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

Bo Liu*

Department of Economics

University of Victoria

October 17, 2023

*Corresponding author. E-mail: boliu1@uvic.ca. I am grateful to my Ph.D. supervisor, Ke Xu, for her invaluable guidance and support throughout my research journey. I would also like to express my appreciation to David Cimon and Jie Zhou for their insightful comments on an earlier draft presented at CEA 2023. I would also like to express my appreciation to Yue Zhao for his suggestions on the initial data processing ideas.

Abstract

In recent years, the emergence of mini flash crashes has become a distinctive concern within contemporary electronic trading markets, garnering attention from scholars, market participants, and regulators alike. These rapid, unforeseen market disruptions have been attributed to the breakneck pace of trading activity. This study employs data from the NYSE Trade and Quote database (TAQ) to examine the efficacy of implementing a "speed bump" mechanism in mitigating the risks associated with mini flash crashes. Utilizing machine learning techniques to estimate the likelihood of mini flash crashes occurring within the NYSE American market, this analysis offers empirical insights. The findings of this research demonstrate that the introduction of a speed bump mechanism can indeed reduce the probability of mini flash crashes. However, it is noteworthy that this mitigation strategy also leads to an influx of noise traders and an increase in short-term market volatility.

JEL Codes: G10, G14, G18

Keywords: speed bump, price discovery, market liquidity, market efficiency, cointegration.

1 Introduction

The literature extensively explores the influence of high-frequency trading (HFT) on flash crashes, 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. Investigated by both the Securities and Exchange Commission (SEC) and the Commodity Futures Trading Commission (CFTC), the precise cause of this event remains elusive. Nevertheless, many market participants suspect that the cancellation of existing buy orders by high-speed trading algorithms upon detecting market imbalances is a primary contributor (Nolte (Nolte)). 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 relies 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 (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 1

In the aftermath of the May 6, 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 (Nolte)).

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 management 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 Staff (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. (2017a), 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 (2014). 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 **?**. 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.

Certain studies have employed a widely recognized rule-of-thumb definition for assessing flash crashes, as initially proposed by Nanex, a trading technology company and data service provider Nanex (2010). This approach often grants access to extensive datasets, facilitating precise statistical inference. Furthermore, advancements in data analysis techniques have ushered in a new era. Beyond finance, the application of machine learning techniques has proven highly effective in analyzing large datasets across various domains, shedding light on the possibility of predicting real-world mini flash crashes using market data.

Figure 2

Liu (2023b) introduced 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. Building 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 (2019)).

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 Liu (2023b), 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, Liu (2023b) 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 (2023a) 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 Liu (2023b). 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 (2023a) 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 (2023a).

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 2 provides an overview of the relevant literature. Section 3 outlines the dataset employed and the data pre-processing techniques employed. Section 4 details the methodology employed, encompassing the problem setup, machine learning models utilized, accuracy evaluation methods, and data imbalance processing strategies. Section 5 presents the empirical results and their implications. Section 6 concludes the findings and summarizes the key takeaways from the study.

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. (2017b) 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 intraday 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 Liu (2023b). 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 (2019), 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 (2018) 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 (2018) 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.

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

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 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

[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. There are 51 stocks commonly traded on NYSE American and Nasdaq, one of them has a significant missing data problem, we use the left 50 stocks.

exclude it from the pool of potential matching stocks. The average matching error observed in this context is 0.0767.

4 Methodology

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

4.1 Response Variables

Differing from many other financial machine learning studies, such as Easley et al. (2021), which employ event windows or information bars for data aggregation, this paper adheres to a time-based approach.

In practice, when a portfolio manager is actively monitoring their investment portfolio in the lead-up to a potential mini flash crash, they cannot afford to wait for additional events to accumulate before the model predicts the probability of such an event. Consequently, to ensure the model's real-time predictive capability for mini flash crashes, I exclusively rely 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 ticks in the same direction;

In this paper, I adopt Nanex (2010)'s definition of mini flash crashes, classifying them as extremely short-term events lasting 1.5 seconds. To gain deeper insights into these market events, I introduce three additional time windows, each with a more relaxed duration: 180 seconds, defined as long-term; 90 seconds, designated as medium-term; and 15 seconds, categorized as short-term.

The long-term mini flash crashes allow us to reasonably assume that both slow and fast traders have had sufficient time to react to market dynamics within 180 seconds. The medium-term window serves as a robust check on the long-term perspective and offers an alternative view, highlighting the involvement of "fast-reactive" slow traders in these events. The 15-second time frame predominantly reflects the influence of fast traders but also provides valuable information about the overall market conditions. Lastly, the extremely short-term category, aligned with Nanex (2010)'s definition, represents the most stringent classification of mini flash crashes.

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 ticks in the same direction and at least 10 ticks in the reversed direction;

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 1, Table 2, Table 3, and Table 4 respectively.

Table 1

4.2 Predictor Variables

To forecast the aforementioned response variables, I consider a diverse set of predictor variables, with most of them constructed in accordance with the approach outlined in Aït-Sahalia et al. (2022). These predictors encompass numerous derived variables, representing nonlinear transformations of historical data at short time intervals. Drawing from existing literature, including Cont et al. (2013) and Kercheval and Zhang (2015), I anticipate that the primary drivers for predicting future short-term events, such as mini flash crashes, will revolve around characteristics of the current Limit Order Book (LOB), including any imbalances and past trade returns at the time of prediction. In addition to the methodology introduced in Aït-Sahalia et al. (2022), I also incorporate microstructure measures, following the approach detailed in Easley et al. (2021). To capture market microstructure noise information, which often carries signals of unexpected events, I introduce three additional predictors related to microstructure noise. A comprehensive description of all our features will be presented in the 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:

$$Breath(T, \Delta_1, \Delta_2) = |D^{txn} \cap Int^{back}(T, \Delta_1, \Delta_2)| \tag{1}$$

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

$$Immediacy(T, \Delta_1, \Delta_2) = \frac{\Delta_1 - \Delta_2}{Breath(T, \Delta_1, \Delta_2)} \tag{2}$$

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

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

$$VolumeAll(T, \Delta_1, \Delta_2) = \sum_{t \in Int^{back}(T, \Delta_1, \Delta_2)} V_t \tag{3}$$

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

$$VolumeAvg(T, \Delta_1, \Delta_2) = \frac{VolumeAll(T, \Delta_1}{Breath(T, \Delta_1, \Delta_2} \tag{4}$$

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

$$VolumeMax(T, \Delta_1, \Delta_2) = max\{V_t : t \in Int^{back}(T, \Delta_1, \Delta_2)\} \tag{5}$$

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:

$$Lambda(T, \Delta_1, \Delta_2) = \frac{P_{max(I)} - P_{min(I)}}{VolumeAll(T, \Delta_1, \Delta_2)} \tag{6}$$

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

$$LobImbalance(T, \Delta_1, \Delta_2) = Average[\frac{S_t^a - S_t^b}{S_t^a + S_t^b}] : t \in Int^{back}(T, \Delta_1, \Delta_2) \tag{7}$$

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

$$TxnImbalance(T, \Delta_1, \Delta_2) = Average[\frac{\sum_{t \in D^{txn} \cap Int^{back}(T, \Delta_1, \Delta_2)} (V_t Dir_t^{LR})}{VolumeAll(T, \Delta_1, \Delta_2)}] \tag{8}$$

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

$$PastReturn(T, \Delta_1, \Delta_2) = 1 - Average\frac{[P_t^{txn} : t \in I]}{P_{max(I)}} \tag{9}$$

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.

$$Turnover(T, \Delta_1, \Delta_2) = \frac{VolumeAll(T, \Delta_1, \Delta_2)}{S} \tag{10}$$

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:

$$\text{AutoCov}\,(T, \Delta_1, \Delta_2) = \text{ Average } \left[\log\left(\frac{P_t^{txn}}{P_{Lt}^{txn}}\right)\log\left(\frac{P_t^{txn}}{P_{L(Lt)}^{txn}}\right) : \text{t} \in D^{txn} \cap \text{Int}\, t^{\text{back}}\ (T, \Delta_1, \Delta_2)\right] \tag{11}$$

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

$$\text{QuotedSpread }\ (T, \Delta_1, \Delta_2) = \text{Average}\left[\frac{P_t^a - P_t^b}{P_t} : \text{t} \in \text{Int}^{\text{back}}\ (T, \Delta_1, \Delta_2)\right] \tag{12}$$

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

$$\text{EffectiveSpread}\,(T, \Delta_1, \Delta_2) = \frac{\sum_{t \in D^{txn} nInt^{\text{back}}\ (T, \Delta_1, \Delta_2)}\left[\log\left(\frac{P_t^{txn}}{P_t}\right)\text{Dir}_t^{LR}\, V_t P_t^{txn}\right)}{\sum_{t \in D^{txn} \cap \ln t^{\text{back}}\ (T, \Delta_1, \Delta_2)}(V_t P_t^{txn})} \tag{13}$$

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:

$$
\begin{aligned}
R_t &= 2\sqrt{|\mathrm{cov}\left(\Delta \boldsymbol{P}_t, \Delta \boldsymbol{P}_{t-1}\right)|} \\
\Delta \boldsymbol{P}_t &= \left[\Delta P_{t-w}, \Delta P_{t-w-1}, \ldots \Delta P_t\right], \\
\Delta \boldsymbol{P}_{t-1} &= \left[\Delta P_{t-w-1}, \Delta P_{t-w}, \ldots \Delta P_{t-1}\right],
\end{aligned} \tag{14}
$$

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:

$$
\tilde{R}_t = \frac{2\sqrt{|\mathrm{cov}\left(\Delta \boldsymbol{P_t}, \Delta \boldsymbol{P}_{t-1}\right)|}}{P_t V_t} \textit{Roll impact:} \tag{15}
$$

Kyle's lambda is given by:

$$
\lambda_t = \frac{P_t - P_{t-w}}{\sum_{i=t}^{t} b_i V_t} \tag{16}
$$

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

Amihud's measure:

$$
\lambda_t^A = \frac{1}{W} \sum_{i=t-W+1}^{t} \frac{|r_i|}{p_i V_i} \tag{17}
$$

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:

$$
\mathrm{VPIN}_t = \frac{1}{W} \sum_{i=\tau-W+1}^{\tau} \frac{\left|P_t^a - P_t^b\right|}{V_i} \tag{18}
$$

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:

$$[P,P]_w = \sum_{t\in W} \left(P_{t_{i+1}} - P_{t_i}\right)^2 \tag{19}$$

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

$$\widehat{\langle P,P\rangle}_T = [Y,Y]_T^{\text{avg}} \; - \frac{\bar{n}}{n}[Y,Y]_T^{\text{all}} \tag{20}$$

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).

VIX: The last one is the daily Volatility Index (VIX) of the previous trading day.

4.3 Machine Learning Methods

4.3.1 Models

In this paper, we evaluate the performance of four primary machine learning models for mini flash crash prediction. These models include regularized logistic regression (LASSO) as a representative of linear parametric methods, penalized support vector machine (penalized-SVM) representing nonparametric techniques, as well as two ensemble models, random forest, and extreme gradient boosting (XGBoost). Further theoretical details regarding these models can be found in Hastie et al. (2009) and Murphy (2013).

Consider the imbalanced classification problem where we aim to predict a response variable Y, which takes on the value 1 to signify the presence of an impending mini flash crash within the look-forward window, or 0 to indicate its absence. This prediction relies on a predictor vector X derived from a random sample $(\mathbf{X}_i,\ Y_i)$. Let $\mathbf{Y} = (y_1, ..., y_n)^T$. Each feature vector $\mathbf{X}_i$ has a dimension of 232, comprising 11 time spans for each of the 21 predictor variables and the final dimension accounting for the volatility index (VIX) from the preceding trading day. Machine learning algorithms can then endogenously generate further

combinations of these predictors or opt for the most informative subsets of predictors.

4.3.2 Penalized Logistic Regression

Among the well-established machine learning models, linear models have always held a prominent position due to their simplicity and interpretability. Logistic regression, a 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:

$$y = \frac{1}{1 + e^{-z}} \tag{21}$$

Through linear regression model:

$$Z = \beta^T \mathbf{X} + \epsilon \tag{22}$$

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:

$$\hat{\beta} = \text{argmin}_{\beta \in R^p} \left\{ \frac{1}{n} \sum_i \left(\widetilde{Z}_l - \beta^T \widetilde{\boldsymbol{X}}_l \right)^2 + \lambda \|\beta\| \right\} \tag{23}$$

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:

$$\widehat{Y_{\text{new}}} = \frac{1}{1 + e^{-\left(\overline{\boldsymbol{Z}} + \hat{\boldsymbol{\beta}}^T \boldsymbol{X}_{\text{new}}\right)}}, \text{ with } \widetilde{\boldsymbol{X}_{\text{new}}} = \text{diag}\left(s_1^{-1}, s_2^{-1}, \ldots, s_p^{-1}\right)\left(\boldsymbol{X}_{\text{new}} - \overline{\boldsymbol{X}}\right) \tag{24}$$

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.

4.3.3 Support Vector Machine

The Support Vector Machine (SVM) stands out as one of the most popular non-parametric algorithms due to its exceptional performance and well-defined mathematical underpinnings. It serves as a representative non-parametric forecasting model in this study. In this section, I provide a brief introduction to SVM, with more comprehensive details available in Hastie et al. (2009)) and Murphy (2013).

A Support Vector Machine constructs a hyperplane or a set of hyperplanes within a high-dimensional or even infinite-dimensional space, making it versatile for both classification and regression tasks (as implemented in scikit-learn).

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

For having this ω, SVM solves the following primal optimization problem:

$$\min_{\omega, b, \zeta 2} \frac{1}{2}\omega^T \omega + C \sum_{i=1}^{n} \zeta_i \tag{25}$$

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

Through the SVM, the goal is to maximize the margin between the two classes by minimizing the norm of the weight vector, $|\omega|^2 = \omega^T\omega$. In an ideal scenario, the hyperplane would perfectly separate all samples, meaning $y_i\left(\omega^T\phi\left(x_i\right)+b\right) \geq 1$ for all samples. However, in the real world, samples are often not perfectly separable by a hyperplane, necessitating the allowance of some samples to deviate by a distance ζ_i from their correct margin boundary. The penalty term C regulates the strength of this penalty. 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 plays a crucial role in mapping the samples into higher-dimensional or even infinite-dimensional spaces. For a more comprehensive understanding, please refer to Hastie et al. (2009). In the context of mini flash crash forecasting presented in this paper, I conducted numerous experiments to fine-tune hyperparameters and found that employing the linear kernel (simple inner product, $< x, x >$) in conjunction with hinge loss for constructing the optimization problem consistently outperformed all other kernel choices. In the subsequent empirical results section, I will exclusively present the results obtained using the linear kernel support vector machine.

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

$$\sum_{i \in SV}^{n} y_i \alpha_i K(x_i, x_{new}) + b \tag{26}$$

Then the predicted class corresponds to its sign.

4.3.4 Random forests

While ensemble learning tree-style models may lack the ideal interpretability of linear models, they often demonstrate impressive forecasting capabilities in various practical problems. In this paper, I employ two ensemble models, with the first being a random forest. Random forest is a scalable nonparametric learning method that builds upon the foundation of a single decision tree. Given that a single-tree method can be prone to instability and may not exhibit strong predictive power, constructing a forest by ensembling multiple individual decision trees becomes a natural approach. By successfully training these independent decision trees and averaging the outcomes of numerous sampled decision trees, we reduce prediction variance, resulting in more stable and reliable predictions.

According to Hastie et al. (2009), random forests are constructed by iteratively growing regression trees through the bootstrap sampling process from the provided dataset. 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 $\hat{C}_{rf}^B(\mathrm{x})$ = majority vote $\left\{\hat{C}_b(x)\right\}_1^B$. (Hastie et al. (2009)).

Predictions derived from bagging methods often exhibit high correlation due to their reliance on samples from the same dataset. However, by employing random forests, which involve bagging of independently trained decision trees from bootstrapped samples, along with variable selection at each decision tree node, we can enhance performance. This process increases the independence among the resulting trees, thereby reducing prediction dependence. Random forests effectively reduce variance compared to a single decision tree, leading to improved predictions.

4.3.5 Extreme Gradient Boosting

The second ensemble model employed in this study is Extreme Gradient Boosting (XGBoost), originally introduced by Friedman (2001). Outside of the realm of deep learning, XGBoost has garnered significant popularity within the machine learning community in recent years. XGBoost, as an ensemble model, leverages a multitude of weak base tree learners. These weak 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 that are more interpretable. XGBoost initiates 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 process continues iteratively, updating $F_1(x) < -F_0(x) + h_1(x)$, until the residuals are minimized as much as possible. 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:

$$f_0(x) = \widehat{\operatorname{argmin}} \sum_{l=1}^{N} L\left(y_l, \theta\right). \tag{27}$$

- For m = 1 to M :

  - Compute the gradients and hessians:

$$\begin{aligned} \hat{g}_m(x_i) &= \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{(m-1)}(x)} \\ \hat{h}_m(x_i) &= \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=f_{(m-1)}(x)} \end{aligned} \tag{28}$$

  - 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 solvi the optimization problem:

$$\begin{aligned} \circ\widehat{\emptyset_m} &= \text{argmin}_{\emptyset} \sum_{i=1}^{N} \hat{h}_m(x_i) \left[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \emptyset(x_i)\right]^2 \\ \hat{f}_{(m)}(x) &= \propto \widehat{\emptyset_m}(x) \end{aligned} \tag{29}$$

  - 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)).

4.3.6 Measuring Prediction Accuracy

I employ Receiver Operating Characteristic (ROC) and Area under the ROC Curve (AUC) as metrics to assess the prediction accuracy of our model. It is evident that predicting mini flash crashes amounts to a supervised anomaly detection problem. In virtually any scenario, the probability of a mini flash crash occurrence is exceedingly low. As I mentioned earlier, our dataset is significantly imbalanced, rendering the commonly used accuracy score as an inappropriate performance measure.

To illustrate this point, let's consider a hypothetical situation where only 0.1% of the look-forward window (data) is labeled as a mini flash crash occurrence. In such a case, a simple yet misguided prediction approach would be to classify all look-forward windows as having no mini flash crashes, resulting in an accuracy score of 99.9%. However, this high accuracy score would be misleading.

ROC is a much more suitable choice for evaluating prediction accuracy in the context of imbalanced data problems. This concept originated from signal analysis technology development during World War II and has since found extensive use in fields like medical issue detection and others. An ROC curve effectively illustrates the performance of a classification model across all classification thresholds. This curve plots 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 3

An ROC curve illustrates the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) at various classification thresholds. When the classification threshold is lowered, more items are classified as positive, leading to an increase in both False Positives and True Positives. The ROC curve arranges all test samples in descending order based on the predicted probability of being True, with the model predicting each sample as True one by one to compute TPR and FPR. This process results in samples with a high probability of being True, such as those indicating the potential arrival of a mini flash crash, being predicted as True first. Consequently, in a well-performing model, the curve exhibits a steep initial increase from 0, followed by a gradually slowing growth towards 1. Two plots from my tuned models are displayed below, with the red line representing a "random guess."

AUC, or "Area under the ROC Curve," quantifies the entire two-dimensional area beneath the ROC curve, extending from (0, 0) to (1, 1), similar to calculating the integral

of a function. AUC offers a comprehensive measure of performance across all possible classification thresholds. One straightforward interpretation of AUC is that it represents the probability of the model ranking a randomly selected positive example higher than a randomly selected negative example.

4.3.7 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 the data imbalance issue is to remove surplus samples from the majority class (those corresponding to instances where no mini flash crash occurs) until a balanced sample set is achieved in comparison to the minority class. This is the simplest strategy for dealing with imbalanced data. However, it comes with a drawback as it results in the removal of a substantial portion of our data. This approach may lead to the loss of valuable information present in the majority class. It has the potential to diminish the impact of certain majority cases, particularly those that fall between the two classes and carry informative value. For example, in support vector machine algorithms, the deleted samples may consist of support vectors that are situated close to the margin hyperplane on the side of the majority class, potentially causing the model to exhibit higher bias.

Oversampling: Another strategy involves randomly duplicating synthetic samples in the minority class (corresponding to instances where a mini flash crash occurs). This approach generates numerous synthetic samples to balance the class sizes. However, it carries a certain level of risk, as introducing noise samples into the minority class can result in an increase in noise, potentially leading the model to overfit. According to Liu (2023b), this strategy does not yield optimal performance and can be extremely resource-intensive. As a result, I have opted not to employ this strategy in my empirical analysis.

Synthetic Minority Oversampling Technique (SMOTE): SMOTE represents a more nuanced approach to oversampling, employing the K-Nearest Neighbors (KNN) algorithm to generate new samples within the minority class. These newly created samples are

designed to be similar to, yet distinct from, the original samples. In comparison to simple oversampling, SMOTE can be seen as a form of ensemble learning, which serves to mitigate variance and the risk of overfitting. However, it's worth noting that SMOTE, while effective in addressing class imbalance, has its own set of limitations. As SMOTE relies on the minority class to generate synthetic samples, it can magnify the impact of noise due to changes in the data distribution. Additionally, SMOTE's computational complexity is significantly higher than both oversampling and undersampling methods.

Threshold Moving: This strategy involves threshold adjustment to increase the model's sensitivity to the minority class. In a balanced data problem, the threshold is typically set at 0.5. Models learn to calculate the probability of each sample belonging to class 1 or 0 and then compare it with the 0.5 threshold. However, in imbalanced data scenarios, it is a straightforward approach 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 substantial number of majority samples to achieve a balanced dataset, which can result in the wastage of valuable data. An alternative approach is to perform multiple rounds of undersampling. For instance, if there are 10,000 samples in the majority class and only 50 samples in the minority class, one can create 200 sets of new samples. Each set would consist of all 50 minority class samples and an additional randomly selected 50 majority samples. These 200 models are then trained independently, and their results are combined in an ensemble. It's important to note that while this strategy offers advantages, such as potentially more reliable performance, it also comes with drawbacks. It significantly increases computational complexity, and the risk of overfitting remains, as minority class samples are used repeatedly. However, in many anomaly detection scenarios, such as ours, successful detection of mini flash crashes is of paramount importance. Therefore, when assuming that the minority class data is of good quality, this strategy may yield dependable performance.

5 Empirical Analysis

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 4.1.

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, 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 (2023a), 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.

5.1 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 5, and the control group's tuning results can be found in Panel A of Table 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 Figure 4a and Figure 4b. 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 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 Figure 4c and Figure 4d. 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 9 shows the daily average of mini flash crashes without reversal of all traded stocks in NYSE American and the control group.

Table 9

For mini flash crashes with reversal, the tuning results for NYSE American are presented in Panel A of Table 7, while the control group tuning results are displayed in Panel A of Table 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 Figure 8a and Figure 8b. 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 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 Figure 8c and Figure 8d.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 10 shows the daily average of mini flash crashes with reversal of all traded stocks in NYSE American and the control group.

Table 10

5.2 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:

$$P_{i,t,180s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{30}$$

$$P_{i,t,180s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{31}$$

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 11

Table 11 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 (2023a), 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.

5.3 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 5 and the control group tuning results in Panel B of Table 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 Figure 5a for the pre-speed bump period and Figure 5b 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 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 Figure 5c and Figure 5d.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 9 shows the daily average of mini flash crashes in 90s without reversal of all traded stocks in NYSE American and the control group.

Table 9

For mini flash crashes with reversal, the tuning results for NYSE American are displayed in Panel B of 7, while the control group tuning results are presented in Panel B of 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 Figure 9a and Figure 9b. 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 $\hat{f}_b(xi, j, t, NYSE \text{ American }, 90s, Rev)$ 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 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 Figure 9c and Figure 9d. 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 10 shows the daily average of mini flash crashes with reversal of all traded stocks in NYSE American and the control group.

Table 10

5.4 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:

$$P_{i,t,90s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \qquad (32)$$

$$P_{i,t,90s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \qquad (33)$$

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 12

Table 12 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.

5.5 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 4.1, 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 5 and the control group tuning results are shown in Panel C of 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 Figure 6a and Figure 6b. 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 subsections. 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 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 Figure 6c, and Figure 6d 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 9 shows the daily average of mini flash crashes without reversal of all traded stocks in NYSE American and the control group.

Table 9

For mini flash crashes with reversal, the tuning results of NYSE American are shown in Panel C of Table 7 and the control group tuning results are shown in Panel C of Table 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 Figure 10a and Figure 10b. 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 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 Figure 10c, and Figure 10d 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 10 shows the daily average of mini flash crashes with reversal of all traded stocks in NYSE American and the control group.

Table 10

5.6 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:

$$P_{i,t,15s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{34}$$

$$P_{i,t,15s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{35}$$

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 13

Table 13 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.

5.7 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 5, while the control group's tuning results are displayed in Panel D of Table 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 Figure 7a and Figure 7b. 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 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 Figure 7c and Figure 7d. 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 9 shows the daily average of mini flash crashes without reversal of all traded stocks in NYSE American and the control group.

Table 9

For mini flash crashes with reversal, the tuning results of NYSE American are shown in Panel D of Table 7 and the control group tuning results are shown in Panel D of Table 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 Figure 11a and Figure 11b. 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 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 Figure 11c and Figure 11d. 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 10 shows the daily average of mini flash crashes with reversal of all traded stocks in NYSE American and the control group.

Table 10

5.8 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:

$$P_{i,t,1.5s} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{36}$$

$$P_{i,t,1.5s,Rev} = \alpha_i + \beta SpeedBump_t + \gamma \text{Volatility}_t + \sum_{q=1}^{2} \delta_q \text{ControlVariable}_{i,t,q} + \varepsilon_{i,t} \tag{37}$$

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 14

Table 14 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 (2023a), 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.

6 Conclusion

To the best of our knowledge, this paper represents the pioneering effort to assess the impact of a speed bump on the probability of mini flash crashes. This study introduces an empirical approach for estimating the likelihood of mini flash crashes using machine learning techniques

applied to real-time limit order book data.

Our analysis, which draws upon data from the NYSE American, demonstrates that the implementation of a speed bump can indeed decrease the probability of identifiable patterns leading to mini flash crashes. Concurrently, the speed bump transforms the dynamics of today's fragmented financial market, drawing more noise traders into the NYSE American—a stock exchange employing this mechanism to safeguard slow traders from the high-frequency arms race.

Through our empirical analysis, we identify mini flash crashes as market microstructure events displaying nonlinear patterns within the real-time limit order book, where high-frequency trading executes nano-second-level algorithms. Although pinpointing the specific trading behaviors responsible for triggering mini flash crashes or even market-wide flash crashes remains a formidable challenge, it is reasonable to posit that the patterns leading to market turmoil can be discerned. The introduction of an order delaying policy significantly reduces the probability of these disruptive patterns emerging.

Simultaneously, when considered in the context of existing literature, our findings highlight the speed bump's pronounced impact on the short-term, spanning minutes and seconds. Complementing prior research, which indicates diminished price discovery and reduced adverse selection costs, our results suggest that the speed bump effectively attracts more uninformed, noise traders into the market.

In summary, the "speed bump" policy may initially appear somewhat nebulous or enigmatic, but our study demonstrates its remarkable effectiveness in mitigating the probability of mini flash crashes while reshaping market dynamics.

References

Alizadeh, S., M. W. Brandt, and F. X. Diebold (2002, December). Range-Based Estimation of Stochastic Volatility Models or Exchange Rate Dynamics are More Interesting Than You Think. *Journal of Finance 57*(3), 1047–1091.

Andersen, T. G. and O. Bondarenko (2014). Reflecting on the VPIN dispute. *Journal of Financial Markets 17*(C), 53–64.

Aoyagi, J. (2019, February). Strategic Speed Choice by High-Frequency Traders under Speed Bumps. ISER Discussion Paper 1050, Institute of Social and Economic Research, Osaka University.

Aït-Sahalia, Y., J. Fan, L. Xue, and Y. Zhou (2022, August). How and When are High-Frequency Stock Returns Predictable? NBER Working Papers 30366, National Bureau of Economic Research, Inc.

Bellia, Mario; Christensen, K. K. A. P. L. R. R. (2020). High-frequency trading during flash crashes: Walk of fame or hall or shame. *SAFE Working Paper No. 270.*, 68.

Bouchaud, J.-P. (2021, dec). Radical complexity. *Entropy 23*(12), 1676.

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), 253–265.

Brolley, M. and D. Cimon (2018). Order flow segmentation, liquidity and price discovery: The role of latency delays. *Staff Working Paper 2018-16, Bank of Canada.*

Chakrabarty, B., J. Huang, and P. K. Jain (October 25, 2020). Effects of a speed bump on market quality and exchange competition. *Available at SSRN 3280645*.

Chakrabarty, B., B. Li, V. Nguyen, and R. A. Van Ness (2007, December). Trade classification algorithms for electronic communications network trades. *Journal of Banking & Finance 31*(12), 3806–3821.

Christopher, P. (2008). Liquidity is a coward. *Financial Sense Wealth Management*.

Cont, R., A. Kukanov, and S. Stoikov (2013, jun). The price impact of order book events. *Journal of Financial Econometrics 12*(1), 47–88.

Dugast, J. and T. Foucault (2018). Data abundance and asset price informativeness. *Journal of Financial Economics 130*(2), 367–391.

Easley, D., M. L. de Prado, M. O'Hara, Z. Zhang, and W. Jiang (2021). Microstructure in the Machine Age [The risk of machine learning]. *Review of Financial Studies 34*(7), 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), 1457–1493.

Fosset, A., J.-P. Bouchaud, and M. Benzaquen (2020). Non-parametric estimation of quadratic hawkes processes for order book events.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics 29*(5), 1189 – 1232.

Golub, A., J. Keane, and S.-H. Poon (2012). High frequency trading and mini flash crashes.

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), 568–604.

Kearns, M. and Y. Nevmyvaka (2013). Machine learning for market microstructure and high frequency trading.

Keller, A. J. (2012). Regulating high frequency trading after the flash crash of 2010. *Ohio State Law Journal 73*(6), 1457–1483.

Kercheval, A. N. and Y. Zhang (2015). Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance 15*(8), 1315–1329.

Kirilenko, A., A. S. Kyle, M. Samadi, and T. Tuzun (2017a). The flash crash: High-frequency trading in an electronic market. *Journal of Finance 72*(3), 967–998.

Kirilenko, A., A. S. Kyle, M. Samadi, and T. Tuzun (2017b). The flash crash: High-frequency trading in an electronic market. *The Journal of Finance 72*(3), 967–998.

Langton, J. (September 9th, 2010). Circuit breakers.

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.

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, January). A ReMeDI for Microstructure Noise. *Econometrica 90*(1), 367–389.

Liu, Bo; Xu, K. (2023a). Speed bump and stock market quality: evidence from nyse american.

Liu, Bo; Xu, K. Z. X. (2023b). Predicting mini flash crash by machine learning.

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), 277–302. PMID: 34612758.

Mohajon, J. (May 28, 2020). Confusion matrix for your multi-class machine learning model.

Murphy, K. P. (2013). *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press.

Nanex (November 29, 2010). Flash equity failures in 2006, 2007, 2008, 2009, 2010, and 2011. *Nanex*.

Nolte, D. Sec takes initial measures to avoid a second flash crash. *HGExperts.com*.

O'Hara, M. (2015). High frequency market microstructure. *Journal of Financial Economics 116*(2), 257–270.

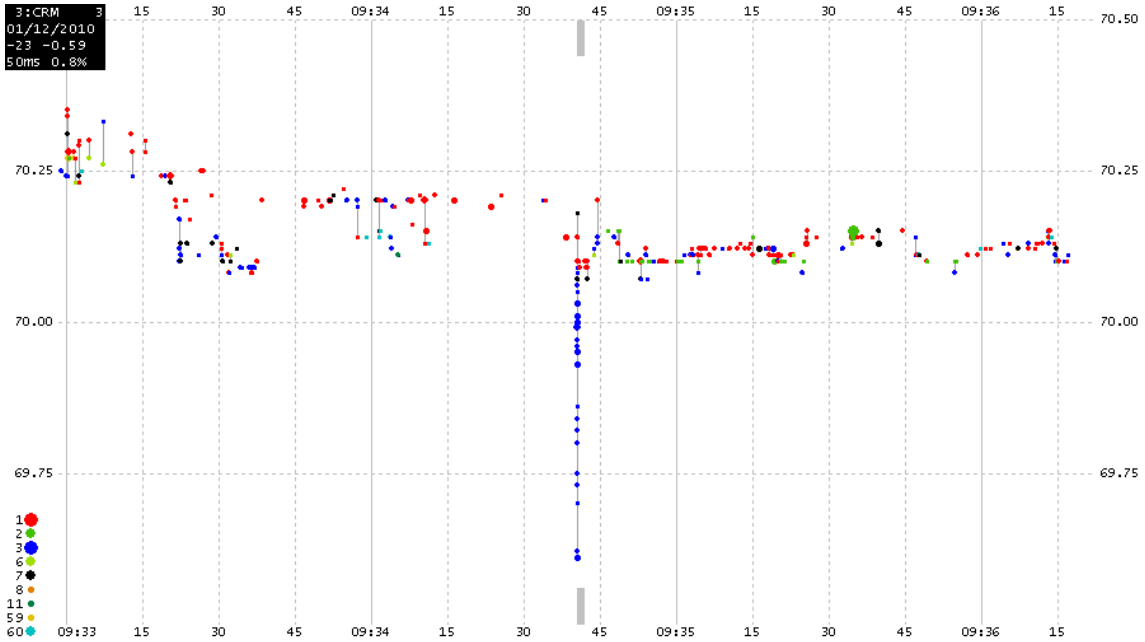
Staff, R. (Octorber 12, 2010). Canada's "flash crash" mild due to mkt structure.

XGBoostContributors (June 19, 2023). Xgboost.

Zhang, L., P. A. Mykland, and Y. Aït-Sahalia (2005). A tale of two time scales. *Journal of the American Statistical Association 100*(472), 1394–1411.

Source: Levine (2015)

Figure 1: Flash Crash, 06 May 2010



Source: Nanex (2010)

Figure 2: Mini Flash Crash, Nanex

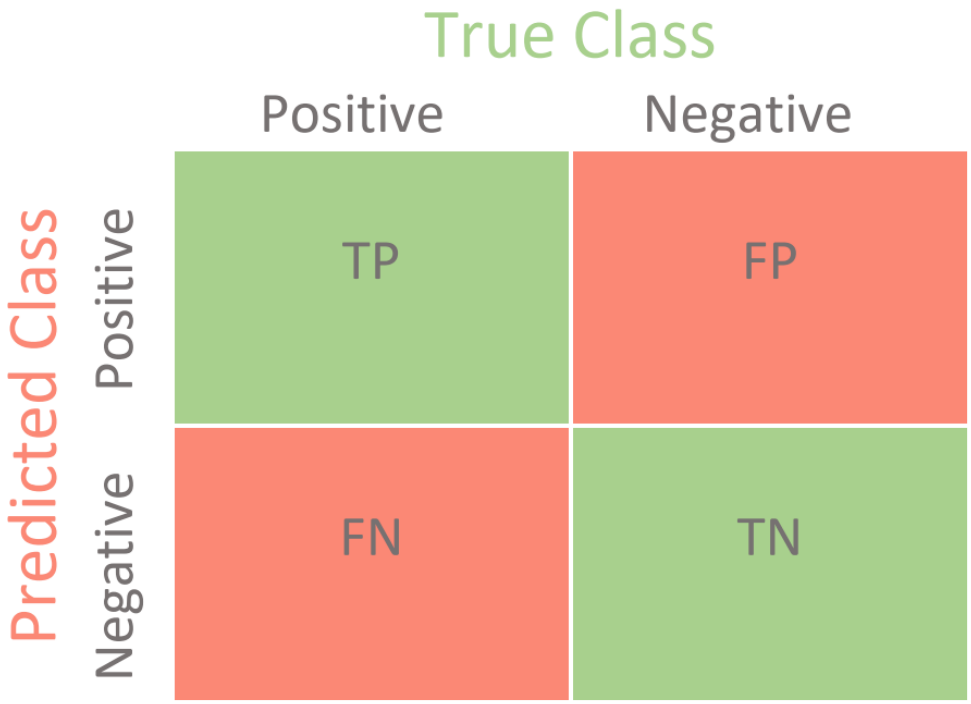


Figure 3: Confusion Matrix

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:	538409	967456	5502495	41654214	517452	931803	5319008	40265897
1:	23524	20047	3705	1386	23328	19796	3652	1183
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:	522536	939230	5319332	40266290	528162	948557	5318353	40266252
1:	22497	19181	3328	790	20913	17445	2711	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 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:	525850	927682	4867484	41127174	510225	901880	4740318	40026033
1:	16887	29235	18141	1746	15577	27867	17806	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:	507619	893250	4686171	39244403	489941	859511	4520810	37504603
1:	15383	27110	15202	1237	13767	23896	15082	1397
Ratio:	3.03%	3.03%	0.324%	3.15×10^{-3}%	2.81%	2.78%	0.334%	3.72×10^{-3}%

Table 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:	478590	724132	5504037	41654947	460088	695510	5320410	40266538
1:	6764	4528	2163	653	6600	4426	2250	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:	465068	701475	5320875	40266817	469750	708836	5319605	40266773
1:	6039	3724	1785	263	5082	3135	1459	307
Ratio:	1.30%	0.531%	0.034%	6.53×10^{-4}%	1.08%	0.442%	0.027%	7.62×10^{-4}%

Table 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:	462724	704720	3246135	21034603	451606	690915	3169370	20619802
1:	11541	7517	4917	677	10663	6276	3478	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:	447121	685425	3157324	20635919	433563	665982	3043622	20284569
1:	11073	6009	2756	361	9513	5231	3142	591
Ratio:	2.48%	0.877%	0.087%	1.75×10^{-3}%	2.19%	0.785%	0.103%	2.91×10^{-3}%

Table 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

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

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

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

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 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

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

Panel C: 15s

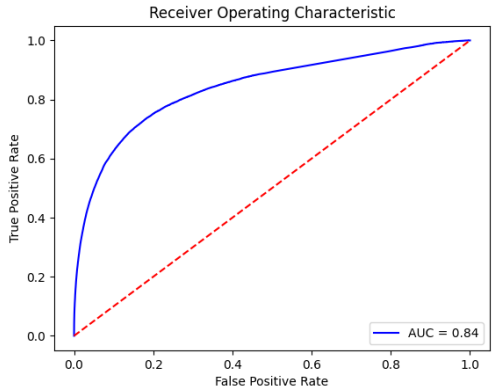
	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.691	0.691	0.692	0.688	0.720	0.719	0.721	0.718
SVM	0.686	0.688	0.688	0.684	0.716	0.716	0.718	0.715
RF	0.572	0.728	0.613	0.648	0.601	0.749	0.636	0.666
XGB	0.716	0.720	0.792	0.593	0.737	0.744	0.811	0.533

Panel D: 1.5s

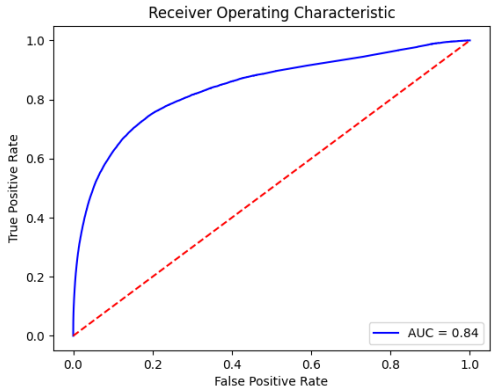
	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.823	0.823	0.821	0.820	0.824	0.800	0.822	0.815
SVM	0.826	0.821	0.825	0.822	0.817	0.804	0.818	0.809
RF	0.763	0.838	0.799	0.829	0.632	0.838	0.701	0.767
XGB	0.822	0.823	0.905	0.813	0.792	0.826	0.902	0.809

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.

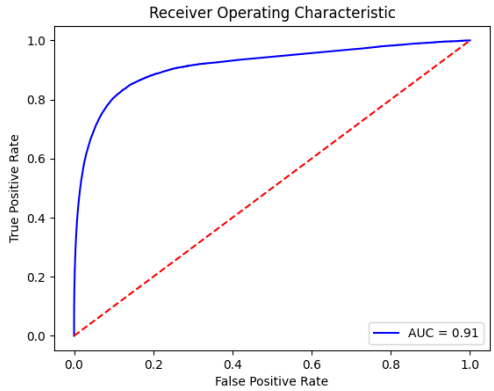
Table 6: Mini Flash Crash without reversal tuning result: Control Group



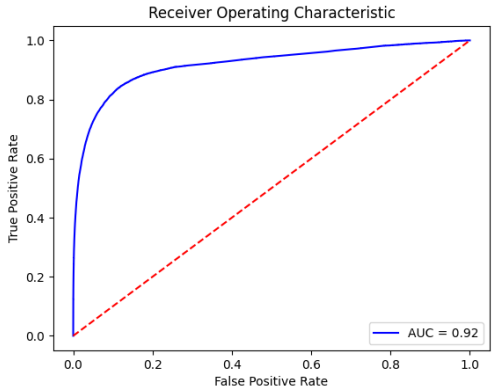
(a) Before



(b) After

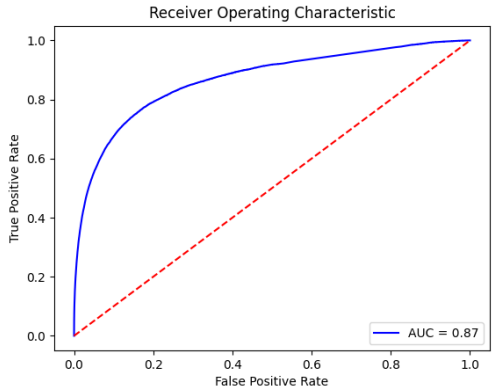


(c) Before (Control Group)

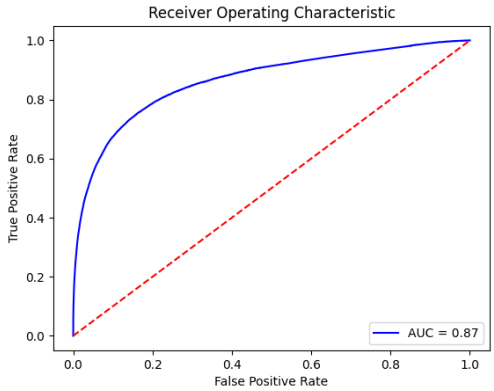


(d) After (Control Group)

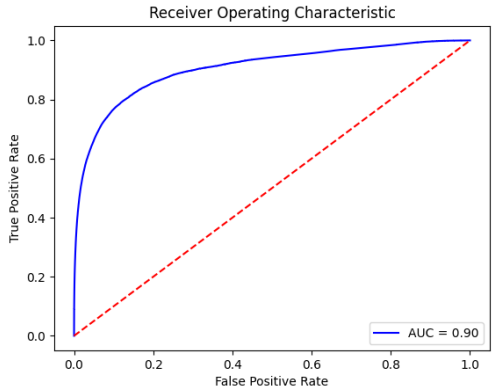
Figure 4: ROC curves of 180s without reversal



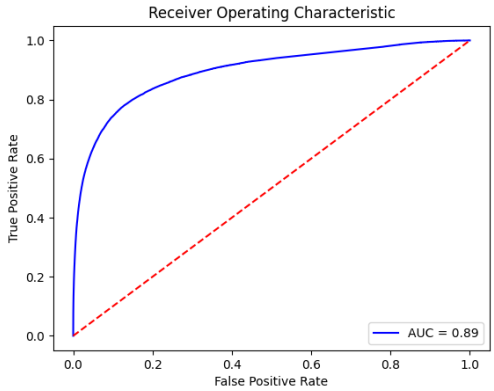
(a) Before



(b) After

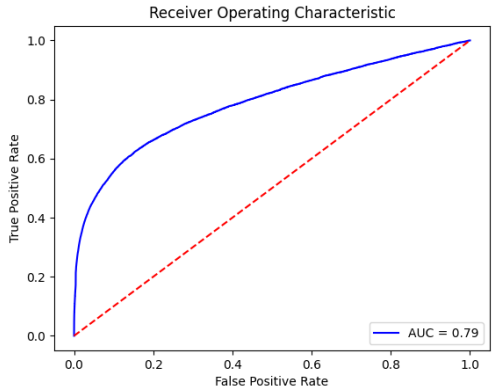


(c) Before (Control Group)

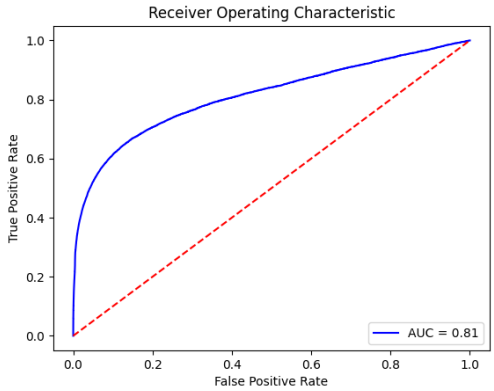


(d) After (Control Group)

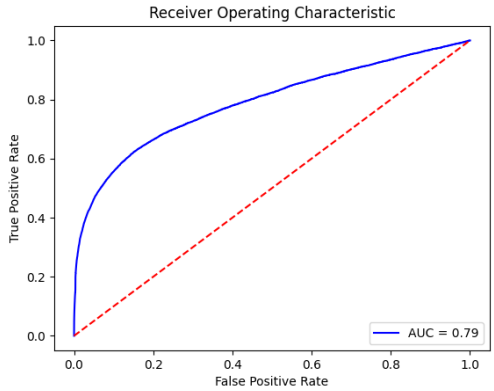
Figure 5: Tuning ROC curves of 90s without reversal



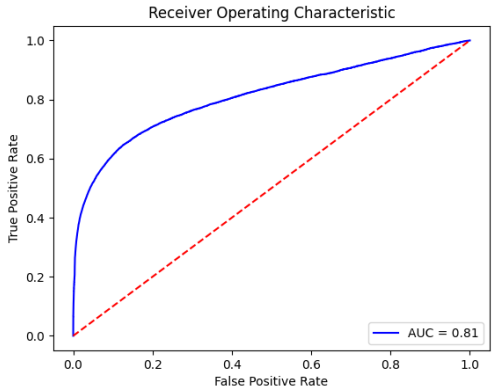
(a) Before



(b) After

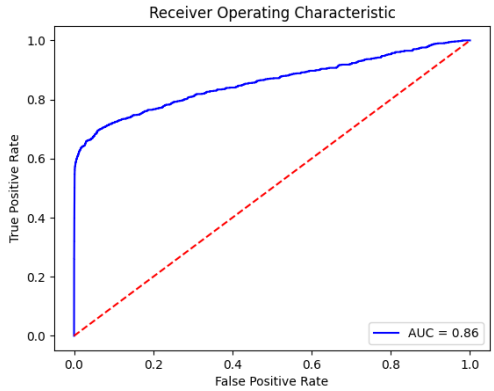


(c) Before (Control Group)

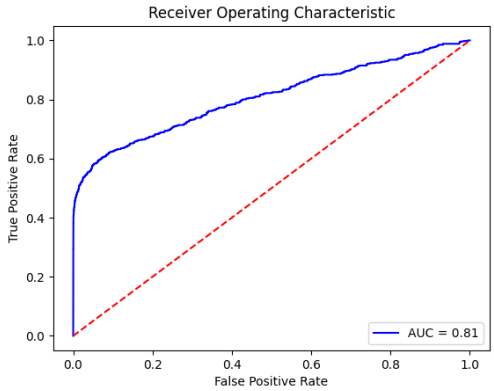


(d) After (Control Group)

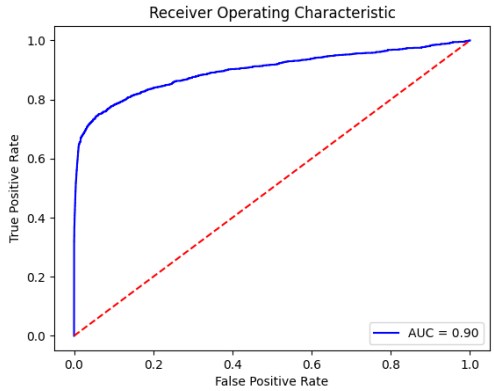
Figure 6: Tuning ROC curves of 15s without reversal



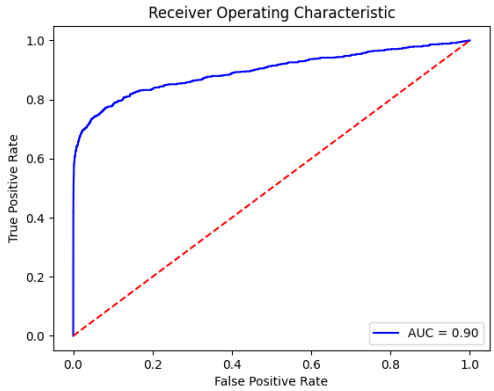
(a) Before



(b) After



(c) Before (Control Group)



(d) After (Control Group)

Figure 7: Tuning ROC curves of 1.5s without reversal

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

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

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

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

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 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

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

Panel C: 15s

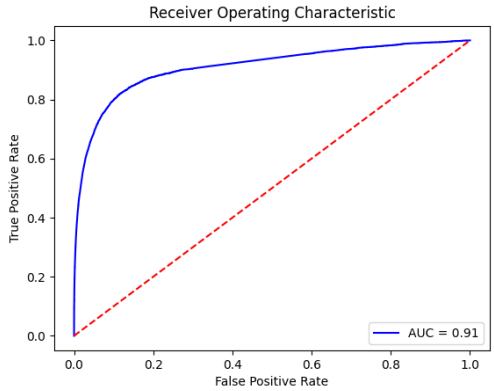
	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.816	0.815	0.817	0.806	0.786	0.773	0.782	0.781
SVM	0.813	0.810	0.814	0.804	0.778	0.774	0.783	0.774
RF	0.610	0.827	0.649	0.703	0.579	0.802	0.641	0.666
XGB	0.715	0.818	0.882	0.681	0.740	0.798	0.873	0.711

Panel D: 1.5s

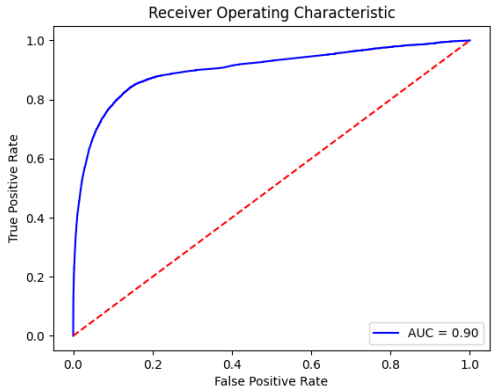
	Before				After			
	TM	UD	EN	SM	TM	UD	EN	SM
LOG	0.886	0.882	0.886	0.885	0.710	0.692	0.729	0.706
SVM	0.884	0.886	0.891	0.882	0.711	0.675	0.709	0.706
RF	0.891	0.856	0.891	0.891	0.620	0.757	0.645	0.651
XGB	0.891	0.842	0.892	0.886	0.669	0.739	0.822	0.652

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.

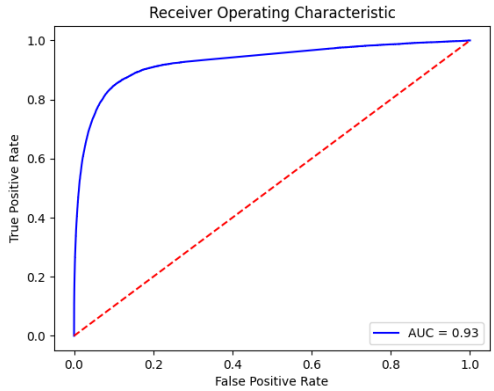
Table 8: Mini Flash Crashes with reversal tuning results: Control Group



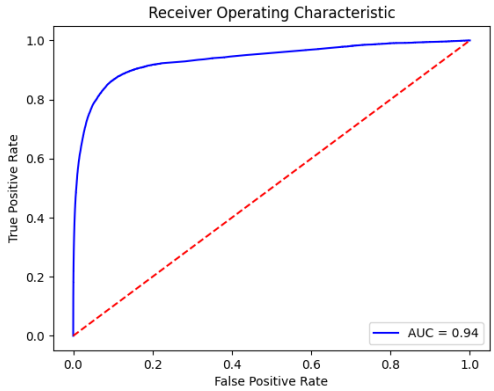
(a) Before



(b) After

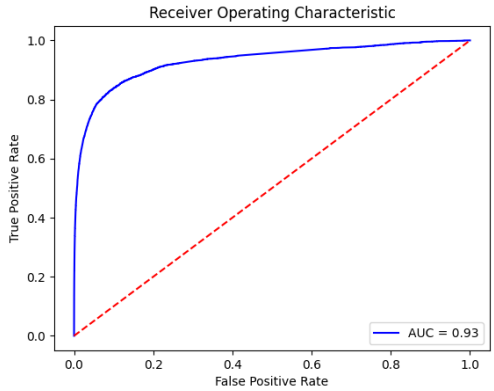


(c) Before (Control Group)

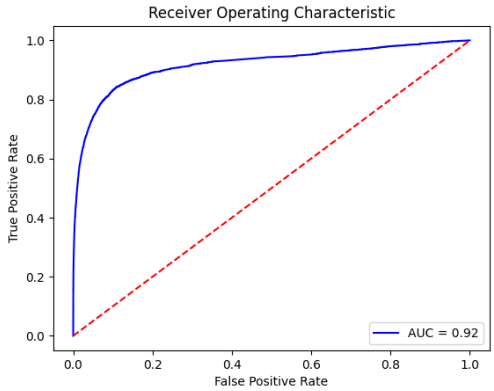


(d) After (Control Group)

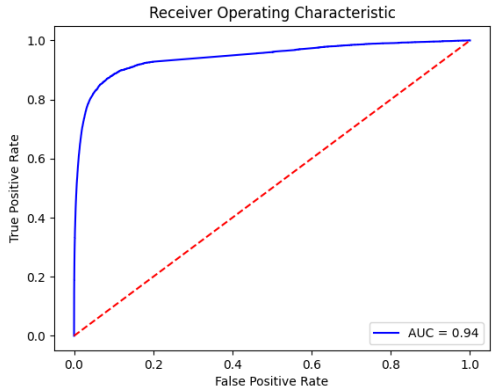
Figure 8: ROC curves of 180s with reversal



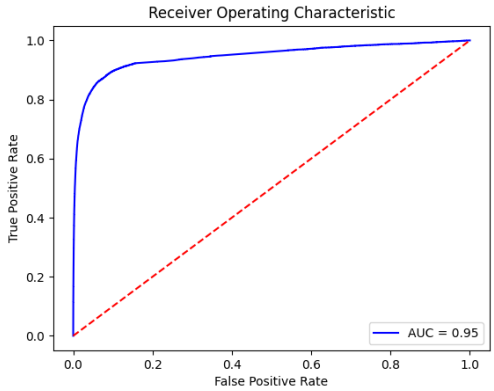
(a) Before



(b) After

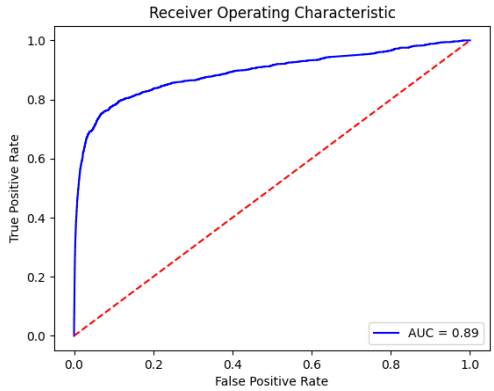


(c) Before (Control Group)

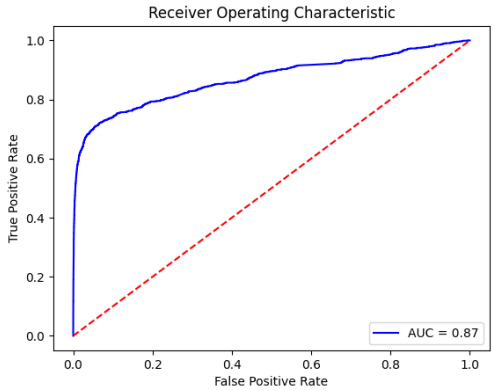


(d) After (Control Group)

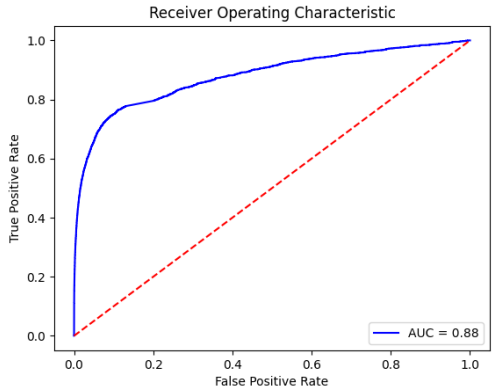
Figure 9: Tuning ROC curves of 90s with reversal



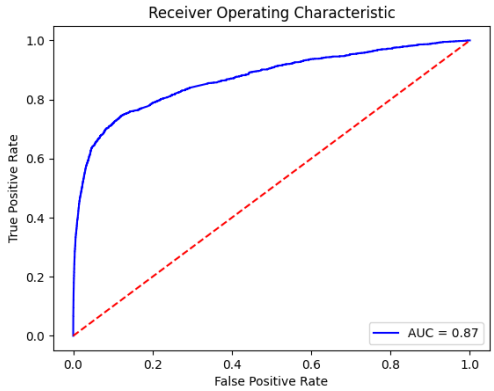
(a) Before



(b) After

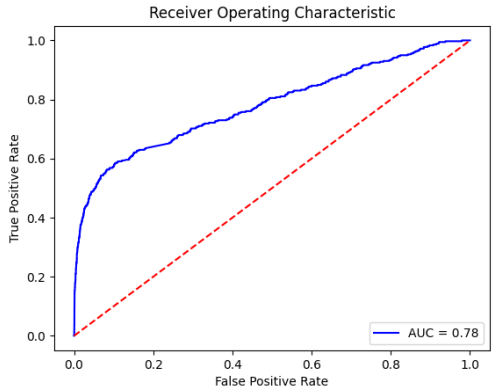


(c) Before (Control Group)

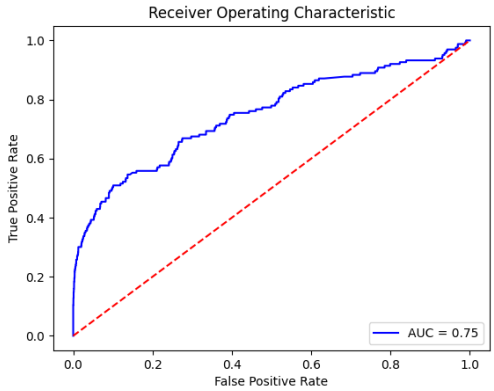


(d) After (Control Group)

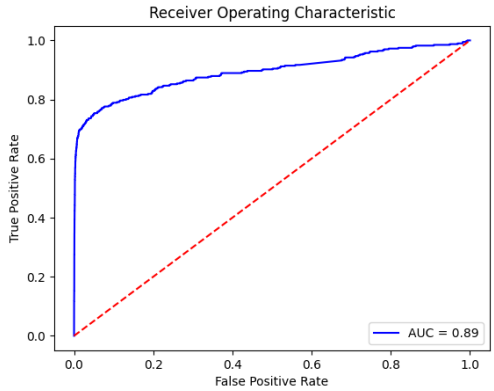
Figure 10: Tuning ROC curves of 15s with reversal



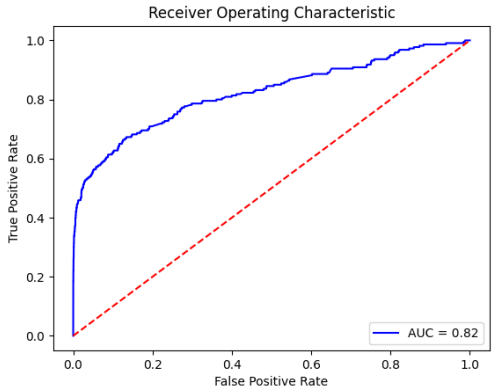
(a) Before



(b) After



(c) Before (Control Group)



(d) After (Control Group)

Figure 11: Tuning ROC curves of 1.5s 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	4486	4334	152	4486	4334	152

Panel B: 90s

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	3988	3846	142	3988	3846	142

Panel C: 15s

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	3284	3147	101	3284	3147	101

Panel D: 1.5s

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	2439	2343	96	2439	2343	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 9: Probablity 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	3958	3849	109	3958	3849	109

Panel B: 90s

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	3017	2948	69	3017	2948	69

Panel C: 15s

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	1960	1902	58	1960	1902	58

Panel D: 1.5s

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	1274	1264	10	1274	1264	10

Notes: This table reports the average values of daily average probablity 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 10: Probability of mini flash crashes with reversal

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 180 seconds mini flashs 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. 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 11: Panel regression of probability 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 90 seconds mini flashs 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. 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 12: Panel regression of probability 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 15 seconds mini flashs 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. 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 13: Panel regression of probability 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 1.5 seconds mini flashs 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. 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 14: Panel regression of probability of mini flash crashes