, particularly in individual stocks. However, when multiple stocks are affected by a flash crash, HFT liquidity demand surpasses their supply, as noted by Brogaard et al. (2018), who argue that HFT is not the root cause of flash crashes. Additionally, Leal et al. (2014) posit that a higher rate of order cancellations by HFT participants increases the likelihood of flash crashes but diminishes their duration. Notably, this paper contributes by uncovering a positive impact on the likelihood of mini flash crashes through the implementation of a speed bump.

The second strand of literature revolves around the domain of machine learning, which has seen a significant proliferation of research in recent years. In Kearns and Nevmyvaka (2013), three case studies illustrate the application of machine learning within high-frequency trading. These studies encompass the optimization of trade execution, the prediction of price movements based on order book dynamics, and the enhancement of execution strategies within dark pools through censored exploration techniques.

Predicting stock returns remains an enduringly popular subject of keen interest

within both the financial industry and the machine learning academic community. Numerous works in the literature have demonstrated the superiority of machine learning as a tool for uncovering nonlinear data patterns. In addition to the algorithms developed by the machine learning research community, financial research places a distinct emphasis on identifying informative features and refining feature engineering methods to enhance stock prediction performance. In a study conducted by Easley et al. (2021), a random forest algorithm was employed to assess the predictability of classical microstructure variables, including Roll measure, Roll impact, Kyle lambda, Amihud, VPIN, and UX (VIX). In alignment with this approach, our predictive models also incorporate these essential microstructure measures.

The third strand of literature centers around the concept of a speed bump. In the work by Aoyagi (2022), a random speed bump is examined through the establishment of a theoretical market model comprising competitive but slow uninformed market makers, risk-neutral high-frequency traders, and liquidity traders who encounter a liquidity shock in a perfectly transparent market. The study forecasts the optimal behavior of market makers and how high-frequency traders will fine-tune their trading speeds. The key conclusion drawn is that when high-frequency traders strategically choose their speed levels by factoring in the impact of their speed decisions on market dynamics, the presence of a speed bump exacerbates issues related to adverse selection and widens the bid-ask spread.

In a related study, Brolley and Cimon (2020) develop a model to forecast the impact of a speed bump on competition among exchanges in a multi-exchange setting. Their framework assumes the initial identity of two markets, each trading the same risky security with a random payoff. Following the implementation of a speed bump on one of the exchanges, Brolley and Cimon (2020) anticipate that informed traders will shift their activities to the conventional exchange without a speed bump. As far as my knowledge extends, this paper represents a pioneering empirical investigation into the effects of speed bumps on mini flash crashes.

4.3 Data description and pre-processing

In this study, I analyze the NYSE TAQ dataset, spanning from March 15th to September 1st, 2017. This comprehensive dataset encompasses 120 trading days, covering a period of 90 days leading up to the implementation of the NYSE American speed bump on July 24th, followed by 30 trading days thereafter. Given that the stock order flow and trade data represent multivariate time series, traditional cross-validation methods are unsuitable for hyperparameter tuning due to their potential to create a "future-to-past" predictive bias. To mitigate this issue and effectively train and fine-tune the model, I partition the entire dataset into four distinct segments:

Period 1: First 30 trading days, training data, 2017/03/15-2017/04/26

Period 2: Second 30 trading days, tuning data, 2017/04/27-2017/06/8

Period 3: Third 30 trading days, testing data 1, which are 30 trading days before speed bump, 2017/06/09-2017/07/23

Period 4: Last 30 trading days, testing data 2, which are 30 trading days after speed bump, 2017/07/23-2017/09/05

The timeline of my working procedure is shown in Figure 4.4.

Figure 4.4

To estimate the probability of detecting pattern-observable mini flash crashes prior to the implementation of the speed bump, I employ data from three distinct periods. Over these periods, I utilize the data from period 1 as the initial training dataset. Then, I employ the data from period 2 to fine-tune the model's parameters. Subsequently, I retrain the model using the data from period 2 to achieve the best prediction performance. Once I have identified the optimal model and parameter settings, I apply the retrained model to calculate the probability of mini flash crashes within each time interval in period 3. Period 3 encompasses precisely 30 days preceding the introduction of the speed bump on NYSE American. For estimating the probability of detecting pattern-observable mini flash crashes subsequent to the speed bump's implementation, I replicate the same procedure using data from periods 2, 3,

and 4.

Dealing with the issue of class imbalance is crucial in addressing the rare occurrence of mini flash crashes and ensuring the accuracy of my machine learning analysis. Initially, there were 217 actively traded stocks on NYSE American before March 15th. To address this imbalance, I systematically filtered out stocks that ceased trading activity between March 15th and September 5th, retaining only those stocks that exhibited mini flash crashes consistently throughout the entire training, tuning, and evaluation periods. Consequently, my dataset comprises a total of 193 stocks.

Through this meticulous approach and effective handling of imbalanced data, my objective is to offer valuable insights into how the speed bump impacts the probability of mini flash crashes, utilizing machine learning techniques to achieve meaningful results.

Following Hendershott and Moulton (2011), I use the same standard to clean the quote and transaction data.[1] I also follow the same way to one-to-one matching without replacement, I use all stocks out of NYSE American based on CRSP market capitalization and closing price. I measure the matching criteria at the 2017/03/14 which is the last date before my research period. I also randomize the order of matching by sorting NYSE American stocks alphabetically by the symbol. Then I calculate the following matching error for each NYSE American stock i and each remaining stock out of NYSE American stock j:

$$matchingerror = \frac{\left|\left(\frac{MCAP_i}{MCAP_j}\right)-1\right|+\left|\left(\frac{PRC_i}{PRC_j}\right)-1\right|}{2}$$

In the selection process, $MCAP$ represents the stock market capitalization, and

[1] As we want to escape from any other events that may hit the market, we restrict our time range to 3 months. More than that, because we use nanosecond timestamp data, 3 months has covered a big bunch of data. We expect they are enough. When we do the data filtering, we restricted the trading data in regular trading hours from 9:30 am to 4 pm. We use only trades for which TAQ's CORR filed is zero, one, or two and for COND field is either blank or equal to @, E, F, I, J, or K. Obviously, we eliminate trades with nonpositive prices or quantities. We also remove trades with prices more than(less than) 150%(50%) of the previous trade price. After that, we restrict quotes for which TAQ's MODE field is equal to 1, 2, 6, 10, 12, 21, 22, 23, 24, 25, or 26. Then we eliminate quotes with nonpositive prices or sizes or with bid prices greater than the asking price. We also exclude quotes when the quoted is greater than 25% of the quote midpoint or when the asking price is more than 150% of the bid price.

PRC signifies the stock's closing price immediately preceding period 1. I identify the stock with the lowest matching error as the most suitable match for NYSE American stock and subsequently exclude it from the pool of potential matching stocks. The average matching error observed in this context is 0.0767.

4.4 Methodology

This section outlines the machine learning methodology and empirical strategy employed in this paper.

4.4.1 Response Variables

Unlike many other financial machine learning studies, such as Easley et al. (2021), which utilize event windows or information bars for data aggregation, this paper employs a time-based approach. In practical scenarios, a portfolio manager actively monitoring their investment portfolio prior to a potential mini flash crash cannot afford to wait for the accumulation of additional events for model prediction. Therefore, to ensure the model's real-time predictive capability for mini flash crashes, this study exclusively relies on a time-based clock for variable detection.

Mini flash crashes have been widely studied in the literature, the most commonly used definition of it is from Nanex (2010). Nanex (2010) defines Mini Flash Crashes as the following:

- The time window does not exceed 1.5 seconds;
- Price change exceeds 0.8%;
- At least 10 times tick down before ticking up OR at least 10 times tick up before ticking down;

In this paper, I adopt Nanex (2010)'s definition of mini flash crashes as events lasting 1.5 seconds. To further analyze these events, I introduce three additional

time windows: 180 seconds (long-term), 90 seconds (medium-term), and 15 seconds (short-term). The long-term window assumes sufficient reaction time for all traders within 180 seconds. The medium-term window validates the long-term perspective and highlights "fast-reactive" slow traders. The 15-second window captures the influence of fast traders and provides insights into overall market conditions. The extremely short-term category, aligned with Nanex (2010), remains the most stringent classification.

To comprehensively capture all mini flash crashes and build a prediction model capable of real-time forecasting, I assess all time intervals within each trading day. I employ a forward-looking window to identify mini flash crashes:

$\text{Int}^{\text{forward}}(T, T+\Delta) = \{t \in R : T < t \leq T+\Delta\}$, a span Δ equals 180s/90s/15s/1.5s.

In addition to Nanex (2010)'s characterization, Dugast and Foucault (2018) delineate mini flash crashes as instances involving substantial, abrupt price drops or spikes followed by rapid price reversals, often resembling "V-shaped" or "inverted V-shaped" price movements. Drawing inspiration from this concept, I adopt and implement a complementary version of mini flash crashes that focuses on these reversal patterns:

- The time window does not exceed 180 seconds/90 seconds/15 seconds/1.5 seconds;
- Price change exceeds 0.8%;
- At least 10 times tick down before ticking up AND at least 10 times tick up before ticking down;

The total number of mini flash crashes on NYSE American throughout the entire experimental period is depicted in Figure 4.5.

Figure 4.5

The total number of mini flash crashes with reversals on NYSE American throughout the entire experimental period is illustrated in Figure 4.6.

Figure 4.6

Considering the various time interval definitions, I include all stocks that experienced mini flash crashes, both with and without reversals, across all periods for both NYSE American stocks and their corresponding control group. The summary statistics for mini flash crashes without reversals, those with reversals, as well as the control groups without and with reversals, are presented in Table 4.1, Table 4.2, Table 4.3, and Table 4.4 respectively.

Table 4.1

4.4.2 Predictor Variables

To forecast the response variables, I employ a diverse set of predictor variables, primarily constructed following the methodology of Aït-Sahalia et al. (2022). These predictors include nonlinear transformations of historical data over short intervals. Based on the literature, including Cont et al. (2013) and Kercheval and Zhang (2015), I focus on characteristics of the current Limit Order Book (LOB), such as imbalances and past trade returns, as key drivers for predicting short-term events like mini flash crashes. Additionally, I incorporate microstructure measures from Easley et al. (2021). To capture market microstructure noise, which often signals unexpected events, I introduce three related predictors. A detailed description of all features will be provided in subsequent sections.

In a manner analogous to the forward-looking intervals used for predicting the response variable of mini flash crashes, I establish lookback intervals in terms of calendar time, employing the current timestamp T and defining lookback spans as (Δ_1, Δ_2). For constructing my predictor variables, I utilize lookback windows denoted as $I = Int(T - \Delta_2, T - \Delta_1)$. More precisely, the pairs of (Δ_1, Δ_2) assume values such as $(0s, .1s)$, $(.1s, .2s)$, and so forth, up to $(102.4s, 204.8s)$. The longest span, which is $204.8s$, is designed to cover a slightly over a 3-minute horizon following each

timestamp T, preventing the model from relying excessively on transient information to predict long-term outcomes.In total, there are 11 available lookback windows, and features can be directly computed after each interval is specified.

Let D^{txn} represent the set of all timestamps, $t \in D^{txn}$, corresponding to trade transactions, and D^{qt} represent its quote counterpart. We define the combined set as $D = D^{txn} \cup D^{qt}$. The National Best Bid and Offer (NBBO) prices, indexed by $t \in D$, are denoted as $\left(P_t^b, P_t^a\right)$, where P_t^b represents the best bid price and P_t^a represents the best ask price. The mid-price is calculated as the simple average, given by $P_t = \frac{P_t^b + P_t^a}{2}$. I denote P_t^{txn} as the transacted price if $t \in D^{txn}$. The best bid and ask sizes are represented as S_t^b and S_t^a, respectively, for the record indexed by t.

Volume and duration: Predictors are associated with a stock's trading intensity within a specified look-back window. For instance, the presence of block trades or a high frequency of transactions may suggest a recent surge in trading activity. Although this heightened trading activity may not inherently reveal the direction of the trend, it can interact with other predictors in a nonlinear manner, providing additional support for the trend's formation. The specific definition is provided below.

Breadth measures the number of transactions in the interval:

$$Breath(T, \Delta_1, \Delta_2) = |D^{txn} \cap Int^{back}(T, \Delta_1, \Delta_2)| \tag{4.1}$$

Immediacy measures the average time between successive transactions in the interval:

$$Immediacy(T, \Delta_1, \Delta_2) = \frac{\Delta_1 - \Delta_2}{Breath(T, \Delta_1, \Delta_2)} \tag{4.2}$$

VolumeAll measures the total number of shares transacted in the interval:

$$VolumeAll(T, \Delta_1, \Delta_2) = \sum_{t \in Int^{back}(T, \Delta_1, \Delta_2)} V_t \tag{4.3}$$

VolumeAvg measures the average number of shares transacted for each transaction

in the interval:

$$VolumeAvg(T, \Delta_1, \Delta_2) = \frac{VolumeAll(T, \Delta_1, \Delta_2)}{Breath(T, \Delta_1, \Delta_2)} \tag{4.4}$$

VolumeMax measures the maximum number of shares transacted in one transaction in the interval:

$$VolumeMax(T, \Delta_1, \Delta_2) = max\{V_t : t \in Int^{back}(T, \Delta_1, \Delta_2)\} \tag{4.5}$$

Return and imbalances: Predictors are linked to the recent trading asymmetry of the stock. These predictors may provide insights into the short-term trend. I possess information from both trades and quotes that can shed light on this trend. For instance, if a significant majority of trades are categorized as buying trades that match with limit sell orders, or if the bid substantially outweighs the ask in the Level I quotes, it suggests upward pressure on the price. I define the following variables to capture these aspects.

Lambda measures the price change in the interval proportional to total volume. Let $I = D^{txn} \cap Int^{back}(T, \Delta_1, \Delta_2)$, then:

$$Lambda(T, \Delta_1, \Delta_2) = \frac{P_{max(I)} - P_{min(I)}}{VolumeAll(T, \Delta_1, \Delta_2)} \tag{4.6}$$

LobImbalance is the average imbalance in the depth of the limit order book over the lookback interval:

$$LobImbalance(T, \Delta_1, \Delta_2) = Average[\frac{S_t^a - S_t^b}{S_t^a + S_t^b}] : t \in Int^{back}(T, \Delta_1, \Delta_2) \tag{4.7}$$

TxnImbalance measures the asymmetry of buy and sells volumes in recent transactions. Denote by Dir_t^{LR} the binary transaction direction at time t signed using the algorithm of Chakrabarty et al. (2007). Then transaction imbalance is calculated as

$$TxnImbalance(T,\Delta_1,\Delta_2) = Average[\frac{\sum_{t\in D^{txn}\cap Int^{back}(T,\Delta_1,\Delta_2)}(V_t Dir_t^{LR})}{VolumeAll(T,\Delta_1,\Delta_2)}] \quad (4.8)$$

PastReturn is the past return in the lookback window. Let $I = D^{txn}\cap Int^{back}(T,\Delta_1,\Delta_2)$:

$$PastReturn(T,\Delta_1,\Delta_2) = 1 - Average\frac{[P_t^{txn} : t\in I]}{P_{max(I)}} \quad (4.9)$$

Speed and cost This set of predictors I employ measure the speed and cost inherent in the stock's trading.

Turnover is the speed of transactions to the stock's total number of shares outstanding.

$$Turnover(T,\Delta_1,\Delta_2) = \frac{VolumeAll(T,\Delta_1,\Delta_2)}{S} \quad (4.10)$$

AutoCov is the autocovariance of transaction returns in the interval. For any $t \in D^{txn}$, denote by $L_t = \text{argmax}_s\{s : s < t, s \in D^{txn}\}$ the timestamp of the transaction right before time t. Then the autocovariance is:

$$\text{AutoCov}\,(T,\Delta_1,\Delta_2) = \text{ Average }\left[\log\left(\frac{P_t^{txn}}{P_{Lt}^{txn}}\right)\log\left(\frac{P_t^{txn}}{P_{L(Lt)}^{txn}}\right) : \text{t}\in D^{txn}\cap \text{Int}\,t^{\text{back}}\ (T,\Delta_1,\Delta_2)\right] \quad (4.11)$$

QuotedSpread is the average proportional nominal spread in the quotes over the lookback interval:

$$\text{QuotedSpread }\ (T,\Delta_1,\Delta_2) = \text{Average}\left[\frac{P_t^a - P_t^b}{P_t} : \text{t}\in \text{Int}^{\text{back}}\ (T,\Delta_1,\Delta_2)\right] \quad (4.12)$$

EffectiveSpread is the dollar-weighted percent effective spread over the interval:

$$\text{EffectiveSpread}\left(T,\Delta_1,\Delta_2\right)=\frac{\sum_{t\in D^{txn}nInt^{\text{back }\left(T,\Delta_1,\Delta_2\right)}}\left[\log\left(\frac{P_t^{txn}}{P_t}\right)\text{Dir}_t^{LR}V_tP_t^{txn}\right)}{\sum_{t\in D^{txn}\cap\ln t^{\text{back }\left(T,\Delta_1,\Delta_2\right)}}\left(V_tP_t^{txn}\right)} \tag{4.13}$$

Microstructure Measures This set of measures aligns with the approach in Easley et al. (2021), encompassing several established market microstructure variables. As previously mentioned, the forecasting of mini flash crashes is valuable when the model can deliver real-time predictions. To ensure this, I compute all of these microstructure measures utilizing lookback windows. More specifically, I include the following:

Roll measure:

$$\begin{aligned} R_t &= 2\sqrt{\left|\text{cov}\left(\Delta\boldsymbol{P}_t,\Delta\boldsymbol{P}_{t-1}\right)\right|} \\ \Delta\boldsymbol{P}_t &= \left[\Delta P_{t-w},\Delta P_{t-w-1},\ldots\Delta P_t\right], \\ \Delta\boldsymbol{P}_{t-1} &= \left[\Delta P_{t-w-1},\Delta P_{t-w},\ldots\Delta P_{t-1}\right], \end{aligned} \tag{4.14}$$

Where ΔP_t is the change in close price between bars $t-1$ and t and W is the lookback window size.

Roll impact, which is the Roll measure divided by the value traded over the lookback window, is:

$$\tilde{R}_t=\frac{2\sqrt{\left|\text{cov}\left(\Delta\boldsymbol{P_t},\Delta\boldsymbol{P}_{t-1}\right)\right|}}{P_tV_t}\textit{Roll impact:} \tag{4.15}$$

Kyle's lambda is given by:

$$\lambda_t=\frac{P_t-P_{t-w}}{\sum_{i=t}^{t}b_iV_t} \tag{4.16}$$

Where b_i is the trade indicator inferred by Chakrabarty et al. (2007), which is computed through one lookback window.

Amihud's measure:

$$\lambda_t^A = \frac{1}{W} \sum_{i=t-W+1}^{t} \frac{|r_i|}{p_i V_i} \tag{4.17}$$

Where r_i, p_i, V_i are the return, price, and volume at look back window i and W is the lookback window size in terms of the number of trades.

Volume-synchronized probability of informed trading is estimated as:

$$\text{VPIN}_t = \frac{1}{W} \sum_{i=\tau-W+1}^{\tau} \frac{\left|P_t^a - P_t^b\right|}{V_i} \tag{4.18}$$

P_t^a and P_t^b are bid and ask quotes.

Microstructure Realized Volatility and Noise The final set of measures draws inspiration from a body of literature on microstructure noise, in line with the theoretical framework proposed by Zhang et al. (2005). This set comprises three distinct measures:

Realized Volatility in each lookback window:

$$[P, P]_w = \sum_{t \in W} \left(P_{t_{i+1}} - P_{t_i}\right)^2 \tag{4.19}$$

Two-Scales Realized Volatility (TSRV) in each lookback window:

$$\widehat{\langle P, P \rangle}_T = [Y, Y]_T^{\text{avg}} \ - \frac{\bar{n}}{n} [Y, Y]_T^{\text{all}} \tag{4.20}$$

The combination is of two time scales, "all" and "average" sampling. More details could be found in Zhang et al. (2005).

The third one is $RealizedmoMentsofDisjointIncrements(ReMeDI)$ which is a new-developed measure to estimate microstructure noise, see Li and Linton (2022).

The last one is the daily Volatility Index (**VIX**). **VIX** has no look-back window; it is the CBOE Volatility Index, a popular measure of the stock market's expectation of volatility. It is based on S&P 500 index options and is provided daily by the Chicago Board Options Exchange.

4.4.3 Machine Learning Methods

Models

In this paper, we evaluate the predictive performance of five essential machine learning models for detecting mini flash crashes. These models include regularized logistic regression (LASSO) for linear parametric methods, penalized support vector machine (penalized-SVM) for nonparametric techniques, and two ensemble techniques: random forest and extreme gradient boosting (XGBoost). Additionally, we include a Multi-layer Perceptron (NN) as a representative of neural network models. Further theoretical details on these models are available in Hastie et al. (2009) and Murphy (2013).

We address an imbalanced classification problem where the response variable Y indicates an impending mini flash crash (1) or its absence (0) within the look-forward window. The prediction is based on a predictor vector X from a random sample ($\mathbf{X}_i$, Y_i), with $\mathbf{Y} = (y_1, ..., y_n)^T$. Each feature vector $\mathbf{X}_i$ has a dimension of 232, covering 11 time spans for each of the 21 predictor variables, plus the volatility index (VIX) from the preceding trading day. Machine learning algorithms can generate further combinations of these predictors or select the most informative subsets.

Penalized Logistic Regression

Among well-established machine learning models, linear models are notable for their simplicity and interpretability. Logistic regression, a widely used classification model, employs the sigmoid function to map the linear regression's value range into the interval $[0, 1]$. Specifically, the response variable is defined as follows:

$$y = \frac{1}{1 + e^{-z}} \tag{4.21}$$

Through linear regression model:

$$Z = \beta^T \mathbf{X} + \epsilon \tag{4.22}$$

Here, $\mathbf{X}$ represents the predictor vector. In the absence of regularization, standard Ordinary Least Squares (OLS) in a high-dimensional setting often leads to poor out-of-sample predictive performance due to in-sample overfitting. To address this issue, a common approach is to introduce regularization through a penalty function applied to normalized variables. Penalized least squares with an L_1 penalty is commonly known as the Least Absolute Shrinkage and Selection Operator (LASSO). Specifically, let $\bar{X} = \frac{1}{n}\sum_i X_i$ and $s_i = \sqrt{\frac{1}{n}\sum_i (x_i - \bar{x}_\imath)^2}$ be the mean vector and standard deviations of the predictor variables. Let $\bar{Z} = \frac{1}{n}\sum_i Z_i$ be the mean of the linear regression response variable. Define the centered regression response $\tilde{Z}_i = Z_i - \bar{Z}$ and standardized predictors $\widetilde{\boldsymbol{X}}_l = \text{diag}\left(s_1^{-1}, s_2^{-1} \dots s_p^{-1}\right)\left(\boldsymbol{X}_i - \overline{\boldsymbol{X}}\right)(i = 1, \dots, n)$. LASSO then fits the centered response on standardized predictors by solving the following optimization problem:

$$\hat{\beta} = \text{argmin}_{\beta \in R^p}\left\{\frac{1}{n}\sum_i \left(\widetilde{Z}_l - \beta^T \widetilde{\boldsymbol{X}}_l\right)^2 + \lambda\|\beta\|\right\} \tag{4.23}$$

This optimization problem can be effectively solved using convex optimization techniques. In this paper, I employ the coordinate descent algorithm, implemented within the Scikit-learn software in Python, for this purpose.

After I solve the coefficient $\hat{\beta}$, I can predict each new data $\boldsymbol{X}_{new}$ as:

$$\widehat{Y_{\text{new}}} = \frac{1}{1 + e^{-\left(\overline{\boldsymbol{Z}} + \widehat{\boldsymbol{\beta}}^T \boldsymbol{X}_{\text{new}}\right)}}, \text{ with } \widetilde{\boldsymbol{X}_{\text{new}}} = \text{diag}\left(s_1^{-1}, s_2^{-1}, \dots, s_p^{-1}\right)\left(\boldsymbol{X}_{\text{new}} - \overline{\boldsymbol{X}}\right) \tag{4.24}$$

LASSO is a straightforward and highly interpretable model that effectively shrinks the coefficients of less informative predictors toward zero. This enables us to assess the relevance of various predictors for the prediction problem. Consequently, I can identify which features exhibit significant predictive power and can serve as key signals for mini flash crash prediction.

Support Vector Machine

The Support Vector Machine (SVM) is a highly regarded non-parametric algorithm known for its strong performance and solid mathematical foundations. In this study, it represents non-parametric forecasting models. For a detailed discussion, see Hastie et al. (2009) and Murphy (2013).

An SVM constructs hyperplanes in a high-dimensional space to handle both classification and regression tasks. Given training vectors $x_i \in \mathbb{R}^p, i = 1, \ldots, n$ in two classes, and a response variable $y \in (1, -1)^n$, the goal is to find $\omega \in \mathbb{R}^p$ and $b \in \mathbb{R}$ such that $\omega^T \phi(x) + b$ predicts the sign of new input vectors x_{new}. For having this ω, SVM solves the following primal optimization problem:

$$\min_{\omega, b, \zeta 2} \frac{1}{2} \omega^T \omega + C \sum_{i=1}^{n} \zeta_i \tag{4.25}$$

$$\text{Subject to } y_i \left(\omega^T \phi(x) + b \right) \geq 1 - \zeta_i, \zeta_i \geq 0, i = 1, \ldots, n$$

The SVM aims to maximize the margin between two classes by minimizing the norm of the weight vector, $|\omega|^2 = \omega^T \omega$. Ideally, the hyperplane would perfectly separate all samples, satisfying $y_i \left(\omega^T \phi \left(x_i \right) + b \right) \geq 1$ for all i. In practice, perfect separation is often unattainable, so some samples are allowed to deviate by a distance ζ_i from their correct margin boundary. The penalty term C regulates the penalty strength. The dual problem corresponding to the primal problem is as follows:

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

Subject to $y^T \alpha = 0, 0 \leq \alpha_i \leq C, i = 1, \ldots, n$, where e is the vector of all ones, Q is a matrix: $Q_{ij} \equiv y_i y_j K \left(x_i x_j \right)$, where $K \left(x_i x_j \right) = \phi \left(x_i \right)^T \phi \left(x_j \right)$ is the kernel.

The kernel function is crucial for mapping samples into higher or infinite-dimensional spaces. For further details, refer to Hastie et al. (2009). Extensive experiments in this paper fine-tuned hyperparameters. Results showed that employing the linear kernel (simple inner product, $< x, x >$) with hinge loss consistently outperformed other

choices. In subsequent empirical results, only results from the linear kernel support vector machine will be presented. When solving the optimization problem, a support vector can predict a new sample x_{new} by:

$$\sum_{i \in SV}^{n} y_i \alpha_i K(x_i, x_{new}) + b \tag{4.26}$$

Then the predicted class corresponds to its sign.

Random forests

In this paper, ensemble learning tree-style models are explored for their forecasting capabilities, despite their comparative lack of interpretability compared to linear models. Specifically, the random forest ensemble model is employed. Random forests, as scalable nonparametric learning methods, improve upon single decision trees' potential instability and limited predictive power by ensembling multiple individual trees. By training independent decision trees and averaging their outcomes, prediction variance is reduced, resulting in more stable and reliable forecasts. As outlined in Hastie et al. (2009), random forests are constructed iteratively by growing regression trees through bootstrap sampling from the dataset. This iterative process enhances the model's robustness and predictive performance. The algorithm is outlined as follows in Hastie et al. (2009):

- For $b = 1$ to B
  - Draw a bootstrap sample Z* of size N from the training data.
  - Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
    - Select m variables at random from the p variables.
    - Pick the best variable/split-point among the m.
    - Split the node into two daughter nodes.

– Output the ensemble of trees $\{T_b\}_1^B$.

For new data, I want to predict x : Let $\hat{C}_b(x)$ be the class prediction of the b th random-forest tree. Then $\hat{C}_{rf}^B$(x) = majority vote $\left\{\hat{C}_b(x)\right\}_1^B$. (Hastie et al. (2009)).

Predictions from bagging methods often show high correlation because they rely on samples from the same dataset. However, by utilizing random forests, which involve bagging independently trained decision trees from bootstrapped samples alongside variable selection at each node, performance can be enhanced. This approach boosts independence among the trees, thereby reducing prediction dependence. Random forests effectively lower variance compared to a single decision tree, resulting in improved predictions.

Extreme Gradient Boosting

The second ensemble model utilized in this study is Extreme Gradient Boosting (XGBoost), originally introduced by Friedman (2001). XGBoost, a popular ensemble model outside the realm of deep learning, leverages multiple weak base tree learners. These learners typically exhibit high bias, with predictive performance only marginally better than random guessing.

In contrast to bagging techniques like Random Forest, which grow trees to their maximum depth, boosting aims to create small, shallow trees for interpretability. XGBoost begins with an initial model, denoted as F_0, to predict the target variable y. Then, the residual, represented as $y - F_0$, is computed. A new model, h_1, is subsequently fitted to this residual. By combining h_1 with F_0, the mean squared error is reduced. This iterative process continues, updating $F_1(x) < -F_0(x) + h_1(x)$, until the residuals are minimized. The algorithm follows these steps:

Given training set $\{(x_i, y_i)\}_{i=1}^N$, and a well-defined differentiable loss function $L(y, F(x))$, several weak tree learners M and a learning rate $\propto$.

- Initialize model with a constant:

$$f_0(x) = \widehat{\text{argmin}} \sum_{l=1}^{N} L\left(y_l, \theta\right). \tag{4.27}$$

- For m = 1 to M :

  - Compute the gradients and hessians:

$$\begin{aligned} \hat{g}_m\left(x_i\right) &= \left[\frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)}\right]_{f(x)=f_{(m-1)}(x)} \\ \hat{h}_m\left(x_i\right) &= \left[\frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)^2}\right]_{f(x)=f_{(m-1)}(x)} \end{aligned} \tag{4.28}$$

  - Fit a weak tree leaner using the training set $\left\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right\}_{i=1}^{N}$ by solve the optimization problem:

$$\begin{aligned} \circ\widehat{\emptyset_m} = \text{argmin}_{\emptyset} \sum_{i=1}^{N} \hat{h}_m\left(x_i\right)\left[-\frac{\hat{g}_m\left(x_i\right)}{\hat{h}_m\left(x_i\right)} - \emptyset\left(x_i\right)\right]^2 \\ \hat{f}_{(m)}(x) = \propto \widehat{\emptyset_m}(x) \end{aligned} \tag{4.29}$$

  - Update the model:

$$\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) + \hat{f}_{(m)}(x)$$

- Output $\hat{f}(x) = \sum_{m=0}^{M} \hat{f}_{(m)}(x)$

XGBoostContributors (2023)

More details could be found in Friedman (2001) and XGBoost documents (XGBoostContributors (2023)).

Neural Network model (Supervised)

The Neural Network model(NN) is a supervised learning algorithm designed to learn a function $f(\cdot) : R^m \to R^o$ through training on a dataset. Here m represents the number of input dimensions, while o denotes the number of output dimensions. Utilizing a feature set $X = x_1, x_2, \ldots, x_m$ and a target y, the NN can approximate non-linear functions for either classification or regression tasks. Unlike logistic regression, the NN incorporates one or more non-linear layers, known as hidden layers, between the input and output layers. Figure 4.7 illustrates an NN with a single hidden layer and scalar output. In my implementation, I only considered single layer with 100 neurons due to the complexity. The prediction outcomes are binary: a result of 1 indicates the occurrence of a mini flash crash, while a result of 0 signifies its absence.

Figure 4.7

More details could be found in Hastie et al. (2009) and sklearn documents (Pedregosa et al. (2011)).

Measuring Prediction Accuracy

I utilize Receiver Operating Characteristic (ROC) and Area under the ROC Curve (AUC) as metrics to assess our model's prediction accuracy. Predicting mini flash crashes essentially amounts to a supervised anomaly detection problem, given the exceedingly low probability of occurrence. With our dataset significantly imbalanced, the commonly used accuracy score becomes an inappropriate performance measure.

To illustrate, suppose only 0.1% of the look-forward window is labeled as a mini flash crash occurrence. In such cases, a naive approach might classify all windows as having no mini flash crashes, yielding a misleadingly high accuracy score of 99.9%. However, ROC provides a more suitable evaluation metric for imbalanced data problems. Originating from signal analysis technology, ROC curves have found extensive use in fields such as medical issue detection. These curves effectively illustrate a classification model's performance across all classification thresholds, plotting two key

parameters:

- True Positive Rate (TPR): $TPR = \frac{TP}{TP+FN}$
- False Positive Rate (FPR): $FPR = \frac{FP}{TP+TN}$

TP, TN, FN, and TN are all from the confusion matrix, see Mohajon (2020):

Figure 4.8

An ROC curve depicts the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) at different classification thresholds. Lowering the threshold results in more items classified as positive, increasing both False Positives and True Positives. Test samples are arranged in descending order based on predicted probabilities of being True, with the model sequentially predicting each sample as True to compute TPR and FPR. In a well-performing model, the curve exhibits a steep initial increase from 0, followed by a gradually slowing growth towards 1. AUC (Area under the ROC Curve) quantifies the area beneath the ROC curve, akin to calculating the integral of a function. It provides a comprehensive measure of performance across all thresholds, representing the probability of the model ranking a randomly selected positive example higher than a randomly selected negative example.

Imbalanced Data Processing Strategies

As mentioned earlier, our problem involves imbalanced data. In addition to utilizing the raw data directly, I will employ five additional strategies to address the data imbalance issue.

Undersampling: One approach to address data imbalance is to achieve a balanced sample set by removing surplus samples from the majority class (instances where no mini flash crash occurs). While this is the simplest strategy, it comes with drawbacks, including the loss of a substantial portion of data and valuable information inherent in the majority class. This approach may diminish the impact of certain majority cases, particularly those carrying informative value. For instance,

in support vector machine algorithms, deleted samples may include support vectors near the margin hyperplane on the majority class side, potentially leading to higher bias in the model.

Oversampling: Another strategy involves randomly duplicating synthetic samples in the minority class (instances where a mini flash crash occurs) to balance class sizes. However, this approach carries a risk of introducing noise samples, potentially leading to overfitting. As discussed in Chapter 3, this strategy does not yield optimal performance and can be resource-intensive. Consequently, I have chosen not to employ this strategy in my empirical analysis.

Synthetic Minority Oversampling Technique (SMOTE): SMOTE represents a nuanced oversampling approach, utilizing the K-Nearest Neighbors (KNN) algorithm to generate new samples within the minority class. These samples are designed to be similar to, yet distinct from, the original ones. Compared to simple oversampling, SMOTE functions akin to ensemble learning, mitigating variance and the risk of overfitting. However, while effective in addressing class imbalance, SMOTE has limitations. Relying on the minority class to generate synthetic samples can amplify noise impact due to data distribution changes. Moreover, SMOTE's computational complexity surpasses that of both oversampling and undersampling methods.

Threshold Moving: This strategy entails threshold adjustment to enhance the model's sensitivity to the minority class. In balanced data problems, the threshold is usually set at 0.5. Models calculate the probability of each sample belonging to class 1 or 0 and compare it with the 0.5 threshold. However, in imbalanced data scenarios, a straightforward approach is to shift the threshold to reflect the ratio of some minority class samples to some majority class samples in the training dataset.

Ensemble Undersampling: As mentioned earlier, the undersampling strategy involves deleting a significant number of majority samples to achieve a balanced dataset, potentially wasting valuable data. An alternative approach is multiple rounds of undersampling. For instance, with 10,000 majority class samples and only 50 mi-

nority class samples, one can create 200 sets of new samples. Each set would include all 50 minority class samples and an additional randomly selected 50 majority samples. These 200 models are then independently trained, and their results are combined in an ensemble. While offering potential advantages such as potentially more reliable performance, this strategy increases computational complexity significantly and maintains the risk of overfitting, as minority class samples are repeatedly used. However, in scenarios like anomaly detection, such as ours, successful detection of mini flash crashes is crucial. Therefore, assuming good quality minority class data, this strategy may yield dependable performance.

4.5 Empirical Analysis

Analyzing all detected mini flash crashes within various time intervals (180s/90s/15s/1.5s) in my dataset, I initially aggregate the mini flash crash events in both NYSE American and the control group. Subsequently, I conduct a difference-in-differences panel data regression to assess the impact of the speed bump on mini flash crashes. As illustrated below, none of these results exhibit high significance, suggesting that a direct observation of the speed bump's impact does not yield a clear effect.

However, this prompts a critical question: does the lack of immediate significance imply that the speed bump is entirely ineffective in mitigating mini flash crashes? Similar to a policy targeting traffic accidents, the speed bump may not immediately reduce the number of incidents, but it could influence and discourage dangerous driving patterns. The positive effects may manifest in the long term, emphasizing the need for a more nuanced understanding of the speed bump's impact on mini flash crashes.

Hence, I employ machine learning techniques to identify patterns in mini flash crashes and assess the impact of the speed bump. In addition to using the original imbalanced data, I apply all the strategies mentioned earlier, excluding oversampling, in combination with four models: logistic regression with an l_1 penalty (LASSO),

random forest, support vector machine, and XGBoost. As defined in the problem setup section, I consider eight scenarios of mini flash crashes based on the looking forward window, aiming to detect crashes within 180s/90s/15s/1.5s, without taking into account reversal and with strict reversal as defined in Section 4.4.1.

One key motivation behind Nanex (2010)'s definition of mini flash crashes is the requirement for a short time interval. Nanex (2010) uses a 1.5-second time window, in which even slow traders like humans cannot react quickly enough to cause stock prices to rise or fall significantly. In my empirical analysis, I not only cover short time intervals like 1.5 seconds and 15 seconds, where slow traders have limited impact on mini flash crash events, but I also consider longer time intervals such as 180 seconds and 90 seconds. For the 180-second and 90-second mini flash crashes, slow traders, including individuals, can have a more substantial effect on market dynamics.

As previously outlined in Section 4.4, this paper utilizes ROC and AUC as the evaluation metrics to gauge the performance of machine learning predictions, particularly due to the challenge posed by imbalanced data.

As suggested by Liu (2023), a speed bump can indeed exert a substantial impact on market quality. It is reasonable to speculate that the probability of long-term mini flash crashes may also be influenced by such policies, particularly when the market reaches a state of equilibrium between slow and fast traders, resulting in their respective feedback effects on the market.

4.5.1 Number of long term mini flash crashes regression

I begin by tallying the occurrences of mini flash crashes, both with and without reversal, for NYSE American and the control group before and immediately after the implementation of the speed bump. The summarized statistics of without reversal are presented in Panel A of Table 4.9.

Table 4.9

The summarized statistics of without reversal are presented in Panel A of Table 4.10.

Table 4.10

For each trading day t, I ultimately compute the variance in the number of mini flash crashes as follows:

$$NumMiniCrash_{i,t,180s} =$$
$$NumMiniCrash_{i,t,NYSE\text{ American },180s} - NumMiniCrash_{i,t,\text{ control},180s}.$$

$$NumMiniCrash_{i,t,180s,Rev} =$$
$$NumMiniCrash_{i,t,NYSE\text{ American },180s,Rev} - NumMiniCrash_{i,t,\text{ control},180s,Rev}.$$

Next, I conduct the following two regressions using a panel data approach, incorporating fixed effects for both date and stock:

$$NumMiniCrash_{i,t,180s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{4.30}$$

$$NumMiniCrash_{i,t,180s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{4.31}$$

where $NumMiniCrash_{i,t,180s}$ denotes the difference in the number of mini flash crashes between NYSE American and the control group without strict reversal, while $NumMiniCrash_{i,t,180s,Rev}$ signifies the difference in the number of mini flash crashes between NYSE American and the control group with strict reversal for stock i on day t. The term α_i incorporates stock and date fixed effects. Additionally, $SpeedBump_t$ is an indicator variable with a value of 1 after speed bump implementation and 0 otherwise. The variable "Volatility" corresponds to the opening value of the CBOE's VIX index on day t. Other independent variables serve as control variables. The $ControlVariable_{i,t,q}$ includes two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as per Alizadeh et al.

(2002)). In the linear regression without fixed effects, we also introduce the market capitalization of the stock.

Table 4.13

Table 4.13 presents the outcomes of our regression analysis, conducted with and without fixed effects, considering mini flash crashes both with and without the reversal requirement. In our linear regression models, we estimated the coefficients of the speed bump dummy variable to assess its causal impact on the probability of mini flash crashes. The coefficients suggest a positive effect of the speed bump policy on $NumMiniCrash_{i,t,180s}$, indicating an increased number of mini flash crashes. However, this result is not consistent between mini flash crashes without the strict reversal criterion and those with reversals in the regressions. Importantly, all results lack statistical significance.

4.5.2 Long term mini flash crashes prediction

In this subsection, I investigate the probability of long-term (180s) mini flash crashes, considering both those without reversal and those with reversal.

To train the models for calculating the probability of mini flash crashes before the speed bump, I utilize data spanning from March 15, 2017, to April 26, 2017 (Period 1, comprising 30 trading days). The performance of these models is then assessed from April 27, 2017, to June 8, 2017 (Period 2, spanning 30 trading days). This evaluation period serves as a basis for strategy and model selection, including hyperparameter tuning. For models forecasting the probability of mini flash crashes immediately following the speed bump, I employ data from April 27, 2017, to June 8, 2017 (Period 2, comprising 30 trading days) for training, while the subsequent 30 trading days from June 9, 2017, to July 23, 2017 (Period 3, spanning 30 trading days) are reserved for hyperparameter tuning.

After selecting the best model and the most suitable imbalanced data processing strategy during the tuning process described above, I train the chosen model on Period 2 and Period 3 data. This approach enables the model to capture the most

up-to-date market information immediately before the test period before the speed bump (Period 3) and the test period following the speed bump (Period 4).

For mini flash crashes without reversal, the tuning results of NYSE American are displayed in Panel A of Table 4.5, and the control group's tuning results can be found in Panel A of Table 4.6. In both tuning processes, conducted both before and after the speed bump, the ensemble undersampling combined with XGBoost demonstrates the most outstanding performance. The corresponding ROC curves are presented in Panel (a) and Panel (b) in Figure 4.9. Consequently, I employ this strategy to calculate the probability as follows:

$$P_{i,j,t,NYSE\text{ American },180s} = \frac{\sum\left\{\hat{f}_b(x_{i,t,j,NYSE\text{ American },180s})\right\}_1^B}{N_B}$$

The probability, denoted as $\hat{f}_b(x_{i,t,j,NYSE\text{ American },180s})$, is estimated through XGBoost for each balanced undersampling dataset constructed by pairing the minority class with an equal number of randomly selected majority samples. Here, B represents the number of XGBoost models I have trained. The final probability is calculated as the aggregate voting ratio across all models for a specific stock i, within time interval j, on date t, throughout the entire Period 3, which comprises the days leading up to the implementation of the speed bump on NYSE American. As shown in Panel A of Table 4.6, the control group exhibits similar tuning results, with ensemble undersampling combined with XGBoost delivering the best performance. The ROC curve for the control group can be found in Panel (c) and Panel (d) in Figure 4.9. Using the same methodology, I calculate the probabilities for all stocks $Pi, j, t,$ control within the control group as follows:

$$P_{i,j,t,Control,180s} = \frac{\sum\left\{\hat{f}_b(x_{i,t,j,Control,180s})\right\}_1^B}{N_B}$$

Then, I calculate the daily average probability of NYSE American as:

$$P_{i,t,NYSE\text{ American },180s} = Average(P_{i,j,t,NYSE\text{ American },180s}) \text{ for all } j \text{ in date } t.$$

and its control group as:

$$P_{i,t,Control,180s} = Average(P_{i,j,Control,180s}) \text{ for all } j \text{ in date } t.$$

for each trading day t. Finally, I calculate relative probability

$$P_{i,t,180s} = \ln\left(\frac{P_{i,t,NYSE \text{ American },180s}}{P_{i,t,\text{ control},180s}}\right).$$

for each stock and each day.

Panel A of Table 4.11 shows the daily average of mini flash crashes without reversal of all traded stocks in NYSE American and the control group.

Table 4.11

For mini flash crashes with reversal, the tuning results for NYSE American are presented in Panel A of Table 4.7, while the control group tuning results are displayed in Panel A of Table 4.8. Similar to the no-reversal version, in both tuning processes, before and after the speed bump, the ensemble undersampling combined with XGBoost achieves the best performance. The ROC curve for this strategy can be found in Panel (a) of Figure 4.9 for the pre-speed bump period and post-speed bump period in Panel (b) of Figure 4.9. Consequently, I utilize the same ensemble undersampling combined with XGBoost to calculate the probability, as follows:

$$P_{i,j,t,NYSE \text{ American },180s,Rev} = \frac{\sum\left\{\hat{f}_b(x_{i,t,j,NYSE \text{ American },180s,Rev})\right\}_1^B}{N_B}$$

The $\hat{f}b(xi, t, j, \text{NYSE American}, 180s, \text{Rev})$ is estimated using XGBoost for each balanced undersampling dataset, created by pairing the minority class with an equal number of randomly selected majority samples. In this context, B represents the number of XGBoost models that have been trained. The final probability is computed as the aggregated voting ratio across all models for a specific stock i, in time interval j, on date t throughout period 3, which encompasses the days leading up to the implementation of the speed bump on NYSE American. As the Panel A of Table 4.6 shows, my control group shows similar tuning results, ensemble under-sampling combined XGBoost shows best performance. I can use same to way to calculate control group probability. The control group ROC curve is shown in Panel (c) of

Figure 4.13 for the pre-speed bump period and post-speed bump period in Panel (d) of Figure 4.13.Through the same way, I calculate all $P_{i,j,t,control,Rev}$ stocks in the control group in the same way as:

$$P_{i,j,t,Control,180s,Rev} = \frac{\sum\left\{\hat{f}_b(x_{i,t,j,Control,180s,Rev})\right\}_1^B}{N_B}$$

Then, I calculate the daily average probability of NYSE American stock's mini flash crashes as:

$$P_{i,t,NYSE \text{ American },180s,Rev} = Average(P_{i,j,t,NYSE \text{ American },180s,Rev}) \text{ for all } j \text{ in date } t.$$

and its control group as:

$$P_{i,t,Control,180s,Rev} = Average(P_{i,j,t,Control,180s,Rev}) \text{ for all } j \text{ in date } t.$$

for each trading day t. Finally, I calculate relative probability of mini flash crashes with quick reversal as:

$$P_{i,t,180s,Rev} = \ln\left(\frac{P_{i,t,NYSE \text{ American },180s,Rev}}{P_{i,t,\text{ control },Rev}}\right).$$

for each stock and each day.

Panel A of 4.12 shows the daily average of mini flash crashes with reversal of all traded stocks in NYSE American and the control group.

Table 4.12

4.5.3 Probability of long term mini flash crashes regression

Having successfully predicted the probability of mini flash crashes without and with strict reversal, I proceed with the following two regressions, employing a panel data approach with fixed effects by date and stock:

$$P_{i,t,180s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (4.32)$$

$$P_{i,t,180s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (4.33)$$

where $P_{i,t,180s}$ is the daily average probability of mini flash crashes without strictly reversal and $P_{i,t,180s,Rev}$ is the daily average probability of mini flash crashes with strictly reversal for stock i on day t, α_i is the stock and date fixed effect, and $SpeedBump_t$ is an indicator variable taking the value of 1 after speed bump implementation, 0 otherwise. Volatility is the opening value of CBOE's VIX index on day t. The other independent variables are control variables. The $ControlVariable_{i,t,q}$ represents two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as Alizadeh et al. (2002)). For the linear regression without fixed effect, we also add market capitalization of the stock.

Table 4.14

Table 4.14 presents the results of our regression analysis, conducted both with and without fixed effects, considering mini flash crashes with and without the reversal requirement. In our linear regression models, we estimated the coefficients of the speed bump dummy variable to assess its causal impact on the probability of mini flash crashes. Surprisingly, our findings suggest that the speed bump policy had a positive effect on $P_{i,t,180s}$, indicating an increase in the long-term probability of mini flash crashes. This outcome holds consistently for both mini flash crashes without the strict reversal criterion and those with reversals in the regressions that incorporate stock fixed effects.

Interestingly, these results align with empirical evidence from Liu (2023), which indicates that the speed bump implementation on NYSE American significantly reduces relative price discovery while simultaneously increasing both the cost of immediacy and market volatility. As NYSE American becomes a slower market with all incoming orders delayed by 350 microseconds, it is reasonable to assume that more uninformed noise traders have migrated to this exchange since the policy's adoption. This influx

of noise traders may have contributed to the reduced informativeness of NYSE American's prices and the increased prevalence of microstructure noise. Consequently, the heightened presence of noise in NYSE American may explain our findings, indicating an elevated probability of long-term mini flash crashes in the 180-second interval following the speed bump's implementation.

4.5.4 Number of medium term mini flash crashes regression

Same as the 180s long term version, I begin by tallying the occurrences of mini flash crashes, both with and without reversal, for NYSE American and the control group before and immediately after the implementation of the speed bump. The summarized statistics of without reversal are presented in Panel B of Table 4.9.

Table 4.9

The summarized statistics of without reversal are presented in Panel B of Table 4.10.

Table 4.10

For each trading day t, I ultimately compute the variance in the number of mini flash crashes as follows:

$$NumMiniCrash_{i,t,90s} = NumMiniCrash_{i,t,NYSE\text{American},90s} - NumMiniCrash_{i,t,\text{ control},90s}$$

$$NumMiniCrash_{i,t,90s,Rev} = NumMiniCrash_{i,t,NYSE\text{American},90s,Rev} - NumMiniCrash_{i,t,\text{control},90s,Rev}$$

Next, I conduct the following two regressions using a panel data approach, incorporating fixed effects for both date and stock:

$$NumMiniCrash_{i,t,90s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{4.34}$$

$$NumMiniCrash_{i,t,90s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{4.35}$$

where $NumMiniCrash_{i,t,90s}$ denotes the difference in the number of mini flash crashes between NYSE American and the control group without strict reversal, while $NumMiniCrash_{i,t,90s,Rev}$ signifies the difference in the number of mini flash crashes between NYSE American and the control group with strict reversal for stock i on day t. The term α_i incorporates stock and date fixed effects. Additionally, $SpeedBump_t$ is an indicator variable with a value of 1 after speed bump implementation and 0 otherwise. The variable "Volatility" corresponds to the opening value of the CBOE's VIX index on day t. Other independent variables serve as control variables. The $ControlVariable_{i,t,q}$ includes two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as per Alizadeh et al. (2002)). In the linear regression without fixed effects, we also introduce the market capitalization of the stock.

Table 4.15

Table 4.15 presents the outcomes of our regression analysis, conducted with and without fixed effects, considering mini flash crashes both with and without the reversal requirement. In our linear regression models, we estimated the coefficients of the speed bump dummy variable to assess its causal impact on the probability of mini flash crashes. The coefficients suggest a positive effect of the speed bump policy on $NumMiniCrash_{i,t,90s}$, indicating an increased number of mini flash crashes. Like the long-term version, this outcome exhibits inconsistency between mini flash crashes that lack the strict reversal criterion and those with reversals in the regressions. Notably, none of the results demonstrate statistical significance.

4.5.5 Medium term mini flash crashes prediction

In this subsection, I replicate the same procedure to investigate the probability of mini flash crashes in the medium term (90 seconds). Just like the 180-second analysis, this 90-second examination offers a robust check of our findings, as it provides another timeframe in which slow traders, including individual investors, have ample time to react to market developments. I conduct empirical assessments for both mini flash crashes without the reversal requirement and those with reversals, mirroring the approach used in the previous subsection.

To train the models and calculate the probability of mini flash crashes before the speed bump, I utilize data from Period 1 and evaluate the models' performance during Period 2. This evaluation phase facilitates the selection of strategies, model selection, and hyperparameter tuning. For models aimed at estimating the probability of mini flash crashes immediately following the speed bump, I employ data from Period 2 as the training set and data from the subsequent 30 trading days in Period 3 for hyperparameter tuning.

Following the same methodology used in the previous section, I utilize the best-performing model and optimal imbalanced data processing strategy selected through the tuning process. This approach allows us to leverage the most up-to-date market information in Period 3 to train the model, which is then applied to both the test period preceding the speed bump (Period 3) and the post-speed bump period (Period 4).

For mini flash crashes without the reversal requirement in the medium term (90 seconds), I present the tuning results for NYSE American in Panel B of Table 4.5 and the control group tuning results in Panel B of Table 4.6. As observed in my analysis for the 180-second timeframe, the ensemble undersampling combined with XGBoost consistently outperforms other methods in both the period before and after the speed bump. The ROC curve for this model is illustrated in Panel (a) of Figure 4.10 for the pre-speed bump period and in Panel (b) of Figure 4.10 for the post-speed bump

period. Consequently, I still employ this strategy to calculate the probability for mini flash crashes without reversal in the medium term.

$$P_{i,j,t,NYSE \text{ American },90s} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,NYSE \text{ American },90s})\right\}_1^B}{N_B}$$

The probability, denoted as $\hat{f}_b(x_{i,t,j,NYSE \text{ American },90s})$, is estimated through XGBoost for each balanced undersampling dataset constructed by pairing the minority class with an equal number of randomly selected majority samples. Here, B represents the number of XGBoost models I have trained. The probability is calculated for stock i, in time interval j, date t in the whole period 3 , which are days before the implementation of the speed bump in NYSE American. As the Panel B of Table 4.6 shows, my control group shows similar tuning results, ensemble under-sampling combined XGBoost shows best performance. The control group ROC curve is shown in Panel (c) of Figure 4.10 for the pre-speed bump period and post-speed bump period in Panel (d) of Figure 4.10. I can use same to way to calculate control group probability. Through the same way, I calculate all $P_{i,j,t,control,90s}$ stocks in the control group in the same way as:

$$P_{i,j,t,Control,90s} == \frac{\sum\left\{\hat{f}_b(x_{i,j,t,Control,90s})\right\}_1^B}{N_B}$$

Then, I calculate the daily average probability of NYSE American as:

$$P_{i,t,NYSE \text{ American },90s} = Average(P_{i,j,t,NYSE \text{ American },90s}) \text{ for all } j \text{ in date } t.$$

and its control group as:

$$P_{i,t,Control,90s} = Average(P_{i,j,Control,90s}) \text{ for all } j \text{ in date } t.$$

for each trading day t. Finally, I calculate relative probability

$$P_{i,t,90s} = \ln\left(\frac{P_{i,t,NYSE \text{ American },90s}}{P_{i,t,\text{ control},180s}}\right).$$

for each stock and each day.

Panel B of 4.11 shows the daily average of mini flash crashes in 90s without reversal of all traded stocks in NYSE American and the control group.

Table 4.11

For mini flash crashes with reversal, the tuning results for NYSE American are displayed in Panel B of 4.7, while the control group tuning results are presented in Panel B of 4.8. Similar to the no-reversal version, for NYSE American, the ensemble undersampling combined with XGBoost performs best in both tuning processes before and after the speed bump. However, for the control group, direct undersampling combined with Random Forest exhibits the best performance. Therefore, I will use undersampling combined with Random Forest to calculate the probability of mini flash crashes in 90s. The ROC curves for these models are depicted in Panel (a) of Figure 4.14 for the pre-speed bump period and post-speed bump in Panel (b) of Figure 4.14. Consequently, I employ the ensemble undersampling combined with XGBoost for NYSE American and the undersampling combined with Random Forest for the control group to calculate the probabilities.:

$$P_{i,j,t,NYSE\text{ American },90s,Rev} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,NYSE\text{ American },90s,Rev})\right\}_1^B}{N_B}$$

The $\hat{f}_b(xi, j, t, NYSE$ American $, 90s, Rev)$ is estimated using XGBoost combined with undersampling, as described in the previous section. This estimation is performed for stock i, in time interval j, on date t, spanning the entire period 3, which encompasses the days leading up to the implementation of the speed bump on NYSE American. As shown in Panel A of Table 4.6, my control group exhibits similar tuning results, with ensemble undersampling combined with XGBoost showing the best performance. I employ the same method to calculate the probability for the control group. The ROC curves for the control group are displayed in Panel (c) of Figure 4.14 for the pre-speed bump period and postt-speed bump period in Panel (d) of Figure 4.14. Using the same approach, I calculate all $P_{i,j,t,control,Rev}$ values for stocks in the control group.

$$P_{i,j,t,Control,90s,Rev} = \frac{\left\{ \hat{C}_b(x) \right\}_1^B}{N_B}$$

The $\left\{ \hat{C}_b(x) \right\}_1^B$ is estimated by each decision tree votes as positive in random forest, N_B is the number of trees. Then, I calculate the daily average probability of NYSE American stock's mini flash crashes as:

$$P_{i,t,NYSE \text{ American },90s,Rev} = Average(P_{i,j,t,NYSE \text{ American },90s,Rev}) \text{ for all } j \text{ in date } t.$$

and its control group as:

$$P_{i,t,Control,90s,Rev} = Average(P_{i,t,j,Control,90s,Rev}) \text{ for all } j \text{ in date } t.$$

for each trading day t. Finally, I calculate relative probability of mini flash crashes with quick reversal as:

$$P_{i,t,90s,Rev} = \ln\left(\frac{P_{i,t,NYSE \text{ American },90s,Rev}}{P_{i,t, \text{ control },90s,Rev}}\right).$$

for each stock and each day.

Panel B of 4.12 shows the daily average of mini flash crashes with reversal of all traded stocks in NYSE American and the control group.

Table 4.12

4.5.6 Probability of medium term mini flash crashes regression

After successfully predicting the probability of mini flash crashes, both with and without strict reversal, I proceed to conduct the following two regressions using a panel data approach with fixed effects for both date and stock:

$$P_{i,t,90s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (4.36)$$

$$P_{i,t,90s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (4.37)$$

In these equations, $P_{i,t,90s}$ represents the daily average probability of mini flash crashes without strict reversal, and $P_{i,t,90s,Rev}$ represents the daily average probability of mini flash crashes with strict reversal for stock i on day t. The variable α_i denotes the stock and date fixed effects, while $SpeedBump_t$ is an indicator variable that takes the value of 1 after the implementation of the speed bump and 0 otherwise. The term "Volatility" corresponds to the opening value of the CBOE's VIX index on day t. Additionally, there are control variables denoted as $ControlVariable_{i,t,q}$, which encompass two stock-level control variables: the daily turnover difference and the daily stock volatility difference, as calculated following the method of Alizadeh et al. (2002). In the linear regression model without fixed effects, we also include the market capitalization of the stock as an independent variable.

Table 4.18

Table 4.18 displays the results of regression analyses conducted with and without fixed effects for both scenarios, considering mini flash crashes without strict reversal and those with reversal requirements. In the linear regression models, I estimated the coefficients of the dummy variable "speed bump" enabling us to discern the causal effect of the speed bump on the probability of mini flash crashes. The findings indicate that the speed bump had a positive effect on $P_{i,t,90s}$, signifying an increase in the long-term probability of mini flash crashes. These results consistently hold for both mini flash crashes without strict reversal and those with reversals when accounting for stock fixed effects.

The results for the 90s analysis align comprehensively with the 180s analysis. For both scenarios, encompassing mini flash crashes without reversal and those with reversal, we observe that the speed bump has a positive impact on the long-term probability of mini flash crashes in minutes. In summary, in the context of relatively

long-term intervals at the minute level, the speed bump appears to elevate the likelihood of impending mini flash crashes. This trend may be attributed to reduced price discovery in a condition where more noise traders execute their trading activities in NYSE American.

4.5.7 Number of short term mini flash crashes regression

Using the same methodology as in the previous section, I compute the count of mini flash crashes, both with and without reversal, for NYSE American and the control group before and immediately after the implementation of the speed bump. The summarized statistics without reversal are detailed in Panel C of Table 4.9.

Table 4.9

The summarized statistics of without reversal are presented in Panel C of Table 4.10.

Table 4.10

For each trading day t, I ultimately compute the variance in the number of mini flash crashes as follows:

$$NumMiniCrash_{i,t,15s} = NumMiniCrash_{i,t,NYSE\text{ American },15s} - NumMiniCrash_{i,t,\text{ control},15s}.$$

$$NumMiniCrash_{i,t,15s,Rev} = NumMiniCrash_{i,t,NYSE\text{ American },180s,Rev} - NumMiniCrash_{i,t,\text{ control},15s,Rev}.$$

Next, I conduct the following two regressions using a panel data approach, incorporating fixed effects for both date and stock:

$$NumMiniCrash_{i,t,15s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{4.38}$$

$$NumMiniCrash_{i,t,15s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{4.39}$$

where $NumMiniCrash_{i,t,15s}$ denotes the difference in the number of mini flash crashes between NYSE American and the control group without strict reversal, while $NumMiniCrash_{i,t,15s,Rev}$ signifies the difference in the number of mini flash crashes between NYSE American and the control group with strict reversal for stock i on day t. The term α_i incorporates stock and date fixed effects. Additionally, $SpeedBump_t$ is an indicator variable with a value of 1 after speed bump implementation and 0 otherwise. The variable "Volatility" corresponds to the opening value of the CBOE's VIX index on day t. Other independent variables serve as control variables. The $ControlVariable_{i,t,q}$ includes two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as per Alizadeh et al. (2002)). In the linear regression without fixed effects, we also introduce the market capitalization of the stock.

Table 4.17

Table 4.17 presents the results of our regression analysis, conducted with and without fixed effects, considering mini flash crashes both with and without the reversal requirement. In our linear regression models, we estimated the coefficients of the speed bump dummy variable to assess its causal impact on the probability of mini flash crashes. Surprisingly, our findings suggest a positive effect of the speed bump policy on $NumMiniCrash_{i,t,15s}$, indicating an increased number of mini flash crashes. This result is consistent for both mini flash crashes without the strict reversal criterion and those with reversals in the regressions that incorporate stock fixed effects or not. However, all results lack statistical significance.

4.5.8 Short term mini flash crashes prediction

In this sub-section, I employ the same approach to investigate the short-term probability of mini flash crashes in 15 seconds, considering both scenarios of mini flash crashes with and without reversal. In the definition of mini flash crashes outlined in Section 4.4.1, we characterize mini flash crashes as involving at least a tenfold increase or decrease in stock prices. Within a 15-second timeframe, it becomes challenging for slow traders, such as human investors, to react swiftly enough and induce such rapid price crashes. Hence, it is reasonable to assume that this level of flash crashes is primarily attributed to fast traders, such as high-frequency traders.

To train the models for calculating the probability of mini flash crashes before the speed bump, I adhere to the same procedure, utilizing data spanning Period 1 and evaluating model performance in Period 2. This evaluation phase serves as a means for strategy and model selection, as well as hyperparameter tuning. For models aiming to predict the probability of mini flash crashes occurring immediately after the speed bump, I employ data from Period 2 as the training set, and the subsequent 30 trading days from Period 3 are designated for hyperparameter tuning.

By leveraging the best-performing model and optimal imbalanced data processing strategy selected through the tuning process outlined above, I employ data from Period 2 and Period 3 to train the model, capturing the most recent market information immediately preceding the test period before the speed bump (Period 3) and the test period following the speed bump (Period 4).

For mini flash crashes without reversal, the tuning results of NYSE American are shown in Panel C of Table 4.5 and the control group tuning results are shown in Panel C of 4.6. For both tuning process for Period before the speed bump and after the speed bump, the ensemble undersampling combined XGBoost achieve the best performance. Its ROC curve is shown in Panel (a) of Figure 4.11 for the pre-speed bump period and post-speed bump period in Panel (b) of Figure 4.11. Therefore, I use the strategy to calculate the probability through:

$$P_{i,j,t,NYSE\text{ American },15s} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,NYSE\text{ American },15s})\right\}_1^B}{N_B}$$

The $\hat{f}_b(x_{i,j,t,\text{NYSE American},15s})$ is estimated using XGBoost in conjunction with ensemble undersampling, following the same methodology as described in the preceding two sub-sections. This estimation is conducted for each stock i, within time interval j, and on date t throughout Period 3, which encompasses the days leading up to the implementation of the speed bump on NYSE American. As indicated by the results in Panel A of Table 4.6, the control group exhibits similar tuning outcomes, with ensemble undersampling combined XGBoost yielding the most favorable performance. The ROC curve for the control group is illustrated in Panel (c) of Figure 4.11 for the pre-speed bump period and post-speed bump period in Panel (d) of Figure 4.11 displays its performance after the speed bump. I employ a similar approach to calculate the control group probability. Using this method, I compute the probabilities for all $P_{i,j,t,\text{control}}$ stocks within the control group in the same manner.

$$P_{i,j,t,Control,15s} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,Control,15s})\right\}_1^B}{N_B}$$

Then, I calculate the daily average probability of NYSE American as:

$$P_{i,t,NYSE\text{ American },15s} = Average(P_{i,j,t,NYSE\text{ American },15s}) \text{ for all } j \text{ in date } t.$$

and its control group as:

$$P_{i,t,Control,15s} = Average(P_{i,j,Control,15s}) \text{ for all } j \text{ in date } t.$$

for each trading day t. Finally, I calculate relative probability

$$P_{i,t,15s} = \ln\left(\frac{P_{i,t,NYSE\text{ American },15s}}{P_{i,t,\text{ control},15s}}\right).$$

for each stock and each day.

Panel C of Table 4.11 shows the daily average of mini flash crashes without reversal of all traded stocks in NYSE American and the control group.

Table 4.11

For mini flash crashes with reversal, the tuning results of NYSE American are shown in Panel C of Table 4.7 and the control group tuning results are shown in Panel C of Table 4.8. Same as the no reversal version, both tuning process for Period before the speed bump and after the speed bump, the ensemble undersampling combined XGBoost achieve the best performance. Its ROC curve is shown in Panel (a) of Figure 4.15 for the pre-speed bump period and post-speed bump period in Panel (b) of Figure 4.15. Therefore, I use the same ensemble undersampling combined XGBoost to calculate the probability through:

$$P_{i,j,t,NYSE \text{ American },15s,Rev} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,Control,15s,Rev})\right\}_1^B}{N_B}$$

The $\hat{f}_b(x_{i,j,t,\text{Control},15s,\text{Rev}})$ is estimated using XGBoost combined with ensemble undersampling for each stock i, within time interval j, and on date t during Period 3, which comprises the days preceding the implementation of the speed bump on NYSE American. As indicated by the results in Panel A of Table 4.6, the control group exhibits similar tuning outcomes, with ensemble undersampling combined XGBoost yielding the most favorable performance. The ROC curve for the control group is illustrated Panel (c) of Figure 4.15 for the pre-speed bump period and post-speed bump period in Panel (d) of Figure 4.15 displays its performance after the speed bump. Using a consistent approach, I calculate the probabilities for all $P_{i,j,t,\text{control, Rev}}$ stocks within the control group.

$$P_{i,j,t,Control,15s,Rev} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,Control,15s,Rev})\right\}_1^B}{N_B}$$

Then, I calculate the daily average probability of NYSE American stock's mini flash crashes as:

$$P_{i,t,NYSE \text{ American },15s,Rev} = Average(P_{i,j,t,NYSE \text{ American },15s,Rev}) \text{ for all } j \text{ in date } t.$$

and its control group as:

$$P_{i,t,Control,15s,Rev} = Average(P_{i,t,j,Control,15s,Rev}) \text{ for all } j \text{ in date } t.$$

for each trading day t. Finally, I calculate relative probability of mini flash crashes with quick reversal as:

$$P_{i,t,15s,Rev} = \ln\left(\frac{P_{i,t,NYSE\text{ American },15s,Rev}}{P_{i,t,\text{ control },Rev}}\right).$$

for each stock and each day.

Panel C of Table 4.12 shows the daily average of mini flash crashes with reversal of all traded stocks in NYSE American and the control group.

Table 4.12

4.5.9 Probability of short term mini flash crashes regression

Through the successful prediction of the probability of mini flash crashes without and with strict reversal, I perform the following two regressions using a panel data approach with fixed effects by date and stock:

$$P_{i,t,15s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (4.40)$$

$$P_{i,t,15s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (4.41)$$

where $P_{i,t,15s}$ is the daily average probability of mini flash crashes without strictly reversal and $P_{i,t,15s,Rev}$ is the daily average probability of mini flash crashes with strictly reversal for stock i on day t, α_i is the stock and date fixed effect, and $SpeedBump_t$ is an indicator variable taking the value of 1 after speed bump implementation, 0 otherwise. Volatility is the opening value of CBOE's VIX index on day t. The other independent variables are control variables. The $ControlVariable_{i,t,q}$ represents two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as Alizadeh et al. (2002)). For the linear regression without fixed effect, we also add market capitalization of the stock.

Table 4.18

Table 4.18 presents the results of a regression analysis conducted with and without fixed effects for both scenarios, with and without the reversal requirement. In the linear regression, we estimated the coefficients of the dummy variable speed bump, enabling us to observe the causal effect of the speed bump on the probability of mini flash crashes. We discovered that the speed bump had a negative effect on $P_{i,t,15s}$, signifying a decrease in the short-term probability of mini flash crashes. These results hold true for both mini flash crashes, whether with or without strict reversal, when considering regressions with stock fixed effects.

This outcome provides the first empirical evidence in the literature that the speed bump can significantly reduce the probability of mini flash crashes. As a policy aimed at mitigating the speed advantage of high-frequency trading, the speed bump, in addition to its other effects on price discovery, adverse selection, and market liquidity, appears to effectively fulfill its primary policy objective. By introducing a 350-microsecond delay for incoming orders, NYSE American has successfully reduced the probability of mini flash crashes occurring within 15 seconds.

4.5.10 Number of Nanex (2010) mini flash crashes regression

I begin by tallying the occurrences of mini flash crashes, both with and without reversal, for NYSE American and the control group before and immediately after the implementation of the speed bump. The summarized statistics of without reversal are presented in Panel D of Table 4.9.

Table 4.9

The summarized statistics of without reversal are presented in Panel D of Table 4.10.

Table 4.10

For each trading day t, I ultimately compute the variance in the number of mini flash crashes as follows:

$$NumMiniCrash_{i,t,1.5s} =$$
$$NumMiniCrash_{i,t,NYSE \text{ American },1.5s} - NumMiniCrash_{i,t, \text{ control},180s}.$$

$$NumMiniCrash_{i,t,1.5s,Rev} =$$
$$NumMiniCrash_{i,t,NYSE \text{ American },1.5s,Rev} - NumMiniCrash_{i,t, \text{ control},1.5s,Rev}.$$

Next, I conduct the following two regressions using a panel data approach, incorporating fixed effects for both date and stock:

$$NumMiniCrash_{i,t,1.5s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{4.42}$$

$$NumMiniCrash_{i,t,1.5s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{4.43}$$

where $NumMiniCrash_{i,t,1.5s}$ denotes the difference in the number of mini flash crashes between NYSE American and the control group without strict reversal, while $NumMiniCrash_{i,t,1.5s,Rev}$ signifies the difference in the number of mini flash crashes between NYSE American and the control group with strict reversal for stock i on day t. The term α_i incorporates stock and date fixed effects. Additionally, $SpeedBump_t$ is an indicator variable with a value of 1 after speed bump implementation and 0 otherwise. The variable "Volatility" corresponds to the opening value of the CBOE's VIX index on day t. Other independent variables serve as control variables. The $ControlVariable_{i,t,q}$ includes two stock-level control variables: the daily turnover difference and the daily stock volatility difference (calculated as per Alizadeh et al. (2002)). In the linear regression without fixed effects, we also introduce the market capitalization of the stock.

Table 4.19

Table 4.19 presents the results of our regression analysis, conducted with and

without fixed effects, considering mini flash crashes both with and without the reversal requirement. In our linear regression models, we estimated the coefficients of the speed bump dummy variable to assess its causal impact on the probability of mini flash crashes. Surprisingly, our findings suggest a positive effect of the speed bump policy on $NumMiniCrash_{i,t,1.5s}$, indicating an increased number of mini flash crashes. Similar to the short-term 15s version, this finding remains consistent across mini flash crashes, whether they adhere to the strict reversal criterion or involve reversals in the regressions with or without the incorporation of stock fixed effects. Nevertheless, it is important to note that all results lack statistical significance.

4.5.11 Nanex (2010) mini flash crashes prediction

In the final sub-section, I examine the probability of mini flash crashes in the shortest term (1.5s), considering both mini flash crashes without a reversal, as defined by Nanex (2010), and mini flash crashes with a reversal under the same conditions as the previous sections.

Similar to the preceding sub-sections, I utilize data spanning from Period 1, consisting of 30 trading days, to train the initial model. The model's performance is evaluated in Period 2, which serves as the basis for strategy and model selection, as well as hyperparameter tuning. For the models aimed at calculating the probability of mini flash crashes occurring immediately after the speed bump, I employ data from Period 2 for training and the subsequent 30 trading days from Period 3 for hyperparameter tuning.

After identifying the best model and optimal imbalanced data processing strategy through the aforementioned tuning process, I utilize the data from Periods 2 and 3 to train the model, ensuring it captures the most up-to-date market information immediately preceding the test period prior to the speed bump (Period 3) and the test period following the speed bump (Period 4).

For mini flash crashes without a reversal, the tuning results for NYSE American are presented in Panel D of Table 4.5, while the control group's tuning results are

displayed in Panel D of Table 4.6. In both cases, for both the tuning process conducted prior to the speed bump and after it, the ensemble undersampling combined with XGBoost yields the best performance. The ROC curves for this strategy are depicted in Panel (a) of Figure 4.12 for the pre-speed bump period and post-speed bump period in Panel (b) of Figure 4.12. Consequently, I employ this strategy to calculate the probability as follows:

$$P_{i,j,t,NYSE\text{ American },1.5s} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,NYSE\text{ American },1.5s})\right\}_1^B}{N_B}$$

The $\hat{f}_b(x_{i,j,t,\text{NYSE American},1.5s})$ is estimated using XGBoost combined with ensemble undersampling, as described in the previous section. This is done for stock i, within time interval j, and on date t during the entire Period 3, which includes the days leading up to the implementation of the speed bump on NYSE American. As indicated by the results in Panel A of Table 4.6, my control group exhibits similar tuning outcomes, with ensemble undersampling combined with XGBoost delivering the most favorable performance. The ROC curves for the control group are presented in Panel (c) of Figure 4.12 for the pre-speed bump period and post-speed bump period in Panel (d) of Figure 4.12. I employ the same approach to calculate the probability for all $P_{i,j,t,\text{control}}$ stocks within the control group in the following manner:

$$P_{i,j,t,Control,1.5s} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,Control,1.5s})\right\}_1^B}{N_B}$$

Then, I calculate the daily average probability of NYSE American as:

$$P_{i,t,NYSE\text{ American },1.5s} = Average(P_{i,j,t,NYSE\text{ American },1.5s}) \text{ for all } j \text{ in date } t.$$

and its control group as:

$$P_{i,t,Control,1.5s} = Average(P_{i,j,Control,1.5s}) \text{ for all } j \text{ in date } t.$$

for each trading day t. Finally, I calculate relative probability

$$P_{i,t,1.5s} = \ln\left(\frac{P_{i,t,NYSE\text{ American },1.5s}}{P_{i,t,\text{ control},1.5s}}\right).$$

for each stock and each day.

Panel D of 4.11 shows the daily average of mini flash crashes without reversal of all traded stocks in NYSE American and the control group.

Table 4.11

For mini flash crashes with reversal, the tuning results of NYSE American are shown in Panel D of Table 4.7 and the control group tuning results are shown in Panel D of Table 4.8. Same as the no reversal version, both tuning process for Period before the speed bump and after the speed bump, the ensemble undersampling combined XGBoost achieve the best performance. Its ROC curve is shown in Panel (c) of Figure 4.12 for the pre-speed bump period and post-speed bump period in Panel (d) of Figure 4.12. Therefore, I use the same ensemble undersampling combined XGBoost to calculate the probability through:

$$P_{i,j,t,NYSE\text{ American },1.5s,Rev} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,NYSE\text{ American },1.5s,Rev})\right\}_1^B}{N_B}$$

The $f_b(x_{i,t,j,NYSE\text{ American },Rev})$ is estimated by XGBoost for each stock i, within time interval j, and on date t during Period 3, which comprises the days preceding the implementation of the speed bump on NYSE American. As evident from the results in Panel A of Table 4.6, my control group exhibits similar tuning outcomes, with ensemble undersampling combined XGBoost yielding the most favorable performance. The ROC curves for the control group are depicted in Panel (a) of Figure 4.16 for the pre-speed bump period and post-speed bump in Panel (b) of Figure 4.16. Employing the same approach, I calculate the probabilities for all $P_{i,j,t,1.5s,\text{control, Rev}}$ stocks within the control group.

$$P_{i,j,t,Control,1.5s,Rev} = \frac{\sum\left\{\hat{f}_b(x_{i,j,t,Control,1.5s,Rev})\right\}_1^B}{N_B}$$

Then, I calculate the daily average probability of NYSE American stock's mini flash crashes as:

$$P_{i,t,NYSE\text{ American },1.5s,Rev} = Average(P_{i,j,t,NYSE\text{ American },180s,Rev}) \text{ for all } j \text{ in date } t.$$

and its control group as:

$$P_{i,t,Control,1.5s,Rev} = Average(P_{i,t,j,Control,1.5s,Rev}) \text{ for all } j \text{ in date } t.$$

for each trading day t. Finally, I calculate relative probability of mini flash crashes with quick reversal as:

$$P_{i,t,1.5s,Rev} = \ln\left(\frac{P_{i,t,NYSE \text{ American },1.5s,Rev}}{P_{i,t,Control,1.5s,Rev}}\right).$$

for each stock and each day.

Panel D of Table 4.12 shows the daily average of mini flash crashes with reversal of all traded stocks in NYSE American and the control group.

Table 4.12

4.5.12 Nanex (2010) mini flash crashes regression

Having successfully predicted the probability of mini flash crashes without and with strict reversal, I proceed to conduct the following two regressions using a panel data approach with fixed effects by date and stock:

$$P_{i,t,1.5s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (4.44)$$

$$P_{i,t,1.5s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \quad (4.45)$$

In these regressions, $P_{i,t,1.5s}$ represents the daily average probability of mini flash crashes without strict reversal, while $P_{i,t,1.5s,Rev}$ denotes the daily average probability of mini flash crashes with strict reversal for stock i on day t. The parameter α_i captures stock and date fixed effects, and $SpeedBump_t$ is an indicator variable taking the value of 1 after the speed bump implementation and 0 otherwise. Volatility

represents the opening value of CBOE's VIX index on day t. Additionally, the models include control variables, denoted as $ControlVariable_{i,t,q}$, which encompass two stock-level factors: the daily turnover difference and the daily stock volatility difference, calculated following the methodology of Alizadeh et al. (2002). In the linear regression without fixed effects, market capitalization of the stock is also included as an independent variable.

Table 4.20

Table 4.20 presents the results of a regression analysis conducted with and without fixed effects for both cases, with and without the reversal requirement. In these linear regressions, we estimated the coefficients of the dummy variable "speed bump", enabling us to observe the causal effect of the speed bump on the probability of mini flash crashes. Our findings indicate that the speed bump had a positive effect on $P_{i,t,1.5s}$, signifying an increase in the long-term probability of mini flash crashes. These results hold consistently for both mini flash crashes without strict reversal and those with reversals in the regressions that incorporate stock fixed effects.

In light of the empirical evidence presented by Liu (2023), the speed bump has demonstrated its ability to significantly reduce relative price discovery while simultaneously increasing the cost of immediacy and overall volatility on NYSE American. This is attributed to NYSE American becoming a slower market due to the delay of all incoming orders by 350 microseconds. Consequently, it is reasonable to assume that more uninformed noise traders have migrated to NYSE American as a result of the speed bump. This influx of noise traders has likely contributed to making NYSE American's price signals less informative and has introduced more microstructure noise into the market. As a consequence of these factors, there is an increased likelihood of unusual price fluctuations, which aligns with our findings regarding long-term mini flash crashes in the 180s and 90s.

4.6 Conclusion

This pioneering study assesses the impact of a speed bump on the probability of mini flash crashes using machine learning techniques on real-time limit order book data. Our analysis of NYSE American data shows that the implementation of a speed bump decreases the probability of identifiable patterns leading to mini flash crashes and transforms market dynamics by attracting more noise traders. Through empirical analysis, we categorize mini flash crashes as nonlinear market microstructure events influenced by high-frequency trading algorithms. While identifying specific trading behaviors triggering these events is challenging, our findings indicate that order delaying policies significantly reduce the probability of these disruptive patterns. In summary, the speed bump policy effectively mitigates the probability of mini flash crashes and reshapes market dynamics.

4.7 Appendix 5. Figure and Table of Chapter 4

Source: Levine (2015)

Figure 4.1: Flash Crash, 06 May 2010

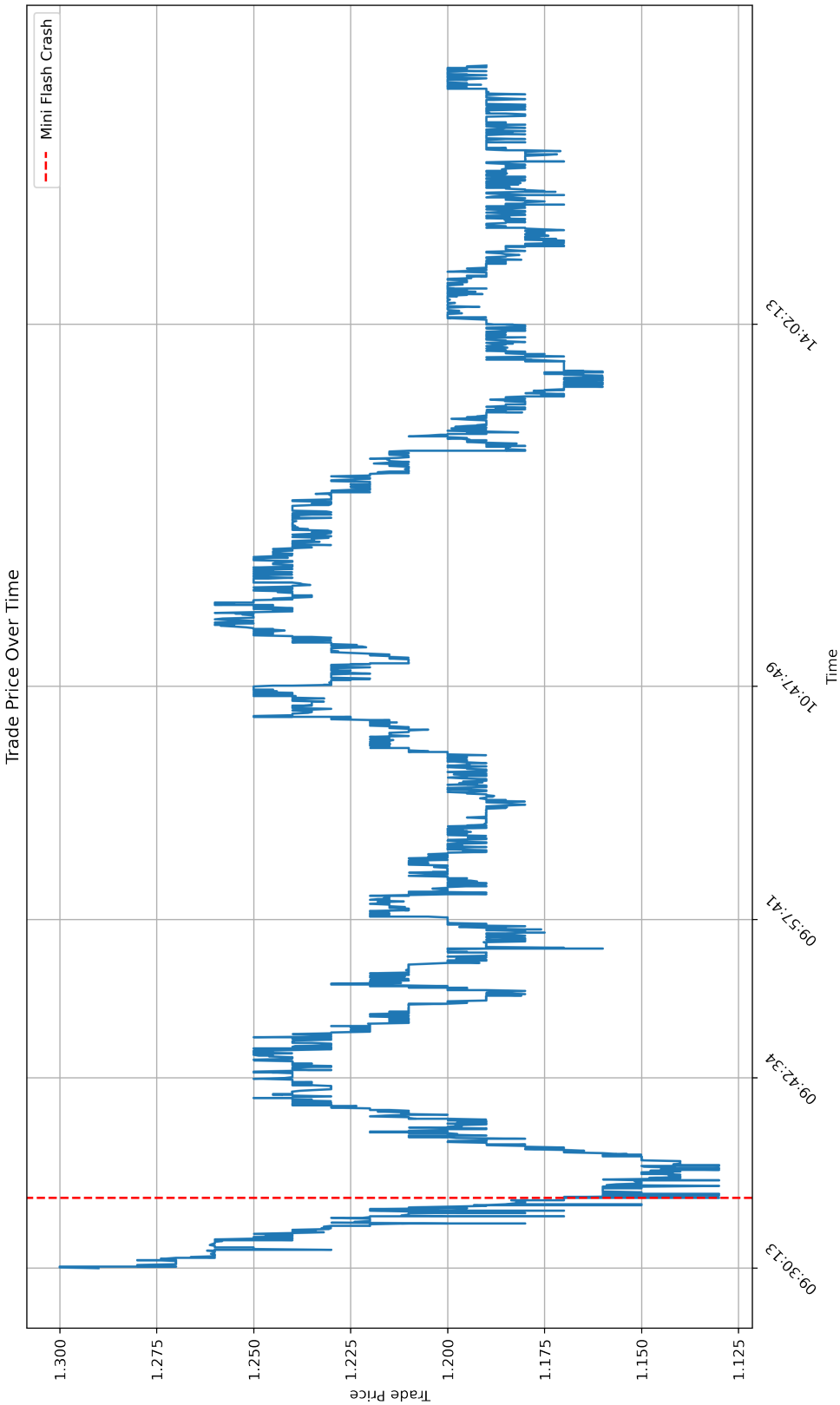


Figure 4.2: Mini Flash Crash, PLX, 03/16/2017

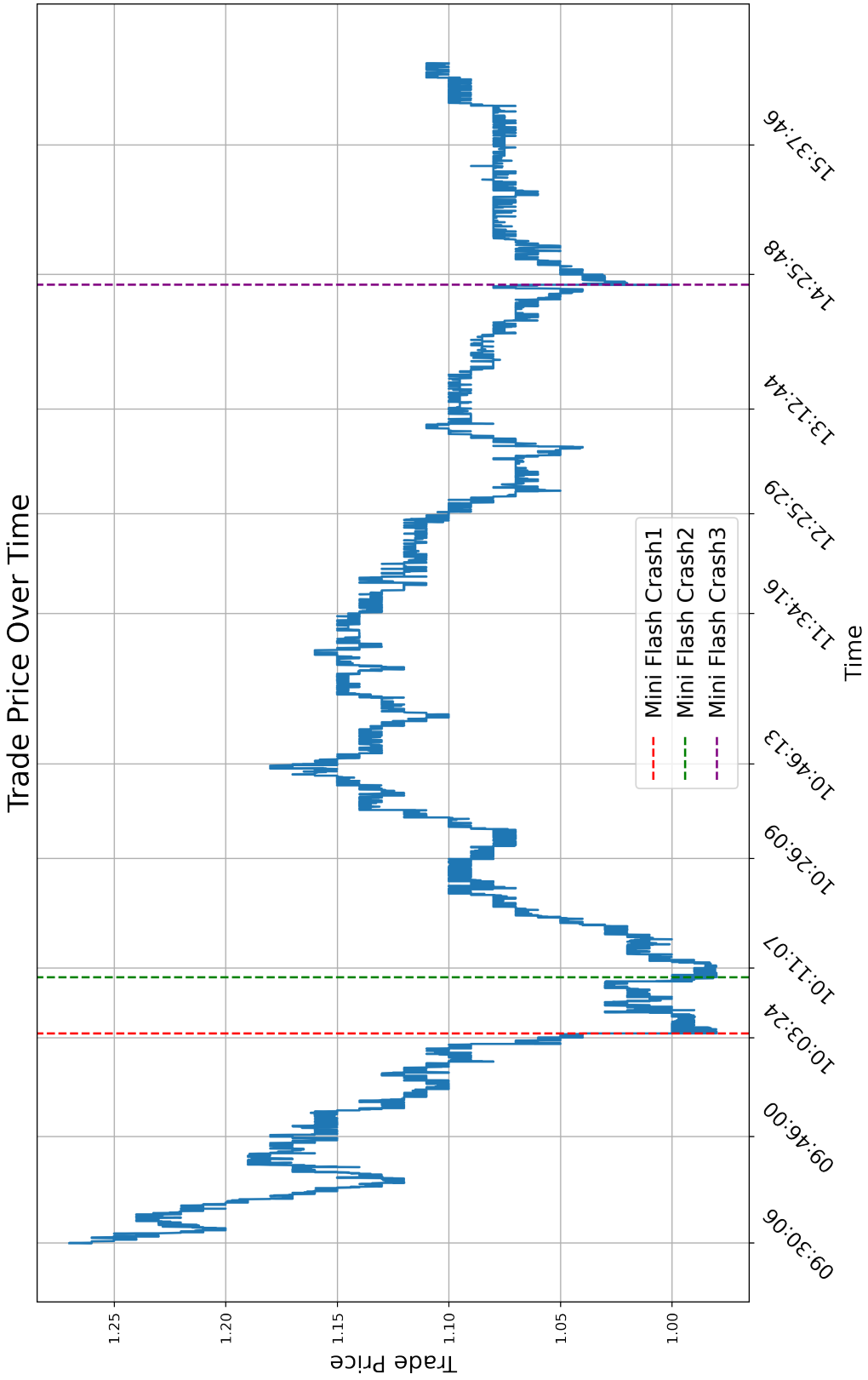


Figure 4.3: Mini Flash Crash, PLX, 04/12/2017

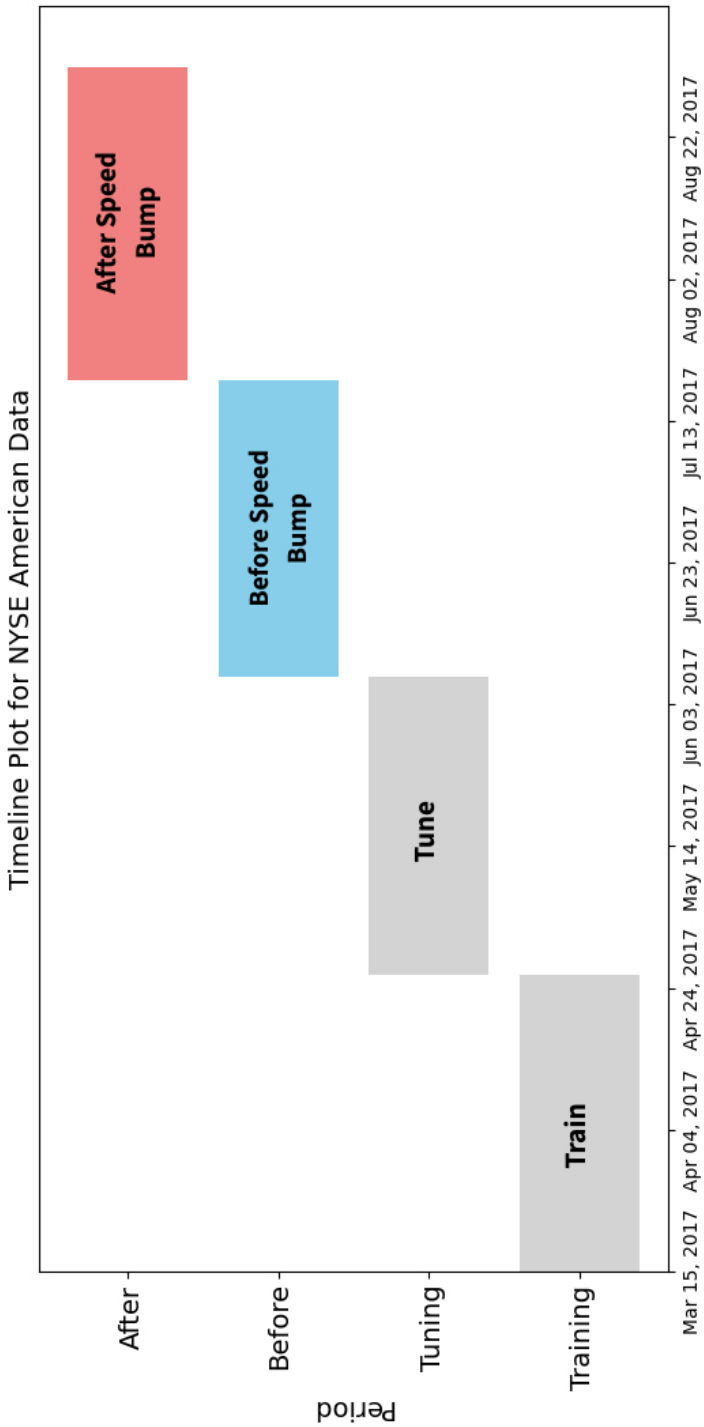


Figure 4.4: Timeline

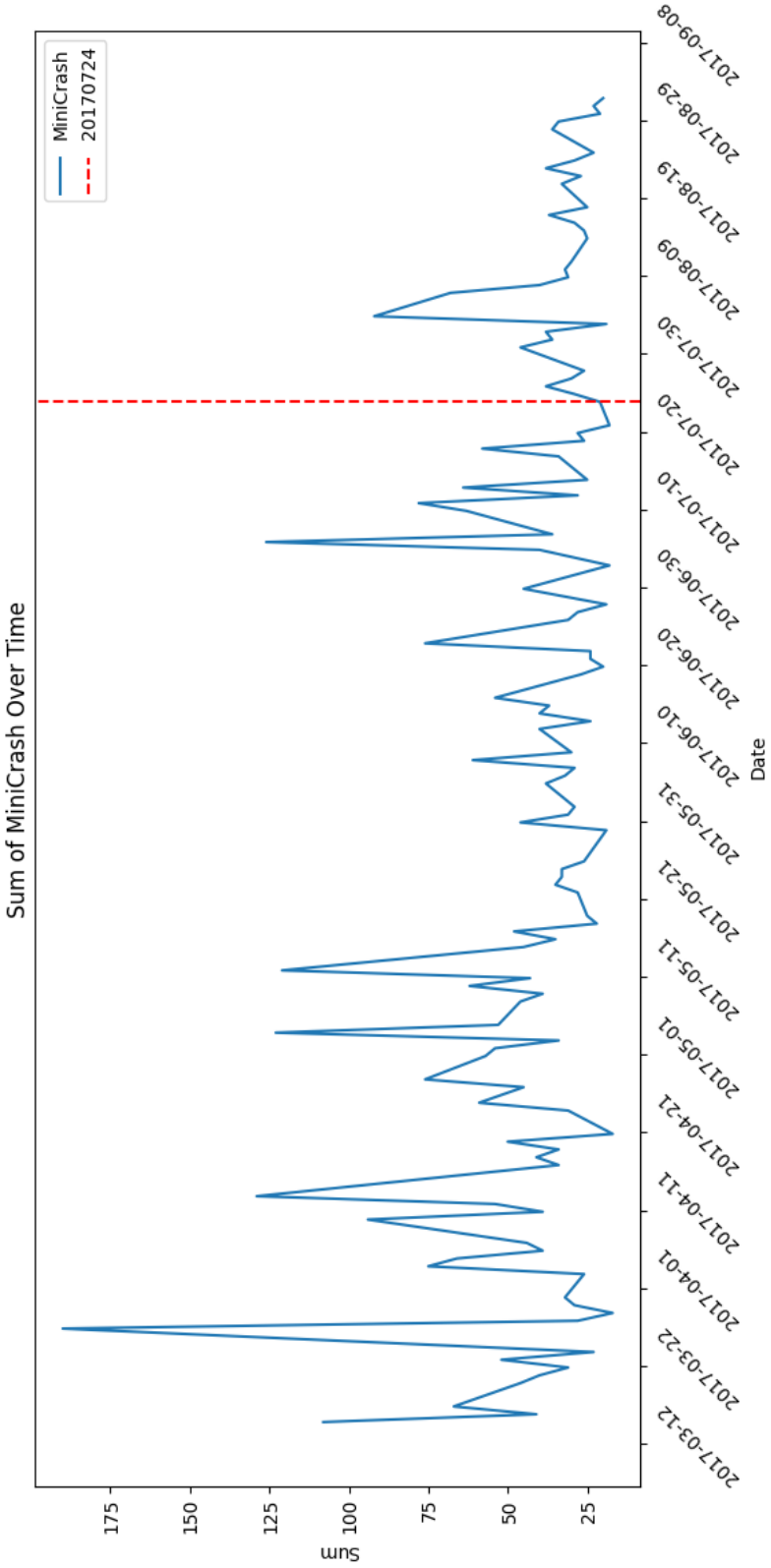


Figure 4.5: Sum of Mini Flash Crashes

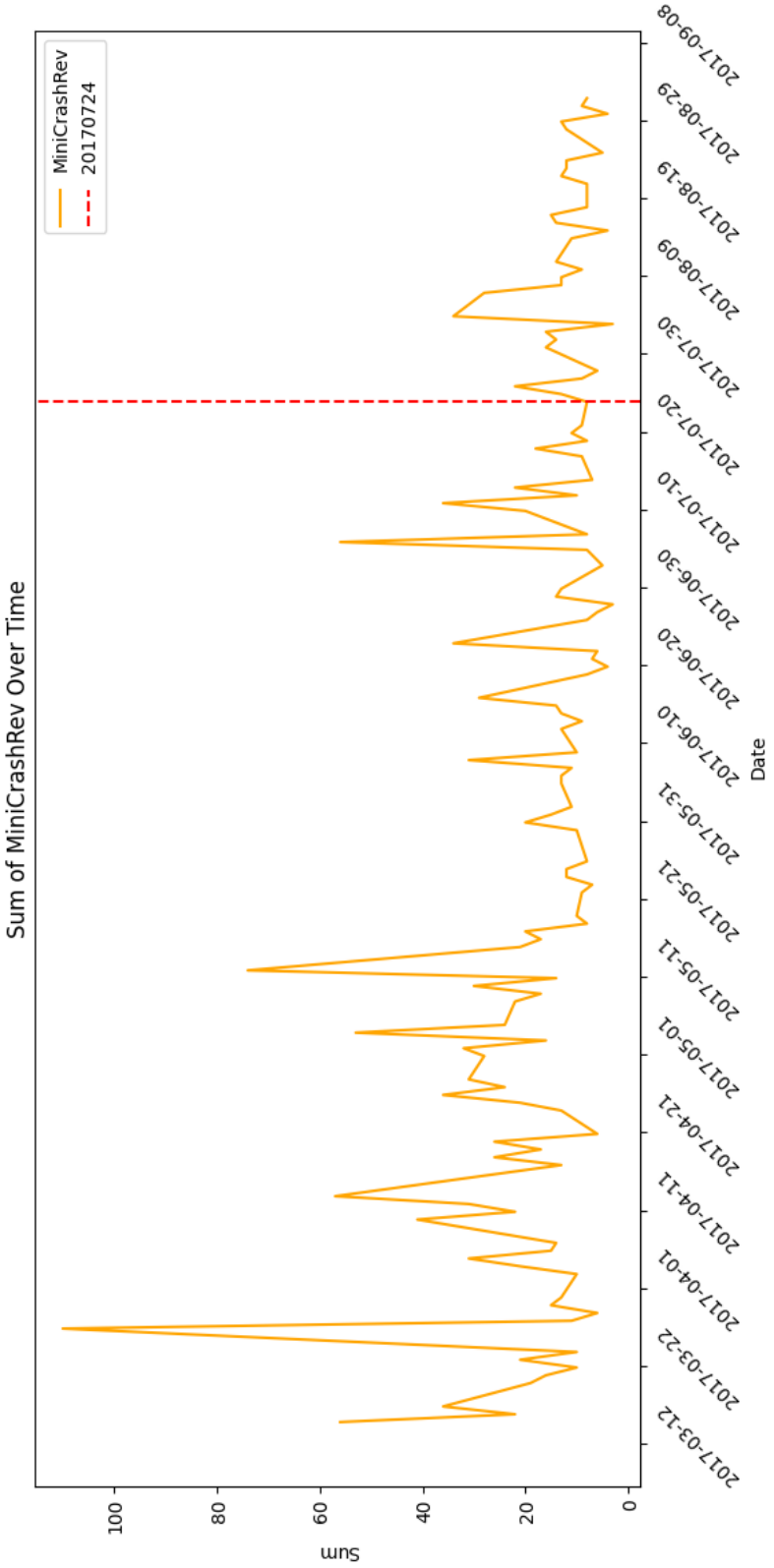
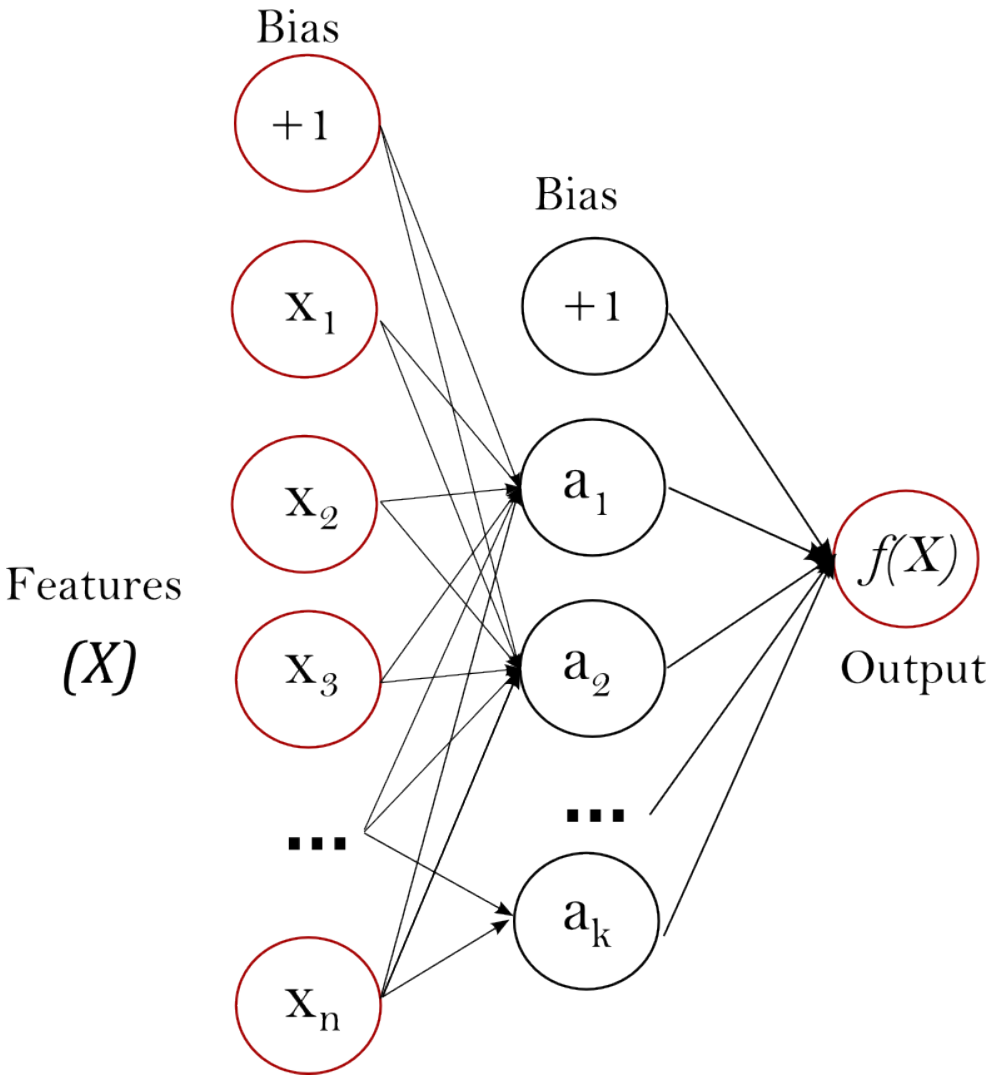
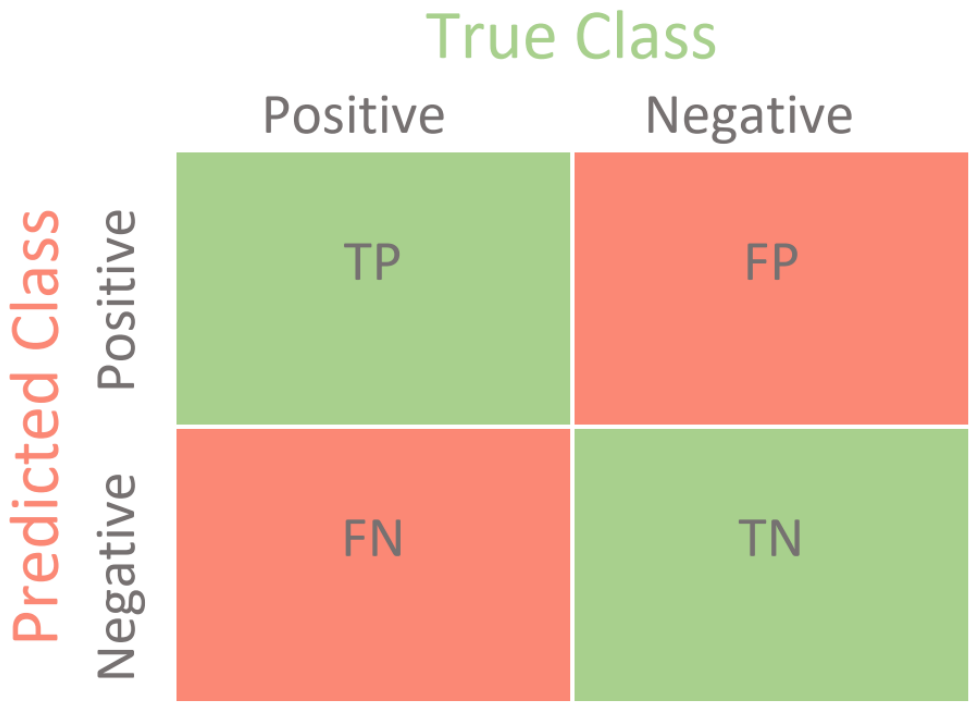


Figure 4.6: Sum of Mini Flash Crashes with reversal



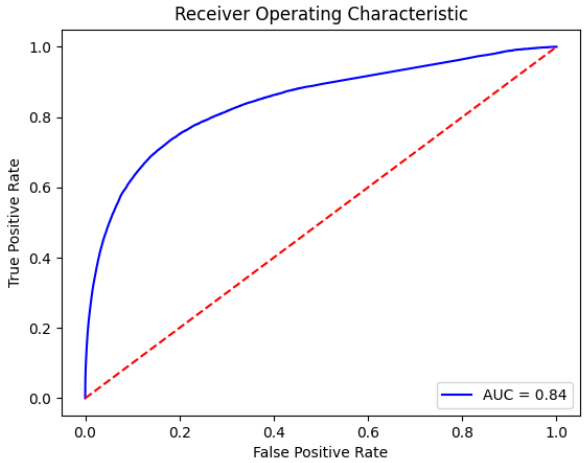
The leftmost layer, known as the input layer, consists of a set of neurons $\{x_i \mid x_1, x_2, \ldots, x_m\}$ representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation $w_1x_1 + w_2x_2 + \ldots + w_mx_m$ followed by a non-linear activation function $g(\cdot) : \mathbb{R} \to \mathbb{R}$ — like the hyperbolic tan function. The output layer receives the values from the last hidden layer and transforms them into output values. Source: Pedregosa et al. (2011)

Figure 4.7: Neural Network Model

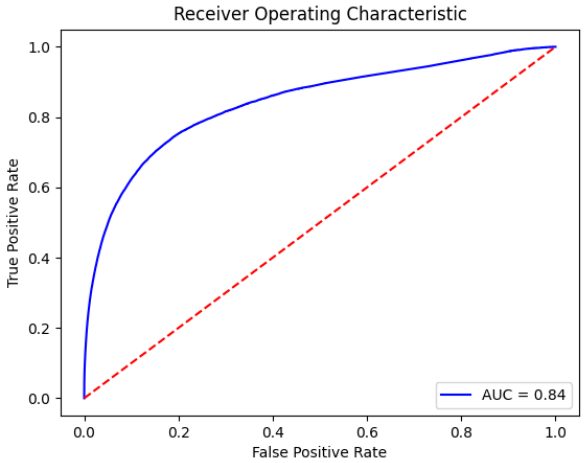


Confusion Matrix is used to evaluate the performance of a classification model. It displays the counts of true positive (TP), false positive (FP), false negative (FN), and true negative (TN) predictions. The rows represent the predicted class, while the columns represent the true class.

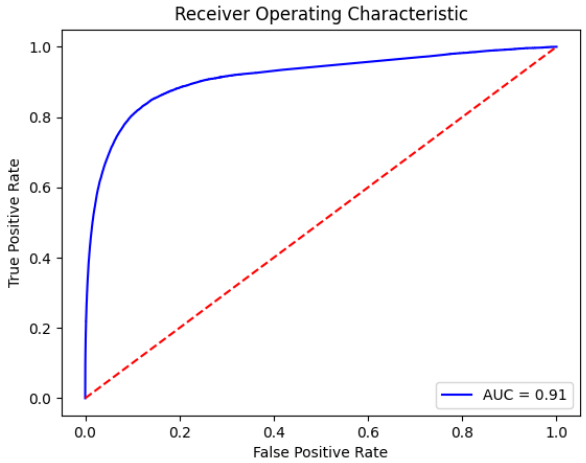
Figure 4.8: Confusion Matrix



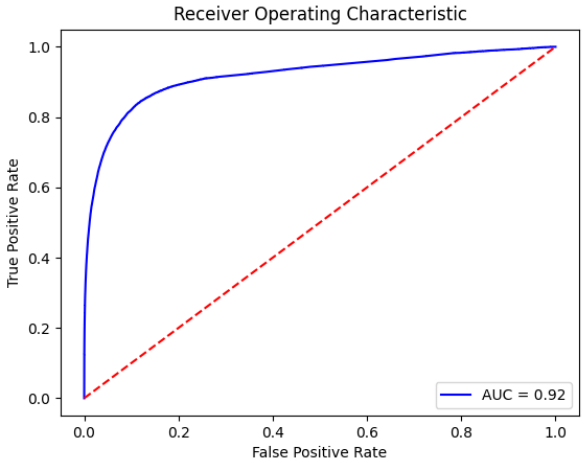
(a) Before



(b) After



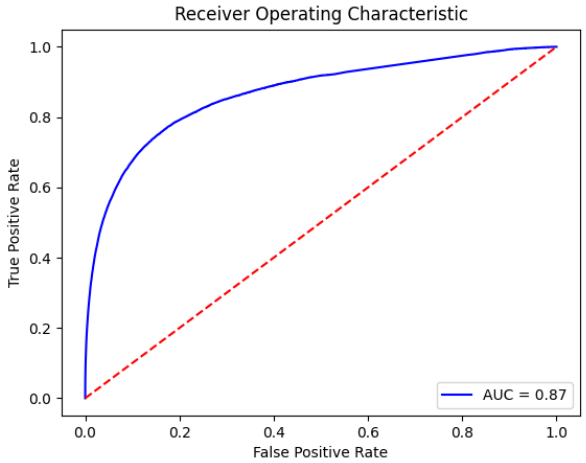
(c) Before (Control Group)



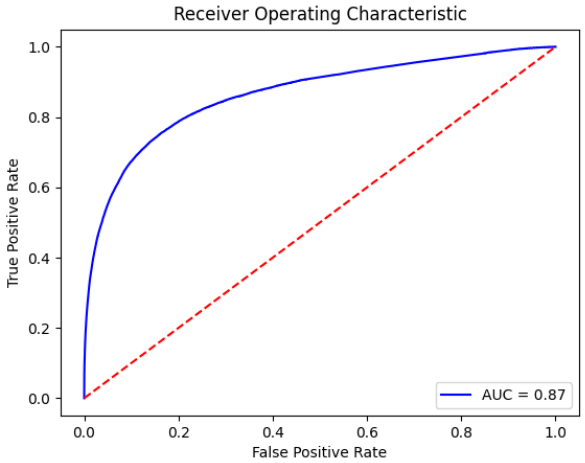
(d) After (Control Group)

The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

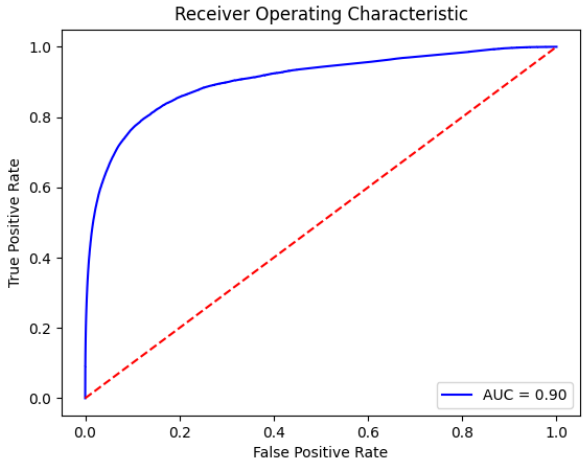
Figure 4.9: ROC curves of 180s without reversal



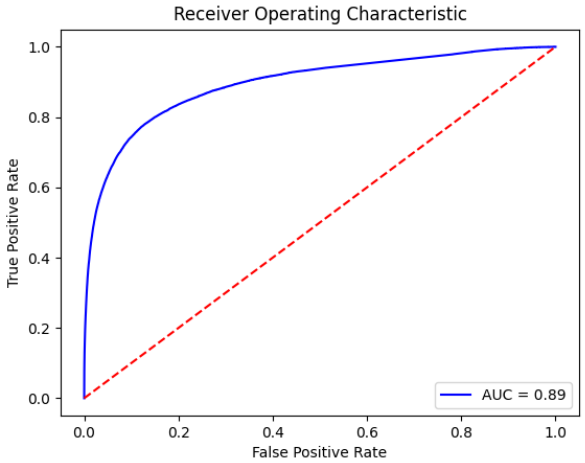
(a) Before



(b) After



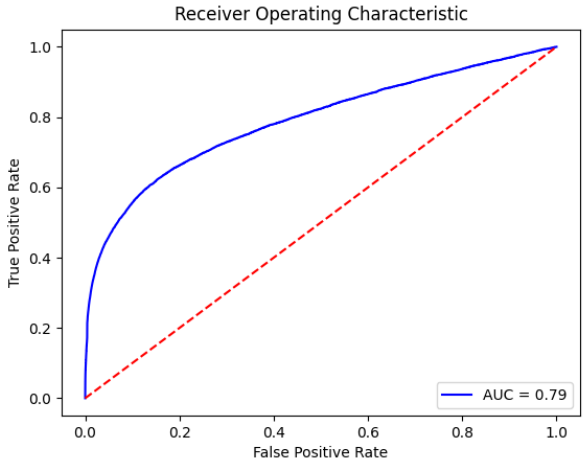
(c) Before (Control Group)



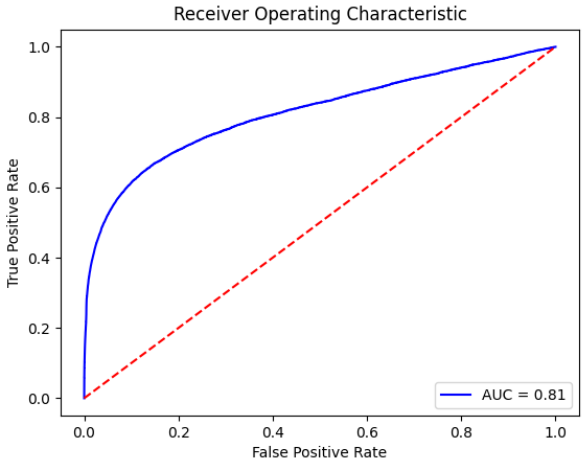
(d) After (Control Group)

The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

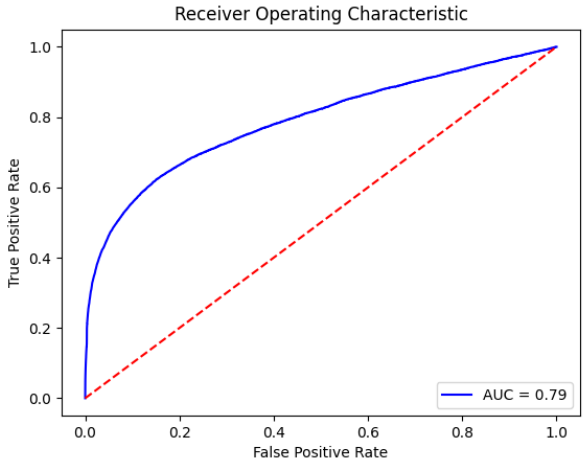
Figure 4.10: ROC curves of 90s without reversal



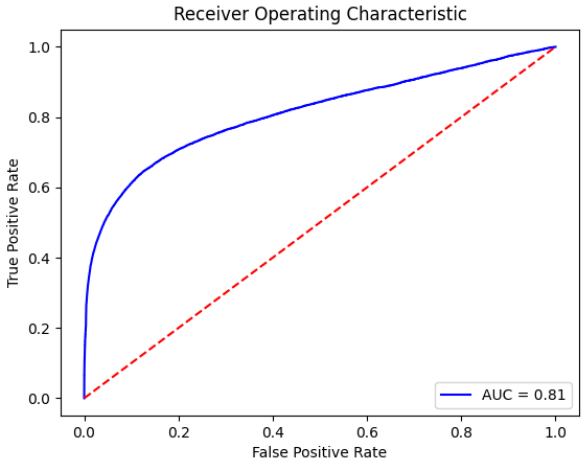
(a) Before



(b) After



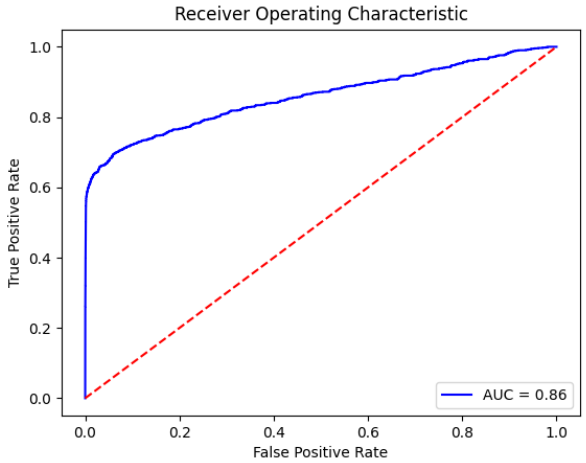
(c) Before (Control Group)



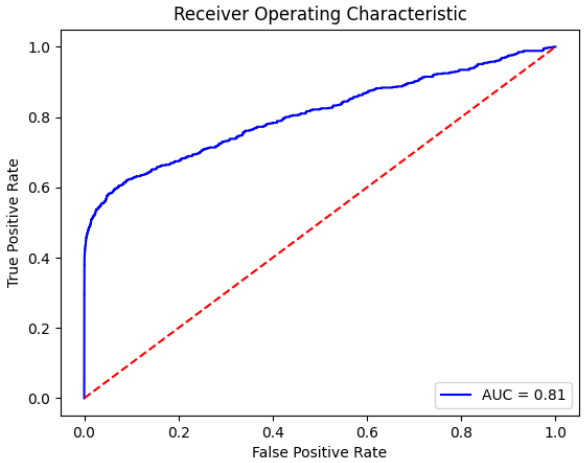
(d) After (Control Group)

The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

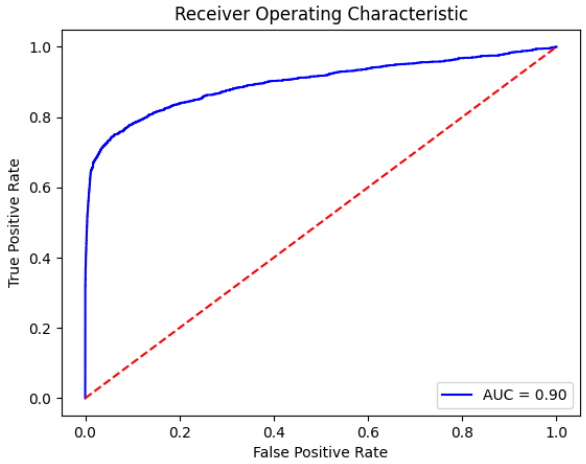
Figure 4.11: ROC curves of 15s without reversal



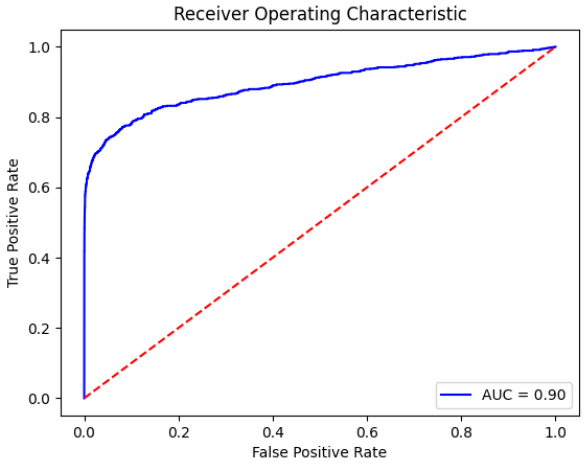
(a) Before



(b) After



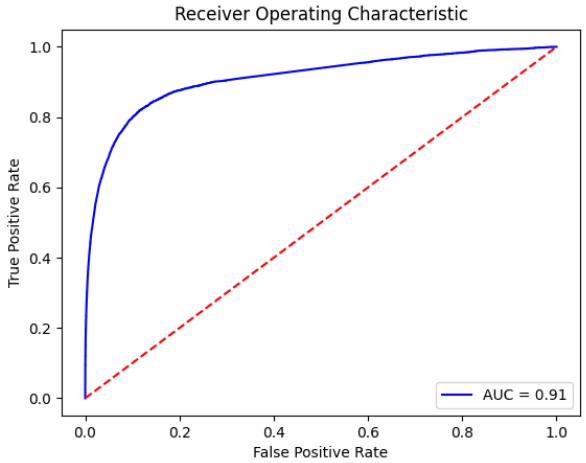
(c) Before (Control Group)



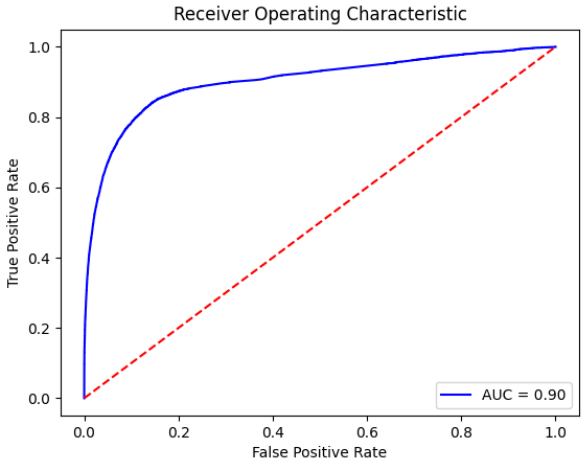
(d) After (Control Group)

The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

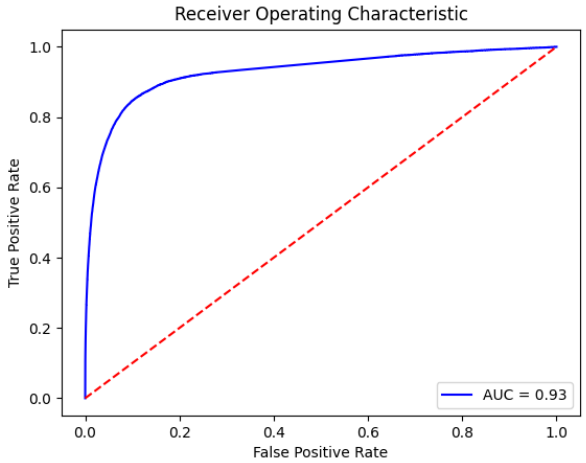
Figure 4.12: Tuning ROC curves of 1.5s without reversal



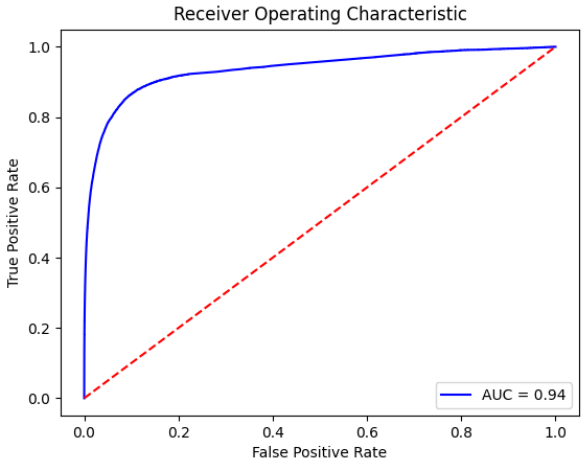
(a) Before



(b) After



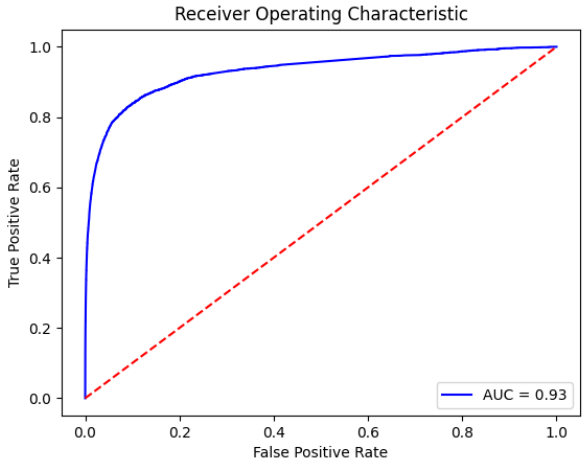
(c) Before (Control Group)



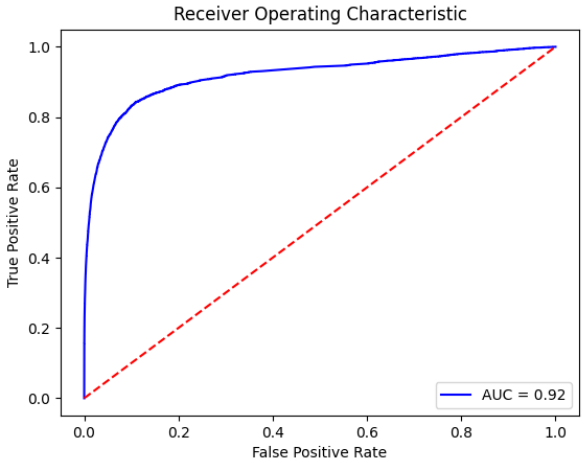
(d) After (Control Group)

The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

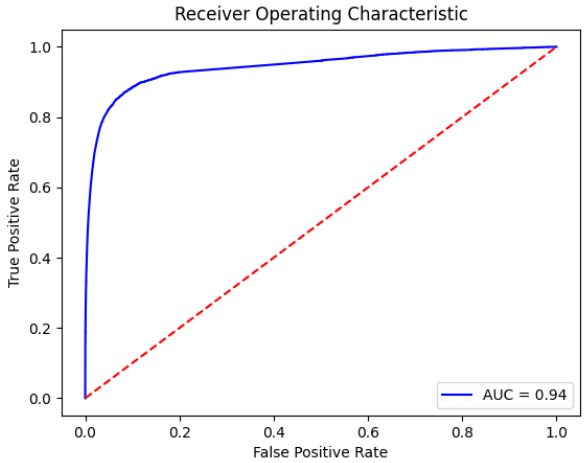
Figure 4.13: ROC curves of 180s without reversal



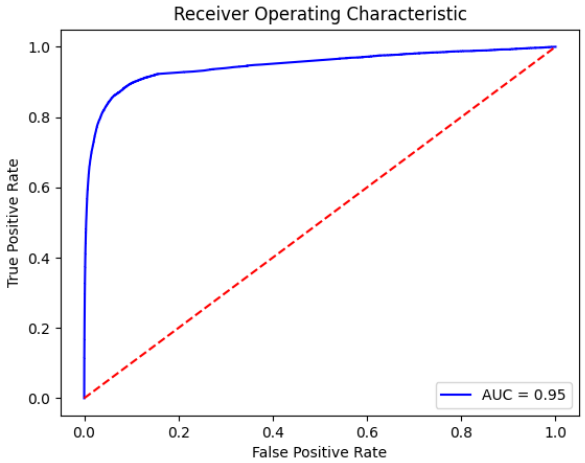
(a) Before



(b) After



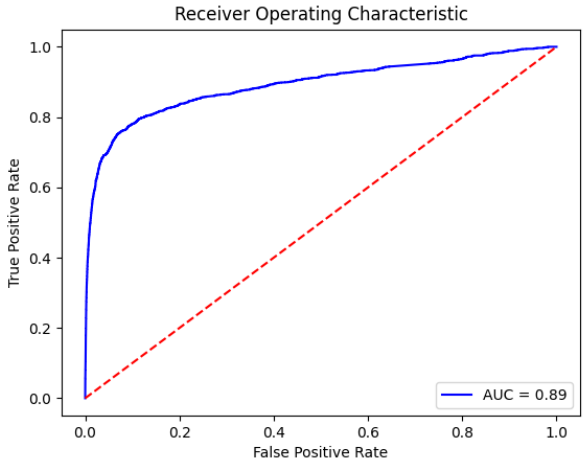
(c) Before (Control Group)



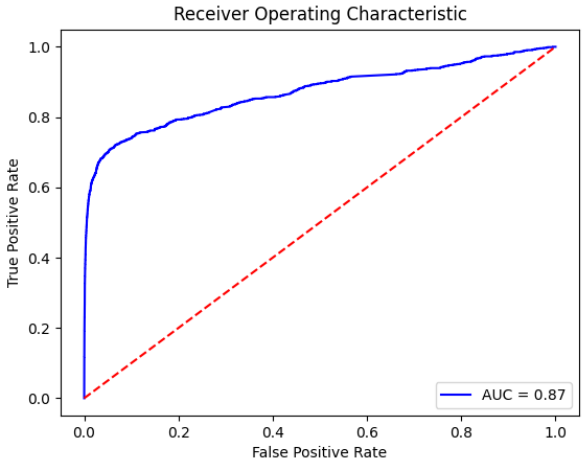
(d) After (Control Group)

The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

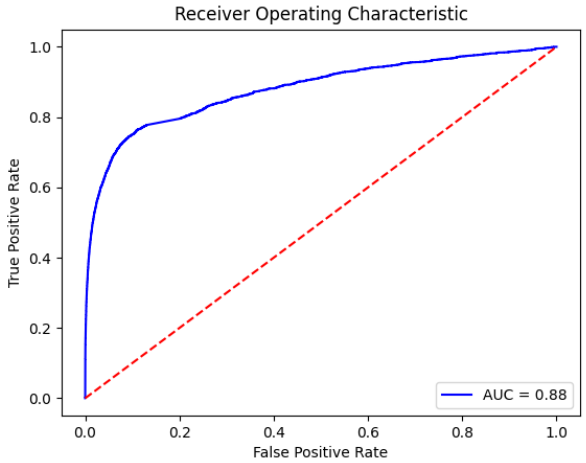
Figure 4.14: ROC curves of 180s without reversal



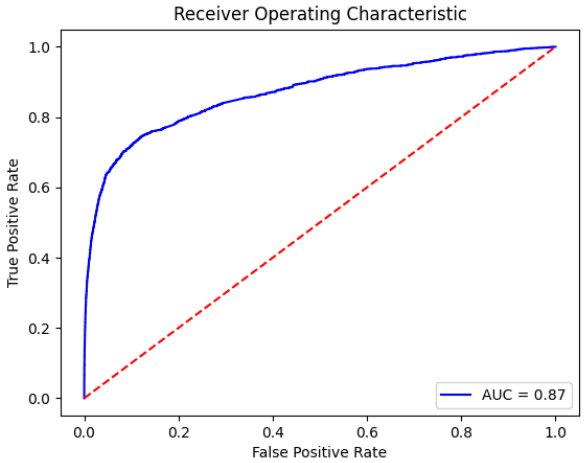
(a) Before



(b) After



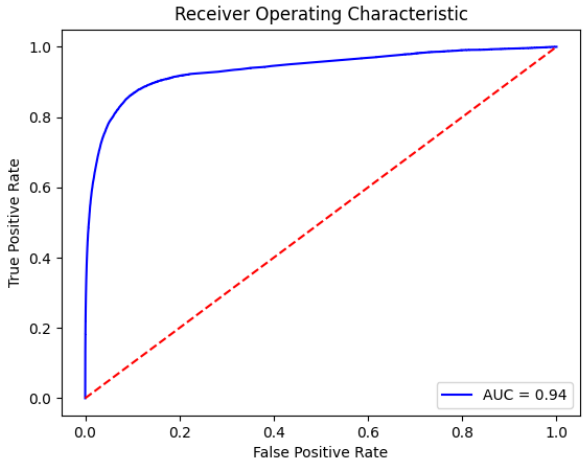
(c) Before (Control Group)



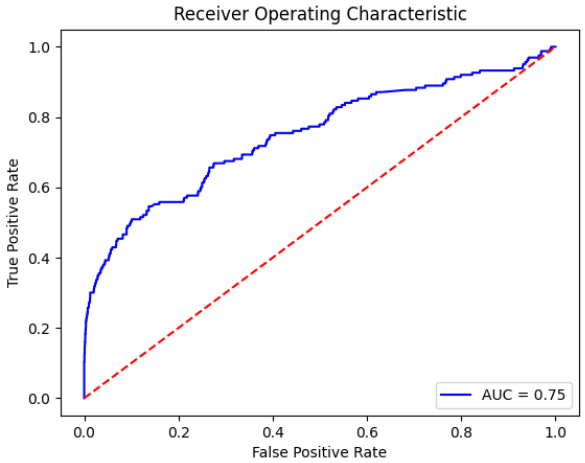
(d) After (Control Group)

The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

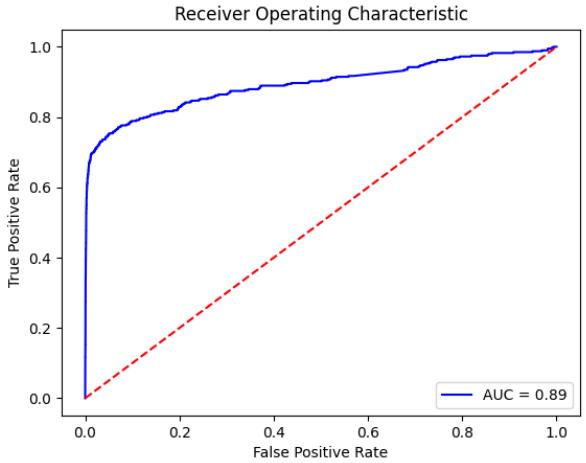
Figure 4.15: ROC curves of 180s without reversal



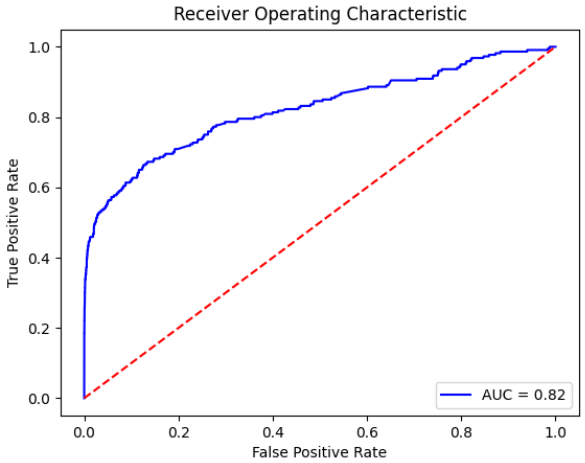
(a) Before



(b) After



(c) Before (Control Group)



(d) After (Control Group)

The blue Receiver Operating Characteristic (ROC) curve graphically represents a classification model's performance by plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at various thresholds. The Area Under the Curve (AUC) value quantifies the model's ability to discriminate between classes, with an AUC of 1 indicating a perfect model and an AUC of 0.5 indicating no predictive ability.

Figure 4.16: ROC curves of 180s without reversal

Panel A: NYSE American

	Train				Tune			
	180s	90s	15s	1.5s	180s	90s	15s	1.5s
Train								
No. Stocks:	160	142	115	87	160	142	115	87
0:	538,409	967,456	55,02,495	41,654,214	517,452	931,803	5,319,008	40,265,897
1:	23,524	20,047	3,705	1,386	23,328	19,796	3,652	1,183
Ratio:	4.37%	2.07%	$6.73 \times 10^{-2}\%$	$3.33 \times 10^{-3}\%$	4.51%	2.12%	$6.87 \times 10^{-2}\%$	$2.94 \times 10^{-3}\%$
			Test1				Test2	
No. Stocks:	160	142	115	87	160	142	115	87
0:	522,536	939,230	53,19,332	40,266,290	528,162	948,557	5,318,353	40,266,252
1:	22,497	19,181	3,328	790	20,913	17,45	2,711	828
Ratio:	4.31%	2.04%	$6.25 \times 10^{-2}\%$	$1.96 \times 10^{-3}\%$	3.95%	1.84%	$5.09 \times 10^{-2}\%$	$2.06 \times 10^{-3}\%$

Table 4.1: Sample descriptive statistics of mini flash crashes without reversal

Panel B: Control Group

			Train				Tune	
	180s	90s	15s	1.5s	180s	90s	15s	1.5s
Train								
No. Stocks:	160	142	115	87	160	142	115	87
0:	525,850	927,682	48,67,484	41,127,174	510,225	901,880	4,740,318	40,026,033
1:	16,887	29,235	18,141	1,746	15,577	27,867	17,806	1647
Ratio:	3.21%	3.15%	0.373%	4.25×10^{-3}%	3.051%	3.08%	0.376%	4.11×10^{-3}%
			Test1				Test2	
No. Stocks:	160	142	115	87	160	142	115	87
0:	507,619	893,250	4,686,171	39,244,403	489,941	859,511	4,520,810	37,504,603
1:	15,383	27,110	15,202	1,237	13,767	23,896	15,082	1,397
Ratio:	3.03%	3.03%	0.324%	3.15×10^{-3}%	2.81%	2.78%	0.334%	3.72×10^{-3}%

Table 4.2: Sample descriptive statistics of mini flash crashes without reversal

Panel A: NYSE American

			Train				Tune	
	180s	90s	15s	1.5s	180s	90s	15s	1.5s
Train								
No. Stocks:	140	107	69	45	140	107	69	45
0:	478,590	724,132	5,504,037	41,654,947	460,088	695,510	5,320,410	40,266,538
1:	6,764	4,528	2,163	653	6,600	4,426	22,50	542
Ratio:	1.41%	0.625%	0.039%	1.57×10^{-3}%	1.43%	0.636%	0.042%	1.35×10^{-3}%
			Test1				Test2	
No. Stocks:	140	107	69	45	140	107	69	45
0:	465,068	701,475	5,320,875	40,266,817	469,750	708,836	5,319,605	40,266,773
1:	6,039	3,724	1,785	263	5,082	3,135	1,459	307
Ratio:	1.30%	0.531%	0.034%	6.53×10^{-4}%	1.08%	0.442%	0.027%	7.62×10^{-4}%

Table 4.3: Sample descriptive statistics of mini flash crashes with reversal

Panel B: Control Group

			Train				Tune	
	180s	90s	15s	1.5s	180s	90s	15s	1.5s
Train								
No. Stocks:	140	107	69	45	140	107	69	45
0:	462,724	704,720	3,246,135	21,034,603	451,606	690,915	3,169,370	20,619,802
1:	11,541	7,517	4,917	677	10,663	6,276	3,478	518
Ratio:	2.49%	1.07%	0.151%	3.21×10^{-3}%	2.36%	0.908%	0.110%	2.51×10^{-3}%
			Test1				Test2	
No. Stocks:	140	107	69	45	140	107	69	45
0:	447,121	685,425	3,157,324	20,635,919	433,563	665,982	30,43,622	20,284,569
1:	11,073	6,009	2,756	361	9,513	5,231	3,142	591
Ratio:	2.48%	0.877%	0.087%	1.75×10^{-3}%	2.19%	0.785%	0.103%	2.91×10^{-3}%

Table 4.4: Sample descriptive statistics of mini flash crashes with reversal

Panel A: 180s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.729	0.730	0.732	0.727	0.731	0.739	0.731	0.728
SVM	0.712	0.718	0.718	0.709	0.715	0.717	0.721	0.714
RF	0.545	0.767	0.597	0.633	0.551	0.770	0.602	0.658
XGB	0.762	0.735	0.841	0.569	0.761	0.769	0.842	0.523
NN	0.578	0.752	0.763	0.632	0.580	0.751	0.760	0.685

Panel B: 90s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.751	0.749	0.755	0.749	0.754	0.752	0.754	0.752
SVM	0.732	0.735	0.740	0.729	0.738	0.744	0.743	0.736
RF	0.535	0.789	0.570	0.602	0.544	0.790	0.582	0.635
XGB	0.779	0.792	0.872	0.549	0.775	0.789	0.873	0.598
NN	0.556	0.776	0.782	0.589	0.569	0.769	0.780	0.623

Panel C: 15s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.691	0.691	0.692	0.688	0.720	0.719	0.721	0.718
SVM	0.686	0.688	0.690	0.684	0.716	0.716	0.718	0.714
RF	0.573	0.726	0.612	0.648	0.600	0.749	0.636	0.664
XGB	0.716	0.720	0.793	0.593	0.737	0.745	0.812	0.533
NN	0.583	0.685	0.707	0.623	0.611	0.696	0.737	0.653

Panel D: 1.5s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.788	0.788	0.789	0.788	0.701	0.719	0.713	0.700
SVM	0.785	0.788	0.788	0.787	0.709	0.715	0.713	0.699
RF	0.723	0.814	0.753	0.792	0.642	0.755	0.688	0.714
XGB	0.785	0.811	0.862	0.782	0.705	0.746	0.813	0.670
NN	0.753	0.771	0.791	0.762	0.690	0.708	0.740	0.706

Notes: This table reports the AUC obtained through a tuning procedure, using both training and tuning data to identify the optimal parameter combination and imbalanced data processing strategy. They are prepared for calculating the probability of mini flash crashes in the period of 30 trading days before and after the implementation of a speed bump on July 24th, 2017.

Table 4.5: Mini Flash Crash without reversal tuning result: NYSE American

Panel A: 180s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.815	0.814	0.816	0.814	0.730	0.730	0.732	0.727
SVM	0.802	0.808	0.808	0.799	0.712	0.718	0.718	0.709
RF	0.585	0.853	0.656	0.709	0.545	0.767	0.598	0.637
XGB	0.833	0.846	0.910	0.635	0.762	0.769	0.921	0.562
NN	0.646	0.824	0.838	0.695	0.653	0.836	0.847	0.735

Panel B: 90s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.782	0.780	0.783	0.782	0.791	0.789	0.792	0.791
SVM	0.762	0.770	0.773	0.760	0.774	0.776	0.780	0.771
RF	0.569	0.824	0.631	0.686	0.610	0.831	0.671	0.725
XGB	0.812	0.820	0.900	0.789	0.820	0.832	0.892	0.789
NN	0.613	0.799	0.815	0.698	0.622	0.810	0.821	0.721

Panel C: 15s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.691	0.691	0.692	0.688	0.720	0.719	0.721	0.718
SVM	0.686	0.688	0.688	0.684	0.716	0.716	0.718	0.715
RF	0.572	0.728	0.613	0.648	0.601	0.749	0.636	0.666
XGB	0.716	0.720	0.792	0.593	0.737	0.744	0.811	0.533
NN	0.583	0.685	0.707	0.623	0.611	0.696	0.733	0.653

Panel D: 1.5s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.823	0.823	0.821	0.820	0.824	0.800	0.822	0.815
SVM	0.826	0.821	0.825	0.822	0.817	0.804	0.818	0.809
RF	0.763	0.838	0.799	0.829	0.632	0.838	0.701	0.767
XGB	0.822	0.823	0.905	0.813	0.792	0.826	0.902	0.809
NN	0.792	0.803	0.801	0.806	0.777	0.781	0.817	0.764

Notes: This table reports the AUC obtained through a tuning procedure, using both training and tuning data to identify the optimal parameter combination and imbalanced data processing strategy. They are prepared for calculating the probability of mini flash crashes in the period of 30 trading days before and after the implementation of a speed bump on July 24th, 2017.

Table 4.6: Mini Flash Crash without reversal tuning result: Control Group

Panel A: 180s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.822	0.822	0.822	0.822	0.806	0.806	0.805	0.806
SVM	0.813	0.817	0.816	0.812	0.796	0.803	0.803	0.796
RF	0.553	0.849	0.602	0.651	0.559	0.842	0.599	0.656
XGB	0.808	0.844	0.912	0.588	0.786	0.838	0.903	0.582
NN	0.587	0.823	0.836	0.660	0.584	0.820	0.824	0.689

Panel B: 90s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.844	0.841	0.842	0.843	0.830	0.827	0.832	0.828
SVM	0.835	0.836	0.836	0.834	0.824	0.828	0.830	0.822
RF	0.557	0.866	0.600	0.605	0.568	0.859	0.606	0.622
XGB	0.813	0.859	0.929	0.592	0.779	0.855	0.922	0.578
NN	0.584	0.834	0.852	0.655	0.611	0.827	0.839	0.664

Panel C: 15s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.814	0.806	0.819	0.810	0.816	0.815	0.815	0.815
SVM	0.813	0.810	0.824	0.812	0.814	0.812	0.813	0.813
RF	0.701	0.816	0.764	0.786	0.706	0.825	0.741	0.681
XGB	0.787	0.779	0.889	0.759	0.775	0.816	0.872	0.651
NN	0.709	0.737	0.801	0.705	0.722	0.807	0.812	0.709

Panel D: 1.5s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.752	0.745	0.763	0.755	0.661	0.679	0.666	0.661
SVM	0.752	0.748	0.767	0.753	0.662	0.670	0.656	0.645
RF	0.792	0.900	0.791	0.790	0.534	0.692	0.562	0.580
XGB	0.791	0.884	0.782	0.773	0.578	0.671	0.753	0.603
NN	0.884	0.835	0.849	0.940	0.580	0.664	0.671	0.576

Notes: This table reports the AUC obtained through a tuning procedure, using both training and tuning data to identify the optimal parameter combination and imbalanced data processing strategy. They are prepared for calculating the probability of mini flash crashes in the period of 30 trading days before and after the implementation of a speed bump on July 24th, 2017.

Table 4.7: Mini Flash Crashes with reversal tuning results: NYSE American

Panel A: 180s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.830	0.83	0.831	0.829	0.840	0.838	0.842	0.838
SVM	0.819	0.823	0.825	0.817	0.827	0.832	0.835	0.824
RF	0.572	0.871	0.638	0.691	0.635	0.879	0.704	0.764
XGB	0.835	0.869	0.932	0.620	0.849	0.876	0.941	0.596
NN	0.611	0.845	0.858	0.708	0.663	0.860	0.868	0.745

Panel B: 90s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.863	0.860	0.865	0.862	0.871	0.867	0.871	0.867
SVM	0.855	0.857	0.860	0.852	0.860	0.864	0.868	0.857
RF	0.577	0.941	0.623	0.670	0.589	0.897	0.670	0.709
XGB	0.837	0.931	0.940	0.630	0.852	0.892	0.948	0.651
NN	0.614	0.864	0.875	0.691	0.606	0.870	0.886	0.686

Panel C: 15s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.816	0.815	0.817	0.806	0.786	0.773	0.782	0.781
SVM	0.813	0.810	0.814	0.804	0.778	0.774	0.783	0.774
RF	0.610	0.827	0.649	0.703	0.579	0.802	0.641	0.666
XGB	0.715	0.818	0.882	0.681	0.740	0.798	0.873	0.711
NN	0.676	0.780	0.806	0.674	0.646	0.729	0.771	0.626

Panel D: 1.5s

	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.886	0.882	0.886	0.885	0.710	0.692	0.729	0.706
SVM	0.884	0.886	0.891	0.882	0.711	0.675	0.709	0.706
RF	0.891	0.856	0.891	0.891	0.620	0.757	0.645	0.651
XGB	0.891	0.842	0.892	0.886	0.669	0.739	0.822	0.652
NN	0.743	0.833	0.844	0.785	0.670	0.713	0.738	0.662

Notes: This table reports the AUC obtained through a tuning procedure, using both training and tuning data to identify the optimal parameter combination and imbalanced data processing strategy. They are prepared for calculating the probability of mini flash crashes in the period of 30 trading days before and after the implementation of a speed bump on July 24th, 2017.

Table 4.8: Mini Flash Crashes with reversal tuning results: Control Group

Panel A: 180s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	2.003	1.636	0.367	3.723	3.353	0.370
std	6.578	6.302	0.276	12.901	11.930	0.971
N	6,170	6,168	2	6,170	6,168	2

Panel B: 90s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	1.434	1.148	0.287	3.333	2.888	0.446
std	6.383	6.394	-0.011	16.832	14.603	2.229
N	6,170	6,168	2	6,170	6,168	2

Panel C: 15s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	0.566	0.444	0.123	1.189	1.681	0.213
std	4.845	4.472	0.102	21.590	15.474	6.116
N	6,170	6,168	2	6,170	6,168	2

Panel D: 1.5s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	0.184	0.151	0.032	0.569	0.456	0.113
std	1.505	0.895	0.610	10.545	5.463	5.082
N	6,170	6,168	2	6,170	6,168	2

Notes: This table presents data on the occurrence of mini flash crashes without reversal for all 225 stocks traded on NYSE American, along with a control group. The information includes the standard deviation and the number of observations recorded between June 9th, 2017, and September 5th, 2017—coinciding with the implementation of the NYSE American speed bump on July 24th, 2017. The table also reports the difference between before and after speed bump.

Table 4.9: Number of mini flashes crashes without reversal

Panel A: 180s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	1.388	1.106	0.282	2.947	2.646	0.301
std	5.326	5.252	0.074	11.783	10.884	0.899
N	6,170	6,168	2	6,170	6,168	2

Panel B: 90s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	0.927	0.711	0.216	2.587	2.215	0.372
std	5.032	5.361	-0.329	15.300	12.928	2.373
N	6,170	6,168	2	6,170	6,168	2

Panel C: 15s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	0.300	0.241	0.059	1.272	1.124	0.148
std	3.372	3.560	-0.188	17.555	12.111	5.444
N	6,170	6,168	2	6,170	6,168	2

Panel D: 1.5s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	0.065	0.056	0.009	0.260	0.205	0.055
std	0.713	5.571	-4.859	0.434	3.206	-2.272
N	6,170	6,168	2	6,170	6,168	2

Notes: This table presents data on the occurrence of mini flash crashes without reversal for all 225 stocks traded on NYSE American, along with a control group. The information includes the standard deviation and the number of observations recorded between June 9th, 2017, and September 5th, 2017—coinciding with the implementation of the NYSE American speed bump on July 24th, 2017. The table also reports the difference between before and after speed bump.

Table 4.10: Number of mini flash crashes with reversal

Panel A: 180s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	32.6 %	31.9%	-0.7%	22.6%	21.9%	-0.7%
std	16.2%	15.1%	1.1%	19.0%	18.3%	0.8%
N	4,486	4,334	152	4,486	4,334	152

Panel B: 90s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	28.8%	36.0%	-7.2%	25.6%	21.7%	3.9%
std	16.1%	16.0%	0.1%	18.7%	18.6%	0.03%
N	3,988	3,846	142	3,988	3,846	142

Panel C: 15s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	28.9%	27.4%	1.5%	23.9%	26.1%	-2.2%
std	5.4%	7.6%	-2.2%	7.0%	7.0%	0.1%
N	3,284	3,147	101	3,284	3,147	101

Panel D: 1.5s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	16.2%	15.7%	0.5%	16.5%	17.2%	-0.6%
std	4.6%	4.7%	-0.06%	6.9%	7.9%	-1%
N	2,439	2,343	96	2,439	2,343	96

Notes: Notes: This table reports the average values of daily average probablity of mini flash crashes without reversal predicted by machine learning model, their standard deviation and number of observations from June 9th 2017 to September 5th 2017, which are around NYSE American speed bump implementation. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump.

Table 4.11: Probability of mini flashes crashes without reversal

Panel A: 180s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	20.9%	20.9%	0.06%	19.7%	18.4%	1.36%
std	16.3%	14.8%	1.49%	18.9%	18.0%	0.9%
N	3,958	3,849	109	3,958	3,849	109

Panel B: 90s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	15.8%	15.7%	0.13%	14.7%	12.8%	1.87 %
std	13.9%	13.0%	0.9%	15.5%	14.5%	1.05%
N	3,017	2,948	69	3,017	2,948	69

Panel C: 15s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	10.7%	11.2%	-0.053%	12.7%	15.8%	-3.1%
std	5.6%	7.8%	-2.2%	6.4%	7.9%	-1.5%
N	1,960	1,902	58	1,960	1,902	58

Panel D: 1.5s

	NYSE American			Control		
	Before speed bump	After speed bump	Diff= Before-After	Before speed bump	After speed bump	Diff= Before-After
Mean	17.0%	20.0%	-3%	13.2%	16.5%	-3.4%
std	8.7%	8.8%	-0.18%	8.8%	9.6%	-0.78%
N	1,274	1,264	10	1,274	1,264	10

Notes: This table reports the average values of daily average probability of mini flash crashes with reversal predicted by machine learning model, their standard deviation and number of observations from June 9th 2017 to September 5th 2017, which are around NYSE American speed bump implementation. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump.

Table 4.12: Probability of mini flash crashes with reversal

Regression results (180 seconds)

	Number of mini flash crashes without reversal			Number of mini flash crashes with reversal		
Speedbump	0.0381	0.0372	-0.1542	-0.0068	-0.0078	-0.1517
	(0.1978)	(0.3052)	(0.2089)	(0.1709)	(0.2569)	(0.1768)
VIX	0.0696	0.0694	0.0903	0.0848	0.0846	0.1001
	(0.7155)	(0.0671)	(0.0755)	(0.0618)	(0.0593)	(0.0639)
Turnover	2.3235***	2.3435***		1.7726***	1.7943***	
	(0.0482)	(0.3683)		(0.0417)	(0.3223)	
StockVolatility	0.1851***	0.1655		0.1314***	0.1100	
	(0.0270)	(0.2280)		(0.0233)	(0.1948)	
MarketCap		0.2288			0.2491	
		(1.0676)			(0.9557)	
Stock F.E.	Yes	No	No	Yes	No	No
N	11,828	11,828	12,338	11,828	11,828	12,338
Adjust R-square	0%	0%	0%	0%	0%	0%

Notes: This table presents the results of a linear regression analysis examining the difference in the number of 180-second mini flash crashes between NYSE American and its control group. The analysis incorporates observations of standard deviation and the number of occurrences recorded between June 9th, 2017, and September 5th, 2017—corresponding to the period surrounding the implementation of the NYSE American speed bump. All of them are in basis points. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 4.13: Panel regression of number of mini flash crashes

Regression results (180 seconds)

	Probability of mini flash crashes without reversal			Probability of mini flash crashes with reversal		
Speedbump	0.0161***	0.0152	-0.0004	0.1075***	0.1165**	0.0960***
	(0.0058)	(0.0114)	(0.0142)	(0.0083)	(0.0135)	(0.0173)
VIX	0.0127***	0.0120***	0.0144***	0.0104***	0.0091***	0.0138**
	(0.0021)	(0.0041)	(0.0051)	(0.0030)	(0.0049)	(0.0062)
Turnover	0.1610***	0.2000***		0.2061***	0.2381***	
	(0.0025)	(0.0028)		(0.0038)	(0.0034)	
StockVolatility	0.0007	0.0057***		0.0014	0.0173***	
	(0.0014)	(0.0016)		(0.0020)	(0.0019)	
MarketCap		0.0472***			0.0821***	
		(0.0050)			(0.0058)	
Stock F.E.	Yes	No	No	Yes	No	No
N	8696	8696	8820	7709	7709	7807
Adjust R-square	35.89%	36.78%	0.08%	37.14%	39.98%	0.61%

Notes: This table reports the linear regression result of the probability of 180 seconds mini flash crashes through the observations from their standard deviation and number of observations from June 9th 2017 to September 5th 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. All of them are in basis points. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 4.14: Panel regression of probability of mini flash crashes

Regression results (90 seconds)

	Number of mini flash crashes without reversal			Number of mini flash crashes with reversal		
Speedbump	0.0501	0.0496	-0.1041	-0.0025	-0.0030	-0.1092
	(0.1821)	(0.2843)	(0.1877)	(0.1455)	(0.2207)	(0.1477)
VIX	0.0835	0.0834	0.0997	0.0715	0.0714	0.0824
	(0.0659)	(0.0608)	(0.0678)	(0.0526)	(0.0487)	(0.0534)
Turnover	1.8330***	1.8429***		1.2953***	1.3078***	
	(0.0444)	(0.3443)		(0.0355)	(0.2752)	
StockVolatility	0.1197***	0.1010		0.0733***	0.0610	
	(0.0249)	(0.2125)		(0.0199)	(0.1674)	
MarketCap		0.1128			0.1437	
		(1.0656)			(0.8670)	
Stock F.E.	Yes	No	No	Yes	No	No
N	11,828	11,828	12,338	11,828	11,828	12,338
Adjust R-square	12.96%	13%	0%	10.38%	10.45%	0%

Notes: This table presents the results of a linear regression analysis examining the difference in the number of 90-second mini flash crashes between NYSE American and its control group. The analysis incorporates observations of standard deviation and the number of occurrences recorded between June 9th, 2017, and September 5th, 2017—corresponding to the period surrounding the implementation of the NYSE American speed bump. All of them are in basis points. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 4.15: Panel regression of number of mini flash crashes

Regression results (90 seconds)

	Probability of mini flash crashes without reversal			Probability of mini flash crashes with reversal		
Speedbump	0.4614***	0.4639***	0.4554***	0.1386***	0.1452***	0.1248***
	(0.0075)	(0.01228)	(0.0149)	(0.0113)	(0.0179)	(0.0208)
VIX	0.0031	0.0026	0.0057	0.0855***	0.0856***	0.0916***
	(0.0027)	(0.0044)	(0.0054)	(0.0040)	(0.0064)	(0.0075)
Turnover	0.1597***	0.1877***		0.2524**	0.2032***	
	(0.0034)	(0.0031)		(0.0052)	(0.0046)	
StockVolatility	-0.0004	0.0077**		-0.0008	0.0219***	
	(0.0018)	(0.0017)		(0.0027)	(0.0025)	
MarketCap		0.0170***			-0.0259***	
		(0.0052)			(0.0074)	
Stock F.E.	Yes	No	No	Yes	No	No
N	7745	7745	7834	5880	5880	5965
Adjust R-square	40.62%	41.15%	12.12%	29.39%	30.42%	4.27%

Notes: This table reports the linear regression result of probability of 90 seconds mini flash crashes through the observations from their standard deviation and number of observations from June 9th 2017 to September 5th 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. All of them are in basis points. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 4.16: Panel regression of probability of mini flash crashes

Regression results (15 seconds)

	Number of mini flash crashes without reversal			Number of mini flash crashes with reversal		
Speedbump	0.0489	0.0485	-0.0032	0.0387	0.0324	0.0040
	(0.0774)	(0.1287)	(0.0780)	(0.0514)	(0.0931)	(0.0511)
VIX	0.0362	0.3610	0.0412	0.0210	0.0210	0.0237
	(0.0280)	(0.0240)	(0.0282)	(0.0186)	(0.0162)	(0.0185)
Turnover	0.6423***	0.6504***		0.3583***	0.3645***	
	(0.0189)	(0.1408)		(0.0125)	(0.0886)	
StockVolatility	0.0268	0.0189		0.0129	0.0068	
	(0.0106)	(0.0851)		(0.0070)	(0.0518)	
MarketCap		0.0927			0.0712	
		(0.4483)			(0.2805)	
Stock F.E.	Yes	No	No	Yes	No	No
N	11,828	11,828	12,338	11,828	11,828	12,338
Adjust R-square	9.07%	9.16%	0%	6.55%	6.66%	0%

Notes: This table presents the results of a linear regression analysis examining the difference in the number of 15-second mini flash crashes between NYSE American and its control group. The analysis incorporates observations of standard deviation and the number of occurrences recorded between June 9th, 2017, and September 5th, 2017—corresponding to the period surrounding the implementation of the NYSE American speed bump. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. All of them are in basis points. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 4.17: Panel regression of number of mini flash crashes

Regression results (15 seconds)

	Probability of mini flash crashes without reversal			Probability of mini flash crashes with reversal		
Speedbump	-0.1629***	-0.1640***	-0.1692***	-0.2073***	-0.2086***	-0.2041***
	(0.0056)	(0.0071)	(0.0080)	(0.0193)	(0.0258)	(0.0260)
VIX	-0.0009	-0.0011	-0.0005	-0.0466***	-0.0451***	-0.0454***
	(0.0020)	(0.0026)	(0.0029)	(0.0069)	(0.0093)	(0.0094)
Turnover	0.0948	0.0653***		0.0788***	-0.0449***	
	(0.0026)	(0.0018)		(0.0090)	(0.0064)	
StockVolatility	-0.0018	0.0107***		0.0054	0.0216***	
	(0.0013)	(0.0009)		(0.0040)	(0.0032)	
MarketCap		-0.0016			-0.0262***	
		(0.0030)			(0.0094)	
Stock F.E.	Yes	No	No	Yes	No	No
N	6395	6395	6395	3862	3862	3862
Adjust R-square	24.08%	26.53%	7.35%	0.37%	5.14%	3.12%

Notes: This table reports the linear regression result of probability of 15 seconds mini flash crashes through the observations from their standard deviation and number of observations from June 9th 2017 to September 5th 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. All of them are in basis points. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 4.18: Panel regression of probability of mini flash crashes

Regression results (1.5 seconds)

	Number of mini flash crashes without reversal			Number of mini flash crashes with reversal		
Speedbump	0.0021	0.0020	-0.0116	0.0016	0.0016	-0.0037
	(0.0194)	(0.0295)	(0.0197)	(0.0088)	(0.0132)	(0.0087)
VIX	0.0117*	0.0117*	0.0131*	0.0067**	0.0067**	0.0070**
	(0.0070)	(0.0066)	(0.0071)	(0.0032)	(0.0030)	(0.0032)
Turnover	0.1750***	0.1761***		0.0609***	0.0616***	
	(0.0047)	(0.0298)		(0.0021)	(0.0104)	
StockVolatility	0.0094***	0.0084		0.0032***	0.0025	
	(0.0026)	(0.0184)		(0.0012)	(0.0058)	
MarketCap		0.0121			0.0084	
		(0.0901)			(0.0300)	
Stock F.E.	Yes	No	No	Yes	No	No
N	11,828	11,828	12,338	11,828	11,828	12,338
Adjust R-square	10.63%	10.68%	0.01%	6.57%	6.64%	0.02%

Notes: This table presents the results of a linear regression analysis examining the difference in the number of 1.5 seconds mini flash crashes between NYSE American and its control group. The analysis incorporates observations of standard deviation and the number of occurrences recorded between June 9th, 2017, and September 5th, 2017—corresponding to the period surrounding the implementation of the NYSE American speed bump. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. All of them are in basis points. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 4.19: Panel regression of number of mini flash crashes

Regression results (1.5 seconds)

	Probability of mini flash crashes without reversal			Probability of mini flash crashes with reversal		
Speedbump	-0.1226***	-0.1239***	-0.1318***	-0.0892***	-0.0839***	-0.1056***
	(0.0131)	(0.0148)	(0.0152)	(0.0308)	(0.0327)	(0.0341)
VIX	0.0820***	0.0822***	0.0826***	-0.0199*	-0.0192	-0.0187
	(0.0047)	(0.0053)	(0.0055)	(0.0110)	(0.0117)	(0.0122)
Turnover	0.0758***	0.0597***		0.1270***	0.1202***	
	(0.0059)	(0.0037)		(0.0141)	(0.0083)	
StockVolatility	0.0022	0.0084***		0.0006	0.0121***	
	(0.0030)	(0.0019)		(0.0057)	(0.0035)	
MarketCap		0.0200			0.0265**	
		(0.0056)			(0.0105)	
Stock F.E.	Yes	No	No	Yes	No	No
N	4782	4782	4782	2503	2503	2538
Adjust R-square	9.57%	10.3%	4.74%	8.21%	8.90%	0.63%

Notes: This table reports the linear regression result of probability of 1.5 seconds mini flash crashes through the observations from their standard deviation and number of observations from June 9th 2017 to September 5th 2017, which are around NYSE American speed bump implementation. Control variables include: daily market volatility measured by VIX index, daily turnover, stock volatility and market capitalization. All of them are in basis points. The speed bump is implemented at July 24th 2017. The table also reports the difference between before and after speed bump. ***, **, and * indicate significance at the 1%, 5%, and 10% levels, respectively.

Table 4.20: Panel regression of probability of mini flash crashes

Bibliography

Aequitas NEO Exchange Inc (2018). Aequitas neo exchange inc. trading functionality guide. *Available at* *https://www.cboe.ca/api/wp-content/uploads/2018/11/NEO-Exchange-Trading-Functionality-Guide_v1.07_Nov_22_2018.pdf*.

Aggarwal, N. Information share and component share weights: R implementation. *Available at* *http://r-forge.r-project.org/projects/ifrogs/*.

Aldrich, E. M. and D. Friedman (2023). Order protection through delayed messaging. *Management Science 69*(2), pp. 774–790.

Alizadeh, S., M. W. Brandt, and F. X. Diebold (2002). Range-based estimation of stochastic volatility models or exchange rate dynamics are more interesting than you think. *Journal of Finance 57*(3), pp. 1047–1091.

Andersen, T. G. and O. Bondarenko (2014a). Assessing Measures of Order Flow Toxicity and Early Warning Signals for Market Turbulence. *Review of Finance 19*(1), pp. 1–54.

Andersen, T. G. and O. Bondarenko (2014b). Reflecting on the VPIN dispute. *Journal of Financial Markets 17*(C), pp. 53–64.

Anderson, L., E. Andrews, B. Devani, M. Mueller, and A. Walton (2022). Speed segmentation on exchanges: Competition for slow flow. *Journal of Financial Markets 58*.

Aoyagi, J. (2022). Strategic High-Frequency Trading, For-Profit Exchanges, and Endogenous Delays. *Available at https://ssrn.com/abstract=3896395*.

Aquilina, M., S. Foley, P. O'Neill, and T. Ruf (2024). Sharks in the dark: Quantifying hft dark pool latency arbitrage. *Journal of Economic Dynamics and Control 158*(C).

Aït-Sahalia, Y., J. Fan, L. Xue, and Y. Zhou (2022, August). How and When are High-Frequency Stock Returns Predictable? *Available at https://ideas.repec.org/p/nbr/nberwo/30366.html*.

Baldauf, M. and J. Mollner (2020). High-frequency trading and market performance. *The journal of finance.*.

Banerjee, A. and P. Roy (2023). High-frequency traders' evolving role as market makers. *Pacific-Basin Finance Journal 82*.

Baron, M., J. Brogaard, B. Hagströmer, and A. Kirilenko (2019). Risk and return in high-frequency trading. *Journal of Financial and Quantitative Analysis 54*(3), pp. 993–1024.

Bellia, Mario; Christensen, Kim; Kolokolov, Aleksey; Pelizzon, Loriana; Reno, Roberto (2020). High-frequency trading during flash crashes: Walk of fame or hall or shame. *SAFE Working Paper No. 270.*, 68.

Benos, E., J. Brugler, E. Hjalmarsson, and F. Zikes (2017). Interactions among high-frequency traders. *The Journal of Financial and Quantitative Analysis*.

Biais, B., T. Foucault, and S. Moinas (2015). Equilibrium fast trading. *Journal of Financial Economics 116*(2), pp. 292–313.

Bossaerts, P., C. Frydman, and J. Ledyard (2013). The Speed of Information Revelation and Eventual Price Quality in Markets with Insiders: Comparing Two Theories. *Review of Finance 18*(1), pp. 1–22.

Bouchaud, J.-P. (2021, dec). Radical complexity. *Entropy 23*(12), 1676.

Breckenfelder, J. (2019, June). Competition among high-frequency traders, and market quality. *Available at https://ideas.repec.org/p/ecb/ecbwps/20192290.html* (2290).

Brogaard, J., A. Carrion, T. Moyaert, R. Riordan, A. Shkilko, and K. Sokolov (2018). High frequency trading and extreme price movements. *Journal of Financial Economics 128*(2), pp. 253–265.

Brolley, M. and D. A. Cimon (2020). Order-flow segmentation, liquidity, and price discovery: The role of latency delays. *Journal of Financial and Quantitative Analysis 55*(8), pp. 2555–2587.

Chaboud, A., D. Rime, and V. Sushko (2023). The foreign exchange market. *Available at https://EconPapers.repec.org/RePEc:bis:biswps:1094*.

Chakrabarty, B., J. Huang, and P. K. Jain (October 25, 2020). Effects of a speed bump on market quality and exchange competition. *Available at https://ssrn.com/abstract=3280645*.

Chakrabarty, B., B. Li, V. Nguyen, and R. A. Van Ness (2007). Trade classification algorithms for electronic communications network trades. *Journal of Banking & Finance 31*(12), pp. 3806–3821.

Chen, H., S. Foley, M. A. Goldstein, and T. Ruf (2017). The value of a millisecond: Harnessing information in fast, fragmented markets. *Available at https://ssrn.com/abstract=2860359*.

Christopher, P. (2008). Liquidity is a coward. *Financial Sense Wealth Management*.

Chuang, H. and H.-C. Ho (2013). Implied price risk and momentum strategy. *Review of Finance 18*(2), pp. 591–622.

Cont, R., A. Kukanov, and S. Stoikov (2013). The price impact of order book events. *Journal of Financial Econometrics 12*(1), pp. 47–88.

Du, S. and H. Zhu (2017). What is the Optimal Trading Frequency in Financial Markets? *The Review of Economic Studies 84*(4), 1606–1651.

Dugast, J. and T. Foucault (2018). Data abundance and asset price informativeness. *Journal of Financial Economics 130*(2), pp. 367–391.

Easley, D., M. L. de Prado, M. O'Hara, Z. Zhang, and W. Jiang (2021). Microstructure in the Machine Age. *Review of Financial Studies 34*(7), pp. 3316–3363.

Easley, D., M. Lopez de Prado, and M. O'Hara (2012). Flow toxicity and liquidity in a high frequency world. *Review of Financial Studies 25*(5), pp. 1457–1493.

Figuerola-Ferretti, I. and J. Gonzalo (2010). Modelling and measuring price discovery in commodity markets. *Journal of Econometrics 158*(1), pp. 95–107.

Fishe, R. P. H., R. Haynes, and E. Onur (2019). Anticipatory traders and trading speed. *The Journal of Financial and Quantitative Analysis 54*(2), pp. 729–758.

Fosset, A., J.-P. Bouchaud, and M. Benzaquen (2020). Non-parametric estimation of quadratic hawkes processes for order book events. *Available at* ***https://arxiv.org/abs/2005.05730***.

Foucault, T. (2016). Where are the risks in high frequency trading? *Financial Stability Review*, pp. 53–67.

Foucault, T., J. Hombert, and I. Roşu (2016). News trading and speed. *Journal of Finance 71*(1), pp. 335–382.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics 29*(5), pp. 1189–1232.

Goldstein, M., A. Kwan, and R. Philip (2023, August). High-Frequency Trading Strategies. *Management Science 69*(8), 4413–4434.

Golub, A., J. Keane, and S.-H. Poon (2012). High frequency trading and mini flash crashes. *Available at* ***https://ssrn.com/abstract=2182097***.

Gonzalo, J. and C. Granger (1995). Estimation of common long-memory components in cointegrated systems. *Journal of Business and Economic Statistics 13*, pp. 27–35.

Gonçalves, J., R. Kräussl, and V. Levin (2019). Do "speed bumps" prevent accidents in financial markets? *CFS Working Paper Series, Center for Financial Studies (CFS)* (636).

Harris, L. (2013). What to do about high-frequency trading,. *Financial Analysts Journal 62*(2), pp. 6–9.

Hasbrouck, J. (1991a). Measuring the information content of stock trades. *Journal of Finance 46*(1), pp. 179–207.

Hasbrouck, J. (1991b). The summary informativeness of stock trades: An econometric analysis. *Review of Financial Studies 4*(3), pp. 571–595.

Hasbrouck, J. (1993). Assessing the quality of a security market: A new approach to transaction-cost measurement. *Review of Financial Studies 6*(1), pp. 191–212.

Hasbrouck, J. (1995). One security, many markets: Determining the contributions to price discovery. *Journal of Finance 50*(4), pp. 1175–1199.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning: data mining, inference and prediction* (2 ed.). Springer.

Hendershott, T. and P. C. Moulton (2011). Automation, speed, and stock market quality: The nyse's hybrid. *Journal of Financial Markets 14*(4), pp. 568–604.

Hu, E. (2019). Intentional access delays, market quality, and price discovery: Evidence from iex becoming an exchange. *Available at* ***https://ssrn.com/abstract=3195001***.

Ke, Z. T., B. T. Kelly, and D. Xiu (2019, August). Predicting Returns With Text Data. *Available at https://ideas.repec.org/p/nbr/nberwo/26186.html*.

Kearns, M. and Y. Nevmyvaka (2013). Machine learning for market microstructure and high frequency trading. *Available at https://api.semanticscholar.org/CorpusID:18058262*.

Keller, A. J. (2012). Regulating high frequency trading after the flash crash of 2010. *Ohio State Law Journal 73*(6), pp. 1457–1483.

Kercheval, A. N. and Y. Zhang (2015). Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance 15*(8), pp. 1315–1329.

Khan, Wasiat; Malik, U. G. M. A. A. M. A. A. K. H. A. A. S. (2020). Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis. *Soft Computing*.

Kirilenko, A., A. S. Kyle, M. Samadi, and T. Tuzun (2017a). The flash crash: High-frequency trading in an electronic market. *The Journal of Finance 72*(3), pp. 967–998.

Kirilenko, A., A. S. Kyle, M. Samadi, and T. Tuzun (2017b). The flash crash: High-frequency trading in an electronic market. *Journal of Finance 72*(3), pp. 967–998.

Kyle, A. S. and A. A. Obizhaeva (2023). Large Bets and Stock Market Crashes. *Review of Finance 27*(6), pp. 2163–2203.

Langton, J. (2010). Circuit breakers, investment executive. *Available at https://www.investmentexecutive.com/newspaper_/news-newspaper/circuit-breakers/*.

Le Moign, C. (2018). Effect of speed bumps: Analysis of the impact of the implementation of eurex's passive liquidity protection on french equity options. *amf-france.org*.

Leal, S. J., M. Napoletano, A. Roventini, and G. Fagiolo (2014). Rock around the clock: An agent-based model of low- and high-frequency trading. *Available at https://doi.org/10.48550/arXiv.1402.2046*.

Lee, C. and M. Ready (1991). Inferring trade direction from intraday data. *Journal of Finance 46*(2), pp. 733–46.

Levine, M. (April 21, 2015). Guy trading at home caused the flash crash. *Bloomberg*.

Lewis, M. (2014). *Flash Boys*. New York: W. W. Norton & Company.

Li, Z. M. and O. Linton (2022). A ReMeDI for Microstructure Noise. *Econometrica 90*(1), pp. 367–389.

Liu, Bo; Xu, K. (2023). Speed bump and stock market quality: evidence from nyse american. *Available at https://ssrn.com/abstract=4593778*.

Mase, B. (1999). The Predictability of Short-Horizon Stock Returns. *Review of Finance 3*(2), pp. 161–173.

Menkveld, A. and M. Zoican (2017). Need for speed? exchange latency and liquidity. *Review of Financial Studies 30*(4), pp. 1188–1228.

Min, B. H. and C. Borch (2022). Systemic failures and organizational risk management in algorithmic trading: Normal accidents and high reliability in financial markets. *Social Studies of Science 52*(2), pp. 277–302.

Mohajon, J. (2020). Confusion matrix for your multi-class machine learning model. *Available at https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826*.

Murphy, K. P. (2013). *Machine learning : a probabilistic perspective*. Cambridge, Mass. The U.S.A: MIT Press.

Nanex (November 29, 2010). Flash equity failures in 2006, 2007, 2008, 2009, 2010, and 2011. *Available at* ***http://www.nanex.net/FlashCrashEquities/FlashCrashAnalysis_Equities.html***.

Nielsen, M. Ø. and M. K. Popiel (2018). A matlab program and user's guide for the fractionally cointegrated var model. *Queen's Economics Department Working Paper 1330*.

Nolte, D. (2010). SEC Takes Initial Measures to Avoid a Second Flash Crash. *Available at* ***https://www.hgexperts.com/expert-witness-articles/sec-takes-initial-measures-to-avoid-a-second-flash-crash-191781***.

OICU-IOSCO (2011). Regulatory issues raised by the impact of technological changes on market integrity and efficiency consultation report. *Available at* ***https://www.iosco.org/library/pubdocs/pdf/IOSCOPD354.pdf***.

Osipovich, A. (2019). More exchanges add 'speed bumps,' defying high-frequency traders. *The Wall Street Journal*.

O'Hara, M. (2015). High frequency market microstructure. *Journal of Financial Economics 116*(2), pp. 257–270.

Pagnotta, E. S. and T. Philippon (2018). Competing on Speed. *Econometrica 86*(3), pp. 1067–1115.

Patel, J., S. Shah, P. Thakkar, and K. Kotecha (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Syst. Appl. 42*(1), pp. 259–268.

Patel, V., T. J. Putniņš, D. Michayluk, and S. Foley (2020). Price discovery in stock and options markets. *Journal of Financial Markets 47*.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Courna-

peau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, pp. 2825–2830.

Peng, Z., D. Guo, and D. Meng (2019). IEX's Speed Bump and Its Effect on Adverse Selection. *Available at https://sbfc.sydney.edu.au/program/papers/P081_Named.pdf*.

Peter Chung, Y. and S. Thomas Kim (2017). Extreme returns and herding of trade imbalances. *Review of Finance 21*(6), pp. 2379–2399.

Putniņš, T. J. (2013). What do price discovery metrics really measure? *Journal of Empirical Finance 23*, pp. 68–83.

Reuters, S. (Octorber 12, 2010). Canada's "flash crash" mild due to mkt structure. *Regulatory News*.

Shkilko, A. and K. Sokolov (2020, December). Every Cloud Has a Silver Lining: Fast Trading, Microwave Connectivity, and Trading Costs. *Journal of Finance 75*(6), pp. 2899–2927.

Sirignano, J. (2016). Deep Learning for Limit Order Books. *Available at https://ideas.repec.org/p/arx/papers/1601.01987.html*.

Sirignano, J. and R. Cont (2018). Universal features of price formation in financial markets: perspectives from deep learning. *Available at https://arxiv.org/abs/1803.06917*.

Technical Committee Of The International Oganization Of Securities Commissions (2011). Regulatory Issues Raised by the Impact of Technological Changes on Market Integrity and Efficiency Final Report. *Available at https://www.iosco.org/library/pubdocs/pdf/IOSCOPD361.pdf*.

Wang, Y., L. Liu, R. Gu, J. Cao, and H. Wang (2010). Analysis of market efficiency for the shanghai stock market over time. *Physica A: Statistical Mechanics and its Applications 389*(8), pp. 1635–1642.

XGBoostContributors (2023). Xgboost, wikipedia. *Available at* ***https://en.wikipedia.org/wiki/XGBoost***.

Xu, K., X. Zheng, D. Pan, L. Xing, and X. Zhang (2020). Stock market openness and market quality: Evidence from the shanghai–hong kong stock connect program. *Journal of Financial Research 43*(2), pp. 373–406.

Yan, B. and E. Zivot (2010). A structural analysis of price discovery measures. *Journal of Financial Markets 13*(1), pp. 1–19.

Yang, L. and H. Zhu (2019). Back-Running: Seeking and Hiding Fundamental Information in Order Flows*. *Review of Financial Studies 33*(4), pp.1484–1533.

Ye, M., C. Yao, and J. Gai (2013). The externalities of high frequency trading. *WBS Finance Group Research Paper No. 180*.

Zhang, L., P. A. Mykland, and Y. Aït-Sahalia (2005). A tale of two time scales. *Journal of the American Statistical Association 100*(472), pp. 1394–1411.