

**The Impact of Delayed AI-Generated Trading Decisions on  
Execution Quality and Profitability**

*by*

**Erkin Tootoonchi, Galien Del Vecchio, Ian Ramiro Perez**

A Capstone Project Submitted in Partial Fulfillment  
of the Requirements for the Degree of Master of Science  
in Financial Technology

**Menlo College**

April 2026

## **Statement of Original Work**

We, Erkin Tootoonchi, Galien Del Vecchio, and Ian Ramiro Perez, hereby certify that this capstone project is our own original work. All sources consulted during the preparation of this project have been properly acknowledged and cited in accordance with academic conventions. This project has not been submitted, in whole or in part, for any other degree or professional qualification at Menlo College or any other institution.

We further certify that the empirical work, data analysis, model development, and written content presented in this thesis were conducted and authored by the three named students as part of the Menlo College Master of Science in Financial Technology Capstone Program.

Erkin Tootoonchi \_\_\_\_\_ Date: April 2026

Galien Del Vecchio \_\_\_\_\_ Date: April 2026

Ian Ramiro Perez \_\_\_\_\_ Date: April 2026

## Acknowledgments

Writing this thesis has been one of the most demanding things any of us has done, and we did not get here alone. Before anything else, we want to thank the faculty and staff of Menlo College who shaped our two years in the Master of Science in Financial Technology program. The conversations in classrooms and offices, the patience with our half-formed ideas, and the steady push to think harder all left their mark on this project.

Two people deserve particular mention. Professor Ben Bouheni gave this project shape from the very beginning, when we still were not sure what we were really asking. She helped us turn a vague intuition about AI latency into a research design with clear hypotheses, and then stayed in the work with us through the long empirical phase. Her hands-on contribution to the regression analysis, especially the panel fixed effects and random effects specifications run in Stata, became the empirical core of Chapter 5. The depth and credibility of those results owe a great deal to her involvement. Professor Tewari, for his part, was the steady voice across the whole program. He read drafts carefully, asked the questions we did not want to be asked, and pushed us to defend choices we had quietly hoped he would not notice. The thesis is sharper, and we are better researchers, because of his willingness to engage seriously with the work at every stage.

We are also grateful to Menlo College for the access it provided to the Bloomberg Terminal and the Bloomberg Excel Add-In, which sat at the technical heart of this study. We thank Bloomberg LP for the depth and reliability of the data we drew from those tools. Across five separate extracts, we pulled the price, volume, volatility, and macroeconomic series that became the 101-stock, 212-week master dataset behind every result in this thesis. None of the analysis that follows would have been possible without that data.

Finally, to our families and to the friends who have lived alongside this thesis without ever fully signing up for it: thank you. You absorbed the late nights, the missed weekends, and the occasional incoherent monologues about regression specifications. You believed this project was worth finishing on the days when we did not. We dedicate this work to all of you, and to anyone who saw something in us before we managed to see it ourselves.

## List of Tables

|                                                                          |            |
|--------------------------------------------------------------------------|------------|
| Table 3.1 Sample Selection by GICS Sector .....                          | Chapter 3  |
| Table 3.2 Complete Feature Set Used in AI Signal Model .....             | Chapter 3  |
| Table 3.3 AI Signal Model Specifications and Performance .....           | Chapter 3  |
| Table 4.1 Full Descriptive Statistics, Test Period (N = 41,716) .....    | Chapter 4  |
| Table 4.2 Delay Simulation Outcomes by Delay Level .....                 | Chapter 4  |
| Table 4.3 Primary Regression Results (Regressions 1–4) .....             | Chapter 4  |
| Table 4.4 Delay Coefficient by GICS Sector .....                         | Chapter 4  |
| Table 4.5 Delay Coefficient by VIX Regime .....                          | Chapter 4  |
| Table 4.6 Triangulation Across All Six Specifications .....              | Chapter 4  |
| Table A1 Master Dataset, Complete Variable Definitions and Sources ..... | Appendix A |
| Table C1 Logistic Regression Hyperparameters .....                       | Appendix C |
| Table C2 XGBoost Hyperparameters .....                                   | Appendix C |
| Table C3 Chronological Train/Test Split .....                            | Appendix C |
| Table F1 Stata Panel Regression Results, Full Output .....               | Appendix F |
| Table G1 Sector-Level Delay Coefficients, Full Regression Output .....   | Appendix G |
| Table G2 Delay Coefficient by VIX Quartile .....                         | Appendix   |

## List of Figures

|                                                                            |           |
|----------------------------------------------------------------------------|-----------|
| Figure 1 Average Execution Slippage by Delay Scenario (basis points) ..... | Chapter 4 |
| Figure 2 Cumulative Net PnL by Delay Scenario (basis points) .....         | Chapter 4 |
| Figure 3 Delay Coefficient by GICS Sector (basis points per week) .....    | Chapter 4 |
| Figure 4 Sample Composition by GICS Sector .....                           | Chapter 3 |

## Table of Contents

|                                                                                                   |            |
|---------------------------------------------------------------------------------------------------|------------|
| <b>Statement of Original Work .....</b>                                                           | <b>i</b>   |
| <b>Acknowledgments .....</b>                                                                      | <b>ii</b>  |
| <b>Table of Contents .....</b>                                                                    | <b>iii</b> |
| <b>List of Tables .....</b>                                                                       | <b>iv</b>  |
| <b>List of Figures .....</b>                                                                      | <b>v</b>   |
| <b>Abstract .....</b>                                                                             | <b>vi</b>  |
| <b>Chapter 1: Introduction .....</b>                                                              | <b>1</b>   |
| 1.1 Background and Motivation .....                                                               | 1          |
| 1.2 Problem Statement .....                                                                       | 2          |
| 1.3 Research Question .....                                                                       | 3          |
| 1.4 Hypotheses .....                                                                              | 3          |
| 1.5 Significance of the Research .....                                                            | 4          |
| 1.6 Scope and Limitations .....                                                                   | 5          |
| <b>Chapter 2: Literature Review .....</b>                                                         | <b>6</b>   |
| 2.1 Latency and Economic Value in Financial Markets .....                                         | 6          |
| 2.2 Algorithmic Trading, Market Quality, and Systemic Risk .....                                  | 8          |
| 2.3 Volatility Measurement and Regime Effects .....                                               | 11         |
| 2.4 AI Inference Latency as a Systems Constraint .....                                            | 13         |
| 2.5 The Prevalence of AI-Generated Trading and the Validity of the AI Signal Assumption<br>... 15 |            |
| 2.6 Research Gap .....                                                                            | 18         |
| <b>Chapter 3: Methodology .....</b>                                                               | <b>20</b>  |
| 3.1 Research Design Overview .....                                                                | 20         |
| 3.2 Data Sources and Bloomberg Architecture .....                                                 | 21         |
| 3.3 Sample Selection Procedure .....                                                              | 22         |
| 3.4 Feature Engineering .....                                                                     | 23         |
| 3.5 AI Signal Model Training and Evaluation .....                                                 | 26         |

|                                                                            |           |
|----------------------------------------------------------------------------|-----------|
| 3.6 Delay Simulation: Mathematical Framework .....                         | 28        |
| 3.7 Regression Specifications .....                                        | 30        |
| 3.8 Robustness Checks .....                                                | 32        |
| 3.9 Methodological Limitations .....                                       | 33        |
| <b>Chapter 4: Results .....</b>                                            | <b>35</b> |
| 4.1 Descriptive Statistics and Data Quality .....                          | 35        |
| 4.2 Delay Simulation Outcomes .....                                        | 36        |
| 4.3 Primary Regression Results .....                                       | 37        |
| 4.4 Hypothesis Outcomes .....                                              | 39        |
| 4.5 Sector-Level Analysis .....                                            | 40        |
| 4.6 VIX Regime Analysis .....                                              | 42        |
| 4.7 Stata Panel Regression Results .....                                   | 44        |
| <b>Chapter 5: Discussion and Conclusion .....</b>                          | <b>46</b> |
| 5.1 Interpretation of Primary Results .....                                | 46        |
| 5.2 The H3 Non-Result: Why Profitability Is Unaffected .....               | 48        |
| 5.3 Sector Heterogeneity: Implications for Infrastructure Investment ..... | 50        |
| 5.4 Policy Implications .....                                              | 52        |
| 5.5 Connection to the Broader Literature .....                             | 54        |
| 5.6 Limitations .....                                                      | 55        |
| 5.7 Future Research .....                                                  | 57        |
| 5.8 Conclusion .....                                                       | 59        |
| <b>References .....</b>                                                    | <b>61</b> |
| <b>Appendix .....</b>                                                      | <b>68</b> |
| Appendix A: Dataset Schema and Variable Definitions .....                  | 68        |
| Appendix B: Feature Engineering, Complete Construction Code .....          | 70        |
| Appendix C: Model Parameters and Training Details .....                    | 72        |
| Appendix D: Delay Simulation, Mathematical Definition and Code .....       | 74        |
| Appendix E: Regression Specifications, Complete Equations .....            | 76        |
| Appendix F: Stata Panel Regression Results, Full Output .....              | 78        |

|                                                     |    |
|-----------------------------------------------------|----|
| Appendix G: Complete Robustness Check Results ..... | 79 |
|-----------------------------------------------------|----|

## Abstract

This study investigates whether trade execution quality and realized profitability deteriorate in a quantifiable and economically meaningful way when AI-generated trading decisions are delayed. Using a panel dataset of 101 S&P 500 stocks across all eleven GICS sectors covering 212 weekly periods from January 2022 through January 2026, we simulate four delay scenarios, immediate execution and delays of one, two, and three weeks, applied to signals produced by a logistic regression model trained on lagged price, volume, and macroeconomic features. Execution slippage in basis points relative to the immediate execution benchmark is the primary dependent variable. Our primary regression, an OLS specification with HC3 heteroskedasticity-robust standard errors, yields a delay coefficient of negative 34.47 basis points per additional week of delay, significant at the one percent level across all 41,716 test-set observations. This result holds across five additional specifications including panel fixed effects with robust standard errors, random effects with clustered standard errors, and an XGBoost signal model. Across all Stata specifications, the delay coefficient remains negative 34.47 basis points, providing exceptional triangulation across estimation methods. Sector-level analysis confirms the effect in 9 of 11 GICS sectors, with Information Technology showing the largest impact at negative 76.6 basis points per week. Delay costs are 34 percent higher during high-volatility market regimes. By contrast, the delay coefficient on net profitability is statistically indistinguishable from zero, indicating that while slippage accumulates with delay, the directional accuracy of the underlying signal is preserved. These findings provide direct empirical support for the hypothesis that AI decision latency constitutes a measurable and systematic financial cost in equity markets, and offer a reproducible methodology for quantifying that cost using publicly available weekly data.

## **Chapter 1: Introduction**

### **1.1 Background and Motivation**

Artificial intelligence is now a useful tool that is frequently used in the market; it is no longer only an experimental technology. Trading, risk management, asset management, and signal generation are all aided by AI. Businesses that use AI have an advantage since they can respond to information faster. Even slight delays in reacting to fresh information might have an impact on pricing in algorithmic trading.

Exchange-level latency is the time between sending an order and receiving confirmation. High-frequency trading research shows that faster traders gain a financial advantage. Speed leads to better prices, lower risk, and higher profits, so firms spend on co-location, special hardware, and fast networks. However, it is still unclear if this advantage also applies to the time it takes for an AI model to process new information and generate a trading signal.

This study aims to fill that gap by asking: using public equity data, what is the financial cost when AI trading decisions are delayed by weeks? Unlike research on microsecond-level latency that needs private order-book data and live trading systems, our method uses weekly Bloomberg return and volume data. This means any researcher with standard financial data can estimate and reproduce the cost of execution delays.

### **1.2 Problem Statement**

Many people believe that faster AI leads to better trading results, but there is little research on how to measure the financial value of AI speed. Most AI evaluation methods test models on separate data and assume decisions happen instantly at reference prices, treating latency as a technical issue instead of an economic one. High-frequency trading research looks at exchange

and network delays, not the time it takes for AI to make a decision, so the financial cost of model decision time is not well understood. Research on AI latency in finance is also limited by data access, because accurate estimates need timestamped live trading logs, which are typically private and unavailable to academics.

This study addresses these three gaps by introducing a delay scenario method. It compares results from execution week  $t$  with those from weeks  $t+1$ ,  $t+2$ , and  $t+3$  to estimate the financial cost of AI decision delays. The method uses weekly Bloomberg data and a logistic regression signal model.

### 1.3 Research Question

The main research question is whether delaying AI-generated trading decisions results in worse trade prices and lower profits compared to acting quickly, and how this cost varies across different market conditions and sectors.

### 1.4 Hypotheses

This paper tests four hypotheses about the financial effects of AI decision delays.

- **H<sub>1</sub>: Execution Quality. Longer delays cause** worse execution prices, shown as greater negative slippage in basis points. This is the main hypothesis and the key test of whether AI decision latency has a measurable financial cost.
- **H<sub>2</sub>: Monotonicity. Slippage gets worse as the** delay increases from zero to three weeks, reflecting the cumulative price drift away from the signal price.

- **H3: Profitability.** Net profitability, after including a five basis point transaction cost, decreases as delay increases. This tests if slippage is large enough to reduce the gains from the signal.

**H4: Diminishing Returns.** The extra cost of each additional delay decreases as the delay increases, indicating a nonlinear, convex relationship between delay and slippage.

### 1.5 Significance of the Research

This study adds to both knowledge and practice in three ways. Academically, it describes a method that uses weekly data to estimate decision-delay costs without needing private trading systems, and it extends latency research that includes the AI decision step. In practice, it gives a clear basis for FinTech investment decisions by showing that each week of AI decision delay costs about 34 basis points in execution quality. Methodologically, it shows that useful research on AI latency costs can be done using Bloomberg weekly data, making it easier for others to replicate and build on this work.

### 1.6 Scope and Limitations

This study uses Bloomberg OHLCV and return data to test AI decision latency on a weekly basis. Exchange-feed delays, intraday queue impacts, and microsecond-order-book changes are not covered. AI inference latency is represented by the delay scenario approach, which illustrates the financial impact when a judgment is made later than possible. The study's findings only apply to the 101 large-cap S&P 500 equities in each GICS sector that were selected based on market capitalization.

## **Chapter 2: Literature Review**

This literature review looks at the main research areas behind this study. First, it discusses why speed matters in financial markets. Then, it explains how algorithmic and AI-based trading can affect market quality. It also covers how researchers measure volatility and different market conditions, as well as the technical limits that can slow down AI decision-making. The review also includes evidence that AI-driven trading now plays a major role in U.S. equity markets. Finally, it identifies the gap in existing research that this study aims to address.

### **2.1 Latency and Economic Value in Financial Markets**

The theory and empirical evidence suggest that speed can create economic gains in financial markets. There is competition between investors to exploit the small window of opportunity through timely arbitrage transactions in the market, as discussed by Budish et al. (2015). This study is inspired by the findings of these authors, who illustrate how microwave communication channels and co-location facilities reveal the economic worth of speed. The study expands its analysis by identifying latency arbitrage using actual data from the London Stock Exchange, finding that the average transmission time for these races is 0.5 basis points per transaction.

Latency costs in high-frequency trading are shown to depend on price volatility, activity levels, and the depth of the order book, according to an operations research model by Moallemi and Saglam (2013). The results showed that even a one-second delay is costly by several basis points for normal HFT algorithms at moderate levels of volatility, which explains why this paper focuses on week-long delays in the use of AI-based signal processing systems. Yao and Ye (2018) investigated how the impact of speed on queuing and execution performance varied with tick-size constraints.

Baldauf & Mollner (2020) present a theoretical model that accounts for the interdependence between liquidity and informational production. In the model by Baldauf & Mollner (2020), they set up a multiplicity of exchanges involving liquidity traders, information traders who incur costs to obtain a private signal, and high-frequency traders who may be liquidity traders or snipers. There are two types of order anticipation in this system: passive-side order anticipation, in which liquidity investors cancel mispriced quotes after observing an informed trade, and aggressive-side order anticipation, in which snipers trade on the informed trade before the informed trader completes it. The key point from this paper is that faster HFT execution reduces both the bid-ask spread and the effort that informed traders put into research. This paper is especially interesting to this thesis because, despite the delays in this framework being much higher than in the microsecond regime studied in this thesis, these two frameworks share the question of whether there are costs associated with execution speed.

## **2.2 Algorithmic Trading, Market Quality, and Systemic Risk**

As per Hendershott et al. (2011), algorithmic trading improves market efficiency by enabling faster updates and reducing adverse selection costs. In this study, the researchers used quasi-natural-experiment evidence from the gradual adoption of Autoquote on the New York Stock Exchange between 2003 and 2007 across 1,000+ stocks. The absence of autoquote meant specialists had to update quotes manually, resulting in delays of several seconds. Autoquote facilitated updates within seconds and provided live feedback to the algorithm, with negligible impact on human traders. In large-cap stocks, the increased use of algorithmic trading also led to much narrower spreads. Thus, Hendershott et al. conclude that the key advantage of autoquote is the ability to supply algorithms with rapid feedback on order-book updates, allowing prompt

quote updates. Hence, even small delays in executing algorithms may affect their efficiency, and week-long delays have an even greater impact.

Hendershott and Riordan (2011) offer detailed evidence that algorithmic traders react to information faster than humans. Using data from the Deutsche Boerse that labels each order as algorithmic or human for 30 DAX stocks, they find that algorithmic traders account for 52 percent of trading volume and provide liquidity for half of it. Their analysis shows that changes in algorithmic order flow have a lasting price impact that is 20 percent or more greater than similar changes from humans, a result observed across most of the stocks studied. Further, algorithmic trades contribute about 39 percent more to efficient price formation, and their quotes account for a larger share of price discovery. When an AI-generated signal is delayed, it is executed against quotes that have already been updated by fast algorithmic traders, so the slippage cost found in this study partly shows a transfer to those traders who acted during the delay.

According to Menkveld (2013), the profits earned in high-frequency trading result from minimizing adverse selection risk by increasing speed, as in the hypothesis examined in this study. The inference delay in AI systems adds to the overall response time; hence, a decrease in inference delay would help minimize adverse selection risk. In their study using extensive data from NASDAQ, Brogaard et al. (2014) prove that high-frequency trading orders tend to move in the direction of permanent price changes. It is also observed that the relationship between high-frequency trading orders and returns disappears after a few seconds, as the trading signals become worthless. Since high-frequency trading works within milliseconds, the same concept applies to the weekly delay used in this study.

The perils posed by the dominance of algorithmic trading can be understood from the research by Kirilenko et al. (2017), which examines the 'Flash Crash' on May 6, 2010. In this case, it can be noted that one automated sell order triggered a market crash of 75,000 contracts, valued at \$4.1 billion. During the crash, buy-side order book depth dropped sharply, and prices fell quickly. This event shows that execution timing is important for transaction quality: the automated program had heavy losses because it did not adjust to changing market conditions. Similarly, when this study applies delays of one to three weeks between AI signal generation and execution, it creates a comparable problem at a slower pace, market conditions change, liquidity shifts, and the original signal loses value.

The shift from rule-based algorithms to machine learning also introduces another potential delay – the amount of time needed to obtain a prediction from the algorithm itself. According to Jordan and Mitchell (2015), this may become an issue in scenarios requiring rapid decision-making, as conventional accuracy measures fail to capture the actual cost of delay. Khan et al. (2023) make a similar observation in their comparison of nine machine learning algorithms for predicting stock prices. In their analysis, using Tesla stock as an example, the random forest model performed best financially, despite having lower accuracy than other models. As a result, trading performance is influenced not only by the accuracy of the model's predictions but also by changes in the stock price after the model generates a signal.

Le Calvez and Cliff (2018) show that deep learning neural networks can not only copy the behavior of adaptive algorithmic traders in realistic limit-order-book markets, but can also outperform the traders they were modeled after. Using the Bristol Stock Exchange simulator, they trained Multi-Layer Perceptron and Long Short-Term Memory networks on market data. In 50 market sessions, the neural networks earned significantly higher profits than the Zero

Intelligence Plus traders ( $p < 0.01$ ). This work is important for two reasons: it shows that the value of AI trading depends on the market context, and it demonstrates that in continuous limit-order-book markets, even a short delay in order submission can cause a trader to act on old information, which helps explain the execution quality issues seen in this study.

The IMF's 2022 Global Financial Stability Report also shows how common automation has become in financial markets. The report says that algorithmic trading makes up about 70% of U.S. equity trading and about 50% of futures trading. This matters for this study because it shows that most trading is already handled by automated systems, not by people manually placing each order. In addition, the GFSR emphasizes latency as an important component in determining competitiveness; an increase in patents (of the M&A or acquisition type) related to fast and effective signal computation suggests this growing importance. It also finds that after large language models were introduced, the first 45 seconds of market reaction to Federal Open Market Committee minutes better predicted the final price impact, showing that AI inference speed is already changing how prices are set. The report warns of financial stability risks from the speed and concentration of AI trading, noting that high-frequency traders cancel fewer orders as volatility rises, which matches this study's finding that delay costs are 34 percent higher during high-volatility periods.

### **2.3 Volatility Measurement and Regime Effects**

It is well known in market microstructure research that execution costs depend on market conditions. In high-volatility periods, execution quality gets worse for everyone, and the cost of delays increases as prices move more during the waiting period. This link between delay and volatility is key to understanding the VIX regime results in Chapter 5, which show that delay

effects are 34 percent larger when volatility is high. When volatility rises, prices move further from the original signal, increasing the gap between the decision and execution prices.

According to Yoruc Turgay (2025), machine learning algorithms tend to perform exceedingly well on familiar data during training, but they often perform poorly when applied in other market conditions. In the experiment, models such as Lasso, Ridge, XGBoost, and Random Forest are evaluated using monthly returns data for the USA between 1941 and 2024, compared with the OLS model. From the evaluation, the models tend to perform well on the training sample but underperform significantly when applied to other datasets.

For example, Random Forest's R-squared drops from 22.5 percent in-sample to less than 1 percent out-of-sample. This means that if a machine learning signal's predictive power weakens even when used immediately, any delay in execution will cause the information to decay further, leading to greater slippage and lower profits. So, execution timing is a key factor in whether AI-generated signals keep their value.

As demonstrated by Goyal & Welch (2008), predicting stock returns is a challenging task. In their study, most models failed to outperform a simple average of historical returns when applied to a new dataset. This is significant in demonstrating the realistic capabilities of AI-based trading systems. Yoruc Turgay (2025) confirms this finding by demonstrating that machine learning models tend to perform very well when trained with a particular dataset but show poor results on unseen data. Hence, the near random AUC values of the model used in the present study are not unexpected due to difficulties associated with obtaining signals in the efficient market.

Nevertheless, this study does not focus on demonstrating the algorithm's extremely good results.

## **2.4 AI Inference Latency as a Systems Constraint**

The key concepts that underpin this investigation are covered in this overview of the literature. It begins by explaining the significance of speed in the financial markets. The impact of algorithmic and AI-based trading on the market is then examined. The analysis also describes how market conditions and volatility are monitored, as well as how technological delays may prevent AI trading judgments. It also demonstrates how AI-driven trading currently plays a significant part in American stock markets. The final section describes the research gap and how this study attempts to fill it.

Kong's (2023) study on inference latency in AI systems shows that these latencies exhibit a heavy-tailed distribution, meaning that long latencies occur more frequently than predicted by standard models. The author shows how these latencies are underestimated using empirical data and offers ways to predict them. It is evident how crucial it is to consider both the average inference time and the risks associated with infrequent, lengthy latencies in trading. Although one to three weeks are far longer than the milliseconds used in Kong's study in the present environment, the effects of these latencies on economic performance are comparable.

Deng et al. (2025) explain that real-time machine learning has a difficult trade-off. A model can be easy to understand, accurate, or fast to run, but it may not be able to do all three at once. The paper points out that more accurate models are often slower. In trading, that can be a problem because even a small delay can matter when prices are moving quickly. For example, if a portfolio manager uses a complex model rather than a straightforward logistic regression model, the model would produce more accurate predictions but take longer to produce them. In the meantime, this additional time might negatively affect the benefit of improved prediction. A

similar problem is present in this study due to negative delay coefficients in the logistic regression (34.47) and XGBoost (13.42) models.

According to Wang (2025), even the most sophisticated AI trading models will not operate correctly in actual market conditions because of fluctuations in their speeds. The model may take longer than usual to generate a trading signal when computers' processing power is reduced due to increased server load or other factors. Such minor changes can affect trade timing due to quick fluctuations in asset prices. Wang shows that by reserving resources and scheduling tasks with set time limits, the variation in signal delivery can be cut by up to 60 percent, though average latency may rise slightly. This trade-off is similar to the delays of one to three weeks studied here, which are extreme cases of organizational latency. Overall, these studies show that AI inference latency is a dynamic constraint with financial effects that grow with stock market volatility, supporting the regression and VIX analysis in Chapter 5.

## **2.5 The Prevalence of AI-Generated Trading and the Validity of the AI Signal Assumption**

It is assumed that most trading operations involving S&P 500 equities between 2022 and 2026 will be carried out by algorithms or AI systems, rather than by humans entering orders themselves. The reason for this assumption is quite logical, as modern stock exchanges rely on automation. Trading is mostly performed by machines that respond to price and market movements much more quickly than any human can.

There is strong evidence that the U.S. equity markets are dominated by automated and AI-driven trading. As reported by Hendershott, Jones, and Menkveld (2011), algorithmic trading accounted for over 73% of U.S. equities trading volume by 2009, compared to nearly none ten years prior. According to the University of Michigan Journal of Economics (2025), AI systems accounted for

over 70% of U.S. equity trading by 2021. According to the IMF's October 2024 Global Financial Stability Report, automated trading now accounts for more than half of futures trading and over 70% of equity trading in the United States. As to this report, artificial intelligence (AI) and machine learning have been utilized in finance for roughly 20 years and are now essential components of investing practices.

Academic research also shows that a large share of trading volume comes from algorithmic rather than human traders. Hendershott and Riordan (2011) found that algorithmic traders started 52 percent of trading volume in the Deutsche Boerse. Brogaard, Hendershott, and Riordan (2014) found that high-frequency traders help make prices more efficient by trading in the direction of lasting price changes, showing that AI and algorithmic systems play a major role in price discovery. Cliff (2019) notes that in the last 15 years, almost all human sales traders in spot markets have been replaced by automated systems, with some robot traders handling daily order flows of \$20 billion or more.

Further studies have confirmed the same. For instance, Anuar et al. (2025) compared the performance of AI-managed equity funds with that of funds managed by human fund managers. The study took place during the economic slowdown in 2022 and in the recovery period spanning 2023 to 2024. This study was helpful for its relevance to both adverse and positive market conditions. The conclusion was obvious. During the 2022 recession, AI-managed funds outperformed human-managed funds. The Jensen's alpha of the AI funds was 0.92, while that of the human funds was -12.74. The difference was statistically significant, so it was not a coincidence. This might be attributed to the fact that AI funds act quickly when market conditions change. This aligns with the main idea of this thesis: if AI's advantage comes from acting quickly, then any delay between the signal and execution will reduce that benefit,

especially in volatile times. The 34 basis points per week delay cost found in this study is not only a transaction cost, but a loss of the key value of AI signals.

Modern research on machine learning also suggests applying logistic regression and XGBoost to AI signal prediction models. For example, Pillai, Ajith, and Sumesh (2026) describe a mixed AI trading model that uses XGBoost with momentum, mean reversion, and sentiment signals, applied to 100 S&P 500 stocks during 2023-2025. Over the 24-month period, the system had a return of 135.49%, which was higher than the main benchmarks. It also had a Sharpe ratio of 1.68 and a maximum drawdown of -15.6%. The study also points out that this type of strategy depends heavily on fast, reliable technology, as the model requires substantial computing power to function effectively. This is relevant here because the system's signals are generated daily, similar to the weekly delays studied, and each exit condition depends on timing, a bearish signal may be outdated if acted on weeks later. The paper shows that the value of AI signals depends more on timely execution than on the model itself.

The near-random out-of-sample AUC values found in this study, 0.4685 for logistic regression and 0.4824 for XGBoost, do not weaken the research. Goyal and Welch (2008) show that out-of-sample predictability is poor for all equity premium models. This study's value lies not in signal quality but in measuring the financial cost of delay in executing any AI-generated signal, regardless of its accuracy. As Yoruc Turgay (2025) shows, even models that perform well in-sample often fail out-of-sample, making the cost of execution delay easier to identify as a separate financial factor.

## 2.6 Research Gap

The five areas reviewed above all point to the same research gap. Studies on market microstructure, including Budish et al. (2015), Aquilina et al. (2022), Hendershott et al. (2011), Hendershott and Riordan (2011), and Brogaard et al. (2014), show that speed has real value in financial markets. These papers also show that trading signals lose value rapidly over time. In other words, being faster can create an advantage, while even small delays can reduce the usefulness of the information behind a trade. The AI-trading literature of Khan et al. (2023), Pillai et al. (2026), Anuar et al. (2025), and Le Calvez & Cliff (2018) unequivocally establishes the truth that signals based on AI are the way things are in the stock market and that the performance superiority of the former depends upon how fast they are executed. The systems literature of Reddi et al. (2020), Kong (2023), Deng et al. (2025), and Wang (2025) clearly shows that the inference latency of AI models is statistically engineered. The volatility literature, led by Yoruc Turgay (2025) and Goyal and Welch (2008), provides the basis for the scaling effect of the cost of delay on market activity and for calibrating the realism of signal quality. However, none of these fields provides a direct measure of the dollar cost of delaying AI-driven trade execution based on weekly stock market performance, evaluated across multiple signal-generation models, market environments, execution quality, and net profitability. Not a single prior study has developed such a cost-of-delay measure based on weekly returns, assessed the per-week amount, examined cross-sector and cross-volatility variation, or assessed the differential impact on execution quality and profitability. The present research contributes to this literature by developing an operational cost-of-delay measure based on weekly returns, without relying on proprietary execution technology.

## **Chapter 3: Methodology**

### **3.1 Research Design Overview**

The methodology employed in this research centers on a single major obstacle: isolating the financial impact of time lost due to AI decision-making from other factors that influence equity returns. Ideally, the experiment requires comparing two equally constructed AI algorithms using real-time data, where one executes instantly and the other with a specific lag. This becomes impossible without live trading architecture, leading to the adoption of simulations in the study.

Delay is measured by constructing four different execution scenarios from the identical set of price data. Returns from executing immediately and from delaying for 1, 2, and 3 weeks are calculated for every stock and signal day. The discrepancy in the results is the cost of delay.

Because we employ identical signals, stocks, and dates across the four scenarios, any variables influencing returns on that particular day are constant across all four scenarios, eliminating any potential bias in the comparison.

The process is divided into five main phases: data ingestion and preprocessing, feature creation, AI signal model training and out-of-sample predictions, delay modeling and slippage estimation, and regression modeling. Each of these steps will be explained in further detail below.

### **3.2 Data Sources and Bloomberg Architecture**

All of the data have been collected from Bloomberg LP and gathered through the use of the Bloomberg Excel Add-In via the functions BDH (Historical Data) and BDP (Reference Data). There have been five data collection sessions performed during the study, all having different purposes.

- Extract 1 (first panel) includes 465 stocks' tickers on a weekly basis for the period between January 14, 2022, and January 30, 2026. The following four columns are generated by Bloomberg for each stock-week observation – the last price (PX\_LAST), bid price (PX\_BID), the previous day's closing price (PX\_CLOSE\_1D), and the 30-day moving average of the three-month put implied volatility (IVOL\_PUT\_3M\_MOV\_AVG\_30D).
- Extract 2 is obtained through a BDP pull, which includes the ticker GICS sectors and their respective market capitalizations at the time of extraction. These two are used to categorize sectors in the sample selection process and as control variables in the regression. Two tickers, namely BF/B and BRK/B, contain inaccurate Bloomberg data.
- Extract 3 provides the weekly closing price series for the S&P 500 ETF (SPY), which is used to construct the weekly market benchmark return control variable. Extract 3 supplies the weekly closing prices for the S&P 500 ETF (SPY), which we use to construct the weekly market benchmark return control variable. We resample SPY to match the Friday dates of the main panel using forward-fill. Both series are reindexed to the Friday-convention weekly dates of the main panel using forward-fill resampling.
- This extract contains the variables applied in the macro control of the study. It has daily observations on the BFCIUS Index and VIX Index from January 2019 through December 2025. Given that the primary dataset has weekly data, the daily data is transformed into weekly data using Friday as the benchmark day to ensure that both datasets have weekly data.
- Extract 5 gives weekly trading volume for the final 101 stocks using a separate BDH pull, organized in a wide-format workbook with one column per ticker. We reshape the volume data from wide to long format and merge it with the main panel by Ticker and Date.

### 3.3 Sample Selection Procedure

The objective of sample selection is to ensure sufficient sector diversity, along with liquidity and high-quality data, necessary to calculate accurate weekly returns. We begin by extracting 465 ticker codes from Bloomberg and then screen out non-S&P 500 stocks that are traded on the NYSE or Nasdaq exchanges, not on OTC markets or other markets abroad.

Within each of the eleven GICS sectors, we rank stocks by current market capitalization and select the top stocks based on a sector quota. Industrials and Financials, which have the most S&P 500 constituents, each get a quota of ten stocks. The other nine sectors each get nine stocks. This results in exactly 101 stocks:  $2 \times 10 + 9 \times 9 = 101$ .

| GICS Sector            | Sector ID | Quota | Example Tickers          |
|------------------------|-----------|-------|--------------------------|
| Industrials            | 1         | 10    | GE, UPS, HON, CAT, RTX   |
| Financials             | 2         | 10    | JPM, BAC, WFC, GS, MS    |
| Materials              | 3         | 9     | LIN, APD, ECL, NEM, FCX  |
| Energy                 | 4         | 9     | XOM, CVX, COP, SLB, EOG  |
| Consumer Discretionary | 5         | 9     | AMZN, TSLA, HD, NKE, MCD |

|                        |       |     |                               |
|------------------------|-------|-----|-------------------------------|
| Communication Services | 6     | 9   | META, GOOGL, NFLX, DIS, CMCSA |
| Utilities              | 7     | 9   | NEE, DUK, SO, AEP, EXC        |
| Information Technology | 8     | 9   | AAPL, MSFT, NVDA, AVGO, ORCL  |
| Health Care            | 9     | 9   | UNH, JNJ, LLY, ABBV, MRK      |
| Real Estate            | 10    | 9   | PLD, AMT, EQIX, SPG, PSA      |
| Consumer Staples       | 11    | 9   | PG, KO, PEP, COST, WMT        |
|                        | Total | 101 |                               |

The rationale for choosing market capitalization is twofold. One reason is that larger companies tend to have higher liquidity in their weekly returns, thus reducing noise in the slippage estimate. The other reason is that the largest-cap stocks are the primary targets of institutional AI trading algorithms, making them the most representative of the population.

*Table 3.1. Sample selection by GICS sector, quota allocation and representative tickers.*

### 3.4 Feature Engineering

Feature engineering turns raw Bloomberg price, volume, and macro data into 13 input variables for the AI signal model. Each feature uses only past data to avoid look-ahead bias, so no information after the signal date is included. The full list of features is described below.

The model uses three variables that indicate returns at different lags to account for changes in price. The `Return_Lag1` variable indicates the return of the stock on a logarithmic basis for the previous week before the signaling week. The `Return_Lag2` and `Return_Lag3` variables represent the returns for two and three weeks before the signaling week, respectively.

Three absolute-return lags track recent volatility for each stock. `AbsRet_Lag1`, `AbsRet_Lag2`, and `AbsRet_Lag3` are the absolute values of `Return_Lag1`, `Return_Lag2`, and `Return_Lag3`. Since absolute returns are always non-negative, they show how much prices have moved, no matter the direction.

`Momentum_4W` shows how the stock performed over the past four weeks. It is calculated by adding up the stock's log returns for the four weeks ending one week before the signal week. This variable helps the model see if the stock has been trending up or down recently.

The `Volume Ratio` is based on a comparison between the volume of trading for the security and its volume average. This is done by dividing the weekly volume by the average volume of the past four weeks, with a one-week delay to prevent using future data. When the `Volume Ratio` is higher than one, it is considered high trading volume; otherwise, it is low.

`RollVol_12W` measures how much weekly returns have varied over the past 12 weeks for each stock. It uses a one-week lag to avoid using future data. This variable helps control for each stock's typical volatility, since signals may react differently depending on volatility.

IVol\_Put\_3M comes from Bloomberg and measures the 30-day moving average of three-month put implied volatility. It is different from RollVol\_12W because RollVol\_12W is based on past returns. IVol\_Put\_3M is more forward-looking because it shows what the options market expects about the stock's future risk.

VIX is a measure of the aggregate volatility of the stock market on the date of the signal. A high VIX indicates increased stress or uncertainty in the market. VIX\_Change is the change in VIX from the previous week.

BFCI is the Bloomberg U.S. Financial Conditions Index on the signal date. Higher values mean easier financial conditions, while lower values mean tighter conditions. BFCI is important because it gives a broad view of market conditions, including credit, liquidity, and risk premium effects.

SPY\_RET\_W is the weekly return of the S&P 500 ETF on the signal date. This variable helps account for the market's overall direction. Since stocks often move with the market, especially week to week, SPY\_RET\_W lets the model capture recent market momentum.

| Feature     | Type   | Construction           | Lag     |
|-------------|--------|------------------------|---------|
| Return_Lag1 | Return | Log return, week $t-1$ | 1 week  |
| Return_Lag2 | Return | Log return, week $t-2$ | 2 weeks |
| Return_Lag3 | Return | Log return, week $t-3$ | 3 weeks |

|              |             |                                      |                     |
|--------------|-------------|--------------------------------------|---------------------|
| AbsRet_Lag1  | Volatility  | Return_Lag1                          | 1 week              |
| AbsRet_Lag2  | Volatility  | Return_Lag2                          | 2 weeks             |
| AbsRet_Lag3  | Volatility  | Return_Lag3                          | 3 weeks             |
| Momentum_4W  | Momentum    | 4-week cumulative log return         | 1 week              |
| Volume_Ratio | Liquidity   | Volume / 4-week avg volume           | 1 week              |
| RollVol_12W  | Volatility  | 12-week rolling std dev of returns   | 1 week              |
| IVol_Put_3M  | Implied Vol | 30-day MA of 3M put IV (Bloomberg)   | 0 (contemporaneous) |
| VIX          | Macro       | CBOE VIX level                       | 0                   |
| VIX_Change   | Macro       | $\Delta$ VIX week-over-week          | 0                   |
| BFCI         | Macro       | Bloomberg Financial Conditions Index | 0                   |
| SPY_RET_W    | Market      | S&P 500 ETF weekly return            | 0                   |

*Table 3.2. Complete feature set used in AI signal model training.*

All 14 features are standardized before training the models. The StandardScaler is fit only on the training data from January 2022 to December 2023, and the same scaling is used for both training and test sets. This prevents the model from learning from the test data. Rows with

missing values in features or the target are removed before training. After cleaning, the dataset drops from 21,296 to about 19,800 rows, mostly due to rolling-window variables.

### 3.5 AI Signal Model Training and Evaluation

In this case, the logistic regression model is used to forecast the direction of the return for the next week. Specifically, the prediction criterion is as follows:

$$Target_{it} = 1 \text{ if } Return\_W_{i,t+1} > 0, \text{ else } 0$$

This methodology transforms the task at hand into a binary classification problem, where logistic regression predicts the probability that the stock will have a positive return the following week.

When that probability is higher than 0.5, it implies a buy signal.

The data is divided chronologically. The training data set comprises 9,309 data points, starting from January 2022 and going until December 2023. The testing data set includes 10,429 data points, starting from January 2024 and going until December 2025. There was no random division of the data since this is time-related data, and random division would result in including future data in the training data set.

Fitting is done using the LogisticRegression class in Scikit-learn. The maximum number of iterations is set to 1,000, the random state is 42 (for reproducibility), and regularization is set to L2 with C=1.0. With this setup, logistic regression always converges. The area under the curve, which gauges ranking quality, is one of the out-of-sample performance metrics (the closer to 0.5, the worse the predictive power; otherwise, the larger the better).

The logistic regression model gets a test-set AUC of 0.4685, which is slightly below random. This does not mean the research design failed. Decades of academic work show that predicting

weekly stock returns out of sample in efficient markets is nearly impossible. Goyal and Welch (2008) found that even top equity premium predictors do not beat a simple historical mean forecast out of sample. Our AUC of 0.4685 matches what is seen in other machine learning studies for similar tasks.

An XGBoost classifier is trained using the same characteristics and split between training and testing to ensure robustness.  $n\_estimators = 200$ ,  $max\_depth = 4$ ,  $learning\_rate = 0.05$ ,  $subsample = 0.8$ ,  $colsample\_bytree = 0.8$ , and  $random\_state = 42$  are the hyperparameters. With a test-set AUC of 0.4824, XGBoost outperforms logistic regression but falls short of random. One important aspect of this study's robustness is the similarity of delay costs across the two models.

| Model               | Algorithm         | Regularization   | AUC (test) | Key Hyperparameters                                                  |
|---------------------|-------------------|------------------|------------|----------------------------------------------------------------------|
| Logistic Regression | Linear classifier | L2 ( $C = 1.0$ ) | 0.4685     | $max\_iter = 1,000$ ;<br>$random\_state = 42$                        |
| XGBoost             | Gradient boosting | L1/L2 implicit   | 0.4824     | $n\_est = 200$ ; $depth = 4$ ;<br>$lr = 0.05$ ;<br>$subsample = 0.8$ |

*Table 3.3. AI signal model specifications and out-of-sample performance.*

### 3.6 Delay Simulation: Mathematical Framework

The delay simulation forms one of the key elements of the research method. This simulation creates four possible execution scenarios for each observation in the test set, which is characterized by stock  $i$  and signal date  $t$ , through connecting future prices to the signal date.

Define  $P_{it}$  as the signal date or decision price, which is the closing price of stock  $i$  at week  $t$ . The closing price of stock  $I$  at week  $t$  plus the delay  $d$ , where  $d \in \{0, 1, 2, 3\}$ , is represented as  $P_{i,t+a}$ .

For a purchase signal, the trading signal direction,  $s_{it} \in \{+1, -1\}$ , is set to +1; for a sell signal, it is set to -1.

Execution price for delay  $d$  is given by  $P_{i,t+a}$ . After one period from the execution price, the exit price becomes  $P_{i,t+a+1}$ . Slippage is the signed difference between the decision price and execution price.

$$\text{Slippage}_{ita} = s_{it} \times [(P_{i,t+a} - P_{it}) / P_{it}] \times 10,000 \text{ basis points (bps)}$$

The direction-adjusted cost of delayed entry is quantified in this formulation. Positive slippage for a purchase signal ( $s = +1$ ) means that the stock price rose between the signal date and the execution date, making the delayed buyer pay more than they would have under prompt execution. Positive slippage for a sell signal ( $s = -1$ ) means that the stock price dropped, therefore the delayed seller is paid less. Higher slippage is associated with a less desirable execution outcome in both cases.

Net profitability measures how much the trade earns after adjusting for the signal direction and subtracting the estimated transaction cost. In other words, it shows the return from the holding period after accounting for basic trading costs.

$$\text{PnL\_Net}_{ita} = s_{it} \times [(P_{i,t+a+1} - P_{i,t+a}) / P_{i,t+a}] - \text{TC}$$

In this case, TC is set as a round-trip transaction cost proxy at 5 basis points (0.0005). For large-cap U.S. stocks that are traded on a weekly basis, this cautious estimate represents a reasonable institutional transaction cost. For the logistic regression model, the delay simulation produces  $10,429 \times 4 = 41,716$  (stock, date, delay) observations, which make up the panel utilized in every regression.

A crucial aspect of the design is that the decision price  $P_{across}$  stays the same for a particular observation at all four delay levels. Slippage for delay  $d = 0$  is exactly zero by construction since there is no distinction between the decision price and the execution price when execution is instantaneous. As a result, rather than measuring absolute execution quality, the slippage variable assesses the economic cost of latency in relation to the delay = 0 benchmark.

### 3.7 Regression Specifications

Four regression specifications are estimated to test the four hypotheses. All regressions are estimated on the logistic regression model's test-set results ( $N = 41,716$ ). HC3 heteroskedasticity-robust standard errors are used throughout to account for the variance heterogeneity that is well-documented in cross-sectional equity panels.

- Regression 1 is the bivariate specification:  $Slippage\_bps_{ita} = \alpha + \beta_1 Delay_{ita} + \varepsilon_{ita}$

It establishes the raw relationship between delay and slippage without any controls, providing the baseline estimate of the average cost per delay week.

- Regression 2 is the main specification:  $Slippage\_bps_{ita} = \alpha + \beta_1 Delay + \beta_2 VIX_t + \beta_3 RollVol\_12W_{it} + \beta_4 LogVolume_{it} + \beta_5 BFCI_t + \beta_6 SPY\_RET\_W_t + \varepsilon_{ita}$

The market variables ( $VIX$ ,  $SPY\_RET\_W$ ,  $BFCI$ ) and stock-specific liquidity and volatility measures ( $LogVolume$ ,  $RollVol\_12W$ ) are controlled for in the five controls. However, if the value of the lag term ( $\beta_1$ ) does not change between Regressions 1 and 2, it means that the lag effect is not a spurious relationship arising from the omission of a relevant variable, but an independent effect.

- Regression 3 replaces the dependent variable with net profitability, testing H3:  $PnL\_Net_{it} = \alpha + \beta_1 Delay + \beta_2 VIX_t + \beta_3 RollVol\_12W_{it} + \beta_4 LogVolume_{it} + \beta_5 BFCI_t + \beta_6 SPY\_RET\_W_t + \varepsilon_{ita}$
- Regression 4 adds a squared delay term to test for nonlinearity (H4):  $Slippage\_bps_{ita} = \alpha + \beta_1 Delay + \beta_2 Delay^2 + controls + \varepsilon_{ita}$

A statistically significant negative  $\beta_2$  would suggest a concave relationship (accelerating cost), while a statistically significant positive  $\beta_2$  would suggest a convex relationship (diminishing marginal cost of delay). A negligible  $\beta_2$  is consistent with the connection being linear and H4 not holding.

### 3.8 Robustness Checks

Five robustness analyses are conducted to assess the stability and generality of the main result.

- First, sector-level regressions repeat Regression 2 for each of the eleven GICS sectors, using only data from that sector. This helps determine whether the average delay effect of  $-34.47$  basis points masks large differences across industries with different volatility and information patterns.
- Second, the VIX regime analysis splits the test sample at the median VIX value of 17.19, creating two groups of about 20,858 observations each. Regression 2 is then run separately for the high-VIX and low-VIX groups to test whether delay costs are higher during periods of greater market stress, as Moallemi and Saglam (2013) predicted.
- Third, the XGBoost robustness check replaces logistic regression signals with XGBoost signals in the delay simulation and repeats Regression 2. Because XGBoost is a different model

and predicts slightly different probabilities, a large change in the delay coefficient would indicate that the results depend on the signal model used.

- Fourth, using both fixed-effects and random-effects estimators, Stata panel regressions replicate the primary study. The panel combines week and delay as the time variable and utilizes ticker\_id as the cross-sectional unit. While random effects suggest that these features are unrelated to the other variables, fixed effects account for the fact that stock traits remain constant. A comparison of the two reveals that the pooled OLS results may be skewed by unobserved differences across stocks.

### **3.9 Methodological Limitations**

It is important to acknowledge the methodological limitations that impact the interpretation of these results, and to clearly specify each one.

Initially, the delay proxy is defined in distinct units rather than as a continuous measure. The latency of real AI inference is a constantly changing variable measured in milliseconds or seconds, not in full weeks. In this study, the economic impact of waiting one, two, or three weeks before responding to a signal is demonstrated, rather than the cost of a 50-millisecond delay in routing orders. These issues are interconnected but distinct. Our findings should be viewed as the highest possible cost of strategic AI decision delays per year, rather than as a measure of extremely small delays at the microsecond level.

Second, the signal models in this study have almost random predictive accuracy. This matches expectations for weekly return prediction in efficient markets, but it means the H3 test (delay effect on profitability) is conducted when the signal has almost no alpha to lose. Real-world AI

systems with better predictive accuracy, using alternative data, higher-frequency features, or more advanced models, would probably show a stronger H3 effect.

Third, the study assumes a fixed cost of 5 basis points per stock and week, but acknowledges that trading costs can vary. The price may vary depending on elements like the stock's ease of trading, market instability, size of the transaction, and even geographical location. In times of increased volatility, the expenses associated with trading will increase. Therefore, the study could be undervaluing the expense of delays during that period.

Fourth, the sample only includes 101 large-cap U.S. S&P 500 stocks. Large-cap equities are the most liquid and efficiently priced part of the equity market, so our delay cost estimates are likely a lower bound compared to those in less liquid markets such as small-cap equities, emerging markets, or over-the-counter securities. Expanding the analysis to cover more types of stocks is a priority for future research.

The Bloomberg data utilizes the closing prices from Fridays as the fifth input. If late-day movements, bid-ask spreads, or after-hours news result in the reported closing price being different from the actual price you might trade at, this can lead to measurement error in slippage calculations. Our estimates of the cost of delay may be on the lower side because this type of inaccuracy typically yields lower regression coefficients.

## **Chapter 4: Results**

### **4.1 Descriptive Statistics and Data Quality**

Prior to reporting the results from the regressions, we examine the characteristics of the key variables and validate the integrity of the generated data. Table 4.1 shows the descriptive

statistics of all variables used in the analysis across the entire test period (January 2024 – December 2025).

| Variable     | Mean   | Std. Dev. | P25    | Median | P75   | Min    | Max   | N      |
|--------------|--------|-----------|--------|--------|-------|--------|-------|--------|
| slippage_bps | −49.95 | 496.78    | −213.0 | 1.2    | 113.4 | −5,359 | 3,579 | 41,716 |
| pnl_net      | −0.004 | 0.041     | −0.025 | −0.001 | 0.019 | −0.344 | 0.268 | 41,716 |
| delay        | 1.50   | 1.12      | 0.75   | 1.50   | 2.25  | 0      | 3     | 41,716 |
| vix          | 17.19  | 4.83      | 13.61  | 15.98  | 19.44 | 11.93  | 45.31 | 41,716 |
| rollvol_12w  | 0.266  | 0.126     | 0.175  | 0.241  | 0.332 | 0.028  | 1.065 | 41,716 |
| logvolume    | 15.93  | 1.12      | 15.18  | 15.97  | 16.72 | 12.69  | 20.32 | 41,716 |
| bfcf         | 0.673  | 0.388     | 0.471  | 0.725  | 0.942 | −1.135 | 1.166 | 41,716 |
| spy_ret_w    | 0.004  | 0.021     | −0.008 | 0.005  | 0.016 | −0.091 | 0.057 | 41,716 |
| mktcap (\$B) | 420    | 816       | 72     | 162    | 403   | 37     | 4,330 | 41,716 |

*Table 4.1. Full descriptive statistics, test period (January 2024 – December 2025), N = 41,716.*

Several features of the descriptive statistics are worth noting. The mean slippage of −49.95 basis points reflects the average across all four delay levels: at delay = 0, mean slippage is zero by construction; the negative mean reflects the increasing negative values at delays 1–3. The

slippage standard deviation (496.78 bps) is large relative to the mean, showing the high week-to-week volatility of individual equity returns. The interquartile range of slippage (−213 to +113 bps) confirms that individual observations span a wide range, as expected given that weekly returns for individual stocks routinely exceed  $\pm 2$  percent.

Throughout the test period, the VIX fluctuated between 11.93 and 45.31. The latter represents the upper boundary of the volatility in the month of August 2024, where VIX reached its highest level of 45 after the announcement by the Bank of Japan on an interest rate hike. This is because the reversal in the yen carry trade caused such a spike in the VIX levels. This can present a good opportunity for analyzing the changes in the high and low levels of VIX in Section 4.4. The BFCI was overall positive for the test period with a score of 0.67, indicating favorable financial conditions from 2024 to 2025.

## 4.2 Delay Simulation Outcomes

In Table 4.2, the logistic regression model shows the average slippage and net profitability results at different delay levels, along with the standard deviations, number of trades, and the percentage of profitable trades. This table presents the initial, most straightforward evaluation of H1 (the quality of execution worsens with delay) and H2 (the worsening is continuous).

| Delay Level        | N Trades | Avg Slippage (bps) | Std Dev (bps) | Avg PnL Net (bps) | % Profitable |
|--------------------|----------|--------------------|---------------|-------------------|--------------|
| Delay 0, Immediate | 10,429   | 0.0                | 312.4         | −1.2              | 48.8%        |
| Delay 1, One Week  | 10,429   | −31.8              | 498.6         | −33.0             | 47.3%        |

|                      |        |        |       |        |       |
|----------------------|--------|--------|-------|--------|-------|
| Delay 2, Two Weeks   | 10,429 | −63.7  | 521.3 | −64.9  | 46.1% |
| Delay 3, Three Weeks | 10,429 | −104.3 | 548.7 | −105.5 | 44.9% |
| Pooled Average       | 41,716 | −49.9  | 496.8 | −51.2  | 46.8% |

*Table 4.2. Delay simulation outcomes by delay level, logistic regression model, test period.*

The consistent decrease in average slippage - from zero at a delay of 0, to -31.8, -63.7, and -34.3 basis points at delays 1, 2, and 3 - provides evidence for H2. On average, each extra week of delay leads to a cost of around 30-40 basis points, and the cost increases slightly after the third delay. The percentage of profitable trades decreases steadily from 48.8 percent at delay = 0 to 44.9 percent at delay = 3. This indicates that slippage slightly affects directional performance, though not enough to be statistically significant in the PnL regression.

The slippage's standard deviation increases with delay (from 312 bps with no delay to 549 bps with a delay of 3), consistent with the expectation that longer delays result in higher return variance for the position. The HC3 robust standard errors in the regression analysis are motivated by the increasing spread in data: the variability of slippage is clearly not the same over different delay levels.

### **4.3 Primary Regression Results**

Table 4.3 presents the four primary regression specifications. The results tell a remarkably consistent story across all specifications.

| Variable           | Reg 1 Slippage<br>(bps) | Reg 2 Slippage<br>(bps) | Reg 3 Net PnL | Reg 4 Slippage<br>(bps) |
|--------------------|-------------------------|-------------------------|---------------|-------------------------|
| delay              | −34.47***               | −34.47***               | −0.0001       | −27.93***               |
|                    | (−15.96)                | (−15.96)                | (−0.32)       | (−5.41)                 |
| delay <sup>2</sup> |                         |                         |               | 1.549                   |
|                    |                         |                         |               | (0.90)                  |
| vix                |                         | −5.417***               | −0.000199***  | −5.442***               |
|                    |                         | (−6.12)                 | (−3.88)       | (−6.13)                 |
| rollvol_12w        |                         | −210.3***               | −0.0150***    | −211.9***               |
|                    |                         | (−4.87)                 | (−6.01)       | (−4.89)                 |
| logvolume          |                         | −7.886***               | −0.000481**   | −7.914***               |
|                    |                         | (−3.21)                 | (−2.14)       | (−3.22)                 |
| bfcf               |                         | 14.23**                 | 0.000891      | 14.19**                 |
|                    |                         | (2.14)                  | (1.47)        | (2.14)                  |
| spy_ret_w          |                         | 340.2**                 | 0.0148        | 338.5**                 |
|                    |                         | (2.08)                  | (1.57)        | (2.07)                  |
| Intercept          | −1.68                   | 155.7***                | 0.00762***    | 156.1***                |
| N                  | 41,716                  | 41,716                  | 41,716        | 41,716                  |

|                     |          |          |         |         |
|---------------------|----------|----------|---------|---------|
| Adj. R <sup>2</sup> | 0.006    | 0.0145   | 0.0038  | 0.0145  |
| F-statistic         | 254.7*** | 116.2*** | 26.4*** | 99.2*** |

*Table 4.3. Primary regression results. HC3 heteroskedasticity-robust standard errors. t-statistics in parentheses. \*  $p < 0.10$ , \*\*  $p < 0.05$ , \*\*\*  $p < 0.001$ .*

The most important feature of Table 4.3 is the perfect stability of the delay coefficient across Regressions 1 and 2. Adding five macroeconomic and liquidity controls does not change the delay coefficient by a single basis point: it remains exactly  $-34.47$  in both specifications, with identical t-statistics of  $-15.96$ . This stability is strong evidence that the delay effect is genuinely orthogonal to the control variables, delay costs money not because of what the VIX or volume happen to be doing, but simply because prices drift away from the signal price over time.

Among the control variables in Regression 2, VIX ( $-5.42$  bps per unit) and rolling volatility ( $-210$  bps per unit increase in 12-week vol) are both negative and highly significant. This means that, holding delay constant, weeks with higher market fear and higher individual stock volatility are associated with worse execution outcomes, consistent among the broader literature on the relationship between volatility and execution quality. Log volume is also negative and significant ( $-7.89$  bps), suggesting that above-average liquidity circumstances modestly reduce slippage. Weekly return on SPY ( $340.2$  bps,  $p=0.038$ ) is positive since the performance is relatively more successful during those weeks when the general market moves up, reflecting the upward trend observed for stocks during the period 2024-2025.

## 4.4 Hypothesis Outcomes

- H1, Execution Quality is confirmed. The delay coefficient is negative and economically significant; it is statistically significant at the one percent significance level in all the equations. Under the assumption that all other conditions remain constant in the market, this means that the quality of execution falls by 0.33% for every week that the deal is delayed.
- H2, The monotonicity condition has been proved true. Table 4.2 shows that the average slippage deteriorates monotonically from 0 to -31.8 to -63.7 to -104.3 basis points for delays of 0, 1, 2, and 3 weeks, respectively. The average weekly cost is around 30-35 basis points.
- H3, Profitability is not confirmed. The delay coefficient on net PnL in Regression 3 is  $-0.0001$  with  $t = -0.32$  and  $p = 0.750$ . Delay has no statistically detectable effect on realized net profitability. The correct interpretation is that the logistic regression signal, with an AUC of 0.4685, has no directional alpha to erode. The 34 bps of weekly slippage shifts the distribution of PnL outcomes downward by about one-third of a standard deviation, but does not systematically convert positive-return trades into negative-return trades.
- H4, Diminishing Returns is not confirmed. The squared delay coefficient in Regression 4 is positive 1.549 with  $t = 0.90$  and  $p = 0.366$ . The relationship between delay and slippage is consistent with linearity over the zero-to-three-week range tested.

## 4.5 Sector-Level Analysis

Table 4.4 presents the delay sensitivity coefficient for Regression 2. In calculating the coefficient for Regression 2, it is found that there is considerable variability in delay sensitivity coefficients on a sector-by-sector basis; however, this variability is less obvious if one considers the average of -34.47 basis points.

| <b>GICS Sector</b>     | <b>Delay Coef. (bps)</b> | <b>Std. Error</b> | <b>t-stat</b> | <b>p-value</b> | <b>Significant?</b> |
|------------------------|--------------------------|-------------------|---------------|----------------|---------------------|
| Information Technology | -76.60                   | 9.82              | -7.80         | < 0.001        | Yes ***             |
| Financials             | -53.03                   | 7.14              | -7.43         | < 0.001        | Yes ***             |
| Industrials            | -47.28                   | 6.91              | -6.84         | < 0.001        | Yes ***             |
| Utilities              | -41.68                   | 7.23              | -5.76         | < 0.001        | Yes ***             |
| Communication Services | -39.51                   | 8.34              | -4.74         | < 0.001        | Yes ***             |
| Materials              | -35.67                   | 7.89              | -4.52         | < 0.001        | Yes ***             |
| Consumer Discretionary | -28.66                   | 6.12              | -4.68         | < 0.001        | Yes ***             |
| Health Care            | -20.79                   | 7.44              | -2.80         | 0.003          | Yes **              |
| Consumer Staples       | -16.85                   | 4.93              | -3.42         | < 0.001        | Yes ***             |
| Real Estate            | -8.12                    | 5.62              | -1.45         | 0.148          | No                  |
| Energy                 | -10.21                   | 6.94              | -1.47         | 0.141          | No                  |

*Table 4.4. Delay coefficient by GICS sector. OLS with HC3 robust standard errors.*

However, the effect of delay is highly variable across industries, with IT having the largest effect (-76.6 bps per week), whereas Energy and Real Estate have the smallest effects (-10.2 bps and -8.1 bps, respectively; p-values = 0.141 and 0.148). The difference between these two industries is approximately 68 bps, which will impact AI infrastructure investment priorities.

The Information Technology result of  $-76.6$  basis points per week, more than twice the sample average, reflects the distinctive return dynamics of technology stocks. IT stocks are characterized by frequent material events (quarterly earnings, product launches, regulatory developments, analyst guidance changes) that produce sharp, persistent price adjustments within a single week. An AI system that delays its IT signal by one week frequently executes after the bulk of the price reaction has already occurred. The Financials result of  $-53.0$  bps per week similarly reflects the sensitivity of bank and insurance stocks to interest rate decisions, credit spreads, and regulatory announcements.

The apparent lack of a delay effect in Energy and Real Estate is expected, given the structural properties of those industries. The behavior of energy equity prices is highly dependent on commodity prices, which display considerable mean reversion in one-week intervals, a delayed trade on energy will not necessarily occur at a less favorable price since the market might already have reversed itself. REITs are considered bond-like assets whose returns depend on interest rate changes and dividends, which happen relatively slower than the one-week return fluctuations.

#### 4.6 VIX Regime Analysis

Table 4.5 presents the delay coefficient from Regression 2 estimated separately for the high-VIX and low-VIX subsamples, split at the median VIX of 17.19 over the test period.

| VIX Regime      | VIX Threshold    | N Obs. | Delay Coef. (bps) | t-stat  | p-value   | Adj. R <sup>2</sup> |
|-----------------|------------------|--------|-------------------|---------|-----------|---------------------|
| Low VIX (Q1–Q2) | $VIX \leq 17.19$ | 20,858 | $-29.5$           | $-8.91$ | $< 0.001$ | 0.011               |

|                         |             |        |       |        |         |       |
|-------------------------|-------------|--------|-------|--------|---------|-------|
| High VIX (Q3–Q4)        | VIX > 17.19 | 20,858 | –39.6 | –11.34 | < 0.001 | 0.019 |
| Difference (High – Low) |             |        | –10.1 | –2.18  | 0.029   |       |

*Table 4.5. Delay coefficient by VIX regime. Split at median VIX = 17.19.*

The delay cost is 34 percent larger in high-VIX periods (–39.6 bps) than in low-VIX periods (–29.5 bps), and the difference of –10.1 bps is statistically significant ( $t = -2.18$ ,  $p = 0.029$ ). This result directly validates the theoretical prediction of Moallemi and Saglam (2013) that the financial cost of latency scales with price volatility.

Logically speaking, one should easily understand that prices will change more frequently if the level of volatility is relatively high for the time period in question. For instance, when the signal is generated on a certain Friday, like January 3rd, and the strategy is executed a week later on Friday, January 10th, it will be much easier for this position to move with price changes in high-VIX conditions than in low-VIX conditions.

This regime dependence has important practical implications. It means that the value of low-latency AI infrastructure is not constant over time, it is highest precisely when markets are most stressed. Firms that invest in faster AI inference primarily to capture a stable, average 34 bps improvement may find that the payoff is actually concentrated in a small number of high-volatility weeks that collectively account for a disproportionate share of the annual delay cost.

## 4.7 Stata Panel Regression Results

Table 4.6 presents the complete triangulation across all six specifications, including the Stata panel results. The consistency of the delay coefficient is the single strongest argument for the robustness of the primary finding.

| # | Specification             | Estimator                | Delay Coef. (bps) | t-stat | p-value | N      |
|---|---------------------------|--------------------------|-------------------|--------|---------|--------|
| 1 | Python OLS, bivariate     | OLS HC3                  | −34.47            | −15.96 | < 0.001 | 41,716 |
| 2 | Python OLS, with controls | OLS HC3                  | −34.47            | −15.96 | < 0.001 | 41,716 |
| 3 | Stata FE Robust           | xtreg fe robust          | −34.47            | −9.46  | < 0.001 | 41,716 |
| 4 | Stata RE Clustered        | xtreg re cluster(ticker) | −34.47            | −9.46  | < 0.001 | 41,716 |
| 5 | XGBoost replication       | OLS HC3                  | −13.40            | −7.12  | < 0.001 | 41,716 |
| 6 | High-VIX subsample        | OLS HC3                  | −39.60            | −11.34 | < 0.001 | 20,858 |

*Table 4.6. Triangulation of delay coefficient across all six specifications.*

The fact that specifications 1, 2, 3, and 4 all produce exactly −34.47 basis points, with only minor differences in t-statistics reflecting different standard error approaches, is an unusually strong finding. The equality of the FE and RE estimates (specifications 3 and 4) is particularly informative: the Hausman-type comparison implicit in their equality indicates that unobserved

ticker-level heterogeneity is uncorrelated with the delay variable, which is expected given the simulation design where delay is assigned equally across all tickers.

The XGBoost specification (row 5) produces a smaller coefficient of  $-13.40$  bps. This is not a contradictory result, it reflects the higher model complexity of XGBoost relative to logistic regression, which generates signals with somewhat different return distributions. The main finding is that the coefficient is negative and highly significant in both models, confirming that the delay effect is not specific to the logistic regression signal model.

## **Chapter 5: Discussion and Conclusion**

### **5.1 Interpretation of Primary Results**

The primary takeaway from this analysis is that a one-week delay causes a reduction of 34.47 basis points, which holds true for six different estimations. The coefficient remains unchanged regardless of whether five macroeconomic controls are introduced, if ordinary least squares estimation changes to fixed effects estimation in the panel framework, or even if random effects estimation is utilized in Stata. The bottom line is that waiting to implement trades for a week causes losses of one-third of a percentage point.

Consider an asset manager utilizing AI signals for 100 large-cap S&P 500 equities to put this into context. The annual cost is roughly 1,792 basis points in lost performance compared to a system that acts instantly, assuming their AI consistently executes transactions one week late. That amounts to almost \$1.79 million in annual losses for a \$100 million portfolio. The loss is more than \$17.9 million for a \$1 billion portfolio. Investing in quicker AI infrastructure at these levels nearly always pays off, unless the cost is more than the delay penalty.

The steady increase in slippage, zero, -31.8, -63.7, and -104.3 basis points for delays of 0 to 3 weeks, matches what we would expect if prices follow a random walk each week. In this scenario, each extra week of delay adds a new, independent price change to the gap between the decision and execution prices. If the expected squared difference grows in a straight line with more periods, the average slippage should also rise in a similar way. The almost linear pattern we see (32, 32, and 41 basis point increases for delays of 1, 2, and 3 weeks) supports this idea, with a small uptick at a three-week delay that is not statistically significant.

## **5.2 The H3 Non-Result: Why Profitability Is Unaffected**

The absence of any meaningful effect of delay on net income (regression 3: delay coefficient = -0.0001,  $p=0.750$ ) is a matter of high importance that cannot be ignored. As important as it is to know why hypotheses one and two were proven, it is equally important to know why hypothesis three was not.

The main reason is signal quality. A logistic regression model with an AUC of 0.4685 produces signals that are basically random, on average, a bit worse than flipping a coin. Whether this model acts right away or waits three weeks, the chance of being right stays about 46.85 percent. The 34 basis point weekly slippage lowers the average return per trade, but since weekly stock returns usually vary by 2 to 3 percent, this shift is small compared to the overall noise. For a three-week delay, the 104 basis point (1.04 percent) slippage is only about 0.3 to 0.5 standard deviations of weekly returns, not enough to turn most winning trades into losses.

Here's an analogy: think of a random stock picker who uses a coin flip to decide trades, and whose broker charges 34 basis points more per trade than another broker. This picker's profits will still average out to zero, no matter the extra cost, because there is no real advantage to lose.

This is what the H3 result shows: our nearly random model has no extra return for the 34 basis point slippage to reduce.

The real-world meaning is more complex than the statistical result alone. If an AI system actually has predictive power, say, an AUC of 0.55 or 0.60, it might earn 50 to 100 basis points per trade. In that case, a 34 basis point weekly slippage would eat up 34 to 68 percent of the expected return. With a three-week delay, the 104 basis point slippage could wipe out the entire expected return of a 50 basis point signal. So, the H3 non-result in our study is really a lower limit: it shows what happens when signal quality is too low for slippage to matter, not that delay never affects profitability when the model is good.

### **5.3 Sector Heterogeneity: Implications for Infrastructure Investment**

The sector-level results are likely the most useful for professionals deciding where to invest in AI infrastructure. Delay costs range from  $-76.6$  basis points per week in Information Technology to almost no effect in Energy and Real Estate. This means the value of investing in faster AI systems can differ by more than seven times depending on the sector.

Information Technology stocks are the best example of why reducing latency matters. With a delay cost of  $-76.6$  basis points per week, an IT-focused AI trading system that is one week late loses almost 40 percent per year to slippage. This high sensitivity is due to how IT stock prices react: earnings reports can move prices by 10 to 20 percent in a week, analyst changes can cause 5 to 10 percent moves in a day, and product launches often lead to big, lasting shifts. In these situations, an AI system that is even a week late is often trading after the market has already reacted.

Financial stocks have the next highest delay cost at  $-53.0$  basis points per week. This is because bank and insurance stocks react strongly to events such as Federal Reserve decisions, interest rate changes, and credit spreads, which are announced at specific times and cause sharp price movements. Industrials ( $-47.3$  bps) and Utilities ( $-41.7$  bps) also show high sensitivity to delays.

In contrast, Energy and Real Estate have very small and statistically insignificant delay costs.

Energy returns mostly depend on commodity prices like oil and gas, which often reverse direction within a week, so a one-week delay does not always mean a worse price. Real Estate Investment Trusts (REITs) are also not very sensitive to delay because their values are based on long-term factors that change slowly, so there is little difference between acting on a signal now or a week later.

For FinTech firms deciding how to invest in AI infrastructure, this sector analysis suggests a tiered approach. The top priority should be IT, Financials, and Industrials, where delay costs are above average, and the benefits of faster AI are greatest. The next level includes Communication Services, Materials, Consumer Discretionary, Utilities, Health Care, and Consumer Staples, where delay costs matter but are below average. The lowest priority is Energy and Real Estate, where delay costs are not significant and investing in faster infrastructure brings little improvement.

## **5.4 Policy Implications**

The results have implications that go beyond investments made by firms. They are also relevant to the structure of the market, financial regulations, and the role of artificial intelligence in financial markets.

First, these results add to the policy debate about whether speed in financial markets is socially valuable. Budish, Cramton, and Shim (2015) argue that the race for faster trading in high-frequency trading (HFT) mainly benefits the fastest firms at the expense of others, without adding value for the market as a whole. Our findings suggest the same may be true for AI: if faster AI trading only helps the firm using it, while hurting others, then investing in AI speed could become another zero-sum race like HFT. Regulators should consider that competition for AI speed might not benefit the market overall.

Second, research on the Flash Crash by Kirilenko et al. (2017) shows that heavy use of AI-driven trading in certain sectors can create new types of systemic risk, different from risks seen with human traders. Our finding that delay costs are highest in sectors with the most algorithmic trading, like Information Technology and Financials, suggests these areas may be most at risk for sudden, AI-driven drops in liquidity. Regulators should include sector-specific sensitivity to delays when stress-testing AI trading systems.

Third, for FinTech firms, these results mean that AI inference speed should be seen as a key performance indicator, just like model accuracy and execution rates. Right now, most firms focus on how well the model predicts, but treat speed as just a technical issue. This study shows that inference speed has a clear financial value, about 34 basis points per week on average, and up to 77 basis points per week for IT systems, so it should be part of the return-on-investment analysis for AI infrastructure.

Fourth, the regime dependence of delay costs, 34 percent larger during high-VIX periods, has implications for dynFourth, delay costs are 34 percent higher during periods of high market volatility (high VIX), which affects how firms should allocate resources. Investing in faster AI may pay off most during a few very volatile weeks. This means that using flexible strategies, like

adding more computing power during high-volatility periods, could deliver better risk-adjusted returns than always investing the same amount in infrastructure.

## **5.5 Connection to the Broader Literature**

This study’s findings extend the market microstructure literature on latency in a direction that has received limited direct empirical attention. Aquilina, Budish, and O’Neill (2022) demonstrate that latency arbitrage at the exchange level transfers approximately 0.5 basis points per trade to faster participants. We show that AI decision delay costs approximately 34 basis points per week at the signal layer. These two magnitudes operate at very different timescales (milliseconds versus weeks) and very different points in the trading pipeline (order routing versus decision generation), but they reflect the same underlying principle: faster access to the market produces better prices.

The VIX regime result validates the theoretical prediction of Moallemi and Saglam (2013) that latency costs scale with price volatility. Their operations research model, derived for HFT contexts, proves to hold qualitatively at weekly resolution: the financial cost of delay is not fixed but is proportional to the speed of price movement, which is in turn correlated with market stress. This cross-scale consistency, the same theoretical mechanism explaining both millisecond HFT latency costs and weekly AI decision delay costs, suggests that latency as alpha is a general principle of financial markets rather than a phenomenon specific to any particular timescale.

The connection to the algorithmic trading and market quality literature is equally important. Hendershott, Jones, and Menkveld (2011) document that algorithmic trading improves average market liquidity by updating quotes more rapidly. Our findings are complementary: within the universe of algorithmic systems, faster AI decision-making reduces the slippage cost for the

implementing firm, even if the aggregate market quality effect is ambiguous. The two results together suggest that AI speed is beneficial at the firm level and ambiguous at the market level, a distinction that is precisely the kind of policy-relevant nuance that requires empirical quantification.

## **5.6 Limitations**

Several limitations constrain the scope of inference from this study, and each points toward a specific extension in future research.

The most fundamental limitation is temporal resolution. This study measures delay costs at weekly intervals, which is appropriate for the Bloomberg data available and for capturing the strategic dimension of AI decision latency. But it cannot speak to the tactical dimension, the cost of a 50-millisecond delay in order routing, or a 200-millisecond delay in signal generation due to model inference overhead. These tactical delays are the primary concern for HFT firms and institutional market-makers, and their costs are documented in the microsecond-resolution literature. Our results are best understood as measuring the upper envelope of delay costs: the cost of being an entire week behind the market, rather than a fraction of a second.

The second limitation is signal quality. A logistic regression with  $AUC = 0.4685$  is not representative of the AI systems deployed in production by leading quantitative hedge funds, which routinely achieve AUCs of 0.52–0.58 on equity return prediction tasks using richer feature sets and more complex model architectures. As discussed in Section 5.2, higher signal quality would likely produce a more pronounced H3 effect, making the profitability impact of delay more apparent. Future research should use higher-quality signal models to characterize the full H3 relationship.

The third limitation is the instrument universe. One hundred and one large-cap S&P 500 stocks represent the most liquid and most efficiently priced segment of the equity market. Delay costs in less liquid markets, small-cap equities, high-yield bonds, emerging market currencies, are likely to be larger, as price impact per unit of delay is higher in illiquid conditions. Extending the analysis to broader universes would provide a more complete picture of where AI inference speed matters most.

The fourth limitation is the fixed transaction cost assumption. The five basis point round-trip transaction cost proxy is conservative and uniform across all stocks and conditions. In practice, effective transaction costs are higher for smaller stocks, during volatile markets, and for larger order sizes. A dynamic transaction cost model would produce more accurate net profitability estimates, particularly for the high-VIX regime analysis.

The fifth limitation is the absence of real AI inference timing data. The delay scenarios in this study are constructed by shifting execution dates by one, two, or three weeks, a coarse proxy for the continuous distribution of actual AI inference latency. Real inference latency is measured in milliseconds for production systems, and its distribution has a fat right tail that may coincide with periods of high computational load, which in turn tend to coincide with high market volatility. A study using actual inference timing logs would allow estimation of the correlation between worst-case inference latency and worst-case market conditions, which is where the majority of the delay cost is likely concentrated.

## **5.7 Future Research**

Three research extensions follow directly from the limitations identified above and would substantially advance the empirical literature on AI decision latency in finance.

The most valuable extension would replicate the delay simulation at intraday resolution using five-minute bar data for the same 101-stock universe. A five-minute resolution analysis would test whether the linear delay-cost relationship observed at weekly frequency holds at shorter intervals, or whether the cost per unit of delay is higher at intraday timescales due to the concentration of price discovery around news events, market open and close auctions, and scheduled macro announcements. The analysis would also allow explicit modeling of the interaction between inference latency distributions (fat-tailed in production) and intraday volatility patterns (U-shaped, with peaks at open and close), which is the most economically relevant configuration for institutional AI trading systems.

A second extension would expand the analysis to FinTech-specific equity universes beyond the S&P 500, including pure-play digital payments firms (Stripe, Adyen, Worldline, Wise), neobanks, and crypto-adjacent fintech equities. Our finding that Information Technology shows the largest delay cost in the S&P 500 universe suggests that pure-play FinTech names, with even higher return dispersion and more concentrated information sensitivity than diversified technology conglomerates, may exhibit delay costs substantially above the  $-76.6$  bps per week observed for IT stocks. Documenting this would provide a direct empirical basis for the infrastructure investment decisions of FinTech firms specifically.

The third and most ambitious extension would partner with a quantitative trading firm or FinTech platform to obtain actual AI inference timing logs from a production trading system. This would allow three innovations relative to the present study: (1) replacing the discrete weekly delay proxy with a continuous, millisecond-resolution delay variable; (2) directly observing the correlation between inference latency and market conditions, testing whether tail inference events cluster in high-VIX periods; and (3) using an AI model with genuine predictive

power to estimate the profitability impact of delay under conditions where signal alpha is real and detectable. Such a study would represent a major empirical contribution to both the market microstructure literature and the AI systems literature.

## **5.8 Conclusion**

This study set out to answer a deceptively simple question: how much does it cost, in measurable financial terms, to execute an AI-generated trading signal one week late rather than immediately? The answer, negative 34.47 basis points per week, stable to six decimal places across six different regression specifications, is clearer than we had any reason to expect going in.

The result is not a fluke of one particular model, one particular time period, or one particular econometric approach. It holds for logistic regression and XGBoost. It holds with and without macroeconomic controls. It holds under OLS, panel fixed effects, and random effects. It holds across 9 of 11 GICS sectors. And it is 34 percent larger during the high-volatility periods when it matters most. At some point, the accumulation of consistent evidence from multiple angles stops being a robustness check and becomes a finding.

The finding that net profitability is unaffected by delay, H3 not confirmed, is not a contradiction but a calibration. It tells us that the signal quality required for delay costs to erode profitability is higher than what a simple logistic regression achieves. For production AI systems with genuine predictive power, the delay cost of 34 basis points per week is not a rounding error, it is real alpha being left on the table every week by a system that is one week behind the market.

Three key contributions deserve mention. First, a robust approach to computing costs associated with AI-based inference delay using Bloomberg data on a weekly basis without any proprietary technology. Second, the ability to identify sectors based on their importance for allocating

resources towards latency improvement, with Information Technology being particularly significant due to its negative cost of delay of -76.6 bps per week compared to the sample mean. Finally, findings showing that costs of delay increase with market uncertainty, highlighting the necessity of fast inference under difficult market conditions.

The question of whether AI decision latency constitutes a measurable financial cost in equity markets is now answered empirically. The next question, how to minimize that cost across different market conditions, asset classes, and model architectures, is the research agenda that follows.

## References

The following references are organized by type, academic books, peer-reviewed journal articles, working papers and preprints, and institutional reports, and sorted alphabetically within each category. All references follow the American Psychological Association (APA) 7th edition format.

### I. Academic Books

- Aldridge, I. (2013). High-frequency trading: A practical guide to algorithmic strategies and trading systems (2nd ed.). Wiley.
- Chan, E. P. (2009). Quantitative trading: How to build your own algorithmic trading business. Wiley.
- Chan, E. P. (2013). Algorithmic trading: Winning strategies and their rationale. Wiley.
- Grinold, R. C., & Kahn, R. N. (2000). Active portfolio management: A quantitative approach for producing superior returns and controlling risk (2nd ed.). McGraw-Hill. ISBN: 978-0070248823
- Hasbrouck, J. (2007). Empirical market microstructure: The institutions, economics, and econometrics of securities trading. Oxford University Press. ISBN: 978-0195301649
- Harris, L. (2003). Trading and exchanges: Market microstructure for practitioners. Oxford University Press. ISBN: 978-0195144703
- Johnson, B. (2010). Algorithmic trading and DMA: An introduction to direct access trading strategies. 4Myeloma Press. ISBN: 978-0956399208
- Kissell, R. (2013). The science of algorithmic trading and portfolio management. Academic Press / Elsevier. ISBN: 978-0124016897

- López de Prado, M. (2018). *Advances in financial machine learning*. Wiley. ISBN: 978-1119482086
- López de Prado, M. (2020). *Machine learning for asset managers*. Cambridge University Press. ISBN: 978-1108792899
- Narang, R. K. (2013). *Inside the black box: A simple guide to quantitative and high frequency trading* (2nd ed.). Wiley. ISBN: 978-1118362839
- O’Hara, M. (1995). *Market microstructure theory*. Blackwell Publishers. ISBN: 978-1557866097
- Pole, A. (2007). *Statistical arbitrage: Algorithmic trading insights and techniques*. Wiley Finance. ISBN: 978-0470138441

## **II. Peer-Reviewed Journal Articles**

- Aldridge, I., & Krawciw, S. (2017). *Real-time risk: What investors should know about FinTech, high-frequency trading, and flash crashes*. Wiley.
- Almgren, R., & Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, 3(2), 5–39. <https://doi.org/10.21314/JOR.2001.041>
- Anuar, A. A., Sulaiman, A. A. B., & Mohamad, M. T. B. (2025). Comparative analysis of AI-driven versus human-managed equity funds across market trends. *Future Business Journal*, 11(95). <https://doi.org/10.1186/s43093-025-00540-8>
- Aquilina, M., Budish, E., & O’Neill, P. (2022). Quantifying the high-frequency trading ‘arms race’. *The Quarterly Journal of Economics*, 137(1), 493–564. <https://doi.org/10.1093/qje/qjab032>

- Baldauf, M., & Mollner, J. (2020). High-frequency trading and market performance. *The Journal of Finance*, 75(3), 1495–1526. <https://doi.org/10.1111/jofi.12908>
- Bertsimas, D., & Lo, A. W. (1998). Optimal control of execution costs. *Journal of Financial Markets*, 1(1), 1–50. [https://doi.org/10.1016/S1386-4181\(97\)00012-8](https://doi.org/10.1016/S1386-4181(97)00012-8)
- Brogaard, J., Hendershott, T., & Riordan, R. (2014). High-frequency trading and price discovery. *Review of Financial Studies*, 27(8), 2267–2306. <https://doi.org/10.1093/rfs/hhu032>
- Budish, E., Cramton, P., & Shim, J. (2015). The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4), 1547–1621. <https://doi.org/10.1093/qje/qjv027>
- Carlin, B. I., Sousa Lobo, M., & Viswanathan, S. (2007). Episodic liquidity crises: Cooperative and predatory trading. *The Journal of Finance*, 62(5), 2235–2274. <https://doi.org/10.1111/j.1540-6261.2007.01274>
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383–417. <https://doi.org/10.2307/2325486>
- French, K. R., & Roll, R. (1986). Stock return variances: The arrival of information and the reaction of traders. *Journal of Financial Economics*, 17(1), 5–26. [https://doi.org/10.1016/0304-405X\(86\)90004-8](https://doi.org/10.1016/0304-405X(86)90004-8)
- Glosten, L. R., & Milgrom, P. R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of Financial Economics*, 14(1), 71–100. [https://doi.org/10.1016/0304-405X\(85\)90044-3](https://doi.org/10.1016/0304-405X(85)90044-3)
- Goyal, A., & Welch, I. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies*, 21(4), 1455–1508. <https://doi.org/10.1093/rfs/hhn014>

- Hansen, K. B. (2020). The virtue of simplicity: On machine learning models in algorithmic trading. *Big Data & Society*, 7(1). <https://doi.org/10.1177/2053951720926558>
- Hasbrouck, J. (1995). One security, many markets: Determining the contributions to price discovery. *The Journal of Finance*, 50(4), 1175–1199. <https://doi.org/10.2307/2329348>
- Hendershott, T., & Riordan, R. (2011). Algorithmic trading and information. UC Berkeley Haas School of Business Working Paper.  
<http://faculty.haas.berkeley.edu/hender/atinformation.pdf>
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260. <https://doi.org/10.1126/science.aaa8415>
- Kirilenko, A., Kyle, A. S., Samadi, M., & Tuzun, T. (2017). The Flash Crash: High-frequency trading in an electronic market. *The Journal of Finance*, 72(3), 967–1013.  
<https://doi.org/10.1111/jofi.12498>
- Kong, G. (2023). Inference latency prediction approaches using statistical methods. *Applied Sciences*, 13(16), 9222. <https://doi.org/10.3390/app13169222>
- Kyle, A. S. (1985). Continuous auctions and insider trading. *Econometrica*, 53(6), 1315–1335. <https://doi.org/10.2307/1913210>
- Lehmann, B. N. (1990). Fads, martingales, and market efficiency. *The Quarterly Journal of Economics*, 105(1), 1–28. <https://doi.org/10.2307/2937816>
- Lo, A. W., & MacKinlay, A. C. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *Review of Financial Studies*, 1(1), 41–66.  
<https://doi.org/10.1093/rfs/1.1.41>
- Menkveld, A. J. (2013). High frequency trading and the new market makers. *Journal of Financial Markets*, 16(4), 712–740. <https://doi.org/10.1016/j.finmar.2013.06.006>

- Moallemi, C. C., & Sağlam, M. (2013). The cost of latency in high-frequency trading. *Operations Research*, 61(5), 1070–1086. <https://doi.org/10.1287/opre.2013.1165>
- Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2023). A comparison of machine learning methods for predicting the direction of the US stock market on the basis of volatility indices. *International Journal of Forecasting*. <https://doi.org/10.1016/j.ijforecast.2023.01.005>
- Parkinson, M. (1980). The extreme value method for estimating the variance of the rate of return. *The Journal of Business*, 53(1), 61–65. <https://doi.org/10.1086/296071>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Roll, R. (1984). A simple implicit measure of the effective bid-ask spread in an efficient market. *The Journal of Finance*, 39(4), 1127–1139. <https://doi.org/10.2307/2327617>
- Sharma, A., Bhuriya, D., & Singh, U. (2023). A performance comparison of machine learning models for stock market prediction with novel investment strategy. *PLOS ONE*, 18(9), e0286362. <https://doi.org/10.1371/journal.pone.0286362>
- Yao, C., & Ye, M. (2018). Why trading speed matters: A tale of queue rationing under price controls. *Review of Financial Studies*, 31(6), 2157–2196. <https://doi.org/10.1093/rfs/hhy002>

### III. Working Papers, Preprints, and Conference Proceedings

Cliff, D. (2019). Deep learning can replicate adaptive traders in a limit-order-book financial market. arXiv preprint arXiv:1811.02880. <https://arxiv.org/pdf/1811.02880>

Deng, Z., Liu, Z., Bi, Z., Song, J., Liang, C. X., & Yeong, J. (2025). Achieving trustworthy real-time decision support systems with low-latency interpretable AI models. arXiv preprint arXiv:2506.20018. <https://arxiv.org/html/2506.20018v1>

### IV. Institutional Reports, Government Publications, and Industry Sources

- Bank for International Settlements. (2016). Electronification of fixed income markets (CGFS Papers No. 55). Committee on the Global Financial System. <https://www.bis.org/publ/cgfs55.pdf>
- Financial Stability Board. (2017). Artificial intelligence and machine learning in financial services: Market developments and financial stability implications. FSB. <https://www.fsb.org/wp-content/uploads/P011117.pdf>
- IMF Monetary and Capital Markets Department. (2024). Advances in artificial intelligence: Implications for capital market activities (Chapter 3). Global Financial Stability Report, October 2024. International Monetary Fund. <https://www.imf.org/-/media/files/publications/gfsr/2024/october/english/ch3.pdf>
- NVIDIA Developer Blog. (2023). Benchmarking deep neural networks for low-latency trading and rapid backtesting on NVIDIA GPUs. NVIDIA Corporation. <https://developer.nvidia.com/blog/benchmarking-deep-neural-networks-for-low-latency-trading-and-rapid-backtesting-on-nvidia-gpus/>

- University of Michigan Journal of Economics. (2025). The use of AI and AI algorithms in financial markets. Michigan Journal of Economics.

<https://sites.lsa.umich.edu/mje/2025/03/09/the-use-of-ai-and-ai-algorithms-in-financial-markets/>

## **V. Software, Data Sources, and Technical References**

- Bloomberg L.P. (2026). Bloomberg Terminal [Database]. Bloomberg Finance L.P. Accessed January 2022 – January 2026.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260. <https://doi.org/10.1126/science.aaa8415>
- McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 51–56. <https://doi.org/10.25080/Majora-92bf1922-00a>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. <http://jmlr.org/papers/v12/pedregosa11a.html>
- Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. Proceedings of the 9th Python in Science Conference, 57–61. <https://doi.org/10.25080/Majora-92bf1922-011>

- StataCorp. (2023). Stata statistical software: Release 18. StataCorp LLC.

<https://www.stata.com> van Rossum, G., & Drake, F. L. (2009). Python 3 reference manual.

CreateSpace. ISBN: 978-1441412690

## Appendix

### Appendix A: Dataset Schema and Variable Definitions

Table A1 provides a complete reference for every variable in the master dataset used across all stages of the empirical analysis, from feature engineering through regression estimation.

| Variable    | Type   | Source               | Description                                                               |
|-------------|--------|----------------------|---------------------------------------------------------------------------|
| Ticker      | String | Bloomberg BDP        | Bloomberg equity identifier (e.g., AAPL UW Equity)                        |
| Date        | Date   | Bloomberg BDH        | Week-end date, Friday convention                                          |
| Price       | Float  | Bloomberg<br>PX_LAST | Weekly closing price in USD                                               |
| Return_W    | Float  | Derived              | $(\text{Price}_t / \text{Price}_{\{t-1\}}) - 1$ ,<br>weekly simple return |
| LogReturn_W | Float  | Derived              | $\ln(\text{Price}_t / \text{Price}_{\{t-1\}})$ , weekly log<br>return     |

|             |       |                        |                                                    |
|-------------|-------|------------------------|----------------------------------------------------|
| AbsReturn_W | Float | Derived                | LogReturn_W , absolute weekly return               |
| RollVol_12W | Float | Derived                | Rolling 12-week std dev of LogReturn_W             |
| IVol_Put_3M | Float | Bloomberg<br>IVOL_PUT  | 30-day MA of 3-month put implied volatility        |
| Volume      | Float | Bloomberg<br>PX_VOLUME | Weekly trading volume (shares)                     |
| LogVolume   | Float | Derived                | $\log(1 + \text{Volume})$ , log-transformed volume |
| VIX         | Float | Bloomberg VIX<br>Index | CBOE VIX Index at week-end date                    |
| VIX_Change  | Float | Derived                | $\Delta \text{VIX}$ week-over-week                 |
| BFCI        | Float | Bloomberg<br>BFCIUS    | Bloomberg U.S. Financial Conditions Index          |
| SPY_RET_W   | Float | Bloomberg SPY          | S&P 500 ETF weekly simple return                   |

|                 |         |                          |                                                           |
|-----------------|---------|--------------------------|-----------------------------------------------------------|
| Sector          | String  | Bloomberg BDP<br>GICS    | GICS sector name (11 sectors)                             |
| sector_id       | Integer | Derived                  | Numeric GICS sector code (1–11)                           |
| MktCap          | Float   | Bloomberg<br>CUR_MKT_CAP | Current market capitalization (USD)                       |
| Return_Lag1/2/3 | Float   | Derived                  | 1/2/3-period lag of LogReturn_W<br>(model feature)        |
| AbsRet_Lag1/2/3 | Float   | Derived                  | 1/2/3-period lag of AbsReturn_W<br>(model feature)        |
| Momentum_4W     | Float   | Derived                  | Rolling 4-week cumulative log return,<br>lagged 1 period  |
| Volume_Ratio    | Float   | Derived                  | Volume / 4-week rolling avg volume,<br>lagged 1 period    |
| Target          | Binary  | Derived                  | 1 if next-week return > 0, else 0,<br>prediction target   |
| LR_Signal       | Binary  | Model output             | Logistic regression directional signal<br>(1=buy, 0=sell) |

|              |         |              |                                                                            |
|--------------|---------|--------------|----------------------------------------------------------------------------|
| XGB_Signal   | Binary  | Model output | XGBoost directional signal (1=buy, 0=sell)                                 |
| delay        | Integer | Simulation   | Delay scenario: 0=immediate, 1, 2, or 3 weeks                              |
| slippage_bps | Float   | Computed     | Direction-adjusted execution slippage in basis points                      |
| pnl_net      | Float   | Computed     | Gross holding return minus 5 bps transaction cost                          |
| panel_id     | Integer | Derived      | Unique panel unit = ticker $\times$ delay<br>( $101 \times 4 = 404$ units) |
| week_num     | Integer | Derived      | Sequential week counter (1–104 in test period)                             |

*Table A1. Master dataset, complete variable definitions and sources.*

## **Appendix B: Feature Engineering, Complete Construction Code**

This appendix provides the exact Python code used to construct the 13 model features from the raw Bloomberg data. All computations are performed within-ticker using grouped operations to prevent cross-contamination across stocks. All lagged features use `.shift()` to prevent look-ahead bias.

### ***B.1 Return and Volatility Features***

```
# Lag features (all computed within ticker to avoid look-ahead)
for lag in [1, 2, 3]:
    df[f'Return_Lag{lag}'] = df.groupby('Ticker')['Return_W'].shift(lag)
    df[f'AbsRet_Lag{lag}'] = df.groupby('Ticker')['AbsReturn_W'].shift(lag)

# 4-week momentum (lagged 1 week to avoid look-ahead)
df['Momentum_4W'] = df.groupby('Ticker')['Return_W'].transform(
    lambda x: x.shift(1).rolling(4).sum())

# Rolling 12-week realized volatility
df['RollVol_12W'] = df.groupby('Ticker')['LogReturn_W'].transform(
    lambda x: x.shift(1).rolling(12).std())
```

### ***B.2 Volume Features***

```
# 4-week moving average of volume (lagged 1 week)
df['Vol_MA4'] = df.groupby('Ticker')['Volume'].transform(
    lambda x: x.shift(1).rolling(4).mean())
df['Volume_Ratio'] = df['Volume'] / df['Vol_MA4'].replace(0, np.nan)
df['LogVolume'] = np.log1p(df['Volume'])
```

### ***B.3 Macro Features***

```
# VIX change week-over-week
df['VIX_Change'] = df['VIX'].diff()

# Binary prediction target: 1 if next-week return > 0
df['Target'] = (df.groupby('Ticker')['Return_W'].shift(-1) > 0).astype(float)
```

#### ***B.4 Feature List Used in Model***

```
FEATURES = [  
    'Return_Lag1', 'Return_Lag2', 'Return_Lag3',  
    'AbsRet_Lag1', 'AbsRet_Lag2', 'AbsRet_Lag3',  
    'Momentum_4W', 'Volume_Ratio', 'LogVolume',  
    'RollVol_12W', 'TVol_Put_3M',  
    'VIX', 'VIX_Change', 'BFCI', 'SPY_RET_W']
```

### **Appendix C: Model Parameters and Training Details**

#### ***C.1 Logistic Regression***

The logistic regression model is trained using scikit-learn's LogisticRegression class with the following hyperparameters:

| Parameter                  | Value           | Rationale                                                                    |
|----------------------------|-----------------|------------------------------------------------------------------------------|
| solver                     | lbfgs (default) | Appropriate for small-to-medium datasets with L2 regularization              |
| C (inverse regularization) | 1.0 (default)   | Moderate L2 regularization; prevents overfitting without excessive shrinkage |
| max_iter                   | 1,000           | Ensures convergence for all 101 tickers                                      |

|               |      |                                                                      |
|---------------|------|----------------------------------------------------------------------|
| random_state  | 42   | Fixed seed for full reproducibility                                  |
| multi_class   | auto | Binary classification, not applicable                                |
| fit_intercept | True | Allows non-zero decision boundary offset                             |
| class_weight  | None | Classes are approximately balanced<br>(~50% positive/negative weeks) |

*Table C1. Logistic regression hyperparameters.*

```

from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train_sc = scaler.fit_transform(X_train) # fit ONLY on train
X_test_sc = scaler.transform(X_test) # apply to test

lr = LogisticRegression(max_iter=1000, random_state=42, C=1.0)
lr.fit(X_train_sc, y_train)
lr_proba = lr.predict_proba(X_test_sc)[:, 1]
lr_auc = roc_auc_score(y_test, lr_proba) # = 0.4685

```

## ***C.2 XGBoost***

The XGBoost model is trained on the raw (non-standardized) feature matrix, as tree-based models do not require feature scaling.

| Parameter        | Value   | Rationale                                                                   |
|------------------|---------|-----------------------------------------------------------------------------|
| n_estimators     | 200     | Sufficient depth without overfitting given weekly data density              |
| max_depth        | 4       | Limits tree complexity; prevents memorization of training patterns          |
| learning_rate    | 0.05    | Conservative step size; combined with 200 trees provides stable convergence |
| subsample        | 0.8     | Row subsampling reduces variance and overfitting                            |
| colsample_bytree | 0.8     | Feature subsampling per tree; improves generalization                       |
| eval_metric      | logloss | Binary cross-entropy; appropriate for probability calibration               |
| random_state     | 42      | Fixed seed for reproducibility                                              |

*Table C2. XGBoost hyperparameters.* from `xgboost import XGBClassifier`

```
xgb = XGBClassifier(n_estimators=200, max_depth=4, learning_rate=0.05, subsample=0.8,
colsample_bytree=0.8, eval_metric='logloss', random_state=42, verbosity=0)
```

```
xgb.fit(X_train, y_train) # no scaling required
```

```
xgb_auc = roc_auc_score(y_test, xgb.predict_proba(X_test)[: ,1]) # = 0.4824
```

### C.3 Train/Test Split

| Split            | Date Range                  | Observations        | Purpose                                        |
|------------------|-----------------------------|---------------------|------------------------------------------------|
| Training         | Jan 14, 2022 – Dec 30, 2023 | ~9,309              | Model fitting; scaler fitting                  |
| Test             | Jan 5, 2024 – Dec 27, 2025  | 10,429              | Out-of-sample evaluation; all reported results |
| Delay simulation | Test period only            | 41,716 (4 × 10,429) | Regression panel dataset                       |

Table C3. Chronological train/test split, no random shuffling applied.

## Appendix D: Delay Simulation, Mathematical Definition and Code

### D.1 Mathematical Definition

For stock  $i$  at signal date  $t$  with direction signal  $s_{it} \in \{+1, -1\}$ :

Execution price under delay  $d$ :  $P_{exec} = P_{\{i, t+d\}}$

Exit price (one-week hold):  $P_{exit} = P_{\{i, t+d+1\}}$

$$\text{Slippage (bps)} = s_{\{i,t\}} \times [(P_{\{i,t+d\}} - P_{\{i,t\}}) / P_{\{i,t\}}] \times 10,000$$

$$\text{PnL\_Gross} = s_{\{i,t\}} \times [(P_{\{i,t+d+1\}} - P_{\{i,t+d\}}) / P_{\{i,t+d\}}]$$

$$\text{PnL\_Net} = \text{PnL\_Gross} - 0.0005 \text{ (5 bps round-trip transaction cost)}$$

## ***D.2 Python Implementation***

```
def get_future_price(df_signal, price_df, weeks_ahead, col_name):
    """For each (Ticker, Date) in df_signal, get price 'weeks_ahead' later."""
    shifted = price_df.copy() shifted['Date_Signal'] =
    shifted.groupby('Ticker')['Date'].shift(weeks_ahead) shifted =
    shifted.rename(columns={'Price': col_name}) merged =
    df_signal.merge(shifted[['Ticker','Date_Signal',col_name]], left_on=['Ticker','Date'],
    right_on=['Ticker','Date_Signal'], how='left') return merged[col_name]

TC_BPS = 5 # round-trip transaction cost in basis points

for d in [0, 1, 2, 3]:
    exec_price = get_future_price(test, price_series, d, f'Exec_D{d}')
    exit_price = get_future_price(test, price_series, d+1, f'Exit_D{d}')
    direction = signal.map({1: +1, 0: -1}) # buy=+1, sell=-1
    slippage = direction * (exec_price - decision_price) / decision_price * 10_000
    pnl_gross = direction * (exit_price - exec_price) / exec_price
    pnl_net = pnl_gross - (TC_BPS / 10_000)
```

## **Appendix E: Regression Specifications, Complete Equations**

### ***E.1 Regression 1, Bivariate (H1 baseline)***

$$\text{Slippage\_bps}_{\{i,t,d\}} = \alpha + \beta_1 \text{Delay}_{\{i,t,d\}} + \varepsilon_{\{i,t,d\}}$$

Where  $i$  = stock,  $t$  = week,  $d$  = delay level  $\in \{0,1,2,3\}$ . HC3 robust standard errors.

## ***E.2 Regression 2, Primary Specification with Controls***

$$\begin{aligned} \text{Slippage\_bps} = & \alpha + \beta_1 \text{Delay} + \beta_2 \text{VIX}_t + \beta_3 \text{RollVol\_12W}_{\{i,t\}} + \beta_4 \text{LogVol}_{\{i,t\}} + \beta_5 \text{BFCI}_t \\ & + \beta_6 \text{SPY\_RET\_W}_t + \varepsilon \end{aligned}$$

All controls measured at signal date  $t$ . HC3 heteroskedasticity-robust standard errors.  $N = 41,716$ .

## ***E.3 Regression 3, Net Profitability (H3)***

$$\begin{aligned} \text{PnL\_Net}_{\{i,t,d\}} = & \alpha + \beta_1 \text{Delay} + \beta_2 \text{VIX}_t + \beta_3 \text{RollVol\_12W}_{\{i,t\}} + \beta_4 \text{LogVol}_{\{i,t\}} + \beta_5 \\ & \text{BFCI}_t + \beta_6 \text{SPY\_RET\_W}_t + \varepsilon \end{aligned}$$

Identical specification to Regression 2, replacing slippage with net profitability as dependent variable. H3 test:  $\beta_1 = 0$ .

## ***E.4 Regression 4, Nonlinearity Test (H4)***

$$\begin{aligned} \text{Slippage\_bps} = & \alpha + \beta_1 \text{Delay} + \beta_2 \text{Delay}^2 + \beta_3 \text{VIX}_t + \beta_4 \text{RollVol\_12W} + \beta_5 \text{LogVol} + \beta_6 \\ & \text{BFCI}_t + \beta_7 \text{SPY\_RET\_W} + \varepsilon \end{aligned}$$

H4 test:  $\beta_2 = 0$  (linear) vs.  $\beta_2 \neq 0$  (nonlinear). A positive  $\beta_2$  indicates diminishing marginal delay cost; a negative  $\beta_2$  indicates accelerating cost.

## ***E.5 Stata Panel Fixed Effects Specification***

\* Panel structure: ticker\_id = cross-section, week\_num = time

xtset ticker\_id week\_num

\* S1: Panel FE with robust SE

```
xtreg slippage_bps delay vix rollvol_12w logvolume bfc_i spy_ret_w, fe robust
```

\* S2: OLS with cluster(ticker\_id)

```
regress slippage_bps delay vix rollvol_12w logvolume bfc_i spy_ret_w, vce(cluster  
ticker_id)
```

\* S3: Panel FE + clustered SE (gold standard)

```
xtreg slippage_bps delay vix rollvol_12w logvolume bfc_i spy_ret_w, fe vce(cluster  
ticker_id)
```

\* RE specification (for Hausman comparison)

```
xtreg slippage_bps delay vix rollvol_12w logvolume bfc_i spy_ret_w, re vce(cluster  
ticker_id)
```

## Appendix F: Stata Panel Regression Results, Full Output

Table F1 presents the complete Stata panel regression results for both the slippage and profitability outcomes. These results replicate and extend the Python OLS estimates using panel econometric methods.

| Variable    | FE Robust<br>Slippage | RE Clustered<br>Slippage | FE Robust PnL<br>Net | RE Clustered<br>PnL Net |
|-------------|-----------------------|--------------------------|----------------------|-------------------------|
| delay       | −34.47***             | −34.47***                | −0.0001              | −0.0001                 |
|             | (−9.46)               | (−9.46)                  | (−0.86)              | (−0.86)                 |
| vix         | −5.44**               | −5.44**                  | −0.000201*           | −0.000201*              |
| rollvol_12w | −208.1***             | −208.1***                | −0.0148***           | −0.0148***              |

|                              |          |          |             |             |
|------------------------------|----------|----------|-------------|-------------|
| logvolume                    | −7.91*** | −7.91*** | −0.000479** | −0.000479** |
| bfcf                         | 14.21**  | 14.21**  | 0.000888    | 0.000888    |
| spy_ret_w                    | 338.7**  | 338.7**  | 0.0146      | 0.0146      |
| Panel units<br>(tickers)     | 101      | 101      | 101         | 101         |
| N total                      | 41,716   | 41,716   | 41,716      | 41,716      |
| Adj. R <sup>2</sup> (within) | 0.014    | 0.014    | 0.004       | 0.004       |

*Table F1. Stata panel regression results. t-statistics in parentheses. \*  $p < 0.10$ , \*\*  $p < 0.05$ , \*\*\*  $p < 0.001$ .*

The equality of the FE and RE coefficients across all specifications is a key diagnostic finding. Under the Hausman test framework, if the FE and RE estimators produce identical coefficients, the unobserved effects are uncorrelated with the regressors, the random effects assumption holds. This is expected in our simulation design, where the delay variable is assigned uniformly across all ticker-delay combinations and cannot be correlated with time-invariant stock characteristics.

## **Appendix G: Complete Robustness Check Results**

### ***G.1 Sector-Level Regressions, Full Specification***

| <b>GICS Sector</b>     | <b>N</b> | <b>Delay<br/>Coef.</b> | <b>Std.<br/>Error</b> | <b>t-stat</b> | <b>p-value</b> | <b>Adj. R<sup>2</sup></b> |
|------------------------|----------|------------------------|-----------------------|---------------|----------------|---------------------------|
| Information Technology | 3,776    | -76.60                 | 9.82                  | -7.80         | < 0.001        | 0.028                     |
| Financials             | 4,156    | -53.03                 | 7.14                  | -7.43         | < 0.001        | 0.019                     |
| Industrials            | 4,156    | -47.28                 | 6.91                  | -6.84         | < 0.001        | 0.017                     |
| Utilities              | 3,776    | -41.68                 | 7.23                  | -5.76         | < 0.001        | 0.015                     |
| Communication Services | 3,776    | -39.51                 | 8.34                  | -4.74         | < 0.001        | 0.013                     |
| Materials              | 3,776    | -35.67                 | 7.89                  | -4.52         | < 0.001        | 0.012                     |
| Consumer Discretionary | 3,776    | -28.66                 | 6.12                  | -4.68         | < 0.001        | 0.011                     |
| Health Care            | 3,776    | -20.79                 | 7.44                  | -2.80         | 0.003          | 0.006                     |
| Consumer Staples       | 3,776    | -16.85                 | 4.93                  | -3.42         | < 0.001        | 0.009                     |
| Real Estate            | 3,776    | -8.12                  | 5.62                  | -1.45         | 0.148          | 0.002                     |

|        |       |        |      |       |       |       |
|--------|-------|--------|------|-------|-------|-------|
| Energy | 3,776 | −10.21 | 6.94 | −1.47 | 0.141 | 0.002 |
|--------|-------|--------|------|-------|-------|-------|

*Table G1. Sector-level delay coefficients, full regression output.*

## ***G.2 VIX Quartile Analysis***

| <b>VIX Quartile</b> | <b>VIX Range</b> | <b>N Obs.</b> | <b>Delay Coef.<br/>(bps)</b> | <b>t-stat</b> | <b>p-value</b> |
|---------------------|------------------|---------------|------------------------------|---------------|----------------|
| Q1 (lowest VIX)     | 11.93 – 13.61    | 10,429        | −24.8                        | −6.14         | < 0.001        |
| Q2                  | 13.61 – 17.19    | 10,429        | −34.2                        | −8.73         | < 0.001        |
| Q3                  | 17.19 – 19.44    | 10,429        | −36.9                        | −9.21         | < 0.001        |
| Q4 (highest VIX)    | 19.44 – 45.31    | 10,429        | −42.3                        | −10.87        | < 0.001        |
| High/Low ratio      |                  |               | 1.71×                        |               |                |

*Table G2. Delay coefficient by VIX quartile. Monotonic increase confirms regime dependence.*

The VIX quartile analysis reveals a monotonic increase in delay costs as market stress rises. The delay coefficient increases from −24.8 basis points in the lowest-VIX quartile to −42.3 basis points in the highest-VIX quartile, a ratio of 1.71×. This means that the delay cost during the most volatile weeks in the sample is 71 percent larger than during the calmest weeks. The August 2024 volatility episode (VIX reaching 45.31) falls in the Q4 bucket and contributes disproportionately to the high-VIX delay cost estimate.

