

Department of Economics
MSc Business Economics with Analytics

Applied Econometrics for High Frequency Trading:
A Machine Learning Approach

Mikelino Chatigia
Athens, February 2025

Master's Dissertation Submitted by
Mikelino Chatigia

Supervisory Committee

Main Supervisor

Yiannis Dendramis

Associate Professor of Economics, Athens University of Economics & Business

Committee Members

Elias Tzavalis

Professor of Economics, Athens University of Economics & Business

Spyridon Pagkratis

Associate Professor of Economics, Athens University of Economics & Business

Athens, February 2025

Abstract

This thesis is an empirical examination of the predictability of ultra high-frequency stock returns, trade directions, and durations in response to key price, volume, and transaction events using machine learning methods. Unlike the limited predictability found in low-frequency data, our results reveal systematic and short-term predictability in high-frequency markets. Using transaction and quote data from five NASDAQ 100 stocks, we explore classical learning models, ensemble methods, and neural networks. The analysis is conducted using Python to implement and evaluate these machine learning techniques. We identify key predictive factors such as order book imbalances, recent trade patterns, and volume statistics. Our analysis also highlights how stock characteristics like liquidity, volatility, and share price influence predictability. Additionally, we assess the impact of data timeliness and simulate the advantage of anticipating order flow direction, illustrating the benefits of even partial foresight. Overall, our findings underscore the potential of high-frequency data and machine learning to capture short-term market dynamics, with implications for both market efficiency and trading strategies.

Keywords: *High-frequency trading, machine learning, stock returns, trade direction, transaction duration, , order book imbalance, NASDAQ 100, Python*

Contents

1	Introduction	5
2	Response and Predictor Variables	10
2.1	Transactions and Quotes Data	10
2.1.1	Time Clocks	11
2.1.2	Transaction Return	13
2.1.3	Price Direction	14
2.1.4	Transaction Duration	14
2.2	Predictor Variables	15
3	Machine Learning Methods	19
3.1	Model Categories	19
3.1.1	Classical Learning	19
3.1.2	Ensemble Learning	19
3.1.3	Deep Learning	19
3.2	Model Presentation	20
3.2.1	Penalized Regression	20
3.2.2	Extreme Gradient Boosting	22
3.2.3	Long Short-Term Memory Networks	26
3.3	Measuring Prediction Accuracy	28
3.4	Algorithm Tuning and Testing	30
3.4.1	Tuning Hyperparameters	30
3.4.2	Tuning, Training, and Testing Windows	31
4	Predictability Results	32
4.1	Transaction Return Predictability	32
4.2	Price Direction Predictability	34
4.3	Trade Duration Predictability	35
4.4	Performance Consistency Over Time	36
5	Asset Characteristics and Predictability	37
5.1	Liquidity and Volume Trading Intensity	38
5.2	Nominal Share Price and Transaction Trading Intensity	39
5.3	Volatility	40
5.4	Performance Across Stocks and Clock Modes	40
6	The Value of a Millisecond	41
6.1	Predictability Lifespan	42
6.2	Impact of Delay	43
6.3	Peek into Future	45

7	Robustness check	46
7.1	Model Comparison and Performance Across Algorithms	46
7.2	Fine-Tuning of Model Hyperparameters	47
8	Conclusion	48

1 Introduction

Over the years, numerous studies have challenged the classical views of market efficiency by attempting to speculate and capitalize on mispricings, aiming to predict asset returns for various horizons. The foundation of these debates can be traced back to the work of [Fama \(1970\)](#) and [Malkiel \(1973\)](#), who argue that such efforts are inherently limited by the Efficient Market Hypothesis and the Random Walk Hypothesis. According to these theories, market prices fully reflect all available information, making it very hard to consistently achieve returns that exceed the market average.

Contrary to these classical views, [Lo \(2004\)](#) introduced the Adaptive Market Hypothesis, offering a more flexible perspective on market behavior. The AMH suggests that market efficiency is not a static condition but evolves over time as market participants adapt to changing economic conditions, technological advances, and shifts in trading strategies, imposing that long-term predictability is feasible under certain market conditions. [MacKinlay \(2002\)](#) further supports this view, describing long-term predictability as an inherent characteristic of the market, driven by slow-moving factors that influence the underlying economic environment.

It is highly believed that long-term predictability often hinges on fundamental analysis and is best evaluated using economic factors [O’Doherty \(2022\)](#). In contrast, short-term tends to rely more on statistical models, which struggle to cope with the high volatility and noise prevalent in market data [Kyriakou \(2021\)](#).

It is controversial, though, that low-frequency data are sufficient for short-term predictability. [Chang \(2021\)](#) addresses this by demonstrating that combining meta-learning techniques with convolutional neural networks can enhance short-term predictability, even with low-frequency data. [Shen \(2020\)](#) also provides evidence that, with properly tuned machine learning frameworks, it is possible to forecast short-term trends accurately, even when working with lower-resolution data. Despite these advances, the challenges of short-term prediction using low-frequency data remain substantial. Several studies point to issues such as the low signal-to-noise ratio, data insufficiency and overfitting ([Nguyen \(2019\)](#)), as well as the complexities involved in feature extraction ([Y. Chen \(2019\)](#)). Even when short-term predictions are achieved, maintaining consistent and systematic results is difficult ([Timmermann \(2018\)](#)).

The question of predictability using low-frequency data has significant implications, both theoretical and practical. From a theoretical perspective, it touches upon the debate over whether financial markets are truly informationally efficient. Practically, it influences asset allocation strategies, as investors seek to balance risks and returns based on the predictability of market movements. Contrary, high-frequency predictability, has direct implications for the design, operation, and regulation of financial markets, as well as for trading and execution strategies that require rapid adaptation to market conditions.

Given the emerging opportunities arising from the abundant availability of high frequency data and the advancements in computational technologies, an increasing body of research has shifted its focus toward such data, recognizing its potential to uncover short-term market dynamics. Building on this perspective, several studies have demonstrated significant results across different short-term

horizons—ranging from weeks, days to down to a few minutes. [Easley \(2021\)](#), suggests that microstructure dynamics such as realized return correlations, skewness, kurtosis, bid-ask spread, and realized volatility can be predicted efficiently over one week horizon. [Alvim \(2010\)](#), reveals that daily volume can be predicted using factor models. Similarly, [Panayi \(2018\)](#) shows that liquidity demand throughout the trading day exhibits predictable patterns, with significant fluctuations at market openings and closings. [Ntakaris \(2018\)](#) explores the application of advanced machine learning models, demonstrating that, significant accuracy in predicting daily price movements can be achieved using feedforward neural networks.

Our study builds on these foundations by examining not only the broad trends but also the predictability of direct aspects of the next transaction or group of transactions. Specifically, we follow the same methodology used for variable construction as in [Aït-Sahalia \(2022\)](#), where we focus on variables such as the direction, size, and price of upcoming trades, as well as the duration to the next event (such as price change, volume traded, or number of transactions). Price movements play a critical role in shaping trading strategies, whether for aggressive directional trades or more passive market-making activities, while duration variable is crucial for decision-making regarding order placement and cancellation. This is especially relevant for liquidity providers, whose limit orders may be positioned in a queue on the order book, necessitating an estimation of the time required to reach the front of the line. In this context, improved predictability directly enhances profitability. For instance, [Aït-Sahalia and Sağlam \(2021\)](#) present a high-frequency market-making model where the market maker can imperfectly forecast the direction and aggressiveness of incoming orders. Similarly, [Dixon \(2018\)](#) explores methods to evaluate the expected profit and loss of a market maker’s orders under execution constraints based on accurate or inaccurate predictions.

Achieving notable levels of predictability, though, requires methodologies capable of capturing the intricate dynamics of financial markets. A wide range of approaches has been explored, with machine learning models playing a pivotal role in identifying complex, nonlinear relationships within market data. Various data types, also—ranging from traditional price and volume information to alternative sources such as social media sentiment and order book dynamics—have been tested in an effort to harness every available informational edge. [R. D. Huang and Stoll \(1994\)](#), focusing on time series analysis, use a two-equation econometric model, highlighting that lagged returns possess strong predictive power. [Chinco \(2019\)](#), adopts a classical arbitrage approach by utilizing the cross-sectional correlation of stock returns and manages to predict efficiently one-minute-ahead returns by analyzing the past three minutes of the most correlated stocks. [Knoll \(2019\)](#), also tries to exploit statistical arbitrage but in a different context. Specifically, by leveraging social media data he manages to achieve significant twenty-minute-ahead predictions using Twitter sentiment analysis. Limit order book (LOB) data have been also extensively analyzed. [Cont \(2010\)](#) utilizes these data, in order to predict key properties of them using a different approach from regression models. Specifically, by applying continuous Markov processes, treating limit orders as queuing processes waiting to be executed or cancelled, he achieves to predict short-term dynamics of the order book, including the probability of midprice changes and the likelihood of limit order execution before price moves. [Sirignano \(2019\)](#), utilizes the complete set of LOB data to model the joint distribution of bid/ask prices conditional on the current LOB state. By

employing spatial neural networks he demonstrates that deep insights into price movements could be gained through a detailed analysis of the LOB’s intricacies.

In our study, we utilize level-1 quotes, focusing on the best bids and offers in the market,¹ to construct variables that effectively capture volume, imbalance, and spread dynamics by using also various lags, as well as combinations thereof. Specifically, we study the predictability over ultra-short horizons of transaction returns, directions, and durations to specific price, volume or transactions events. Utilizing the complete transactions and quote data for 5 stocks, that were part of the NASDAQ 100 during the period from October 2023 to December 2023, specifically TSLA, NFLX, PEP, AZN and ANSS, we employ a range of machine learning models that span from classical learning methods, such as penalized regression and factor models, to ensemble learning techniques and Artificial Neural Networks. As noted by [Mullainathan \(2017\)](#), these machine learning methods may not be primarily designed for an econometric framework but may be adequate to optimize predictive accuracy.

Using minimal algorithm tuning, we find that out-of-sample predictability for trade duration is notably high, with a median R^2 of approximately 40%, highlighting the strong temporal dependencies in trade arrival patterns. Similarly, we achieve a significant directional accuracy across all clock modes, with a median accuracy over 60%. Although we achieved positive results, the predictability of transaction returns proved to be a considerably more challenging task compared to the other predictive objectives. The results exhibited greater variability and were less consistent across different forecasting horizons and stocks. Nonetheless, we were able to attain out-of-sample R^2 values with a median of approximately 0.6%, indicating the presence of some degree of predictable structure, albeit with more pronounced fluctuations and sensitivity to model specifications. This outcome is partly due to the nature of financial data, which often follows heavy-tailed distributions that exhibit large prediction errors. Stock prices can jump unpredictably, creating substantial outliers in returns and causing R^2 values to fluctuate significantly.² Notably, predictability improves when using a transaction clock approach, where the forecasting horizon is based on transactions, yielding a higher out-of-sample R^2 , just above 1%. In contrast, predicting returns based on a volume-centric horizon appears more difficult, likely due to the variable and order flow-dependent nature of trade arrival times, which introduces additional noise and irregularities in the data. These findings align with the well-documented challenges of return predictability in financial markets.

Besides quantifying the predictability present in the data and assessing how fast it dissipates, we seek to determine which predictor variables are most informative for the next trade and durations. For return and direction predictions, important predictors include the imbalance in the limit order book, recent transaction imbalance, and past trade returns, while statistics derived from recent

¹Our predictions rely solely on widely available trade and quote data. Since fundamental information remains largely unchanged over short time frames, key predictive variables are inferred from recent trades and limit order book dynamics. However, as our information set is a subset of what market participants can access in real time across multiple exchanges, the identified predictability represents a lower bound on what could be achieved by technologically advanced traders.

²Results are obtained under fixed tuning parameter settings to ensure computational efficiency. Consequently, they represent a lower bound on achievable performance, as more advanced computational resources can incorporate minimal tuning adjustments without significant time trade-offs, potentially leading to improved predictive accuracy.

trade volume are most effective for duration predictions.

We then investigate whether variations in predictability can be attributed to stock-level characteristics. Specifically, we try to assess the impact of liquidity, trading intensity based on transactions and volume, nominal share price and volatility while also conducting a comparative analysis of predictive performance across all stocks and clock modes in our sample. Our findings suggest a modest negative correlation between all three examined characteristics (liquidity, share price, volatility) and the predictability of both returns and trade direction, with quite similar patterns, regarding returns. Conversely, these characteristics appear to enhance the predictability of trade duration. However, given the relatively small sample size, these observations should be interpreted with caution. A more notable pattern emerges in the negative correlation between the predictability of returns/direction and that of trade duration. This inverse relationship may be explained by the inherent structure of market activity: when trade duration exhibits higher predictability, market conditions tend to be more stable, reducing short-term inefficiencies that could otherwise be exploited for return and direction forecasting. Conversely, when trade durations are less predictable, heightened market fluctuations and liquidity imbalances may create transient opportunities for return and direction predictability. Furthermore, we observe that directional accuracy is significantly higher under the calendar clock approach for relatively illiquid stocks, whereas for highly liquid stocks, predictability remains more stable across all clock modes and response variables. This highlights the importance of examining and evaluating predictive performance in relation to stock-specific characteristics and differences.

Having established broader patterns of predictability across different stocks, we now shift our focus to Tesla (TSLA), the most liquid and trading-intensive stock in our dataset, as a representative of broader market trends. The selection of TSLA is motivated by its high trading frequency, deep order book liquidity, and significant presence in both retail and institutional trading activity, making it an ideal candidate for testing more refined methodologies.

Our first objective is to examine the lifespan of predictability, determining how long return and direction forecasts remain informative before dissipating. Interestingly, we find that predictability for returns and trade direction peaks at intermediate horizons—approximately 10 seconds, 10,000 total shares traded, and 20 transactions. Beyond these points, return predictability deteriorates sharply, for transaction and volume clock modes, underperforming even the in-sample average after approximately 20,000 shares traded or 50 transactions, while surprisingly, predictability in fixed forward windows (especially 10 seconds or longer) scores a remarkable median out of sample R^2 of 14%. In contrast, while directional accuracy also declines over time, it remains relatively robust even at extended horizons, suggesting that certain microstructural patterns persist despite the broader decay of return predictability. For trade duration, we observe an inverted U-shaped relationship between predictability and horizon length across all clock modes, with substantial predictive power (an out-of-sample R^2 higher than 15%) persisting even at extended horizons, such as 50,000 total shares traded. This finding suggests that although returns become increasingly difficult to forecast at longer horizons, structural patterns in trade durations maintain a more enduring degree of predictability.

Next, we quantify the importance of timeliness in data availability, assessing how delays in

receiving or processing market information affect predictive performance. Our results indicate that return predictability declines significantly after a delay of just 100 milliseconds (ms), and beyond a delay of 1 second, it deteriorates to the point of underperforming in-sample average (our baseline model). Directional accuracy follows a similar trajectory, with a marked decline after 100 ms, yet unlike returns, some level of predictability remains intact even with a 1-second delay. This resilience in trade direction forecasting could stem from the fact that directional shifts often reflect persistent order flow imbalances that unfold over slightly longer time frames, whereas return fluctuations are more sensitive to immediate price changes and high-frequency trading reactions. To provide a more granular understanding of these effects, we systematically map the decline in predictability as a function of delay duration, illustrating the sensitivity of short-term forecasts to real-time data processing constraints.

Lastly, we explore the impact of acquiring partial information on the direction of order flow, simulating the potential advantage gained by a high-frequency trader who, through strategic order placement across multiple exchanges, can infer (albeit imperfectly) the likely direction of the next trade. This approach follows the rationale presented in [Lewis \(2014\)](#), where certain market participants exploit real-time order flow dynamics to enhance short-term predictive accuracy. We analyze how even a limited ability to "look ahead" at incoming trades, by correctly inferring the sign of the next trade with a probability of 65% translates into substantial improvements in return forecasts. Our findings show that such an informational advantage could increase 5-second return predictability by approximately fivefold, reaching an absolute range of 2% to 4%, demonstrating the substantial benefits of even marginal improvements in market awareness.

Finally, we assess the robustness of our findings by conducting a comparative analysis across different machine learning methodologies. This evaluation allows us to determine the extent to which our results are influenced by model selection and whether certain approaches yield more stable and consistent predictive performance. Additionally, we investigate the impact of hyperparameter tuning on model effectiveness, analyzing how variations in parameter configurations affect predictive accuracy.

The paper is structured as follows. Section 2 outlines the various prediction tasks, detailing their respective response variables, the construction of predictor variables, and the evaluation criteria used to assess model performance. Section 3 introduces the dataset and the machine learning methodologies employed, along with the evaluation metrics used for comparison. Section 4 presents the results at the individual stock level, across all response variables and the whole period. In Section 5, we conduct a comparative analysis across all stocks and clock modes, examining the relationship between stock characteristics and predictability while identifying patterns in predictive performance across different time measurement approaches. Section 6 investigates the impact of data timeliness and the potential advantages of anticipating order flow direction on predictability outcomes. Section 7 provides the results of a series of robustness tests to validate our findings. Finally, Section 8 concludes the study.

2 Response and Predictor Variables

2.1 Transactions and Quotes Data

We use standard, widely-available data to construct our variables, specifically derived from the NYSE’s Trade and Quote (TAQ) database for the period of October 2023 till the end of December 2023 (from 02/10/2023 till 29/12/2023), totalling 63 days. This choice of timeframe was made deliberately to observe market behavior after the significant disruptions caused by the COVID-19 pandemic, when its direct effects on market dynamics had slowly been diminishing.

For this analysis, we focus on 5 stocks that were constituents of the NASDAQ 100 index as of December 31, 2023. The NASDAQ 100 includes companies at the forefront of technological innovation, representing industries that are shaping the future of finance, data, software, and more. Unlike traditional indices like the S&P 100, which include a mix of sectors, the NASDAQ 100 provides a concentrated look at companies that are driving digital transformation and automation. Specifically, we analyse the following 5 stocks: Tesla, Inc. (TSLA), Netflix, Inc. (NFLX), PepsiCo, Inc. (PEP), AstraZeneca PLC (AZN), and ANSYS, Inc. (ANSS). These stocks were carefully chosen to provide a representative sample of the broader NASDAQ 100 constituents based on several critical dimensions, including liquidity, trading intensity, nominal share price, and coverage of different technology-driven sectors. TSLA represents the Energy/Automobile sector, NFLX reflects Media/Entertainment, PEP captures Consumer Goods, AZN represents Healthcare, and ANSS covers Software. This selection enables us to project findings to the full NASDAQ 100 index, as these stocks embody the diversity and trading characteristics of the broader index. By focusing on these diverse stocks, we ensure that our analysis captures key market behaviors in a segment characterized by growth potential, technological innovation, and significant trading volumes.

A brief summary of the size of the dataset we use is shown in Table 1. This dataset includes transaction and quote records for the 5 stocks, which span a considerable number of trading days and offer a rich set of information for high-frequency analysis.

Table 2 presents an example of trade and quote data for Tesla Inc. Each row corresponds to an event that occurred at a specific timestamp, identified by its event type—either a trade, a bid quote, or an ask quote. Each event is precisely time-stamped to the nanosecond level³ and includes both the event’s price and size (i.e., quantity). For clarity, the ticker column has been omitted since each dataset is analyzed individually by stock.⁴ Odd-lots are also included, as opposed by O’Hara (2014), as they now account for up to a quarter of total trades and contribute significantly to price discovery⁵. In Table 3 is depicted an example of such trades for NFLX stock, where even

³During model evaluation, we round the timestamp to microseconds for computational efficiency, opposing negligible effect in the results.

⁴The event type TRADE NB is observed only for Tesla (TSLA), whereas for other stocks, trade events are labeled simply as TRADE. A possible reason for this discrepancy could be differences in the data feed or reporting conventions used by the data provider, particularly for trades executed off the main exchange or through specific trading mechanisms.

⁵Quotes are still round-lots, but this should have minimal impact in results.

single share transaction are recorded. Consistent with industry standards, for all stocks considered, both quote and trade prices are reported as multiples of \$0.01.

For each trade, the best bid and ask prices are identified using the most recent quote data available at the exact moment of the transaction. To ensure consistency in our analysis, we exclude all data points that fall outside standard trading hours, strictly considering timestamps between 9:30:00 AM and 4:00:00 PM.⁶ Both the transaction and quote update data play a crucial role in the tuning, training, and testing phases of the algorithms employed in this study.

The summary statistics for our dataset have been adjusted to account for these changes. For a more comprehensive examination of each stock’s summary statistics, we present them individually in Tables 4–8. For each table, the upper panel contains a daily-level summary (covering 63 trading days), while the lower panel of the table contains summary statistics for the response variables computed over all nanosecond-level timestamps for each stock. The total market capitalization refers to the market value of a company’s outstanding shares, calculated using the daily closing price of the stock. Formally, for each stock and date, it is computed as the product of the number of shares outstanding and the daily closing price. This measure serves as a key indicator of a company’s size and market performance.⁷ The total market capitalization, nominal stock price, and daily returns are all based on daily closing prices. Notably, from these tables, we observe that daily returns exhibit low mean values and relatively high standard deviations, indicating modest average gains accompanied by substantial volatility. Skewness and kurtosis values highlight the non-normal distribution of returns, with pronounced positive skewness for returns suggesting occasional large gains, while elevated kurtosis indicates a higher likelihood of extreme return fluctuations. These characteristics align with the volatile nature of high frequency trading and rapid price adjustments observed across the sampled stocks.

2.1.1 Time Clocks

For measuring duration, we utilize three distinct time clocks, each of which captures important aspects of market activity. These clocks are essential for understanding different dimensions of trading activity and market dynamics.

Calendar clock corresponds to the conventional measurement of time, allowing us to relate events in the market to actual time, such as market opening and closing periods. This clock facilitates the comparison of trades and quotes across different assets or trading days. By using the calendar clock, we can analyze time-dependent factors and monitor the real-time behavior of stocks. Its role in providing a standardized framework for timestamping trades ensures consistency in market analyses, making it an essential tool for studying time-specific events.

Complementing this, the *transaction clock* measures time based on the number of transactions, regardless of their size. This approach shifts the focus from the passage of time to the frequency of

⁶Further study could explore the impact of pre-market or after-market hours on trading patterns and model performance.

⁷The number of shares outstanding for each stock was obtained from publicly available sources: TSLA from CompaniesMarketCap.com and NFLX, ANSS, PEP, and AZN from StockAnalysis.com.

trades, capturing the intensity of market activity and revealing patterns in trading frequency, which can be indicative of market liquidity or price movement trends. It is particularly useful in markets where the number of trades holds predictive power for price movements, even when trade volumes fluctuate.

To gain a more detailed understanding of liquidity, we turn to the *volume clock*, which measures time by the total volume of shares traded, irrespective of the number of transactions. This clock, also, provides valuable insights into liquidity dynamics by focusing on the amount of shares exchanged, allowing us to track the impact of large trades or sudden spikes in volume. It helps us uncover the relationship between liquidity and price changes, while capturing significant market shifts that might not be accompanied by frequent transactions but are driven by larger volumes, offering a deeper view into market conditions. This way, we are able to analyze different facets of trading activity, capturing a multi-dimensional view of the market and allowing for more robust and insightful predictions.

All trades and quotes for a given stock are uniquely labeled by the timestamp of their occurrence in calendar time, represented by $t \in \mathbb{R}^+$. Here, t represents the number of seconds since the beginning of the trading day, measured with precision up to a nanosecond (10^{-9}). For instance, $t = 25.335432532$ corresponds to the calendar time of 9:30:25.335432532, which refers to 25 seconds and approximately 335 million nanoseconds after the market opened at 9:30 AM. The number of shares traded at time t is denoted as V_t , where $V_t = 0$ indicates no trade at that time. The indicator function $\mathbf{1}_{\{V_t > 0\}}$ is used to determine whether a trade has occurred at t . The interval of calendar time between two timestamps T_1 and T_2 , where $T_1, T_2 \in \mathbb{R}$, is defined as the half-open, half-closed interval:

$$\text{Int}(T_1, T_2) = \{t \in \mathbb{R} : T_1 \leq t < T_2\}. \quad (1)$$

For simplicity and consistency, we generalize the notion of time intervals to apply to all three time clocks (calendar, transaction, and volume) using a unified definition that works across different response variables. Given a starting timestamp T , a span $\Delta > 0$, and a clock mode $M \in \{\text{calendar, transaction, volume}\}$, the forward-looking time interval is defined as follows:

$$\text{Int}^{\text{forward}}(T, \Delta, M) = \begin{cases} \text{Int}(T, T + \Delta) & \text{if } M = \text{calendar}, \\ \{t > T : \sum_{s \in \text{Int}(T, t)} \mathbf{1}_{\{V_s > 0\}} \leq \Delta\} & \text{if } M = \text{transaction}, \\ \{t > T : \sum_{s \in \text{Int}(T, t)} V_s \leq \Delta\} & \text{if } M = \text{volume}. \end{cases} \quad (2)$$

Under the calendar clock, Δ represents the target time horizon over which the prediction is made. In the transaction clock, Δ refers to the target number of trades: the interval includes all timestamps after T such that the total number of trades does not exceed Δ . Similarly, under the volume clock, Δ represents the target total volume traded: the interval includes all timestamps after T such that the cumulative volume traded does not exceed Δ . The interval in all cases is defined as a set of consecutive timestamps based on the respective clock mode.

Let the set of timestamps represent all the moments when either a trade occurs or when a quote

is updated. These timestamps track both the trades and the changes in quoted prices.

We use the National Best Bid and Offer (NBBO) prices, which represent the best bid price (the highest price someone is willing to pay) and the best ask price (the lowest price someone is willing to accept). These are represented as P_t^b for the best bid price and P_t^a for the best ask price at a given time t . The mid-price, which gives us an average price between the bid and ask, is calculated as:

$$P_t = \frac{P_t^b + P_t^a}{2}$$

The price at which the actual transaction happens is called the transaction price, and the sizes of the best bid and best ask are recorded as S_t^b and S_t^a respectively, for the record indexed by t .

2.1.2 Transaction Return

At time $T \in D$, with a span Δ and a clock mode M , the transaction return is defined as:

$$\text{Return}(T, \Delta, M) = \text{Average} \left[P_t^{\text{txn}} : t \in D^{\text{txn}} \cap \text{Int}^{\text{forward}}(T, \Delta, M) \right] / P_T - 1. \quad (2.4)$$

This expression captures the average return of transactions executed within a forward-looking time window, typically over a short period when Δ is small. By averaging across multiple transactions, this calculation reduces noise compared to using the return from a single transaction at a specific point in time. The result is a smoother and more representative measure of the overall trading behavior during the interval Δ , minimizing the effect of short-term price fluctuations.

Transaction returns are particularly important for market makers. In fast-moving markets, they continuously adjust their quotes based on incoming information and must make quick decisions about whether to execute, modify, or cancel their orders. Since the precise timing of an order's execution is difficult to predict, they benefit from using average returns over short time windows. This helps them assess the likely price movements in the near future, allowing them to provide liquidity while managing the risk of adverse price changes. By considering the transaction return over a forward interval, market makers can more accurately anticipate the impact of trades, improve their order placement, and adjust their strategies to maintain profitability while minimizing exposure to unfavorable price movements.

This measure is not only beneficial for market makers but also for traders looking to speculate or manage risk by analyzing price movements. Traders can leverage transaction return data to identify profitable trading opportunities by focusing on the distribution of returns and using quantile-based strategies. By examining specific quantiles of the transaction return distribution, traders can better understand the likelihood of extreme price movements, allowing them to position themselves accordingly. For example, traders may choose to execute trades based on higher quantiles, where there is a higher probability of significant price increases, or focus on lower quantiles to short the market in anticipation of price declines. This approach enables them to make more informed decisions by considering the range and potential extremes of transaction returns rather than relying on average movements alone. As shown by [Narajewski and Ziel \(2022\)](#), these strategies help traders manage tail risks, as quantile forecasts are used to optimize both speculative

trades and risk mitigation in volatile market conditions.

2.1.3 Price Direction

The next key variable we focus on is the prediction of whether the price will move upwards or downwards over a short horizon. At time $T \in D$, with a given time span Δ and time clock M , the price direction is defined as:

$$\text{Direction}(T, \Delta, M) = 1 \{ \text{Return}(T, \Delta, M) > \bar{R}(\Delta, M) \}. \quad (2.5)$$

This variable transforms the transaction return into a binary indicator that captures whether the price movement is up (1) or down (-1). It helps regularize extreme values and smooth out noise that might be present in the raw return data, making it easier for predictive models to focus on general trends rather than the magnitude of price changes.

By normalizing the transaction return in this way, the model can focus purely on direction, allowing for more robust predictions in volatile environments. While directionality removes the granularity of the return magnitude, it remains a crucial factor in trading strategies, where accurate predictions of price direction can lead to substantial profitability. This is especially true in high-frequency trading, where small directional movements, executed repeatedly over short periods, can result in cumulative gains.

Moreover, market makers and traders alike find directionality essential. For market makers, accurately predicting the direction allows for better positioning of orders in the order book. Traders can use directional signals to either enter or exit trades in anticipation of short-term price movements. The ability to predict even minor directional shifts enhances decision-making, helping both avoid potential losses by exiting unfavorable positions early.

2.1.4 Transaction Duration

At time $T \in D$, with a given time span Δ and clock $M \in \{\text{transaction, volume}\}$, we define the transaction duration as follows:

$$\text{Duration}(T, \Delta, M) = \arg \max_{t \in D} \left\{ t \in \text{Int}^{\text{forward}}(T, \Delta, M) \right\} - T. \quad (2.6)$$

This measures the amount of (calendar) time it takes to observe either Δ transactions or Δ shares traded. Note that by definition, $\text{Duration}(T, \Delta, \text{calendar}) = \Delta$, making this measurement particularly relevant when the time clock is based on transactions or volume.

Predicting transaction duration is crucial as it directly reflects trading intensity, which plays a significant role in optimizing various trading strategies. For traders executing large orders relative to Δ , accurately forecasting the duration enables them to adjust their execution schedule more effectively, splitting orders and timing their actions for better market impact. For market makers, transaction duration prediction is equally important, particularly for quote placement and cancellation strategies. Specifically, they can anticipate how fast orders will execute, which helps them

decide when to modify or cancel quotes before they reach the front of the queue..

2.2 Predictor Variables

To assess how well the response variables can be predicted, we consider a diverse set of predictor (or independent) variables that capture essential features of the short-term trading environment for a specific stock. At this stage, we rely solely on a stock’s own variables to forecast its future movements.⁸

Just like the response variables, the predictor variables are constructed entirely from the time-stamped transaction and quote data. These variables represent various transformations (possibly nonlinear) of the data recorded before the point in time T at which the prediction occurs.

One critical aspect to consider when building these predictors is determining the optimal length of the lookback window before T . We use multiple disjoint lookback windows to ensure that we capture a broad range of market activity—ranging from the most recent events (on the scale of milliseconds) to more distant historical data (on the scale of minutes). This broad range is essential because market behaviors often fluctuate at different time scales. By including both ultra short-term and longer term windows, we avoid prematurely dismissing important predictive patterns that might emerge from different market conditions.

It is also important to note that the length of the most informative lookback window is unlikely to be the same for each predictor. Different variables might derive their predictive power from different time spans. For instance, liquidity-related variables might respond more to short-term fluctuations, while price or volume trends could require a longer historical window to reveal meaningful patterns. Therefore, machine learning algorithms are particularly well-suited to this task, as they can efficiently handle the complexity of choosing the optimal lookback window for each predictor.

Similar to the forward-looking intervals (Section 2.2) for the response variables, we construct lookback intervals to build predictor variables.⁹ For calendar time at the timestamp T , lookback spans (Δ_1, Δ_2) , where $\Delta_1 \leq \Delta_2$ and the time clock M , are defined as:

$$\text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M) = \begin{cases} \text{Int}(T - \Delta_2, T - \Delta_1) & \text{if } M = \text{calendar}, \\ \{t : t \leq T, \sum_{s \in \text{Int}(T, t)} \mathbf{1}_{\{V_s > 0\}} < \Delta_2\} & \text{if } M = \text{transaction}, \\ \{t : t \leq T, \sum_{s \in \text{Int}(T, t)} V_s < \Delta_2\} & \text{if } M = \text{volume}. \end{cases} \quad (2.7)$$

⁸Predictions could potentially be enhanced by leveraging sector-level or industry-wide correlation patterns, where observations from other stocks with similar behaviors might help improve prediction accuracy—a technique often referred to as “statistical arbitrage.”

⁹We exclude the initial observations that do not extend as far back as the longest lookback window, to ensure computational efficiency. This exclusion has a negligible impact on the results, as the total number of observations remains sufficiently large to preserve statistical validity.

For each timestamp T and clock mode M , we set the lookback spans (Δ_1, Δ_2) to generate features at T with multiple disjoint intervals.

For the calendar clock ($M = \text{calendar}$), we now use seven lookback intervals:

$$(\Delta_1, \Delta_2) \in \{(0, 0.1), (0.1, 0.25), (0.25, 0.5), (0.5, 1), (1, 2.5), (2.5, 5), (5, 10)\}$$

These intervals are chosen to capture market behavior at very short time scales, as we expect the bulk of predictability to lie within the first few seconds after an event. In high-frequency trading, the most impactful trades and price changes tend to happen almost immediately following new information, making intervals in the range of milliseconds to a few seconds particularly important. By extending the maximum lookback to 10 seconds, we ensure that we capture both the immediate market reaction and the short-term adjustments that follow.

For the transaction clock ($M = \text{transaction}$), the seven lookback intervals are:

$$(\Delta_1, \Delta_2) \in \{(0, 1), (1, 2), (2, 5), \dots, (50, 100)\}$$

These intervals are based on the number of transactions. The separation from $(0, 1)$ to $(50, 100)$ allows us to analyze the market at varying levels of trading intensity. For instance, the $(0, 1)$ interval captures the market behavior immediately after a single transaction, whereas the larger intervals (like $(50, 100)$) account for patterns that emerge over a much larger number of trades, potentially highlighting slower market trends. This step-wise expansion allows us to differentiate between short bursts of high activity and longer-term transaction flows.

Similarly, for the volume clock ($M = \text{volume}$), the seven spans are:

$$(\Delta_1, \Delta_2) \in \{(0, 100), (100, 250), \dots, (5000, 10000)\}$$

These volume-based intervals measure the number of shares traded over different scales. We begin with smaller increments (like $(0, 100)$ and $(100, 250)$) to focus on periods of low-volume trading that could still have predictive power. The larger intervals, like $(5000, 10000)$, allow us to capture significant shifts in market liquidity or larger block trades, which tend to have a more profound impact on price movements. By scaling up incrementally, we can better observe how changes in volume affect price dynamics over time.

We utilize a total of 13 main predictors, each applied across 7 time spans and 3 time clocks. Additionally, the predictors can interact in a variety of nonlinear ways, with these interactions being selected by the machine learning algorithm as part of the predictive modeling process. The 13 predictors can be grouped into three distinct categories, which we will now outline.

Volume and Duration

The first group of predictors relates to the stock's trading intensity. A higher than usual frequency of transactions or block trades may suggest persistent trading patterns that extend over short time horizons, potentially offering predictive insights. This group of predictors captures various dimensions of trading volume and transaction frequency, which are key indicators of market activity.

1. **Breadth** is the number of transactions in the interval¹⁰:

$$\text{Breadth}(T, \Delta_1, \Delta_2, M) = |\mathcal{D}^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M)| \quad (2.8)$$

2. **Immediacy** is the average time between successive transactions in the interval:

$$\text{Immediacy}(T, \Delta_1, \Delta_2, M) = \frac{\Delta_2 - \Delta_1}{\text{Breadth}(T, \Delta_1, \Delta_2, M)} \quad (2.9)$$

3. **VolumeAll** is the total number of shares transacted in the interval:

$$\text{VolumeAll}(T, \Delta_1, \Delta_2, M) = \sum_{t \in \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M)} V_t \quad (2.10)$$

4. **VolumeMax** is the maximum number of shares transacted in a single transaction within the interval:

$$\text{VolumeMax}(T, \Delta_1, \Delta_2, M) = \max \left\{ V_t : t \in \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M) \right\} \quad (2.12)$$

Return and Imbalance

The second group of predictors is related to the stock's recent trading asymmetry. For example, if a majority of trades are buy trades that hit the limit sell, or if the bid is dominating the ask in the level 1 quotes, we might expect to see upward pressure on the price. It is natural to expect that elements of the limit order book (LOB), including any imbalances, will be predictive of future returns and durations. Imbalances are often indicative of future price movements (see, e.g., [Cont \(2014\)](#) and [Kercheval and Zhang \(2015\)](#)). We define the following variables:

1. **Lambda** measures the price change in the interval relative to the total volume. It is computed as:

$$\text{Lambda}(T, \Delta_1, \Delta_2, M) = \frac{P_{\max(\mathbb{I})} - P_{\min(\mathbb{I})}}{\text{VolumeAll}(T, \Delta_1, \Delta_2, M)} \quad (2.13)$$

2. **LobImbalance** is the average imbalance in the depth of the limit order book over the look-back interval:

$$\text{LobImbalance}(T, \Delta_1, \Delta_2, M) = \text{Average} \left[\frac{S_t^a - S_t^b}{S_t^a + S_t^b} : t \in \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M) \right] \quad (2.14)$$

3. **TxnImbalance** measures the asymmetry between buy and sell volumes in recent transactions. Denoted by Dir_t^{LR} , the binary transaction direction at time t is signed using the algo-

¹⁰for transaction clock mode this predictor is not applicable, so instead, we use the duration of different lookback intervals based on executed trades

rithm of [Lee and Ready \(1991\)](#). Then transaction imbalance is calculated as:

$$\text{TxnImbalance}(T, \Delta_1, \Delta_2, M) = \frac{\sum_{t \in D^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M)} (V_t \cdot \text{Dir}_t^{LR})}{\text{VolumeAll}(T, \Delta_1, \Delta_2, M)} \quad (2.15)$$

4. **PastReturn** is the past return in the interval. It is defined similarly to the transaction return response, except over a lookback window:

$$\text{PastReturn}(T, \Delta_1, \Delta_2, M) = 1 - \text{Average} \left[\frac{P_t^{\text{txn}}}{P_{\max}(\text{I})} : t \in I \right] \quad (2.16)$$

Speed and Cost

The final set of predictors we use measures the speed and cost components inherent in stock trading. These predictors capture various dimensions of transaction speed, price movement correlations, and both the quoted and realized trading costs.

1. **Turnover** represents the speed of transactions relative to the stock's total number of shares outstanding (denoted as S).¹¹ It is defined as:

$$\text{Turnover}(T, \Delta_1, \Delta_2, M) = \frac{\text{VolumeAll}(T, \Delta_1, \Delta_2, M)}{S} \quad (2.17)$$

2. **AutoCov** is the autocovariance of transaction returns in the interval. It captures how quickly information is incorporated into prices by measuring how past price movements relate to future movements. Faster information absorption leads to a quicker decay in autocovariance. It is computed as:

$$\text{AutoCov}(T, \Delta_1, \Delta_2, M) = \text{Average} \left[\log \left(\frac{P_t^{\text{txn}}}{P_{L_t}} \right) \log \left(\frac{P_t^{\text{txn}}}{P_{L_t}} \right) : t \in D^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M) \right] \quad (2.18)$$

where L_t is the timestamp of the transaction right before time t .

3. **QuotedSpread** represents the average proportional nominal spread in the quotes over the lookback interval. This spread is often interpreted as the transaction cost from the perspective of market participants, indicating the difference between the best bid and ask prices. It is calculated as:

$$\text{QuotedSpread}(T, \Delta_1, \Delta_2, M) = \text{Average} \left[\frac{P_t^a - P_t^b}{P_t} : t \in \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M) \right] \quad (2.19)$$

where P_t^a and P_t^b are the ask and bid prices, respectively.

¹¹While turnover is generally expected to exhibit relatively low values, it may exceed unity in rare instances. Such occurrences indicate that the traded volume within a given interval surpasses the total number of shares outstanding, which is typically associated with periods of heightened market activity and rapid trading dynamics

4. **EffectiveSpread** is the dollar-weighted realized effective spread over the interval. It measures the realized transaction cost from the trader’s perspective and the realized profit from the market maker’s perspective. It is defined as:

$$\text{EffectiveSpread}(T, \Delta_1, \Delta_2, M) = \frac{\sum_{t \in D^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M)} \left[\log \left(\frac{P_t^{\text{txn}}}{P_t} \right) \cdot \text{Dir}_t^{LR} \cdot V_t \cdot P_t^{\text{txn}} \right]}{\sum_{t \in D^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, M)} (V_t \cdot P_t^{\text{txn}})} \quad (2.20)$$

where P_t^{txn} is the transaction price, Dir_t^{LR} is the trade direction, and V_t is the volume traded.

3 Machine Learning Methods

3.1 Model Categories

We proceed by organizing our predictive models into three principal categories: Classical, Ensemble, and Deep Learning. Our objective is to conduct a comprehensive analysis across these categories, while also providing a concise overview of the primary models used for predicting stock returns and durations.

3.1.1 Classical Learning

Classical learning methods are divided into supervised and unsupervised learning techniques. In the supervised learning category, we employ penalized regressions models, specifically, LASSO and Adaptive LASSO, which are particularly useful in high-dimensional data environments where variable selection is necessary. These models apply an L1 penalty to prevent overfitting and improve out-of-sample predictive performance. In addition, we explore FarmPredict as an unsupervised learning approach, designed to group similar variables and reduce dimensionality without relying on pre-labeled data, see [Fan \(2020\)](#) for more details.

3.1.2 Ensemble Learning

Ensemble learning combines multiple base learners to improve prediction accuracy, divided into Boosting and Bagging methods. For Boosting, we use Extreme Gradient Boosting (XGBoost), a scalable gradient-boosted decision tree that works by sequentially improving model performance. For Bagging, we implement the Random Forest algorithm, which builds numerous decision trees on random subsets of data and averages their predictions to reduce variance. This method is particularly effective in high-dimensional settings.

3.1.3 Deep Learning

Our third model category, Deep Learning using Artificial Neural Networks (ANN’s), is represented by a recurrent architecture. We employ Recurrent Neural Networks (RNN), specifically

Long Short-Term Memory (LSTM) networks, to capture temporal dependencies in the data. LSTM models are suitable for financial time series due to their ability to learn such short as long-term dependencies and avoid vanishing gradient problems.

3.2 Model Presentation

The models selected as representative for each category and which we briefly describe their methodologies are the Least Absolute Shrinkage and Selection Operator (LASSO), Extreme Gradient Bossting (XGB), and Long Short-Term Memory (LSTM) networks, respectively. In Section 7.1, we present a comprehensive evaluation of these methods, alongside the additional models we mentioned before (FarmPredict using Adaptive Lasso, Random Forest (RF), and a simple OLS regression).

Consider the regression problem of predicting a response variable Y using a predictor vector \mathbf{X} , based on a random sample $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$. Let $\mathbf{Y} = (y_1, y_2, \dots, y_n)^T$. In our data, each \mathbf{X}_i has a dimension of 91, consisting of 7 time spans for each of the 13 predictor variables. The algorithms can then endogenously construct further combinations of these variables, as well as select the most informative subsets of variables.

3.2.1 Penalized Regression

Least Absolute Shrinkage and Selection Operator (LASSO) creates a regression model that is penalised with the L1-norm, which is the sum of the absolute coefficients. Because of the nature of this constraint, it tends to produce some coefficients that are exactly equal to zero and, hence, gives more interpretable models. However, as a least-squares-based method, LASSO lacks robustness to heavy-tailed data. Given that the return variable exhibits substantial kurtosis, as shown in Table , extreme values can disproportionately influence the estimates. To mitigate this issue, we apply clipping only to the return variable, truncating its training responses at the 5th and 95th percentiles to reduce the impact of outliers and enhance model stability.¹²

As originally noted by Tibshirani (1996), the LASSO regression is better suited for predictor selection compared to other penalized regressions, because the former method performs model/predictor selection, keeping those variables which are more suitable for forecasting. The optimisation problem is:

$$\min_{\beta_N} \left\{ \sum_{t=1}^T (y_t - a - \beta_N^T x_{t,N})^2 + \lambda \sum_{i=1}^N |\beta_i| \right\}.$$

Although we cannot write the explicit formulas for the bias and variance of the LASSO estimator, the general trend is that the bias increases as λ increases and the variance decreases as λ increases.

¹²This technique, known as Winsorization, is commonly used in statistics to limit the influence of extreme values. Theoretical justifications for its application in high-dimensional settings can be found in Fan (2020). We also tested alternative thresholds (1st–99th and 10th–90th percentiles), observing that broader clipping enhances stability, while narrower clipping increases the risk of overfitting.

Following [Bühlmann and van de Geer \(2011\)](#), we summarise the key properties and corresponding assumptions for the LASSO. Considering the true model in Equation (2), it is:

$$\frac{1}{T} \sum_{t=1}^T \left(x_{t,N} \left(\hat{\beta}_N^{\text{LASSO}} - \beta \right) \right)^2 = O_P \left(\sum_{i=1}^N |\beta_i| \sqrt{\frac{\log(N)}{T}} \right),$$

where $O_P(\cdot)$ is with respect to $N \geq T \rightarrow \infty$. This implies that we achieve consistency of prediction if $\sum_{i=1}^N |\beta_i| \ll \sqrt{\frac{T}{\log(N)}}$.

Faster convergence rate and estimation error bounds with respect to the L1- or L2-norm can be achieved using the so-called oracle optimality condition:

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \left(x_{t,N} \left(\hat{\beta}_N^{\text{LASSO}} - \beta \right) \right)^2 &= O_P \left(s_0 \phi^{-2} \log(N)/T \right), \\ \sum_{i=1}^N \left(\hat{\beta}_i^{\text{LASSO}} - \beta_i \right)^q &= O_P \left(s_0^{1/q} \phi^{-2} \sqrt{\frac{\log(N)}{T}} \right), \quad q \in \{1, 2\}, \end{aligned}$$

where s_0 equals the true number of non-zero regression coefficients and ϕ^2 is the compatibility constant or restricted eigenvalue, which is a number depending on the compatibility between the design and the L1-norm of the regression coefficient. The above rate is optimal up to the $\log(N)$ factor and the restricted eigenvalue ϕ^2 .

Additionally, to the oracle optimality and assuming the beta-min condition:

$$\min_{i \in S_0} |\beta_i| \gg \phi^{-2} \sqrt{s_0 \log(N)/T},$$

we obtain the screening variable property:

$$P(\hat{S} \supseteq S) \rightarrow 1 \quad (N \geq T \rightarrow \infty),$$

where $\hat{S} = \{i : \hat{\beta}_i^{\text{LASSO}} \neq 0, i = 1, \dots, N\}$ and $S = \{i : \beta_i \neq 0, i = 1, \dots, N\}$. Consistent variable selection then means:

$$P(\hat{S} = S) \rightarrow 1 \quad (N \geq T \rightarrow \infty).$$

The Adaptive Lasso (A-Lasso), introduced by [Zou \(2006\)](#), enhances the traditional Lasso by incorporating adaptive weights into the L1 penalty term. Unlike the original Lasso, which applies the same penalty to all coefficients, A-Lasso adjusts the penalty based on initial estimates of the coefficients, providing more flexibility and accuracy in variable selection. The weights \hat{w}_i are inversely proportional to the absolute value of the initial coefficient estimates. This leads to improved consistency in selecting non-zero coefficients and better handling of high-dimensional data.

The optimisation problem now becomes:

$$\min_{\beta_N} \left\{ \sum_{t=1}^T (y_t - a - \beta_N^T x_{t,N})^2 + \lambda \sum_{i=1}^N \hat{w}_i |\beta_i| \right\},$$

where $\hat{w}_i = 1/|\hat{\beta}_{\text{init},i}|^\gamma$, $\hat{\beta}_{\text{init}}$ is an initial estimator, and $\gamma > 0$. Typically, the initial estimator is the Lasso solution, with the regularization parameter tuned via cross-validation. After obtaining the initial estimates, cross-validation is used again to determine the optimal λ in the Adaptive Lasso formulation. More technical details on the Adaptive Lasso can be found in [J. Huang \(2008\)](#).

3.2.2 Extreme Gradient Boosting

Decision trees are widely used, non-parametric models suitable for both classification and regression tasks. They excel in capturing non-linear relationships between input features X and target variables Y without requiring assumptions about the underlying data distribution. This adaptability makes them powerful tools for datasets with complex variable interactions. Their construction evolves recursive partitioning of the feature space, where the algorithm, in each step, selects a feature j and a corresponding split point s that divides the data into two regions:

$$R_1(j, s) = \{X_i | X_{ij} \leq s\}, \quad R_2(j, s) = \{X_i | X_{ij} > s\}.$$

The objective is to select j and s that minimize the impurity¹³ or error in the resulting partitions. For regression tasks, this is typically expressed as minimizing the sum of squared errors (SSE):

$$(\hat{j}, \hat{s}) = \arg \min_{j, s} \left[\sum_{i \in R_1(j, s)} (y_i - \bar{y}_1)^2 + \sum_{i \in R_2(j, s)} (y_i - \bar{y}_2)^2 \right],$$

where \bar{y}_1 and \bar{y}_2 are the mean target values in regions R_1 and R_2 , respectively.

To prevent overfitting or underfitting, decision trees rely on predefined stopping criteria that govern the growth of the tree. One commonly applied criterion is the limitation of the maximum depth (d_{max}) of the tree, which controls the number of splits and thereby constrains the overall complexity of the model. Another important criterion is the specification of a minimum leaf size, ensuring that each terminal node contains a sufficient number of observations to avoid overly granular partitions that may lead to overfitting. Additionally, the tree construction process may terminate if further splits fail to result in a significant reduction in the error metric, such as the sum of squared errors (SSE) or another chosen metric. This threshold ensures that splits are only performed when they provide meaningful improvements to the model's performance, effectively balancing complexity and generalization.

While decision trees are effective, their simplicity can lead to overfitting when grown deep or

¹³Impurity measures the degree of heterogeneity of the target variable within a partition. Common impurity metrics include Gini Index, entropy, and variance for classification tasks, or the sum of squared errors (SSE) for regression tasks.

underfitting when too shallow. This motivates the development of ensemble methods like Bagging and Boosting.

Bagging, or bootstrap aggregation, is a powerful ensemble learning technique designed to reduce the variance of decision trees by training multiple models independently on bootstrap samples, which are random subsets of the original data created with replacement. Each decision tree is trained on a unique bootstrap sample, and the final prediction is obtained by aggregating the outputs of all trees. For regression tasks, the predictions are averaged, whereas for classification tasks, majority voting is used to determine the final class label. Mathematically, given M decision trees $\{h_1, h_2, \dots, h_M\}$, the prediction for a new instance X_{new} is expressed as:

$$\hat{y}_{\text{new}} = \frac{1}{M} \sum_{m=1}^M h_m(X_{\text{new}}).$$

An exemplary application of bagging is the Random Forest algorithm, introduced by [Breiman \(2001\)](#). Random Forests extend the bagging approach by incorporating random feature selection at each split during tree construction. This additional layer of randomness decorrelates the individual trees, improving model robustness and reducing the likelihood of overfitting while maintaining strong predictive performance. (See [Breiman \(2001\)](#) for more details)

Unlike bagging, where individual models are trained independently in parallel, boosting operates sequentially, with each new model focusing on correcting the errors made by its predecessors. It is a powerful technique that aims to improve the performance of weak learners by combining them into a single strong predictive model. This iterative approach allows boosting to systematically reduce bias and improve the predictive accuracy of the ensemble.

Formally, consider a training dataset $\{(X_i, y_i)\}_{i=1}^N$, where $X_i \in \mathbb{R}^d$ represents the feature vectors, and y_i the corresponding target values. The objective of boosting is to construct a predictive function $F(X)$ that minimizes a predefined loss function $L(y, F(X))$, which measures the discrepancy between the true target y and the model's predictions $F(X)$. The predictive function is represented as an additive model:

$$F(X) = \sum_{m=1}^M \alpha_m h_m(X),$$

where:

- $h_m(X)$ are the individual weak learners, typically decision trees.
- α_m are the weights assigned to each learner, reflecting their contribution to the final model.
- M is the total number of iterations (i.e., the number of weak learners combined).

The boosting algorithm begins with an initial constant prediction $F_0(X)$, which is typically chosen to minimize the loss function over the training data. For example, in regression tasks with a squared error loss function, $F_0(X)$ would be the mean of the target values:

$$F_0(X) = \arg \min_c \sum_{i=1}^N L(y_i, c).$$

In subsequent iterations, the model is updated by adding a new weak learner $h_m(X)$ to the ensemble. Each weak learner is trained to approximate the negative gradient of the loss function with respect to the current model's predictions, which acts as a proxy for the residual error:

$$r_i^{(m)} = -\frac{\partial L(y_i, F_{m-1}(X_i))}{\partial F_{m-1}(X_i)},$$

where $r_i^{(m)}$ represents the pseudo-residuals for observation i at iteration m . The weak learner $h_m(X)$ is then trained to minimize the squared error between its predictions and the pseudo-residuals:

$$h_m(X) = \arg \min_h \sum_{i=1}^N \left(r_i^{(m)} - h(X_i) \right)^2.$$

The contribution of the weak learner is scaled by a learning rate parameter η , which controls the step size of the updates and helps prevent overfitting. The updated predictive function at iteration m is given by:

$$F_m(X) = F_{m-1}(X) + \eta \alpha_m h_m(X).$$

The process continues for M iterations, with each iteration aiming to reduce the loss function further. The final predictive function $F(X)$ is the sum of all the weak learners, each weighted by its respective α_m .

While Gradient Boosting provides a powerful foundation, Extreme Gradient Boosting (XGBoost), introduced by [T. Chen and Guestrin \(2016\)](#), introduces several significant enhancements to address the limitations of GBM. These enhancements improve the efficiency, scalability, and accuracy of the boosting process, especially for high-dimensional and large-scale datasets, such as those found in high-frequency financial data.

A key innovation in XGBoost is the inclusion of a regularization term in the objective function, which helps control model complexity and prevents overfitting. The regularized objective function is defined as:

$$\mathcal{L} = \sum_{i=1}^N L(y_i, \hat{y}_i) + \sum_{t=1}^T \Omega(h_t),$$

where:

- $L(y_i, \hat{y}_i)$ represents the loss function (e.g., squared error for regression or log-loss for classification),

- $\Omega(h_t)$ is the regularization term for tree complexity, defined as:

$$\Omega(h) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2,$$

- T is the number of leaves in the tree,
- γ is a penalty term controlling the number of leaves,
- λ is an L_2 regularization parameter applied to the leaf weights w_j .

XGBoost further improves upon GBM by using a second-order Taylor expansion of the loss function for optimization. This allows XGBoost to leverage both the gradient and the Hessian (second derivative) of the loss function, leading to more precise updates. The second-order approximation is given by:

$$\mathcal{L}^{(t)} = \sum_{i=1}^N \left[g_i h_t(X_i) + \frac{1}{2} h_i h_t^2(X_i) \right] + \Omega(h_t),$$

where:

- $g_i = \frac{\partial L(y_i, F(X_i))}{\partial F(X_i)}$ is the first-order gradient,
- $h_i = \frac{\partial^2 L(y_i, F(X_i))}{\partial^2 F(X_i)}$ is the second-order Hessian.

This second-order approximation enables XGBoost to converge faster and make better use of the training data, particularly for complex datasets. By incorporating both first and second-order derivatives, XGBoost achieves higher optimization precision compared to traditional gradient boosting methods.

After training M trees, the final prediction for a new instance X_{new} is computed as the sum of predictions from all weak learners:

$$\hat{y}_{\text{new}} = \sum_{m=1}^M h_m(X_{\text{new}}).$$

For classification tasks, the raw predictions are transformed into probabilities using the sigmoid function:

$$P(y = 1 | X) = \frac{1}{1 + e^{-\hat{y}}}.$$

For regression tasks, \hat{y} is used directly as the predicted output.

In the context of high-frequency financial data, characterized by high dimensionality, sparsity, and noise, XGBoost excels due to its regularization framework, which mitigates overfitting, and second-order optimization, which captures complex non-linear patterns. The parallelized implementation ensures efficient handling of large volumes of data, making XGBoost particularly suitable for real-time decision-making scenarios in high-frequency trading.

3.2.3 Long Short-Term Memory Networks

Recurrent Neural Networks (RNNs) are widely used for modeling sequential data due to their ability to maintain a hidden state across time steps, allowing them to capture context in data (G. Huang (2002)). In an RNN, the hidden state h_t at any time step t is computed as a function of the current input X_t and the hidden state from the previous time step h_{t-1} . The general formula for this is:

$$h_t = F_h(X_t, h_{t-1}) = \sigma(W_h X_t + U_h h_{t-1} + b_h),$$

where W_h and U_h are weight matrices that control how the current input and previous hidden state contribute to the new hidden state, and b_h is a bias term. The function σ typically represents a non-linear activation function such as the tanh or ReLU function.

For continuous outputs, the RNN computes the final output Y as a transformation of the hidden state at the last time step T :

$$Y = W_y h_T + b_y,$$

where W_y and b_y are the weight matrix and bias for the output layer. If the output is categorical, such as in classification tasks, a softmax function is applied:

$$Y = \text{softmax}(W_y h_T).$$

While RNNs are effective in many applications, when trained with Back-Propagation Through Time (BPTT) (Werbos (1990)) they suffer from a significant drawback known as the vanishing gradient problem. This occurs when gradients, during the process of backpropagation, shrink exponentially as they are propagated back through many time steps. As a result, RNN's struggle to learn longer term dependencies in sequential data.

To address these limitations, Long Short-Term Memory networks were introduced (Hochreiter (1997)). LSTM's are a variant of RNN's that incorporate a more sophisticated architecture capable of learning long-term dependencies while preventing the loss of important information over time. Another difference of the LSTM compared to the vanilla RNN is the use of multiple sigmoidgates that control the information flow through the model. Layer normalization (Ba and Hinton (2016)) is also introduced in our model to prevent neurons from saturating, by keeping their inputs centered.

LSTMs extend RNNs by introducing memory cells that can maintain information over long sequences. These memory cells are regulated by three critical gates: the input gate, the forget gate, and the output gate. These gates control the flow of information in and out of the memory cell and determine which information is stored, discarded, or outputted at any given time.

At each time step t , the LSTM processes the current input X_t , the previous hidden state h_{t-1} , and the previous cell state c_{t-1} . The three gates (input, forget, and output) play distinct roles in regulating the information flow within the LSTM unit.

The input gate controls the amount of new information that enters the memory cell. It is defined

as:

$$i_t = \sigma(W_i X_t + U_i h_{t-1} + b_i),$$

where W_i , U_i , and b_i are the weights and bias for the input gate, and σ is the sigmoid activation function.

The forget gate determines how much of the previous cell state should be retained or discarded. The forget gate is calculated as:

$$f_t = \sigma(W_f X_t + U_f h_{t-1} + b_f),$$

where W_f , U_f , and b_f are the weights and bias for the forget gate.

Once the input and forget gates have been calculated, the LSTM generates a candidate cell state \tilde{c}_t using the current input and previous hidden state:

$$\tilde{c}_t = \tanh(W_c X_t + U_c h_{t-1} + b_c),$$

The new cell state c_t is then updated based on both the forget gate and the input gate:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t,$$

where \odot denotes element-wise multiplication. This update allows the cell to selectively retain information from the past and incorporate new information.

Finally, the output gate controls how much of the cell state should be used to compute the hidden state for the current time step. It is computed as:

$$o_t = \sigma(W_o X_t + U_o h_{t-1} + b_o),$$

and the new hidden state h_t is computed as:

$$h_t = o_t \odot \tanh(c_t),$$

where \tanh is the hyperbolic tangent function, ensuring that the hidden state captures both current and past information.

The key innovation of LSTMs is the gating mechanism, particularly the forget gate, which allows them to effectively control the flow of information through the network. By maintaining a constant error flow through the memory cells, they prevent gradients from vanishing as they are propagated backwards through time, a problem that plagues standard RNNs. This LSTM's ability to retain long-term dependencies and selectively forget irrelevant information makes it a powerful tool for sequence modeling.

3.3 Measuring Prediction Accuracy

Assessing prediction accuracy involves multiple metrics to ensure a comprehensive evaluation of model performance. Each metric captures different aspects of the prediction task, such as the magnitude of prediction errors, directional accuracy, and robustness to outliers. This combination allows us to validate model robustness across varying market conditions, especially given the volatile nature of financial data. Below, we discuss the primary metrics used.

The out-of-sample R^2 is a widely recognized metric in regression modeling, particularly valuable for high-frequency trading applications where we need to measure continuous predictions such as, in our case, transaction returns and durations. This metric compares the model's residual sum of squares with the residual sum of squares of a naive predictor, typically a constant baseline prediction like the in-sample average, in our case. A positive out-of-sample R^2 indicates that the model's predictions are better than random or constant predictions, while $R^2=1$ means that the target can be perfectly predicted. R^2 can also take negative values meaning that it fails to capture any predictability and performs poorer than the baseline model. It is important, here, to emphasize that the in-sample mean average often serves as a strong baseline predictor in financial data modeling. This is because financial returns and durations are typically noisy and exhibit mean-reverting properties, meaning that the in-sample mean captures much of the inherent structure and central tendency in the data. As a result, achieving better predictions than the in-sample mean is inherently challenging, especially in volatile and unpredictable environments like financial markets.

The mathematical formulation of out-of-sample R^2 is defined as:

$$R^2(Y, \hat{Y}) = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n \left(Y_i - \frac{1}{n} \sum_{i=1}^n Y_i \right)^2} \quad (3)$$

where $Y = \{Y_1, Y_2, \dots, Y_n\}$ denotes the actual target values, and $\hat{Y} = \{\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_n\}$ represents the predicted values from the model.

While this metric is useful for understanding the overall magnitude of prediction errors in continuous variables, it may have limitations in financial data due to the presence of outliers. As mentioned in the beginning, prediction errors in financial models often follow heavy-tailed distributions, as stock prices can jump unpredictably, creating substantial outliers in returns. Out-of-sample R^2 aggregates the squared errors for testing data with equal weight, making it sensitive to large deviations caused by extreme prediction errors. Furthermore, trading volume varies significantly over time, with disproportionately large orders appearing sporadically. This variability in transaction size and frequency can also result in outliers in duration predictions.

These considerations lead us to adopt a more robust set of metrics for evaluating directional predictions that are less sensitive to outliers and provide a clearer indication of model performance for directional tasks. Sign accuracy focuses on whether the predicted and actual values share the same sign, emphasizing directional consistency rather than precise magnitude. Specifically, it measures the proportion of instances in which the predicted and actual returns share the same sign, and can be denoted as:

$$\text{Accuracy}(Y, \hat{Y}) = \frac{1}{n} \sum_{i=1}^n 1\{Y_i \cdot \hat{Y}_i > 0\} \quad (4)$$

Here, Y_i is the target, \hat{Y}_i is the prediction and $1\{Y_i \cdot \hat{Y}_i > 0\}$ is an indicator function that equals 1 if the predicted and actual values share the same sign, and 0 otherwise.

In financial data, however, it is also crucial to understand how often our model mislabels the direction of predictions and what kind of mislabeling occur mostly. These kind of information directly impact trading strategies, by providing accurate insights into the rate of correct and incorrect predictions. This way traders can implement strategies like shorting or going long more effectively, adjusting for potential errors in the model's directional output.

The Receiver Operating Characteristic (ROC) curve is mostly used for binary classification tasks, such as predicting price direction, in this case, making it an essential tool, especially in the machine learning age. Unlike the accuracy metric which examines only the directional alignment between the prediction and the actual outcomes, the ROC curve provides a graphical and threshold-independent assessment of a classifier's performance. It plots the True Positive Rate (TPR) ¹⁴ against the False Positive Rate (FPR) ¹⁵ at various threshold settings, capturing the trade-off between sensitivity (correctly identifying upward trends) and specificity (avoiding incorrect upward predictions). This balance is crucial for high-frequency trading strategies, where misclassifications can impact profitability due to transaction costs or losses on false signals.

The ROC curve allows traders to fine-tune their decision thresholds according to market conditions, helping them decide when to short or go long with greater confidence. A high TPR implies that the model captures true upward trends accurately, while a low FPR indicates that the model avoids costly false upward predictions. The Area Under the Curve (AUC) of the ROC provides a single value summary that represents the classifier's ability to distinguish between classes, with a higher AUC indicating better discrimination. An AUC close to 1.0 suggests excellent classification performance, whereas an AUC of 0.5 implies random guessing. In this context, the ROC curve and AUC are especially useful as they allow for the evaluation of classifiers under different market scenarios without relying on a fixed decision threshold.

¹⁴TPR is defined as $\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$, where TP (True Positive) refers to a prediction that is labeled as an upward trend (positive sign) and the actual outcome is also upward. For example, if the model forecasts an increase in stock price, and the price indeed goes up, this is a TP. While FN (False Negative) occurs when the model predicts a downward trend (negative sign) but the actual outcome is upward. For instance, if the model predicts a decline in stock price, but the price actually rises, this is an FN.

¹⁵FPR is defined as $\frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$, where FP (False Positive) refers to a prediction that is labeled as an upward trend (positive sign) and the actual outcome is a downward. For example, if the model forecasts an increase in stock price, but the price actually goes down, this is a FP. While TN (True Negative) occurs when the model predicts a downward trend (negative sign) and the actual outcome is downward. For instance, if the model predicts a decline in stock price, and the price actually decreases, this is an TN.

3.4 Algorithm Tuning and Testing

Each model we employ has a large number of parameters, which are tuned and tested on a rolling window basis. A new model is fitted for every testing day using data from the past 3 days, ensuring that the most recent information is included. Hyperparameters for each model are also tuned every 15 days to keep them up to date and adapt to any structural changes in the data.

3.4.1 Tuning Hyperparameters

To maintain computational efficiency, we focus only on tuning the hyperparameters that have the most significant impact on model performance using a fixed range of values. The tuning process is conducted iteratively and independently for each model.

For LASSO, we tune the l_1 -penalty term λ , which controls the strength of the regularization. We search over a range of values, typically from 10^6 to 10^{-8} , to find the optimal value that minimizes the model's error on a validation set.

For XGBoost, we primarily tune the learning rate (λ), which is one of the most crucial hyperparameters as it controls the step size during gradient descent updates. A smaller learning rate ensures that the model explores the parameter space more thoroughly, capturing subtle dependencies in the data, while a larger learning rate speeds up convergence but risks missing finer patterns. To strike a balance between these trade-offs, we test values of λ from 0.1, 0.01, and 0.001, which provide a wide range of dependencies for exploration. In addition to the learning rate, we also tune the number of estimators (i.e., trees) to values of 10, 50 and 100 and the maximum depth of each to 2, 5, and 10 to explore varying levels of tree complexity. These choices ensure that XGBoost is effectively tuned to balance computational efficiency and predictive performance. For Random Forests, we tune by using the same range for depth and total number of trees. The trees are trained to minimize the total mean squared error (MSE) on the training set for the transaction return and duration variable, and maximize the classification accuracy for direction.¹⁶

For LSTM networks, tuning involves a distinct set of hyperparameters due to their recurrent nature. To ensure computational efficiency while maintaining model performance, we focus on tuning three critical hyperparameters: the number of units (neurons), the learning rate, and the number of hidden layers. Each of these hyperparameters directly impacts the model's ability to capture patterns in the data and its computational cost. The number of units (neurons) is tuned to a fixed set of values consisting of 20, 50, and 100. This hyperparameter controls the model's capacity to learn complex patterns in the sequential data. This range allows us to balance model expressiveness and generalization. The learning rate is tuned to values of 0.1, 0.01, and 0.001, using the same logic we mentioned in XGBoost, while the number of hidden layers is tuned to 2, 3, and 5. By tuning these hyperparameters, we aim to optimize the performance of the LSTM network while maintaining computational efficiency, ensuring the model effectively captures the temporal dependencies in the data.

¹⁶For the Random Forest model, we set the random state to 42, a commonly used convention in programming. This value is frequently chosen as a default seed to ensure reproducibility of results, allowing for consistent model performance across different runs while maintaining randomness in the algorithm's decision processes.

We have also experimented with other choices of hyperparameters across all models, such as tuning a different range of numbers for the layers and learning rates in LSTM, varying λ more widely in LASSO, and exploring deeper trees in Random Forests. The results we obtain from this comprehensive tuning process are quite robust across all models.

3.4.2 Tuning, Training, and Testing Windows

The rolling window structure ensures that models are trained and tested using the most up-to-date data. For each testing day, models are retrained using data from the most recent 3 trading days, and hyperparameters are re-tuned every 15 testing days to maintain optimal performance.

The experiments are conducted using a two-layer rolling window structure. The first layer is the training window, set to a length of 3 trading days. The broader 30-day rolling window includes both tuning and testing phases. Specifically, the first 15 trading days are dedicated to tuning hyperparameters, while the subsequent 15 days are reserved for testing the models. An example of such window is shown in Figure 1

For a given time T , a 30-day rolling window is applied as follows:

1. **Learning:** For each combination of hyperparameters, and at $t = T, T + 3, \dots, T + 12$, the model is trained using data from the previous 3 trading days $(t - 2, t - 1, t)$. The model is then evaluated on the next trading day $t + 4$, and the out-of-sample R^2 is calculated for that day. The out-of-sample testing errors are accumulated over the last 15 days of the tuning window (as shown in orange in the diagram), giving $R^2_{t+4}, \dots, R^2_{t+15}$.
2. **Tuning:** The hyperparameters that produce the highest average out-of-sample R^2 across all testing days in the tuning window are selected. The average R^2 is computed as:

$$\frac{1}{15} \sum_{t=T+3}^{T+14} R_t^2$$

These optimized hyperparameters are then used for the prediction phase.

3. **Predicting:** For each day $t = T + 15, \dots, T + 29$, the model is retrained using the most recent 3 trading days and is used to predict the target value for day t . Predictions are made in a rolling window manner, as indicated by the green line in the diagram, and each prediction result is saved.
4. **Rolling Forward:** After the 30-day window is completed, the entire window is rolled forward by 15 days (i.e., $T \rightarrow T + 15$), and steps 1 to 4 are repeated for the new window of trading days.

This rolling window procedure ensures that the models are always up-to-date with the latest data and allows for regular hyperparameter tuning, which in turn ensures robust and timely predictions.

4 Predictability Results

In this section, we present a comprehensive analysis of our findings for all five NASDAQ 100 stock constituents over the period from October 3 to December 31, 2023. The objective is to assess the extent to which short-term predictability exists across all three response variables—transaction return, trade direction, and trade duration—while identifying the most significant predictive features through LASSO variable selection. As outlined in Section 2, a total of 13 predictor variables are constructed across seven distinct time horizons, incorporating interaction terms, yielding an overall set of 91 features. This structured approach facilitates the identification of the key factors governing short-term price movements, order flow dynamics, and market liquidity.

A central aspect of this investigation involves examining predictability across different time measurement frameworks. Specifically, we forecast future stock returns under the three distinct clock modes as follows: for calendar-based approach, predictions extend 5 seconds ahead; for transaction-based approach, we forecast the subsequent 10 executed trades; while for volume-based framework, the target is set at 2000 total shares traded ahead. These horizons were selected as the most robust and representative configurations, aiming to balance comparability across clock modes while maintaining sensitivity to distinct market microstructure characteristics. However, this selection posed certain challenges, as the average time for x amount of trades varied considerably across stocks, reflecting differences in trading frequency, liquidity conditions, and order flow intensity. Similarly, the time required to accumulate y amount of total shares varied significantly between high-volume and low-volume stocks, further complicating direct comparability across time measurement modes. Despite these limitations, the chosen horizons provide a well-calibrated benchmark for evaluating predictability across heterogeneous market environments.

As mentioned in Section 3, the three representative of each category chosen machine learning models, are LASSO regression, gradient-boosted trees (XGB), and long short-term memory networks (LSTM). To evaluate the relative effectiveness of each model, we conduct a comparative performance analysis through boxplot visualizations, systematically assessing how well each approach captures short-term predictability across different clock modes and forecasting horizons. This comparative assessment provides deeper insights into the strengths and limitations of each methodology in handling the inherent complexity of the data.

Beyond assessing raw predictive performance, we further evaluate the stability and consistency of model outputs over time, investigating whether forecast accuracy is persistent across the sample period or fluctuates in response to minor shifting market conditions.

4.1 Transaction Return Predictability

The predictability of transaction returns across all selected stocks is assessed through the out-of-sample R^2 performance, averaged over the 30-day testing period, with results presented in Figure 2. This figure illustrates the distribution of prediction performance across various forecasting models and time horizons, expressed in the form of boxplots. As outlined in Equation (3.3), an R^2 value of zero establishes the benchmark at which model-based predictions fail to surpass the simple in-

sample average return over the corresponding short horizon. The results unequivocally indicate that none of the employed models demonstrate a substantial ability to outperform this benchmark, underscoring the formidable challenge inherent in forecasting short-term financial returns. Although the mean out-of-sample R^2 remains marginally positive across all configurations, the median values consistently hover around zero, reflecting the inherent unpredictability of transaction returns. This phenomenon can be largely attributed to the profound stochasticity that characterizes high-frequency financial markets, where meaningful patterns are exceedingly difficult to discern due to the predominance of noise, the rapid evolution of market microstructure, and the recurrent presence of extreme outliers. The minimal yet observable predictability that emerges in the transaction clock mode suggests that aligning forecasting models with event-driven time frameworks may provide slight informational advantages.

To ascertain the key determinants of this limited predictability, LASSO variable selection is employed, evaluating the importance of individual predictors through two complementary metrics: the frequency with which each variable is selected across stocks and days, and the magnitude of the corresponding regression coefficients, which have been standardized to ensure comparability. Figure 3 reveals that the most influential predictors are those related to market imbalances, particularly limit order book imbalance (LobImbalance) and transaction imbalance (TxnImbalance). Both variables exert a predominantly negative impact, suggesting that the presence of strong directional imbalances in the order flow often induces short-term mean reversion, thereby attenuating the persistence of price movements and diminishing the scope for predictive modeling. In contrast, other predictors exhibit significantly lower coefficient magnitudes, indicating their comparatively minor role in determining short-term return fluctuations. Furthermore, a critical observation is that the most informative predictors are consistently derived from the most recent time windows, reinforcing the notion that any exploitable return predictability is fleeting and confined to the most immediate past. This is evident from the dominance of predictors constructed over horizons shorter than 0.2 seconds, highlighting the rapid dissipation of informative signals in high-frequency environments.

The stability of predictor selection across different time measurement frameworks is further examined in Figure 4, which reports the frequency with which each variable is retained by the LASSO model under transaction, calendar, and volume clocks. The findings reaffirm the primacy of order imbalance metrics, followed by past returns and certain microstructure indicators such as lambda, whereas volume-based variables—including total volume, average trade size, and maximum trade size—consistently rank among the least selected features. This suggests that transaction volume alone provides negligible incremental value in short-term return forecasting, likely due to its weak association with directional price movements over ultra-short horizons. Additionally, predictor selection is markedly more stable under the transaction clock, where LASSO identifies relevant features with greater consistency, whereas in the volume-based framework, model selection appears highly erratic, frequently shrinking all coefficients toward zero. This aligns with the previously observed lack of predictive power in the volume clock mode, further corroborating the hypothesis that volume-based segmentation fails to encapsulate the underlying structure of price evolution in an informative manner. Overall, these results underscore the inherent limitations

in forecasting transaction returns, emphasizing that while microstructural features such as order imbalance offer the most significant, albeit weak, predictive signals, these effects remain highly ephemeral and difficult to systematically exploit.

4.2 Price Direction Predictability

The ability to accurately forecast the direction of future price movements is a crucial aspect of financial market predictability. In this analysis, we examine whether it is possible to anticipate the next price movement—either upward or downward—across all clock modes. Since the target variable is binary in nature, the benchmark for a purely random outcome, equivalent to a coin toss, is 50%. The results, as depicted in Figure 5, demonstrate that all models achieve an average median accuracy consistently above 60% across all forecasting horizons, significantly exceeding the benchmark, while Figure 6 illustrates the Receiver Operating Characteristic (ROC) curve, which, in our case, was constructed using a slightly different approach than the conventional method. Since our response variable is inherently binary, we did not generate the curve by varying classification thresholds. Instead, for each day in our sample, we directly computed the True Positive Rate (TPR) and False Positive Rate (FPR) and plotted these points to form the ROC curve. Given the limited number of observations, the range of available TPR-FPR values is not sufficient to fully capture a smooth ROC representation, so we performed linear interpolation¹⁷. While this approach does not precisely replicate a traditional ROC curve based on threshold variation, it still offers valuable insights into the models' classification performance in a more general perspective. These findings suggest that short-term price movements exhibit systematic patterns that can be exploited by predictive algorithms.

Although no single model emerges as a dominant performer across all clock modes, a notable difference arises in the distribution of accuracy metrics across different horizons. When predicting price direction over a 5-second horizon, there is a substantial increase in interquartile range, indicating that model performance varies considerably across different stocks and time periods. This suggests that while predictive signals are present in very short time frames, their strength and stability may fluctuate depending on underlying market conditions, such as volatility and liquidity. Conversely, when forecasting the direction of the next 10 executed trades, the spread of prediction accuracy values narrows significantly, demonstrating a greater degree of stability in model outputs. Despite these variations, the findings remain broadly consistent across all clock modes, reinforcing the idea that short-term price direction is significantly predictable using the selected models.

To further explore the determinants of this predictability, we analyze the variable selection process employed by LASSO, presented in Figure 7. The findings indicate that transaction imbalance emerges as the overwhelmingly dominant predictor, with 8 out of the 10 most important variables derived from transaction imbalance measures at various horizons. Notably, these imbalances exert mostly a strong negative effect, aligning with the previously observed results in transaction return predictability. This consistent pattern suggests that large transaction imbalances introduce price

¹⁷This methodology connects the curve from the point (0,0) to the lowest observed (FPR, TPR) and extends it from the highest observed (FPR, TPR) to (1,1), indicating a linear trend in the unobserved thresholds

distortions that subsequently reverse, thereby allowing directional predictability. The prevalence of transaction imbalance as a predictive factor underscores the role of order flow dynamics in shaping short-term price movements, as shifts in liquidity demand and supply appear to systematically influence trade direction. The next most influential predictors are immediacy and LOB imbalance.

The stability of predictor selection across time measurement frameworks is depicted in Figure 8. In contrast to the findings from transaction return predictability, where certain groups of predictors were distinctly favored, the results for directional accuracy reveal a more balanced selection process. Specifically, for both calendar and transaction clocks, the frequency of predictor selection is evenly distributed across multiple predictor groups, with transaction imbalance standing out as the most frequently chosen feature. However, the volume clock mode continues to exhibit low selection stability, with only transaction imbalance and limit order book imbalance showing consistent importance. This aligns with the earlier finding that volume-based segmentation struggles to extract meaningful predictive signals.

Overall, the results indicate that short-term price direction is highly predictable across all tested models and clock modes, with accuracy levels well above the theoretical 50% benchmark. The strong predictive signals observed in this study suggest that such patterns can, in principle, be translated into profitable trading strategies. Given the ultra-short time horizons over which these predictions apply, a high-frequency trader could theoretically execute thousands of trades per day across multiple stocks, achieving a consistent success rate of approximately 60%. Even after factoring in transaction costs, and maybe even inventory expenses, (we will further investigate this in Section 6.2) such predictive advantages could yield substantial cumulative gains.

4.3 Trade Duration Predictability

We continue now with duration, specifically targetting the predictability of the amount of time, necessary for a certain number of transactions to take place, or a certain volume to be traded. The methods employed for forecasting trade duration mirror those used in the prediction of returns and price direction and the out-of-sample R^2 performance is presented in Figure 9.

The results indicate a substantial degree of predictability, with a median out-of-sample R^2 of approximately 40% across both transaction-based and volume-based forecasting horizons. This level of accuracy is notably higher than that observed for transaction return, suggesting that trade duration exhibits a more stable and structured temporal dependency. Unlike in previous tasks, however, model performance seems to vary a bit across models, this time. Notably, the LSTM model underperforms significantly in the volume-based horizon, achieving a median out-of-sample R^2 just below 30%. By contrast, both the LASSO and XGBoost models yield significantly stronger results, with median out-of-sample R^2 values approaching 50%, in the same clock mode.

Further insights are gained by examining the most influential predictive features selected through LASSO, as depicted in Figure 10. A striking observation is that all of the 20 most informative predictors exhibit strongly positive coefficient magnitudes. This result contrasts sharply with the findings from return and direction predictability, where imbalance-related variables played a dominant role, often with negative coefficients. The persistent positive coefficients observed in trade duration

forecasting indicate that past values of trade duration are strongly associated with future durations, suggesting a form of temporal autocorrelation in market activity. This implies that once trading slows down, it tends to remain slow, whereas when trading accelerates, it sustains momentum over short intervals.

The highest-ranked predictors include prior durations measured over different lookback windows, reinforcing the idea that past execution speeds serve as the primary determinant of future execution speeds. Also, the significance of immediacy, defined as the number of trades occurring within a recent interval, further supports this notion, as higher trade frequency in the past strongly correlates with shorter durations in the near future. Unlike in previous predictability tasks, transaction imbalances play a far less prominent role, indicating that duration forecasting is largely an endogenous process governed by past trade execution patterns rather than external liquidity shifts.

Overall, the results demonstrate that trade duration is highly predictable across all tested models and horizons, with past observed durations serving as highly reliable indicators of future trade timing. The high out-of-sample R^2 values achieved underscore the structured nature of trade duration dependencies, making it a very stable and exploitable feature in market microstructure analysis. This finding carries practical implications for algorithmic trading, as execution strategies that incorporate historical trade duration patterns can optimize order placement timing with a high degree of precision.

4.4 Performance Consistency Over Time

A key question arising from the previous analyses is whether the observed predictive performance remains stable over time or if the reported average out-of-sample R^2 values result from a few exceptional days compensating for periods of lower predictive accuracy. Similarly, it is important to determine whether predictability is primarily driven by a subset of stocks, overshadowing the weaker performance of others, or if it is a consistent phenomenon across all assets. To address these concerns, we present the time series of daily out-of-sample R^2 values for all three response variables and all stocks, depicted in Figures 11, 12, and 13. The shaded areas represent the tuning period, during which model parameters were optimized, while the unshaded regions correspond to the testing phase, where predictions were evaluated on unseen data. The more stable performance observed in the tuning period is expected, as models are calibrated to fit the specific market conditions of that segment.

Figure 11 illustrates the temporal evolution of return predictability. The results reveal substantial variation, particularly during the testing phase, with several instances where model performance falls below the benchmark (in-sample mean). This highlights the inherent instability and inconsistency associated with return predictability, reinforcing the notion that achieving robust transaction return forecasting remains a significant challenge. The high sensitivity of return prediction to tuning parameters further suggests that optimal performance is highly dependent on calibration choices, making it difficult to generalize across different market conditions. From a stock-specific perspective, considerable variations are observed in the performance of different assets. Notably, Netflix and Pepsi exhibit the highest fluctuations, whereas Ansys and AstraZeneca

display relatively stable performance, with AstraZeneca consistently achieving the highest predictive accuracy among all stocks. Further insights into stock characteristics and their correlation to predictability will be examined in section 5. However, due to the generally low R^2 values obtained, the reliability of these conclusions remains somewhat limited.

In contrast, Figure 12 demonstrates that the predictability of trade direction remains more stable throughout the entire period, both in the tuning and testing phases. While some variations exist, the overall performance across stocks appears more robust compared to transaction return forecasting. Ansys exhibits the most pronounced fluctuations but also consistently outperforms the other stocks across the sample period.

Figure 13 presents the time series of daily R^2 values for trade duration predictability. Once again, a relatively stable predictive performance is observed across all stocks and days, with only a few instances of negative R^2 values, likely attributable to daily outliers or temporary poor model performance. Occasional downward spikes are present, but these are sporadic and do not indicate systematic model failure. The overall robustness of duration predictability suggests that market timing, as measured through trade duration, follows a more deterministic pattern compared to return fluctuations. Additionally, no single stock appears to dominate the overall results, reinforcing the conclusion that predictability is distributed relatively evenly across assets.

Taken together, these findings indicate that while return predictability remains highly volatile and sensitive to tuning parameters, trade direction and trade duration demonstrate more stable predictive power. The persistence of directional and duration forecasting accuracy highlights the feasibility of leveraging such insights in algorithmic trading strategies, whereas transaction return prediction remains a more elusive and uncertain task.

5 Asset Characteristics and Predictability

In this section, we examine how the predictability achieved, primarily for trade direction and trade duration, varies across different stocks and clock modes. The objective is to identify key asset characteristics that drive these variations and to determine whether specific stock attributes can systematically explain differences in predictive performance. By analyzing the relationship between asset features and forecasting accuracy, we aim to provide deeper insights into the structural factors that contribute to the observed patterns of predictability, while also providing some further inspections of the performance across different clock modes.

Although the sample consists of only five stocks, the substantial variability among them in terms of liquidity, trading intensity, nominal share price, and core sector focus allows us to draw cautious yet meaningful inferences that extend beyond the specific assets analyzed. The diversity within the sample provides a basis for making a limited but safe projection regarding broader market behavior, offering insights into how different stock characteristics influence short-term predictability. The findings in this section will thus contribute to a more granular understanding of how market microstructure elements interact with short-term forecasting models, ultimately enhancing the interpretability of predictability results across assets.

5.1 Liquidity and Volume Trading Intensity

In this section we investigate the effect of stock liquidity on individual predictability and whether trading intensity in terms of volume exhibits substantial patterns. Liquidity is quantified using the percentage spread, computed as the average bid-ask spread divided by the mid-price, sampled at 15-second intervals throughout the trading day. Lower values of this metric indicate higher liquidity. The mean daily spread is calculated for each stock and subsequently averaged over all days in the sample, yielding a single liquidity measure per asset. To assess trading intensity based on volume, we use the total shares traded per second rather than the number of shares per transaction, as the former provides a more stable and robust measure that accounts for varying execution sizes and market conditions.

Figure 14 illustrates the relationship between return predictability and liquidity. The results suggest a slight negative U-shaped relationship, where stocks with moderate liquidity levels tend to exhibit marginally higher predictability compared to those with relatively extreme liquidity values. A similar pattern is observed for trading intensity, where neither the highest nor the lowest trading intensity stocks achieve the best predictive performance.

Turning to trade direction accuracy, Figure 15 provides a clearer pattern in the relationship between predictability and liquidity. In this case, the negative correlation is more pronounced, with Tesla demonstrating the lowest accuracy, just above 60%, while Ansys exhibits the highest accuracy, approaching 70% (keep in mind that the smaller values of spread indicate higher liquidity). A similar trend is observed with trading intensity, reinforcing the notion that lower liquidity and lower trading intensity correspond to enhanced trade direction predictability. What stands out most prominently in these two last figures is the almost identical way in which these two market characteristics influence predictability.

The relationship between trade duration predictability and liquidity is depicted in Figure 16. Unlike the previous metrics, this analysis reveals a nuanced positive correlation, where stocks with higher liquidity exhibit a tendency for more predictable trade duration. This suggests that in more liquid markets, the timing of trade execution follows more structured and systematic patterns, potentially due to the regularity of order arrivals and execution speeds. More importantly, a strong negative correlation emerges between predictability in transaction return and predictability in trade duration, indicating that stocks that perform well in terms of return tend to exhibit lower levels of trade duration predictability. This is particularly evident in the case of AstraZeneca, which achieves, significantly, the highest predictability in return, while also scores quite well in trade direction accuracy, but ranks significantly lower in trade duration forecasting. This finding suggests a fundamental trade-off between the ability to predict price movements and the ability to anticipate the timing of executions, which may be attributed to differences in the underlying drivers of these two dimensions of market activity.

These results collectively demonstrate that liquidity and trading intensity exert a significant influence on short-term predictability, with a clear negative correlation observed between liquidity and both return and trade direction predictability, whereas a positive correlation emerges between liquidity and trade duration predictability. Moreover, trading intensity and liquidity appear to influ-

ence predictability in a remarkably similar manner across all three response variables, suggesting that these two market characteristics are inherently linked in shaping short-term forecastability.

5.2 Nominal Share Price and Transaction Trading Intensity

We proceed by examining the relationship between nominal share price, transaction-based trading intensity, and predictability. The nominal share price is computed as the average stock price over the sample period, while transaction trading intensity is quantified as the total number of transactions per day. Unlike the previous analysis, where trading intensity was measured based on volume, we extend the analysis to transaction-based intensity to provide a more comprehensive assessment of trading activity. We opt not to use transactions per second, as the extreme discrepancy between high-frequency stocks such as Tesla and lower-liquidity stocks such as Ansys and AstraZeneca would have led to a poorly distributed allocation along the x-axis.

Figure 17 presents the relationship between return predictability and nominal share price. Same as liquidity, the results reveal a negative correlation in a U-shaped form, where stocks with mid-range nominal prices tend to exhibit slightly lower return predictability than those with extremely high or low prices. However, in contrast to volume-based trading intensity, transaction-based intensity does not exhibit any significant pattern, suggesting that the mere frequency of trades per day does not systematically influence return predictability.

Turning to trade direction accuracy, Figure 18 demonstrates a nuanced negative correlation between nominal share price and trade direction predictability. This negative relationship, while less pronounced than that observed with liquidity, suggests that stocks with lower nominal prices tend to exhibit slightly better trade direction predictability. More notably, a stronger negative correlation is observed between transaction-based trading intensity and trade direction predictability, implying that stocks with a high number of daily transactions tend to display less predictable directional movements.

Figure 19 presents the relationship between trade duration predictability and nominal share price. Here, a clear positive correlation emerges, more pronounced than that observed with liquidity. This suggests that as nominal share price increases, trade duration predictability also improves, potentially due to higher-priced stocks being less susceptible to rapid, erratic order flow shifts and short-term liquidity fluctuations. This result indicates that execution timing in higher-priced stocks may follow more stable patterns, allowing for more structured duration forecasting.

Overall, the findings indicate that liquidity and nominal share price exert similar influences on predictability across all response variables, albeit with varying degrees of significance. Some relationships, such as the positive correlation between nominal share price and trade duration predictability, appear stronger than their liquidity-based counterparts. Additionally, while volume-based trading intensity showed notable patterns, transaction-based intensity does not display any significant correlation with predictability, underlining the importance of considering more than one metric in order to derive more robust results.

5.3 Volatility

It is natural to expect that cross-sectional differences in volatility influence predictability. Stocks that exhibit higher volatility tend to experience more rapid fluctuations in trading conditions, making it more challenging for predictive models trained on past data to maintain accuracy in an out-of-sample setting. If price movements change erratically, systematic patterns become harder to detect, reducing the effectiveness of forecasting methods. For each stock and each day, volatility is computed as the standard deviation of mid-price returns, measured at 30-second intervals. This measure captures the degree of short-term price variation, providing a robust metric for evaluating the impact of volatility on predictability.

Figures 20 and 21 confirm the patterns observed in previous sections. Return predictability exhibits a negative, closer to U-shaped correlation with volatility, with less volatile stocks performing better, but not in a linear way, since the relationship slightly tends to inverse for more volatile stocks. This result aligns with the general idea that excessive volatility introduces randomness, making return forecasting more challenging.

Conversely, trade duration predictability exhibits a clear positive correlation with volatility, indicating that stocks with higher volatility tend to have more structured and predictable execution timing. This result suggests that increased volatility leads to more systematic order execution patterns, potentially due to the interaction between high-frequency liquidity demand and supply. When market activity is more intense, execution speeds and trade durations tend to follow more regular patterns, making them easier to anticipate.

Overall, these findings reinforce the observation that liquidity, nominal share price, and volatility all impact predictability in a quite similar manner, demonstrating a consistent relationship across different forecasting tasks. Each of these market characteristics exerts comparable effects on short-term forecastability, underscoring the importance of considering multiple factors simultaneously when developing predictive models.

5.4 Performance Across Stocks and Clock Modes

To gain a deeper understanding of how predictability varies across different market conditions, we now examine the performance of individual stocks across different clock modes. By analyzing predictability across transaction-based, volume-based, and calendar-based time frameworks, we aim to identify patterns in forecastability that are driven by the underlying trading characteristics of each stock. These results allow us to assess whether specific assets consistently exhibit higher predictability in certain trading conditions and whether observed differences in predictability across clock modes are driven by systematic factors or merely a result of outliers.

Figure 22 presents the performance of return predictability across all stocks and clock modes. The results indicate quite robust performance across individual stocks, with only a few significant deviations. The most noteworthy observation is the strong upward outliers in AstraZeneca and Netflix under the transaction clock mode, which largely contribute to the overall higher predictability obtained under this framework. This suggests that the superior performance observed

in transaction-based forecasting is primarily driven by these two stocks, rather than being a generalizable pattern across all assets.

Turning to trade direction accuracy, Figure 23 confirms the previously observed stability in performance across different stocks and clock modes. While most assets maintain consistent accuracy levels, two notable exceptions emerge: AstraZeneca and Ansys under the calendar clock mode, where Ansys achieves an exceptionally high accuracy, exceeding 80%. An interesting pattern is that stocks with lower trading intensity, such as Ansys, AstraZeneca, and Pepsi, tend to score higher accuracy when predicting fixed forward intervals in calendar time. However, this observation should be interpreted with caution, as other confounding factors may influence this relationship.

Finally, Figure 24 examines the predictability of trade duration across different clock modes. The results indicate that stocks with lower trading intensity (Ansys, AstraZeneca, and Pepsi) exhibit significantly higher predictability when forecasting duration in volume-based horizons. This finding suggests that in less frequently traded stocks, execution times exhibit more structured patterns when volume-based intervals are considered, potentially due to more regular liquidity provision and lower sensitivity to high-frequency order imbalances.

Overall, these results suggest that while predictability remains highly robust across different stocks and clock modes, certain assets still exhibit substantial deviations that could drive aggregate differences in forecastability. The transaction clock mode's higher predictability appears to be driven mainly by specific outliers rather than a universal advantage, while calendar-based accuracy tends to favor stocks with lower trading intensity. These insights reinforce the importance of considering both market structure and stock-specific attributes when evaluating overall performance.

6 The Value of a Millisecond

The landscape of high-frequency trading is characterized by intense competition, where firms go to extreme lengths to minimize latency in their interactions with stock exchanges. This includes the physical colocation of servers near exchange infrastructure, the deployment of dedicated ultra-low latency transmission technologies, and the use of direct, optimized communication channels between major financial centers. The primary motivation behind these efforts is to ensure rapid transmission of orders and cancellations, but an equally fundamental requirement is the acquisition of real-time market data. Without access to the most up-to-date information, predictive models might lose their effectiveness, and trading strategies become indistinguishable from random speculation.

In this section, we explore the critical role of data timeliness in market predictability. Our objective is to quantify how predictive power evolves over extremely short time intervals and to assess the extent to which predictive value deteriorates as data becomes even slightly outdated. Specifically, we address three key questions. First, how quickly does the substantial predictability identified in earlier sections dissipate? Second, how costly is a delay or lag in acquiring and processing data, in terms of lost forecasting accuracy? Third, is there any additional predictive

advantage to be gained from briefly peeking into the immediate future order flow, even if such foresight is imperfect?

Having established broader predictability patterns across various stocks in Section 5 and confirming the robustness of these results, we now shift our focus to a more granular analysis of high-frequency predictability. To that end, we concentrate on Tesla (TSLA), the most liquid and trading-intensive stock in our dataset, as the best representative of overall market conditions. This selection is driven by its exceptionally high trading frequency, deep order book liquidity, and significant presence among both retail and institutional investors. As one of the most actively traded stocks in the market, it serves as an ideal test case for examining short-term predictability under conditions of high-speed execution and continuous order flow dynamics.

6.1 Predictability Lifespan

In the preceding sections, we examined the extent to which returns, trade direction, and trade duration exhibit predictability. Under the premise of a competitive and efficient market, such predictability is expected to be highly transient. Given that the data utilized in this study is publicly available and widely accessible, any systematic patterns in predictability should be swiftly identified and arbitrated away by market participants employing similar datasets and modeling techniques. The empirical findings reinforce this expectation, although the degree to which predictability diminishes seems to vary across different response variables.

To systematically examine the temporal persistence of return predictability, we evaluate forecasts for $\text{Return}(T, \Delta, M)$ at different horizons. Specifically, we consider $\Delta = 1, 3, 5, 10$, and 30 seconds in the calendar clock, $\Delta = 5, 10, 20, 50$, and 100 transactions in the transaction clock, and $\Delta = 2K, 5K, 10K, 20K$, and 50K total traded volume in the volume clock. Forecasting horizons for trade duration and trade direction are set in a similar manner. Predictor variables remain consistent with previous experiments, and for computational efficiency, we employ the LASSO model, for the period of 15 to 23 of November 2023, where quite robust results were obtained as we saw in section 4.4, with hyperparameters separately reoptimized for each forecasting horizon.

Figure 25 presents the results for return predictability across the three different clock modes. The patterns observed across all time clocks exhibit a consistent trend, where predictability peaks at mid-term horizons before sharply declining. More precisely, return forecasts achieve their highest performance at horizons of approximately 10 seconds, 10 transactions, and 5K total shares traded. Beyond these points, predictive accuracy in both the volume and transaction clock modes diminishes rapidly. Notably, once the forecasting horizon surpasses 20K shares traded, predictability deteriorates entirely, as models begin to underperform relative to the benchmark.

A particularly striking result emerges in the calendar clock mode, where mid-term forecasting horizons demonstrate a significant improvement in predictive power. Specifically, at the 10-second horizon, out-of-sample R^2 reaches an impressive 14%, marking nearly a twentyfold increase in performance relative to shorter horizons. Furthermore, this enhanced predictability remains substantial over extended intervals, as even at the 30-second mark, the mean R^2 remains above 10%. This finding suggests that in calendar time, return predictability does not decay as abruptly as it

does in transaction- or volume-based time clocks, potentially indicating a more stable underlying price formation process over fixed time intervals.

Turning to directional accuracy, Figure 26 exhibits a similar diminishing pattern but with notable differences compared to return predictability. In this case, the sharpest decline occurs in the calendar clock mode, where accuracy reaches its peak at the 10-second mark before falling significantly at the 30-second horizon, ultimately reaching a mean accuracy of approximately 56%. In contrast, transaction and volume clocks demonstrate a more sustained level of predictability. While the highest accuracy is still achieved at mid-range horizons, predictability remains substantial for extended periods. Even at 100 transactions or 50K total shares traded, directional accuracy continues to exceed 60%, indicating that order execution patterns and liquidity-taking behaviors provide a more persistent informational edge compared to purely time-based intervals.

This result appears somewhat conflicting, as return predictability in fixed forward horizons is significantly higher as time progresses compared to event-based horizons, whereas directional accuracy declines more sharply. One possible explanation lies in the underlying distribution of returns across different time clocks. In event-based horizons, there may be a greater presence of extreme price movements and order flow irregularities, leading to a higher incidence of outliers. These outliers can introduce variability into the return predictability estimates, causing greater fluctuations in forecast accuracy as time passes. However, despite this increased variability, event-based horizons may still be better suited to capturing broader price trends, as they dynamically adjust to market conditions rather than relying on fixed forward-looking intervals. Conversely, calendar-based predictions, while achieving stronger return predictability at mid-range horizons, struggle to maintain consistent directional accuracy over time, likely due to the fixed nature of the forecast window failing to adapt to changing order flow conditions.

For duration predictability, Figure 27 reveals a similar diminishing pattern, though with substantial differences between clock modes. In the transaction clock, predictability decreases sharply after the 10-transaction horizon, but the rate of decline slows thereafter, suggesting that beyond the initial threshold, further increases in transaction span lead to a more gradual erosion of predictive power. In contrast, the volume clock exhibits a much smoother decline, with no sharp deterioration observed, indicating a more stable relationship between past and future duration values over volume-based horizons.

Overall, in each of the tests, predictability declines as the forecasting span increases, consistently reaching a peak at mid-range horizons before gradually diminishing. However, the rate and nature of this decline vary significantly across different response variables and time clocks, reinforcing the importance of assessing predictability not only across time horizons but also across different temporal frameworks.

6.2 Impact of Delay

In all previous analyses, predictability was evaluated under the assumption that the most recent market data was immediately available for processing and decision-making. However, in real-world trading environments, delays in data transmission, processing, and execution are inevitable.

The impact of such delays on forecastability is of particular importance, as even milliseconds can hold substantial value in high-frequency trading environments. Reports suggest that major trading firms invest vast resources in reducing communication latency between financial centers, such as the construction of dedicated microwave transmission networks to decrease the time required for data transmission between New York and Chicago by mere fractions of a millisecond. These investments underscore the significance of minimizing delays and, conversely, highlight the cost of any latency introduced in a trading strategy.

Data processing delays can arise from multiple sources. The transmission of the latest trade and quote updates from exchange servers to a trader’s system incurs a non-negligible time lag, after which the trading algorithm must process the new information, compute optimal trading decisions, and transmit orders to the market. The minimization of such computational delays requires significant infrastructure and technological investments. These forms of delay are best analyzed within the calendar time framework, where we evaluate the effect of latencies ranging from milliseconds to full seconds. Additionally, another form of delay arises when an order submitted to an exchange does not execute immediately but instead remains in the order book until sufficient liquidity is available for execution. Such execution delays are more naturally modeled in event-based time clocks, including both transaction and volume-based frameworks, where the delay is measured in terms of the number of trades or the total number of shares transacted before an order is filled.

To formally quantify the impact of delayed information, we introduce a time lag, denoted as δ , and examine the predictability decay when forecasting a delayed version of transaction return and direction, formulated as $Return(T + \delta, \Delta, M)$. All predictor variables remain computed at time T , but the target response is shifted forward by the delay parameter δ . The magnitude of this delay varies based on the time clock employed: in the calendar clock, delays are set to $\delta = 0s, 0.1ms, 1ms, 10ms, 0.1s$, and $1s$; in the transaction clock, the delay corresponds to $\delta = 0, 1, 2, 3, 5$, and 7 trades; in the volume clock, delays are defined as $\delta = 100, 300, 500, 1000$, and 2000 shares. The study focuses on return and trade direction predictions over the same stock, period and evaluation model as examined in Section 6.1, with forecasting horizons of $\Delta = 5$ seconds, 10 trades, or $5K$ total traded volume.

Figure 28 illustrates the effect of delay on return predictability across different time clocks. A significant decline in return forecastability is observed beyond 100 ms or 3 trade delays, with predictability largely vanishing when the delay reaches 1 second or 7 trades. This steep decline highlights the extreme sensitivity of return-based predictability to data freshness, further reinforcing the rationale behind the large investments in minimizing trading latencies.

Turning to trade direction accuracy, the impact of delay is less immediate but still pronounced. As shown in Figure 29, accuracy remains almost unaffected for delays up to 10 milliseconds or 100 shares traded, experiencing little to no decline. However, beyond this threshold, accuracy begins to deteriorate gradually, with a notable drop occurring at delays of 1 second or 2000 shares traded. Despite this decline, a degree of predictability remains, with directional accuracy still hovering around 55% even at the longest delay horizons. This suggests that while ultra-low-latency access to market data is crucial for return predictability, directional accuracy retains some robustness even in slightly delayed settings, likely due to underlying market structure patterns and execution

imbalances persisting over longer periods.

6.3 Peek into Future

The previous section demonstrated the significant cost of delays in data acquisition and processing on predictability. This naturally raises the question: what if, instead of being hindered by latency, a trader had access to advance signals regarding certain aspects of the order flow? Even if this information were imperfect and available only for a brief moment before execution, it could still provide a strategic advantage.

Such foresight into the direction of incoming orders can originate from various sources. A trader might leverage deeper insights from the limit order book, process data from related securities such as futures markets, or interact more efficiently with the market by reacting to quotes posted on other exchanges. Additionally, some traders may benefit from direct data feeds from exchanges, which process transactions faster than publicly available sources, thereby providing a minor yet potentially exploitable informational edge. While the realism of these possibilities remains a subject of debate, it is nonetheless insightful to evaluate whether such information, even if noisy, can enhance predictability.

To model this scenario, we introduce a new predictor alongside those already considered. This additional predictor serves as a noisy estimator of the future order flow direction, computed over a specified time horizon Δ . Formally, we define Dir_t^{LR} as the binary trading direction, determined using the [Lee and Ready \(1991\)](#) algorithm at time t . In Table 9, we provide an illustrative example demonstrating how this algorithm assigns directional labels to individual trades. Furthermore, let X be a Bernoulli-distributed random variable, with probability $P(X = 1) = p$, representing the likelihood that the predictor correctly reflects the actual order flow direction. Consequently, the probability of obtaining a misclassified signal is given by $(1 - p)$. Such a predictor can be considered an approximation of the advance trading signals available in certain optimal market-making strategies, such as those explored in [Aït-Sahalia and Sağlam \(2021\)](#).

The advance signal available at time T is thus a noisy representation of the average future trade direction, mathematically expressed as:

$$FlowDir(T, \Delta, M, p) = \text{sign}(2X - 1) \cdot \text{sign} \left(\sum_{t \in D^{txn}} n_{\text{Int forward}}(T, \Delta, M) Dir_t^{LR} \right). \quad (5)$$

In essence, this variable randomly flips the average trade direction with probability $(1 - p)$. When $p = 1$, the signal provides perfect foresight of future order flow direction, whereas at $p = 0.5$, it becomes entirely random, reducing to a pure noise variable. Notably, this predictor relies exclusively on the sign of future trade directions without incorporating price or volume information.

For this study, we examine the impact of peeking ahead using a forecasting span of $\Delta = 5$ seconds, under the calendar clock framework, aligning it with the return prediction horizons explored earlier. To allow the model to fully leverage this additional information, we also introduce interaction terms with every existing predictor, thereby doubling the total number of predictors from

91 to 182. The empirical evaluation, presented in subsequent sections, investigates the impact of different values of p on forecastability.

Figure 30 illustrates the relationship between the probability of correctly anticipating the future trading direction and the corresponding predictability of 5-second-ahead returns. The results indicate that incorporating a predictive signal regarding the sign of the future average transaction direction significantly enhances return forecastability. Specifically, when the probability of correctly estimating the future trade direction reaches a reasonably attainable level of 70%, the out-of-sample R^2 improves substantially, rising from a baseline of approximately 0.12% to over 4%. Moreover, as the probability approaches 90%, return predictability exhibits an exponential increase, suggesting that even marginal improvements in forecasting accuracy could yield disproportionately large gains in predictability. This finding underscores the substantial informational advantage conferred by the ability to anticipate incoming order flow with high precision, further highlighting its potential strategic value in high-frequency trading environments.

7 Robustness check

To conclude the analysis, we conduct a series of robustness checks to assess the sensitivity of our results to various modeling choices. Specifically, we evaluate the impact of algorithm selection by incorporating a broader range of predictive methods, including Random Forest, PCA regression (referred to as FarmPredict) and OLS. Additionally, we examine the effect of hyperparameter tuning on model performance to determine the extent to which optimization enhances predictive accuracy across different methodologies. In this final chapter we try to assess if our findings remain consistent and reliable under varying modeling configurations.

7.1 Model Comparison and Performance Across Algorithms

To assess the robustness of our predictive results, we conduct a comparative analysis of multiple forecasting models, extending beyond the primary methods used in previous sections. Given that directional accuracy and trade duration exhibited substantial levels of predictability, we include three additional modeling approaches: Ordinary Least Squares (OLS) Serving as a simple linear regression benchmark model. FarmPredict; A factor-based machine learning method that first decomposes observed data into latent factors and idiosyncratic components (Fan (2020)). In this case, the extracted factors are utilized as inputs in a statistical learning framework, while an Adaptive LASSO is applied to the idiosyncratic components to enhance predictive performance. Finally, Random Forest (RF); A non-parametric ensemble learning method that constructs multiple decision trees to improve accuracy and reduce variance.

Figure 31 presents the results for directional accuracy, where we observe a high degree of consistency across all models. Most approaches achieve comparable predictive accuracy, with the exception of OLS, which demonstrates a slight underperformance. Nevertheless, even OLS maintains a relatively strong mean accuracy of approximately 58%, highlighting the robustness of

the directional predictability findings.

In contrast, the results for trade duration reveal more pronounced differences in model performance. Figure 32 illustrates that Random Forest emerges as the most effective model, followed by LASSO and XGBoost, consistently outperforming other approaches. FarmPredict, while slightly lagging behind these three methods, still achieves an R^2 exceeding 25%, demonstrating a reasonable level of predictive power. OLS, however, exhibits significant underperformance in this setting, primarily due to issues of overfitting. The model's sensitivity to noise results in highly erratic predictions, reinforcing the challenges posed by linear regression when applied to complex, high-dimensional financial data.

7.2 Fine-Tuning of Model Hyperparameters

A critical component of predictive modeling lies in the optimization of hyperparameters, as they significantly influence model performance. In this subsection, we analyze the impact of key tuning parameters across the various machine learning models employed in this study.

XGB represents the base nonparametric method utilized throughout the study. Among its three primary hyperparameters—the number of trees, tree depth, and learning rate—the most influential was identified as the learning rate, while the other two produced only marginal variations in predictive accuracy. Notably, increasing the learning rate resulted in higher variance in return predictions, with smaller learning rates producing nearly constant predictions around the sample mean. Specifically, a learning rate of 0.1—the highest value within the selected range—initially, in tuning, achieved remarkably high in-sample R^2 for transaction return. However, when applied in the testing phase, it exhibited substantial errors, leading to poor generalization performance. This phenomenon likely stems from increased sensitivity to outliers. Furthermore, a weak positive correlation was observed between learning rate size and the number of trees, wherein higher learning rates tended to perform better when paired with a larger number of trees, possibly due to enhanced model capacity in capturing complex interactions. For directional accuracy, the learning rate exerted the strongest influence. As depicted in Figure 33, models with higher learning rates consistently outperformed their counterparts, underlining the importance of this hyperparameter in classification tasks.

FarmPredict, a factor-based predictive model, was primarily tuned by selecting the number of principal components (PCs) for regression. As anticipated, the total number of PCs did not substantially impact transaction return predictions, given the aggressive penalization introduced by adaptive LASSO. The penalization shrank most coefficient estimates to zero, highlighting the challenge of dealing with inherent noise and justifying the relatively weaker predictive performance for transaction returns. In contrast, for directional accuracy, a clearer pattern emerged. Prior to applying the adaptive LASSO penalization, an optimal range of 50 to 60 principal components (from a total of 91) was identified, which contributed to improved classification performance.

For the Random Forest model, hyperparameter tuning focused on the total number of trees and their depth. The latter exhibited no substantial influence on predictive accuracy, while the number of trees yielded moderate improvements. Increasing the number of trees from 10 to 20 led

to small gains, while further increases to 50 resulted in only marginal enhancements. A plateau effect was observed at approximately 100 trees, beyond which additional trees offered no tangible improvement and, in some cases, led to slight performance deterioration. Figure 34 illustrates the robustness of performance with respect to the number of trees for the duration prediction variable, further reinforcing this observation.

The LSTM architecture presented the most significant challenges in hyperparameter tuning. Unlike the other models, where a dominant parameter could be identified, LSTM performance was sensitive to multiple factors, including the number of layers, the number of units per layer, and the learning rate. Additionally, optimal hyperparameter configurations varied substantially across different training runs, making it difficult to establish a consistent pattern. While LSTM exhibited superior performance during the tuning phase, this advantage did not persist in the testing phase, where it consistently underperformed relative to the other models. Unlike XGBoost, where a learning rate of 0.1 yielded unstable results, LSTM models favored a lower learning rate of 0.001 when predicting directional accuracy.

Overall, transaction return was found to be particularly challenging both in terms of hyperparameter optimization and in achieving stable predictive performance across different models. Conversely, predictability in direction and duration produced quite robust results both in tuning and testing period. The number of trees as well as their depth did not influence performance significantly, whilst among all hyperparameters, the learning rate was identified as the most influential factor, particularly in complex ensemble learning methods, underscoring the necessity of careful tuning to optimize model generalization and performance.

8 Conclusion

We examined the predictability of three key variables of financial market microstructure over ultra-short high-frequency horizons, namely transaction returns, trade direction, and durations. The analysis was conducted using three different time clocks and a range of machine learning models, including LASSO, random forests, and neural networks, applied to transaction and quote data from five NASDAQ 100 stocks from October to December 2023.

Our results indicate that significant predictability is achieved for both trade durations and price direction, with the former reflecting strong temporal dependencies in trade arrival patterns and the latter demonstrating notable accuracy across various conditions. In contrast, while positive results are also observed for transaction returns, their predictability is considerably lower and less consistent. Key predictors driving short-term predictability include order book imbalances, recent transaction imbalances, and past trade returns.

Notably, predictability for trade durations improves under more liquid and volatile conditions, while the predictability of returns is more pronounced during periods of heightened market fluctuations. While all the examined characteristics—liquidity, nominal share price, volatility, and trading intensity—appear to be inversely associated with return predictability and directional accuracy, what stands out more distinctly is the trade-off between the predictability of trade durations

and that of returns and trade direction, suggesting that as durations become more predictable, the short-term predictability of returns and direction tends to decrease.

The pattern of predictability across all three variables exhibits a similar inverse U-shaped relationship over time, with performance peaking at intermediate horizons before declining as the prediction window extends. Notably, transaction return predictability diminishes rapidly beyond a certain threshold, becoming negligible at longer horizons. In contrast, while both trade direction and duration experience a decline in predictability over time, they maintain relatively higher levels of predictability for extended periods, suggesting a more enduring temporal structure in these variables compared to returns. The timeliness of data, also, plays a crucial role, as predictability declines sharply with even slight delays. Most predictability lies within the first few milliseconds or trades diminishing rapidly beyond a one-second delay or a few thousands lots traded. Simulating the advantage of anticipating order flow direction—akin to the capabilities of high-frequency traders—reveals significant improvements in return forecasts, highlighting the benefits of even limited foresight.

We also find that alternative machine learning methods, such as PCA-penalized regression (FarmPredict) and random forests, yield comparable results, whereas OLS fails to capture substantial predictability and consistently underperforms. This outcome underscores the importance of penalization techniques in mitigating overfitting, a key advantage of most machine learning algorithms. Among the examined tuning parameters, the learning rate emerges as the most influential factor in enhancing predictive performance for both XGB and LSTM. While increasing the number of layers and units in LSTM, as well as the number of trees in random forests and XGB, improves performance, their marginal benefits diminish beyond a certain threshold, indicating limited gains from excessive model complexity.

References

- Aït-Sahalia, Y. (2022). How and when are high-frequency stock returns predictable? *NBER Working Paper No. 30366*. <https://doi.org/10.3386/w30366>
- Aït-Sahalia, Y., & Sağlam, M. (2021). High frequency market making: The role of speed. *Journal of Econometrics*, 239, 105421. <https://doi.org/10.1016/j.jeconom.2023.105421>
- Alvim, L. G. M. (2010). Daily volume forecasting using high frequency predictors. *Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Applications*, 47–54. <https://doi.org/10.2316/P.2010.674-047>
- Ba, J. L., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*. <https://arxiv.org/abs/1607.06450>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Bühlmann, P., & van de Geer, S. (2011). *Statistics for high-dimensional data: Methods, theory and applications*. Springer. <https://doi.org/10.1007/978-3-642-20192-9>
- Chang, S.-H. (2021). Short-term stock price-trend prediction using meta-learning. *arXiv preprint arXiv:2105.13599*. <https://arxiv.org/abs/2105.13599>
- 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>
- Chen, Y. (2019). Optimized input for cnn-based hyperspectral image classification using spatial transformer network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12), 7179–7188. <https://doi.org/10.1109/TGRS.2019.2926774>
- Chinco, A. (2019). Sparse signals in the cross-section of returns. *The Journal of Finance*, 74(1), 449–492. <https://doi.org/10.1111/jofi.12733>
- Cont, R. (2010). A stochastic model for order book dynamics. *Operations Research*, 58(3), 549–563. <https://doi.org/10.1287/opre.1090.0755>
- Cont, R. (2014). The price impact of order book events. *Journal of Financial Econometrics*, 12(1), 47–88. <https://doi.org/10.1093/jjfinec/nbt003>
- Dixon, M. F. (2018). A high frequency trade execution model for supervised learning. *High Frequency*, 1(1), 3–19. <https://doi.org/10.1002/hf2.10016>
- Easley, D. (2021). 3950 lecture 26 [Accessed: 2025-02-23].
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2), 383–417. <https://doi.org/10.2307/2325486>
- Fan, J. (2020). *Statistical foundations of data science*. Chapman; Hall/CRC. <https://www.routledge.com/Statistical-Foundations-of-Data-Science/Fan-Li-Zhang-Zou/p/book/9781466510845>
- Hochreiter, S. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Huang, G. (2002). A generalized growing and pruning rnn for modeling complex systems. *IEEE Transactions on Neural Networks*, 13(3), 622–634. <https://doi.org/10.1109/TNN.2002.1000132>

- Huang, J. (2008). Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18(4), 1603–1618. <https://www.jstor.org/stable/24308572>
- Huang, R. D., & Stoll, H. R. (1994). Market microstructure and stock return predictions. *The Review of Financial Studies*, 7(1), 179–213. <https://doi.org/10.1093/rfs/7.1.179>
- Kercheval, A. N., & Zhang, Y. (2015). Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, 15(8), 1315–1329. <https://doi.org/10.1080/14697688.2015.1032546>
- Knoll, J. (2019). Exploiting social media with higher-order factorization machines: Statistical arbitrage on high-frequency data of the sp 500. *Quantitative Finance*, 19(4), 571–585. <https://doi.org/10.1080/14697688.2018.1521002>
- Kyriakou, I. (2021). Short-term exuberance and long-term stability: A simultaneous optimization of stock return predictions for short and long horizons. *Mathematics*, 9(6), 620. <https://doi.org/10.3390/math9060620>
- Lee, C. M. C., & Ready, M. J. (1991). Inferring trade direction from intraday data. *The Journal of Finance*, 46(2), 733–746. <https://doi.org/10.1111/j.1540-6261.1991.tb02683.x>
- Lewis, M. (2014). *Flash boys: A wall street revolt*. W. W. Norton & Company. <https://www.amazon.com/Flash-Boys-Wall-Street-Revolt/dp/0393351599>
- Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *The Journal of Portfolio Management*, 30(5), 15–29. <https://doi.org/10.3905/jpm.2004.442611>
- MacKinlay, A. C. (2002). *A non-random walk down wall street*. Princeton University Press. <https://press.princeton.edu/books/paperback/9780691092560/a-non-random-walk-down-wall-street>
- Malkiel, B. G. (1973). *A random walk down wall street*. W. W. Norton & Company.
- Mullainathan, S. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, 31(2), 87–106. <https://doi.org/10.1257/jep.31.2.87>
- Narajewski, M., & Ziel, F. (2022). Optimal bidding in hourly and quarter-hourly electricity price auctions: Trading large volumes of power with market impact and transaction costs. *Energy Economics*, 110, 105974. <https://doi.org/10.1016/j.eneco.2022.105974>
- Nguyen, S. (2019). A novel approach to short-term stock price movement prediction using transfer learning. *Applied Sciences*, 9(22), 4745. <https://doi.org/10.3390/app9224745>
- Ntakaris, A. (2018). Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37(8), 852–866. <https://doi.org/10.1002/for.2543>
- O’Doherty, M. S. (2022). The impact of analyst forecast errors on fundamental indexation. *Asian Business Management*, 23(5), 1–25. <https://doi.org/10.1057/s41260-022-00276-y>
- O’Hara, M. (2014). What’s not there: Odd lots and market data. *The Journal of Finance*, 69(5), 2199–2236. <https://doi.org/10.1111/jofi.12185>
- Panayi, E. (2018). Designating market maker behaviour in limit order book markets. *Journal of Banking and Finance*, 87, 174–194. <https://doi.org/10.1016/j.jbankfin.2017.10.004>

- Shen, J. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, 7(1), 66. <https://doi.org/10.1186/s40537-020-00333-6>
- Sirignano, J. (2019). Deep learning for limit order books. *Quantitative Finance*, 19(4), 549–570. <https://doi.org/10.1080/14697688.2018.1546053>
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288. <https://www.jstor.org/stable/2346178>
- Timmermann, A. (2018). Forecasting methods in finance. *Annual Review of Financial Economics*, 10(1), 449–479. <https://doi.org/10.1146/annurev-financial-110217-022713>
- Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10), 1550–1560. <https://doi.org/10.1109/5.58337>
- Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476), 1418–1429. <https://doi.org/10.1198/016214506000000735>

Tables

Table 1: Basic description of the size of the data used

Description	Value
Securities included	TSLA, NFLX, ANSS, PEP, and AZN
Date range of data	2023.10.02 to 2023.12.29
Number of trading days included	63
Number of tickers available on all days	5
Total disk size	13.97 GB

Table 2: Example of Tesla (TSLA) trade and quote events from dataset.

Date	Timestamp	EventType	Ticker	Price
20231229	10:11:28.680298656	QUOTE ASK NB	TSLA	253.35
20231229	10:11:28.769871254	TRADE NB	TSLA	253.3
20231229	10:11:28.777538304	QUOTE BID NB	TSLA	253.32
20231229	10:11:28.777538304	QUOTE ASK NB	TSLA	253.35
20231229	10:11:28.778397101	TRADE NB	TSLA	253.3
20231229	10:11:28.779588485	QUOTE BID NB	TSLA	253.32
20231229	10:11:28.779588485	QUOTE ASK NB	TSLA	253.35
20231229	10:11:28.780025510	QUOTE BID NB	TSLA	253.32
20231229	10:11:28.780025510	QUOTE ASK NB	TSLA	253.3
20231229	10:11:28.794160459	QUOTE BID NB	TSLA	253.32

Note: "NB" refers to the *National Best Price (NBP)*, which represents the highest bid price and lowest ask price available across multiple trading venues in compliance with SEC regulations.

Table 3: Example of odd-lots transaction recorded for NFLX.

Date	Timestamp	EventType	Price	Quantity
20231025	09:30:00.534098344	TRADE	415.52	1
20231025	09:30:00.570152324	TRADE	416.03	64
20231025	09:30:00.572346236	QUOTE BID NB	415.4	200
20231025	09:30:00.572346478	QUOTE ASK NB	416.15	200
20231025	09:30:00.589711862	TRADE	415.52	1
20231025	09:30:00.611061332	TRADE	415.84	1

Table 4: Summary statistics: ANSS

Data	Mean	Std	Skewness	Kurtosis
Data rows	93.9K	59.2K	2.95	0.99
Transactions	16.2K	8.7K	4.51	2.57
Quote Updates	77.7K	40.8K	2.31	6.84
Volume (# shares)	245K	214.1K	5.17	3.1
Volume (\$)	74.8M	74.6M	5.40	3.3
Market Cap (\$)	25.9B	1.84B	2.11	4.38
Nominal Price (\$)	296.04	20.97	0.50	4.37
Daily Return (%)	-0.11	1.49	0.86	3.63
5 seconds returns	-1.1e-06	1.9e-04	0.71	117.5
10 trades returns	4.1e-07	3.1e-04	0.49	38.16
2000 volume returns	1.9e-05	1.2e-03	0.12	1.68
10 trades duration (seconds)	14.6	20.7	2.20	14.26
2000 volume duration (seconds)	630	372	0.06	-0.76

Table 5: Summary statistics: AZN

Data	Mean	Std	Skewness	Kurtosis
Data rows	536K	176K	2.6	0.11
Transactions	30.5K	15K	4.53	2.6
Quote Updates	506K	164K	2.35	9.83
Volume (# shares)	1.1M	838K	2.81	8.17
Volume (\$)	71.4M	54.8M	2.88	6.84
Market Cap (\$)	100B	2.58B	0.59	-0.73
Nominal Price (\$)	65.00	1.66	0.59	-0.73
Daily Return (%)	-0.08	0.80	-0.15	0.94
5 seconds returns	-7.2e-07	1.4e-04	-2.50	588.8
10 trades returns	-2.8e-07	1.3e-04	-0.79	177.5
2000 volume returns	-4.3e-06	5.2e-03	-0.04	8.32
10 trades duration (seconds)	7.79	10.59	2.05	10.71
2000 volume duration (seconds)	178	102	0.64	0.59

Table 6: Summary statistics: NFLX

Data	Mean	Std	Skewness	Kurtosis
Data rows	294K	105K	2.17	0.83
Transactions	93.4K	37.5K	3.69	1.9
Quote Updates	201K	71.2K	1.35	3.21
Volume (# shares)	1.6M	907K	3.92	2.1
Volume (\$)	682M	353M	4.02	2.2
Market Cap (\$)	195B	19.5B	-0.47	-1.06
Nominal Price (\$)	436.53	43.59	-0.45	-1.04
Daily Return (%)	0.03	1.39	0.23	0.26
5 seconds returns	-4.0e-06	2.8e-04	4.01	431.02
10 trades returns	4.7e-07	2.2e-04	6.65	835.43
2000 volume returns	2.4e-05	6.1e-04	0.10	33.76
10 trades duration (seconds)	2.55	3.75	2.26	10.18
2000 volume duration (seconds)	94.3	59.53	0.78	0.81

Table 7: Summary statistics: PEP

Data	Mean	Std	Skewness	Kurtosis
Data rows	626K	184K	0.79	1.25
Transactions	73.5K	20.4K	0.49	0.02
Quote Updates	553K	170K	0.91	1.46
Volume (# shares)	1.65M	679K	2.33	6.85
Volume (\$)	267M	110M	2.38	7.19
Market Cap (\$)	228B	4.6B	-0.70	-0.76
Nominal Price (\$)	165.83	3.36	-0.70	-0.76
Daily Return (%)	-0.16	1.04	-1.8	0.67
5 seconds returns	-3.0e-06	1.5e-04	3.74	560.54
10 trades returns	-3.5e-07	1.2e-04	1.84	766.87
2000 volume returns	3.1e-06	3.7e-04	0.39	9.85
10 trades duration (seconds)	3.12	4.46	1.96	13.28
2000 volume duration (seconds)	92.7	59.02	0.35	-0.09

Table 8: Summary statistics: TSLA

Data	Mean	Std	Skewness	Kurtosis
Data rows	3.29M	679K	-0.07	-0.03
Transactions	336K	55.8K	0.35	0.74
Quote Updates	2.95M	650K	0.00	-0.21
Volume (# shares)	96.8M	15.8M	0.03	-0.12
Volume (\$)	22.8B	3.6B	0.09	0.19
Market Cap (\$)	760B	56.6B	-0.51	-0.78
Nominal Price (\$)	237.65	17.79	-0.51	-0.78
Daily Return (%)	0.02	2.37	-0.17	-0.48
5 seconds returns	-5.2e-07	2.3e-04	0.27	23.35
10 trades returns	-5.9e-08	8.4e-05	-1.59	213.99
2000 volume returns	4.9e-07	1.2e-04	0.81	137.69
10 trades duration (seconds)	0.62	0.65	2.03	6.33
2000 volume duration (seconds)	1.38	1.29	1.87	5.17

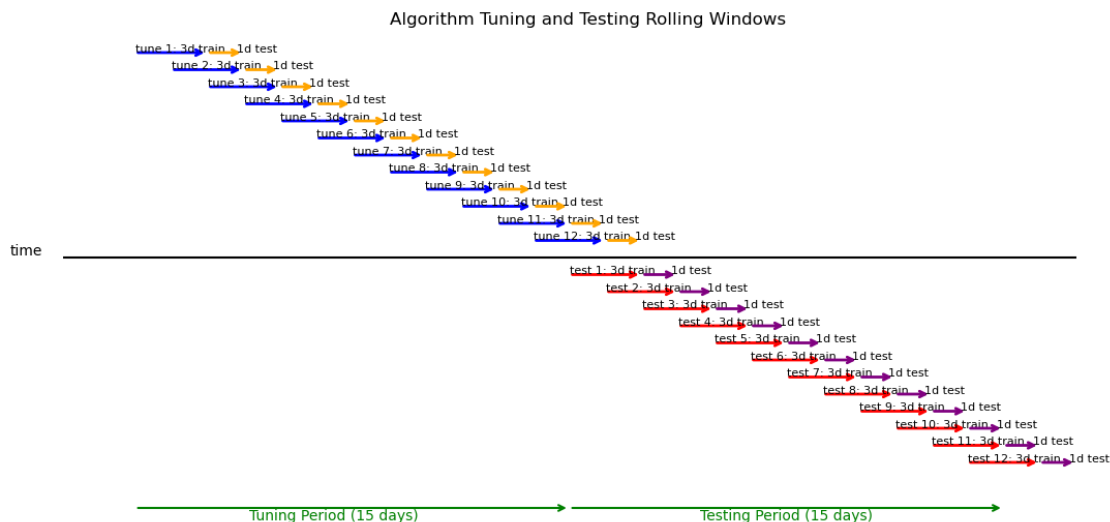
Table 9: Example of trade direction assignment using the Lee and Ready (1991) algorithm for TSLA.

Date	Timestamp	EventType	Price	Quantity	TradeDirection
20231201	09:30:00.059320	TRADE NB	233.14	100	
20231201	09:30:00.087025	TRADE NB	233.10	100	-1
20231201	09:30:00.087256	TRADE NB	233.10	100	-1
20231201	09:30:00.141223	TRADE NB	233.03	100	-1
20231201	09:30:00.141390	TRADE NB	233.05	100	1
20231201	09:30:00.141390	TRADE NB	233.04	163	-1
20231201	09:30:00.315829	TRADE NB	233.10	335355	1
20231201	09:30:00.316593	TRADE NB	233.01	100	-1

Note: The timestamp is rounded in microsecond for computational efficiency during model evaluation

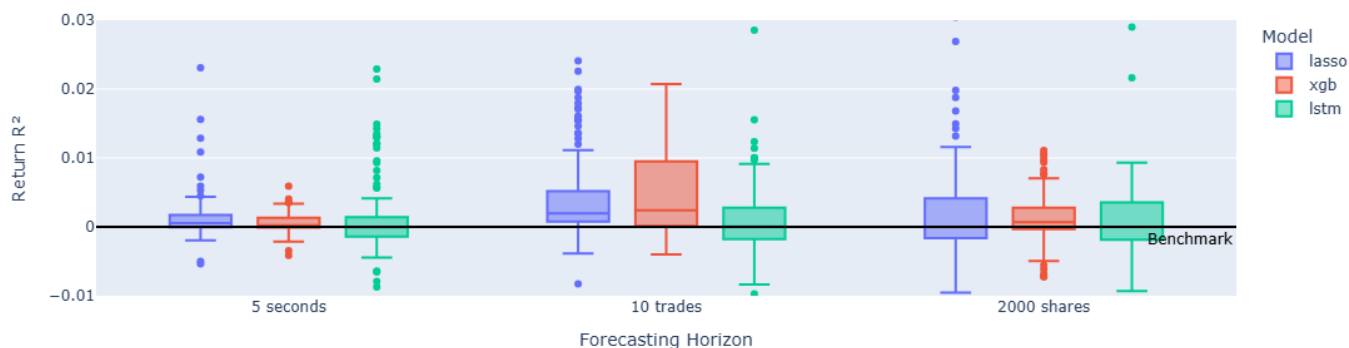
Figures

Figure 1: Algorithm tuning and testing rolling window



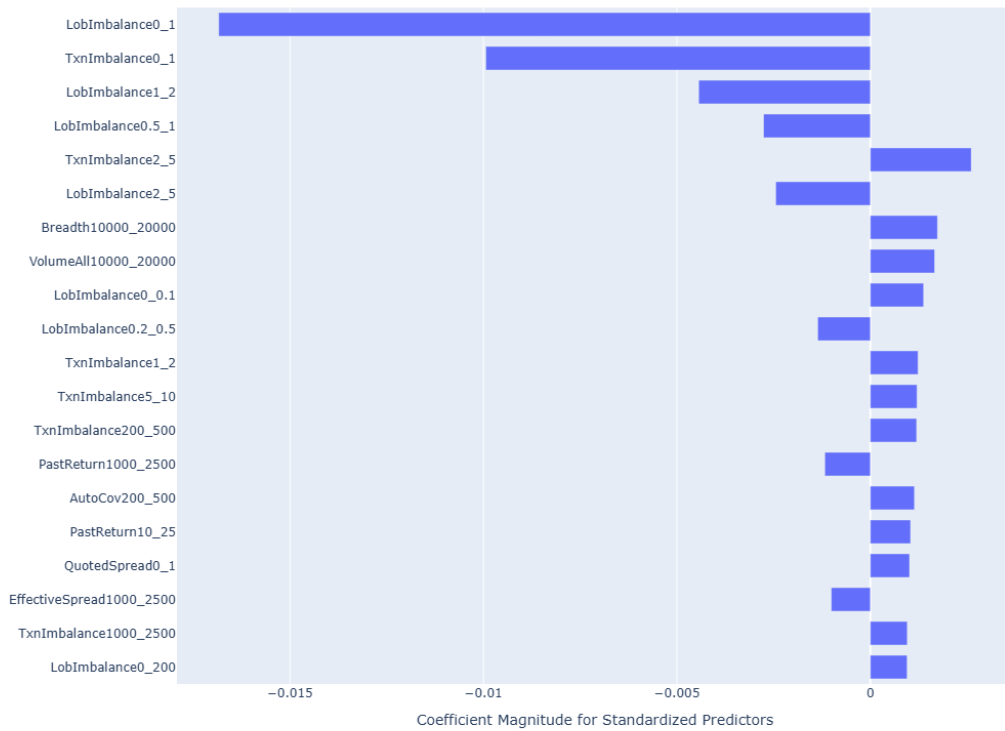
Note: The tuning parameters are optimized twice during the whole period, hence every 15 days. For every given set of parameters, we use the past 3 days of trading data to train the model and compute the testing errors for the next day. The testing errors during the 15-day period then define the fixed parameters we are going to use for the next set of 15 days (testing period). Following the same procedure, using the past 3 days of data, we evaluate the model's performance on the next target day, for the upcoming 15 days. This cycle then repeats for the next 30 days.

Figure 2: Out-of-sample R^2 performance for transaction return across different forecasting horizons and models



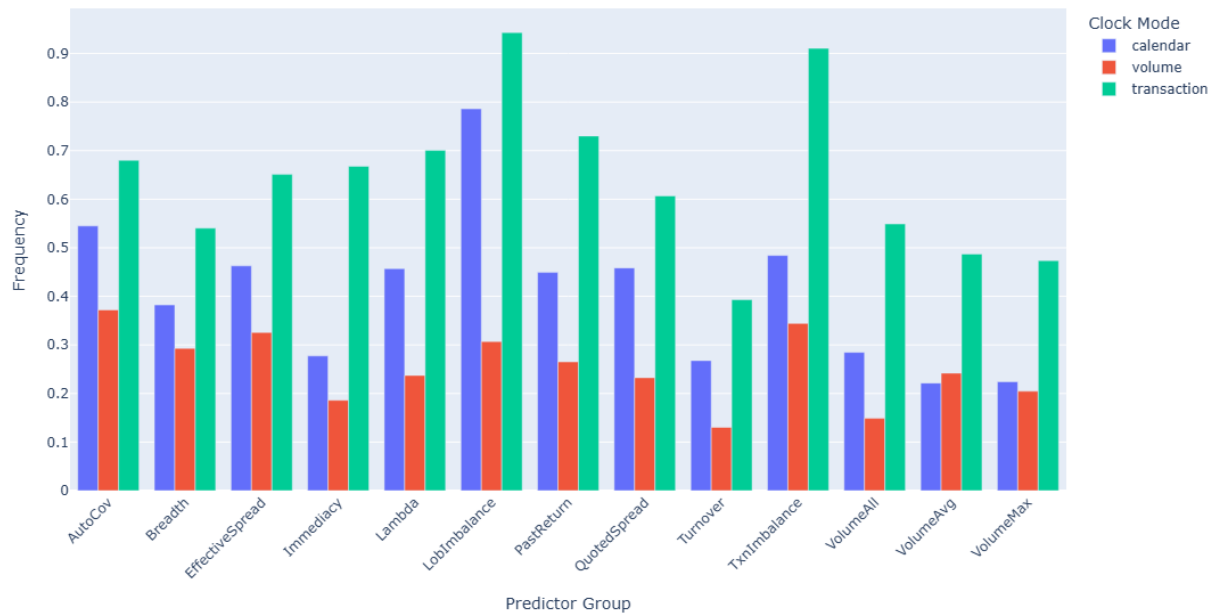
Note: The x-axis shows the look-forward intervals for all three clock modes. Each boxplot summarizes the distribution of the average out-of-sample R^2 , which measures the performance during the testing period of 30 days between October 2023 to December 2023, across all five stocks. The black horizontal line represents the performance of the in-sample average.

Figure 3: LASSO coefficient magnitudes for standardized predictors across all stocks and days for transaction return



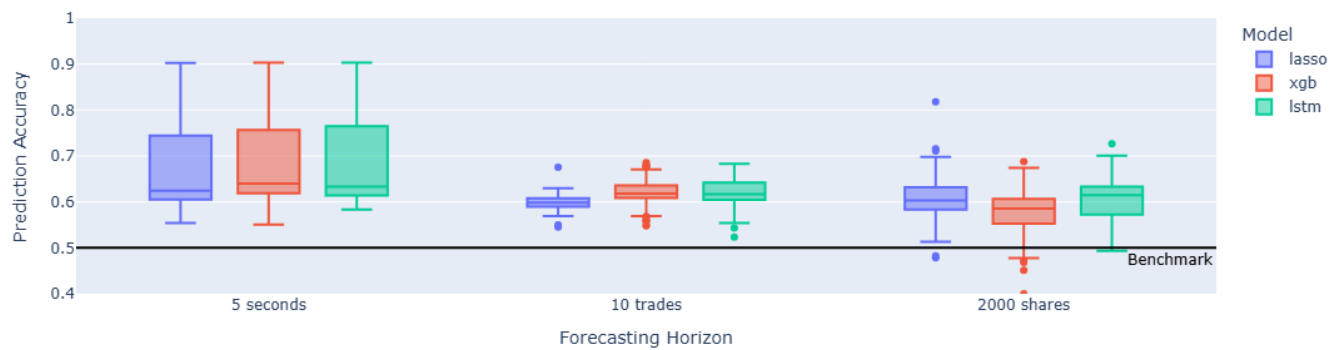
Note: This plots shows the average coefficients across all tests for the 20 most significant predictors, based on absolute average values, after standarzitation. Y-axis shows the selected variables and x their coefficient's magnitude sorted from highest to lowest. The values next to the predictors name, indicate the lookback intervals. For example LobImbalance0_1 refers to the imbalance of current time t and 1 trade ago, while LobImbalance1_2 to the time needed 2 trades ago to 1 trade ago, from current time t.

Figure 4: Frequency of predictors group selection in LASSO for transaction return



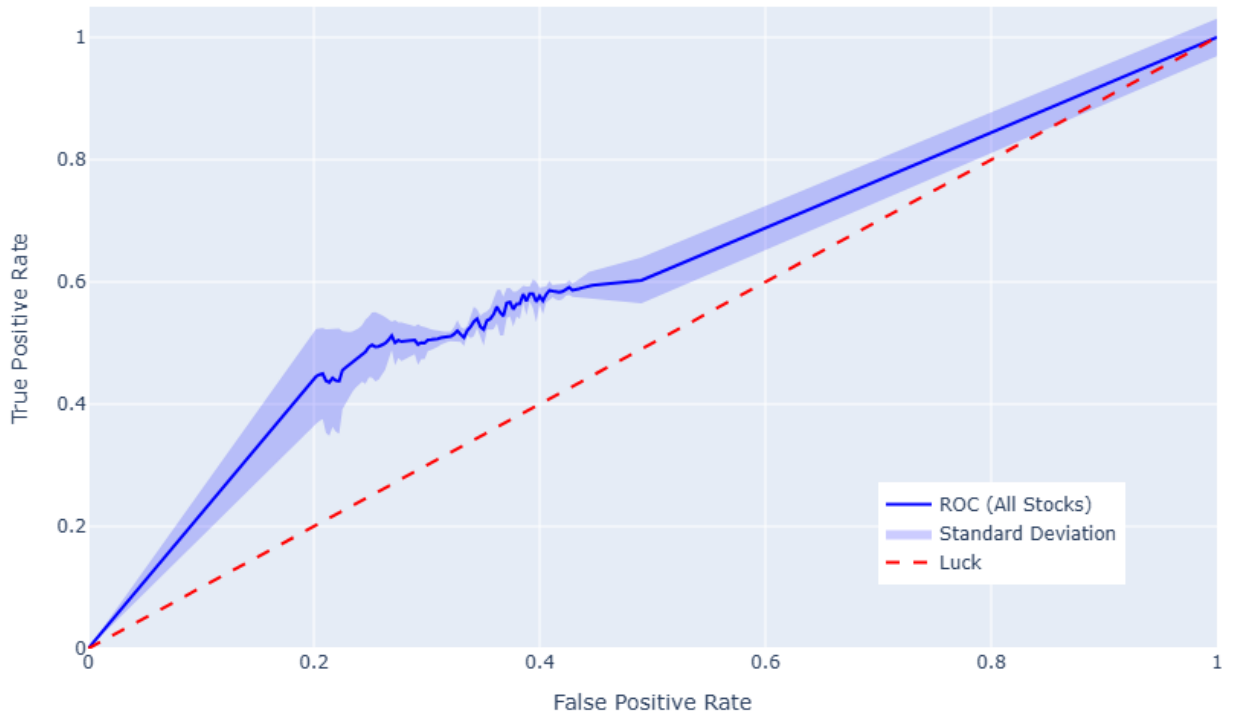
Note: This plots shows the frequency of mostly used group of predictor variables, out of the 13. Frequency of a group is calculated over the testing period of all 5 stocks. A group is counted as selected if at least one of the predictors of the specific group is selected, hence differs from zero, for a given trading day.

Figure 5: Prediction accuracy for price direction across different clock modes and forecasting horizons



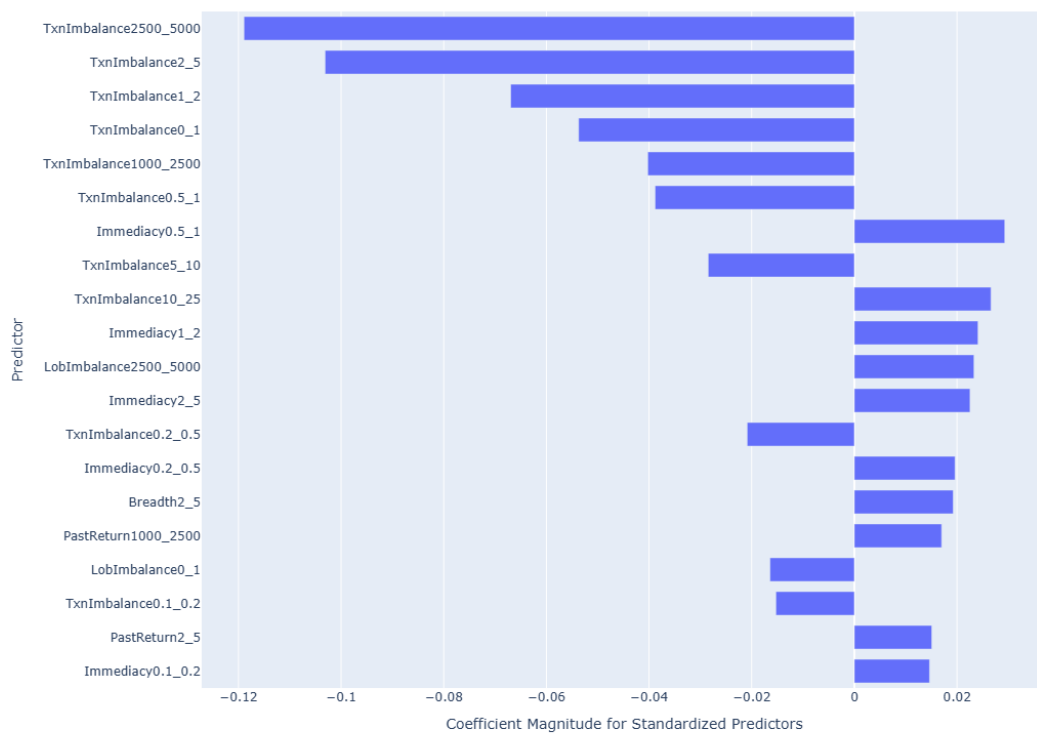
Note: Similar captions to that in Figure 1.

Figure 6: Receiver Operator Curve for all stocks and clock modes using XGB



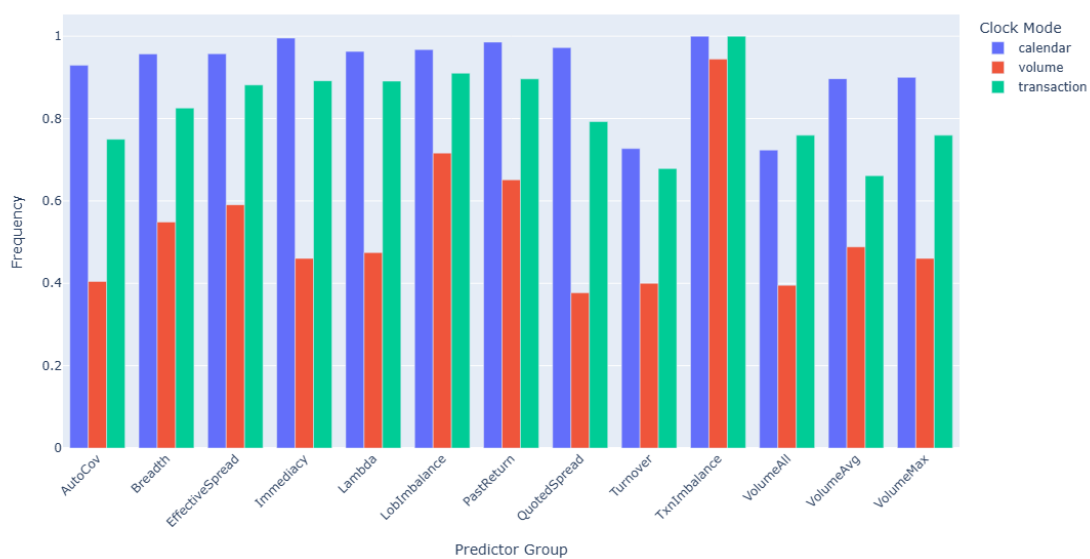
Note: ROC is constructed as follows: For each day in our sample, we directly computed the True Positive Rate (TPR) and False Positive Rate (FPR) and plotted these points to form the ROC curve, using XGB model for all stocks. Given the limited number of observations, the range of available TPR-FPR values is not sufficient to fully capture a smooth ROC representation, so we performed linear interpolation between the last available (minimum and maximum) pair of values and (0,0),(1,1) respectively. The shaded area depict the 95% confidence intervals of the mean accuracy.

Figure 7: LASSO coefficient magnitudes for standardized predictors across all stocks and days in price direction prediction



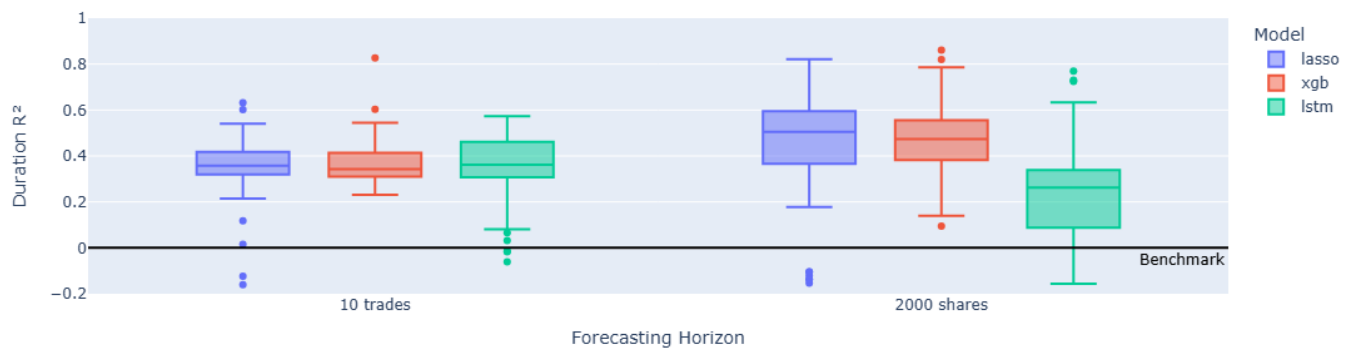
Note: Similar Caption as in Figure 3.

Figure 8: Frequency of predictors group selection in LASSO for Price Direction



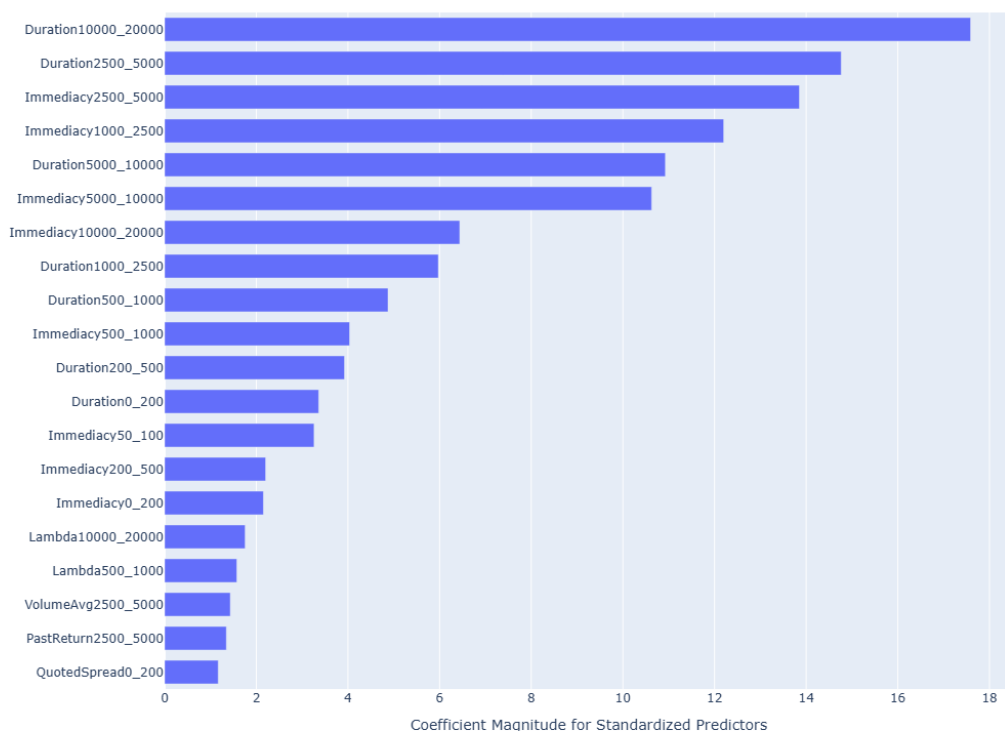
Note: Similar Caption as in Figure 4.

Figure 9: Out-of-sample R^2 performance for trade duration prediction across different clock modes and forecasting horizons



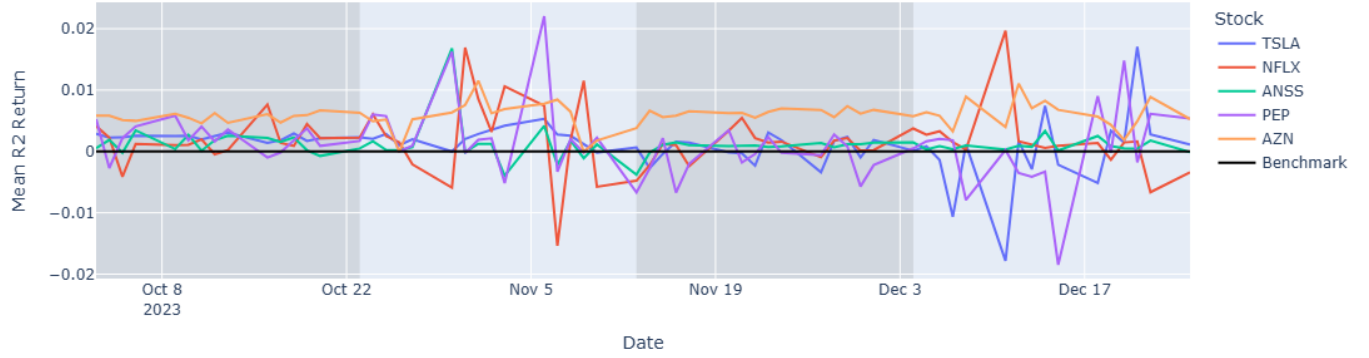
Note: The x-axis shows the look-forward intervals for both clock modes, transaction and volume. Each boxplot summarizes the distribution of the average out-of-sample R^2 for duration, which measures the performance during the testing period of 30 days between October 2023 to December 2023, across all five stocks. The black horizontal line represents the performance of the in-sample average.

Figure 10: LASSO coefficient magnitudes for standardized predictors across all stocks and days in trade duration prediction



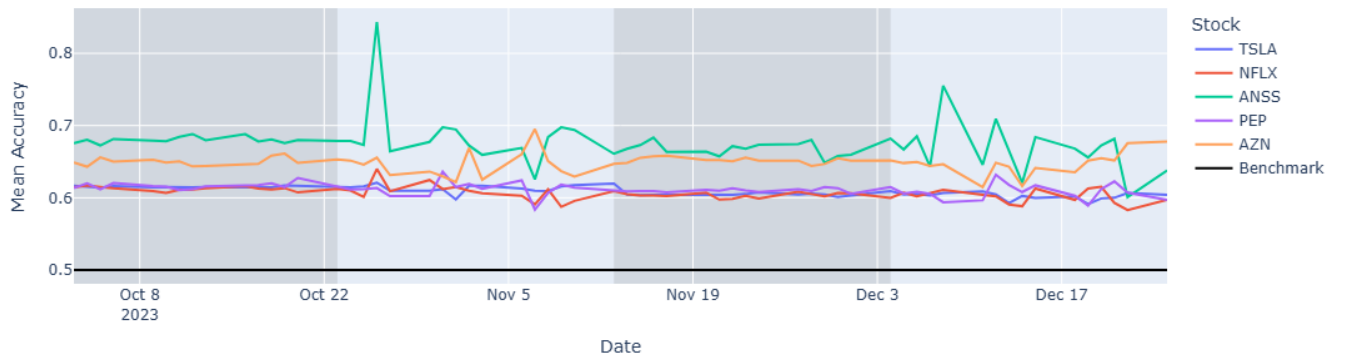
Note: All the explanatory variables are standardized before fitting LASSO and the variables with largest 20 absolute coefficients are displayed in this plot. As mentioned in section 2, Breadth variable is not applicable for transaction clock interval, so instead, duration variable is used for different lookback intervals, as the time of certain amount of transactions to occur. Specifically the Duration10000_20000 refers to the lookback window of 100 trades before current time t , to 50 trades before. Similarly Duration2500_5000 refers to the lookback window of 25 trades before current time t , to 10 trades before, Duration5000_10000 refers to 50 trades before current time t to 25, and so on.

Figure 11: Time series of daily out-of-sample R^2 values for transaction return predictability



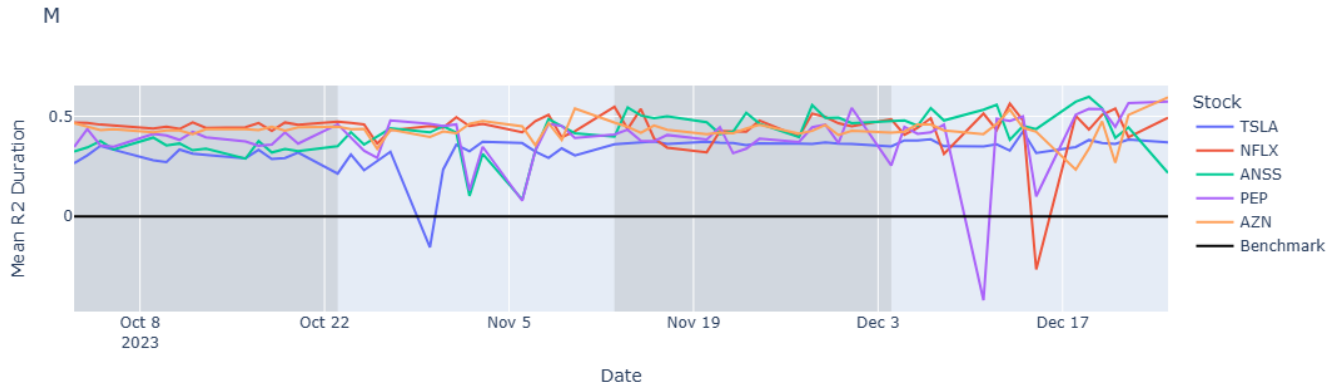
Note: This plot shows the average out of sample R^2 for Transaction Return throughout all models, clock modes and all 63 days ranging from October to December of 2023. The shaded are represents the tuning periods, while the blue area the testing.

Figure 12: Time series of daily accuracy for trade direction predictability across different stocks



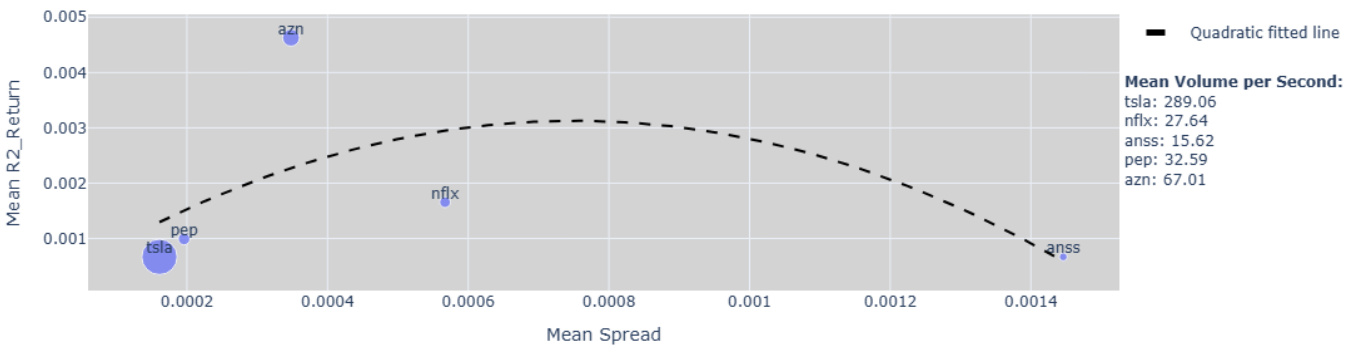
Note: This plot shows the average out of sample accuracy for Price Direction, throughout all models, clock modes and all 63 days ranging from October to December of 2023. The shaded are represents the tuning periods, while the blue area the testing.

Figure 13: Time series of daily out-of-sample R^2 values for trade duration predictability across different stocks



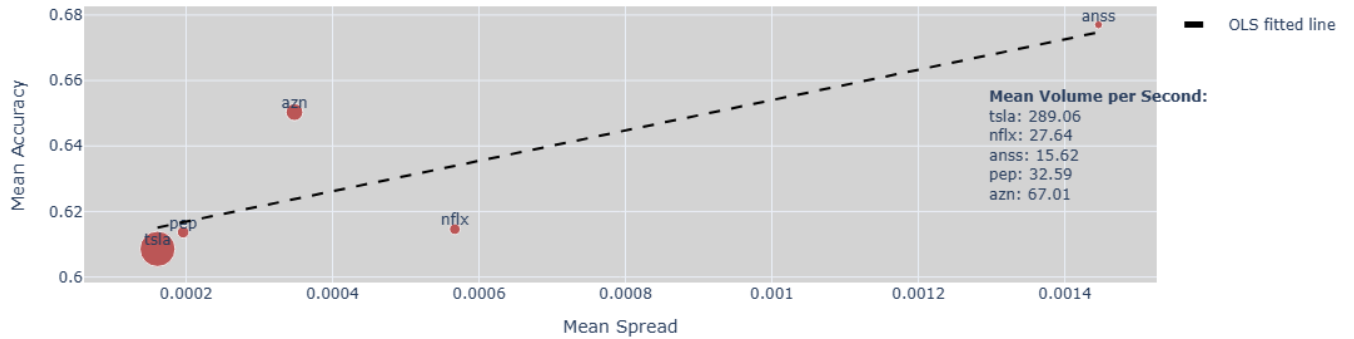
Note: Similar Caption as in Figure 11.

Figure 14: Relationship between return predictability and liquidity



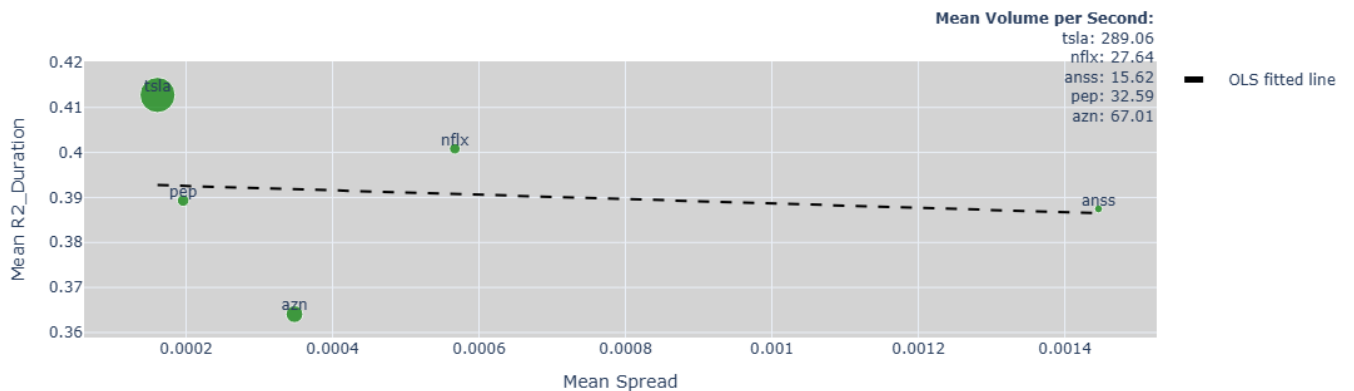
Note: This plot shows how each stocks performance based on the mean out-of-sample R^2 for transaction return across all clock modes, models and testing days, is correlated to its liquidity. For measuring liquidity, we compute the percentage spread, calculated as the average bid-ask spread divided by the mid-price, sampled at 15-second intervals throughout the day. In the plot, it is also depicted the trading intensity in terms of volume, represented by the size of each stock's point on the plot. Trading intensity is calculated as the mean total shares traded per second throughout day, averaged for all 63 days for the period of October 2023 to December 2023. The black dotted line represents the Ordinary Least Squares (OLS) fitted quadratic line.

Figure 15: Relationship between trade direction and liquidity



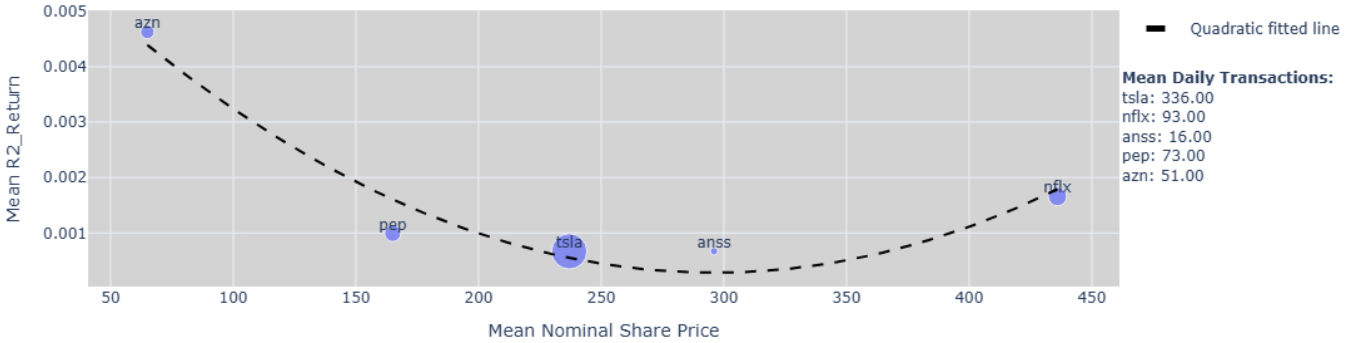
Note: This plot shows how each stocks performance based on the mean accuracy for price direction across all clock modes, models and testing period, is correlated to its liquidity. For measuring liquidity, we compute the percentage spread, calculated as the average bid-ask spread divided by the mid-price, sampled at 15-second intervals throughout the day. In the plot, it is also depicted the trading intensity in terms of volume, represented by the size of each stock's point on the plot. Trading intensity is calculated as the mean total shares traded per second throughout day, averaged for all days for the period of October 2023 to December 2023. The black dotted line represents the Ordinary Least Squares (OLS) fitted line.

Figure 16: Relationship between trade duration predictability and liquidity



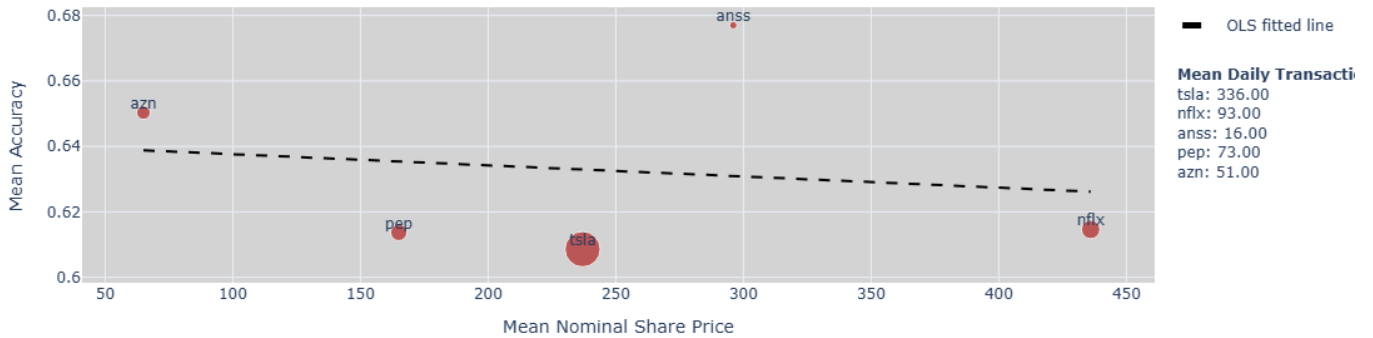
Note: Similar captions as in Figures 14 and 15.

Figure 17: Relationship between return predictability and nominal share price



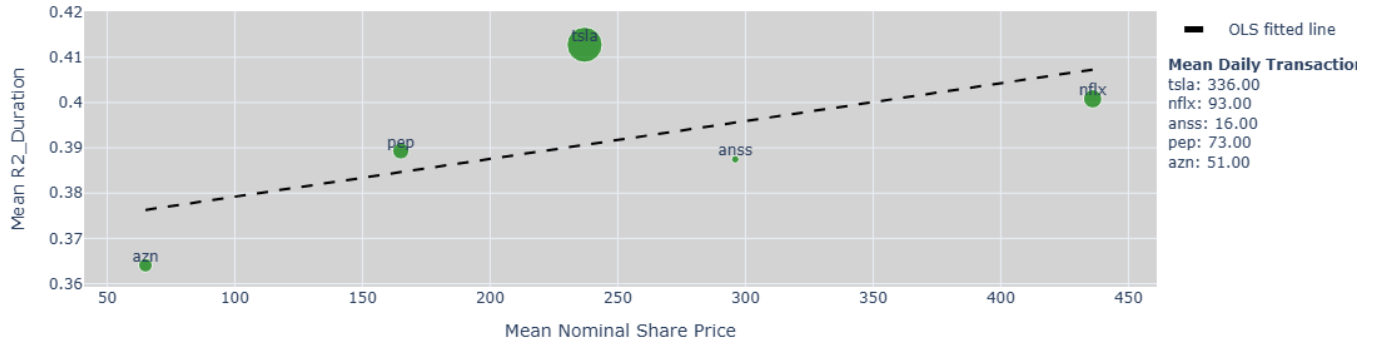
Note: This plot shows how each stocks performance based on the mean out-of-sample R^2 for transaction return across all clock modes, models and testing period, is correlated to its nominal share price. For measuring share price, we compute the average stock's closing price each day for the period of October 2023 to December 2023. In the plot, it is also depicted the trading intensity in terms of transactions, represented by the size of each stock's point on the plot. Trading intensity is calculated as the mean total shares traded per day throughout the period of October 2023 to December 2023 (as multiples of 1000). For example Mean Daily Transaction for Tsla is 336k total trades. The black dotted line represents the Ordinary Least Squares (OLS) quadratic fitted line.

Figure 18: Relationship between direction accuracy and nominal share price



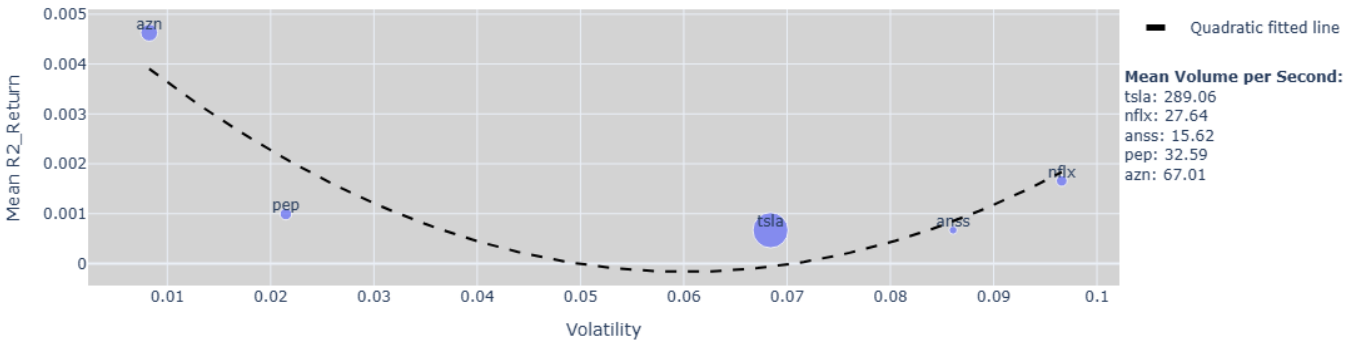
Note: This plot shows how each stocks performance based on the mean accuracy for price direction across all clock modes, models and testing period, is correlated to its nominal share price. For measuring share price, we compute the average stock's closing price each day for the period of October 2023 to December 2023. In the plot, it is also depicted the trading intensity in terms of transactions, represented by the size of each stock's point on the plot. Trading intensity is calculated as the mean total shares traded per day throughout the period of October 2023 to December 2023 (as multiples of 1000). For example mean daily Transaction for Nflx is 93k total trades. The black dotted line represents the Ordinary Least Squares (OLS) fitted line.

Figure 19: Relationship between trade duration predictability and nominal share price



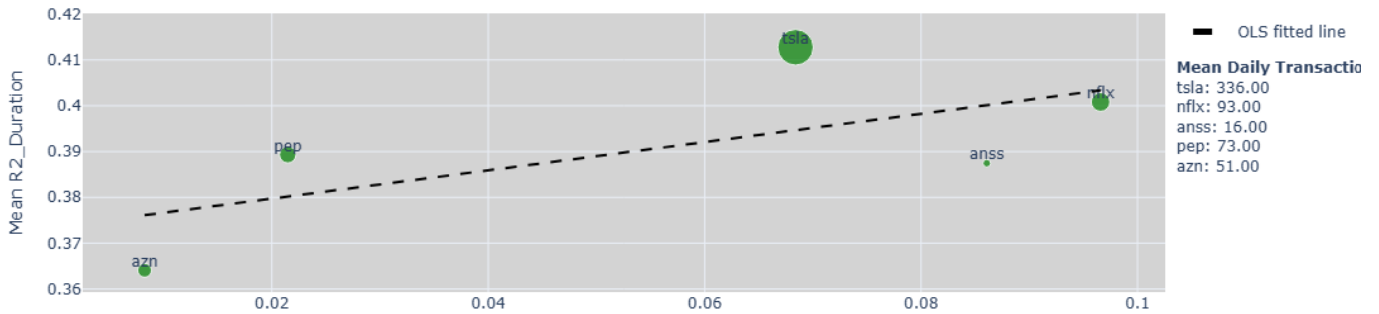
Note: Similar captions as in Figures 17 and 18.

Figure 20: Relationship between return predictability and volatility



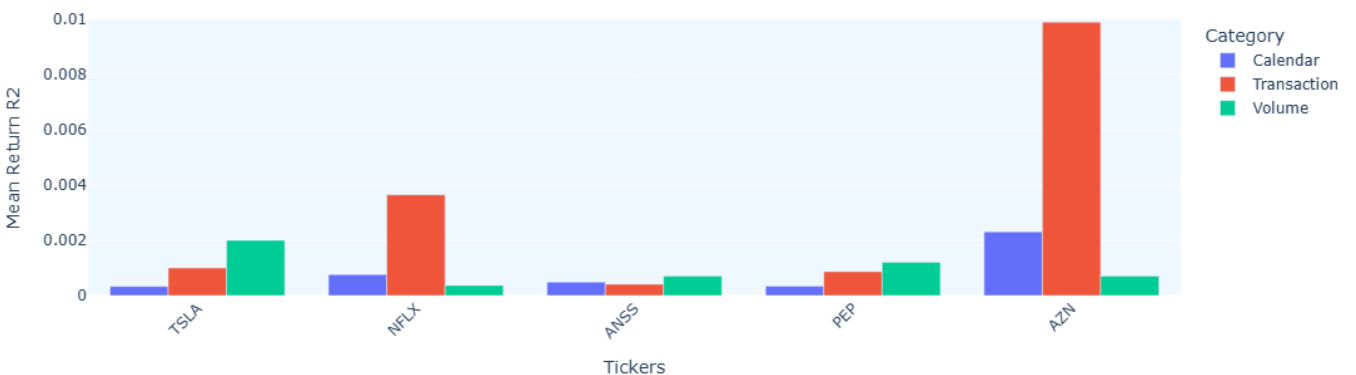
Note: This plot shows how each stocks performance based on the mean out-of-sample R^2 for transaction return across all clock modes, models and testing period, is correlated to volatility. For measuring volatility, we compute the average stock's standard deviation of mid-price returns measured at 30-second intervals throughout the day. In the plot, it is also depicted the trading intensity in terms of volume, represented by the size of each stock's point on the plot, calculated as mentioned previously. The black dotted line represents the Ordinary Least Squares (OLS) quadratic fitted line.

Figure 21: Relationship between trade duration predictability as a function of volatility



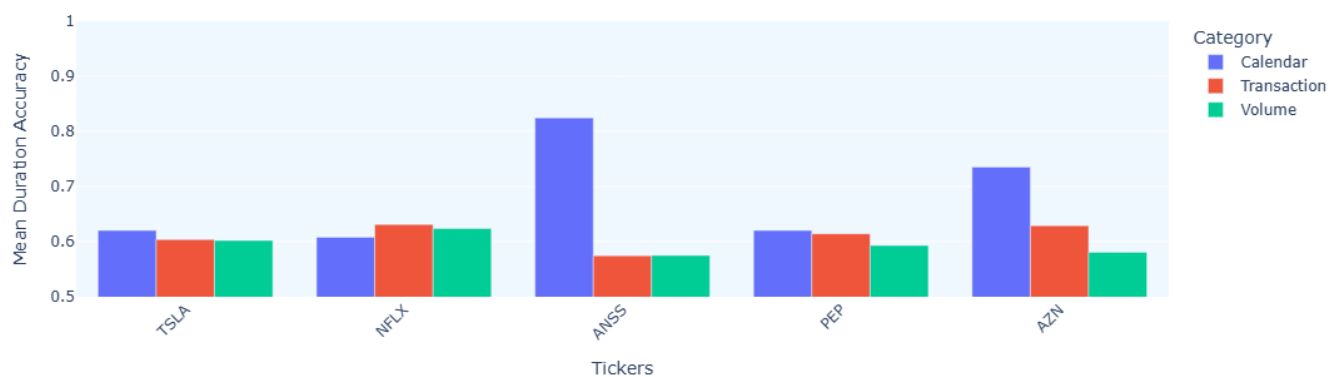
Note: This plot shows how each stock's performance based on the mean out-of-sample R^2 for trade duration across all clock modes, models and testing period, is correlated to its volatility. For measuring volatility, we compute the average stock's standard deviation of mid-price returns measured at 30-second intervals throughout the day. In the plot, it is also depicted the trading intensity in terms of transactions, represented by the size of each stock's point, calculated as mentioned previously. The black dotted line represents the Ordinary Least Squares (OLS) fitted line.

Figure 22: Return predictability across all stocks and clock modes



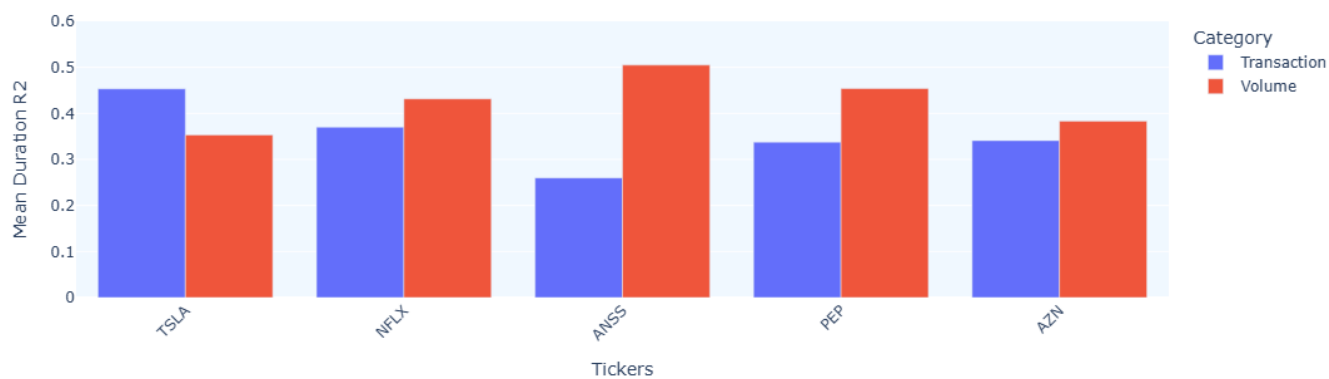
Note: This bar chart illustrates the mean out-of-sample R^2 for transaction return predictability, across different clock modes and stocks, throughout all 30 testing days.

Figure 23: Trade direction accuracy across stocks and clock modes



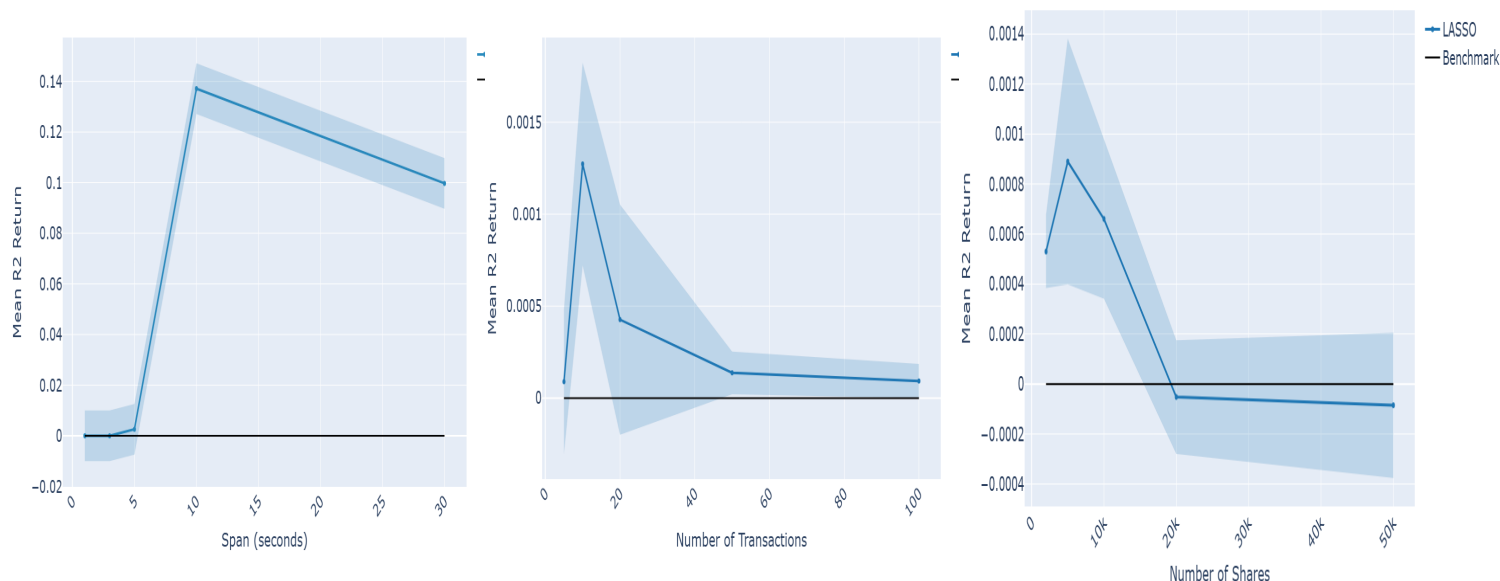
Note: This bar chart illustrates the mean accuracy for price direction predictability across all clock modes, for all 5 stocks, throughout all 30 testing days.

Figure 24: Trade duration predictability across stocks and clock modes



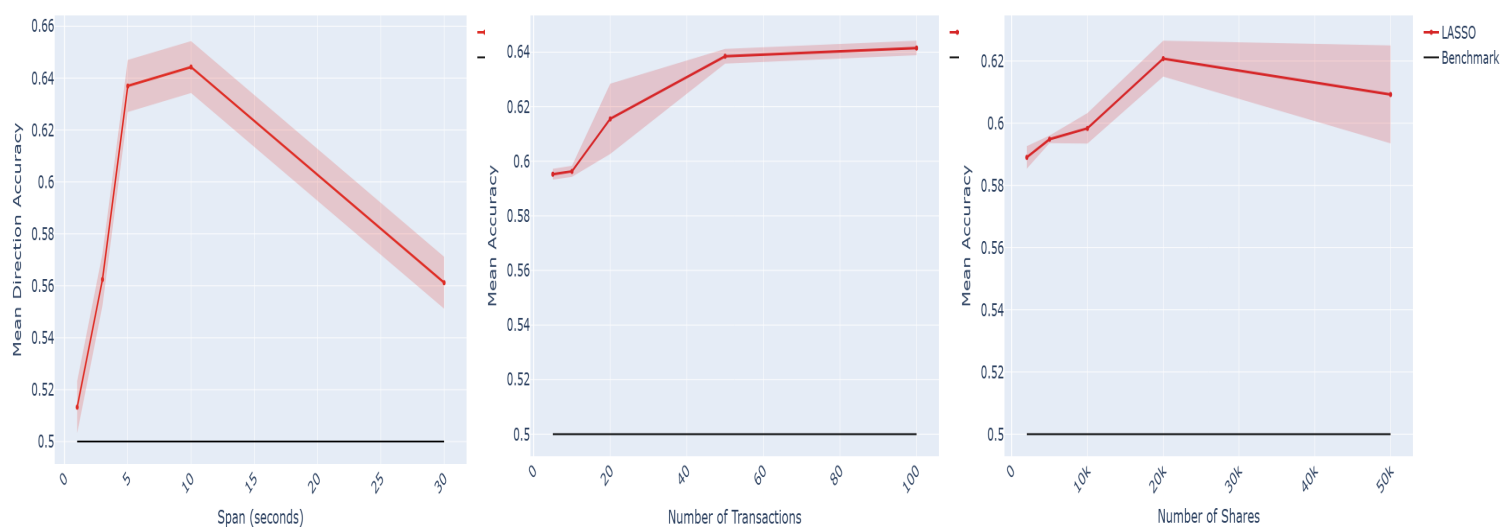
Note: Similar to caption 22.

Figure 25: Predictability lifespan: Return prediction performance as a function of the time horizon



Note: The shaded areas depict the 95% confidence intervals of the mean out-of-sample R^2 over the 3 testing days of 21-23 November for Tesla. The plots depict the results for return predictions in calendar, transaction and volume clocks respectively.

Figure 26: Predictability lifespan: Direction prediction performance as a function of the time horizon



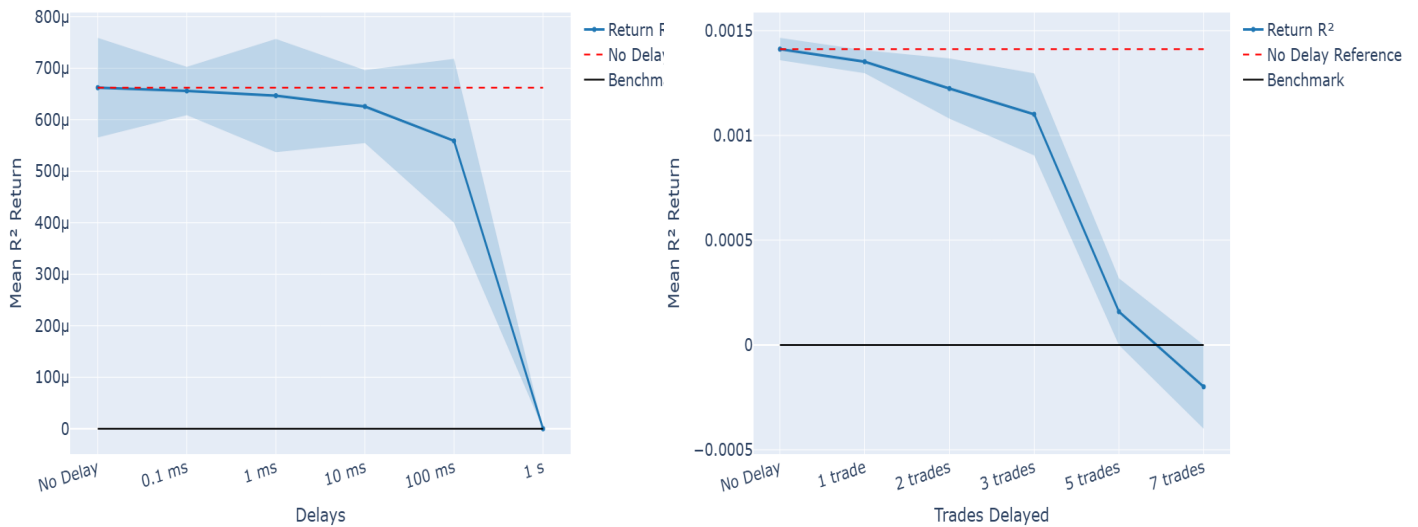
Note: The shaded areas depict the 95% confidence intervals of the mean accuracy over the 3 testing days of 21-23 November for Tesla. The plots depict the results for Direction accuracy in calendar, transaction and volume clocks respectively.

Figure 27: Predictability lifespan: Duration prediction performance as a function of the time horizon



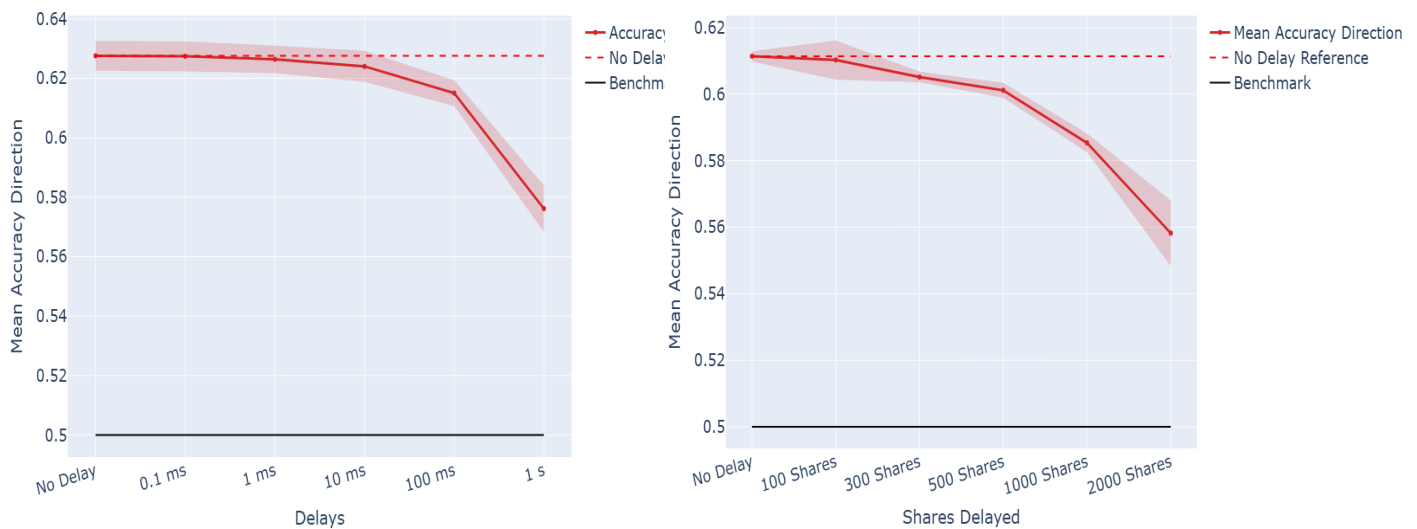
Note: Left and right panel are predictability of duration in transaction and volume clock respectively, for Tsla. The shaded area represents the 95% confidence intervals of the mean out-of-sample R^2 for the 3 testing days of 21-23 November.

Figure 28: The cost of delay: Returns predictability as a function of lags in data processing



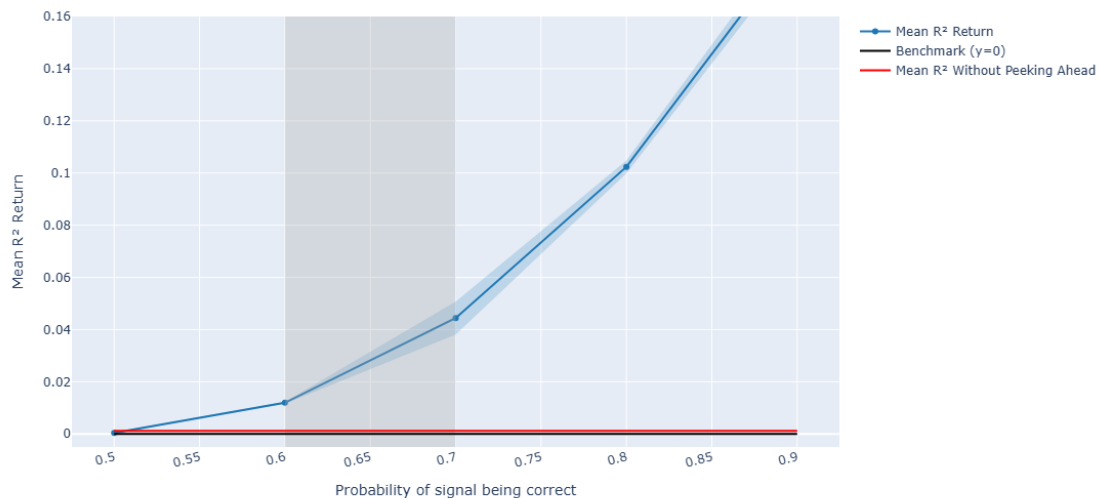
Note: Predictability of Return for different levels of delay for Tsla. Left and right panel depict the decay of out-of-sample R^2 for calendar and transaction clock modes, respectively. The forecasting horizons are 5 seconds and 10 transactions, respectively, while the shaded area indicates the 95% confidence intervals of mean daily predictability for 3 days of 21-23 November.

Figure 29: The cost of delay: Direction predictability as a function of lags in data processing



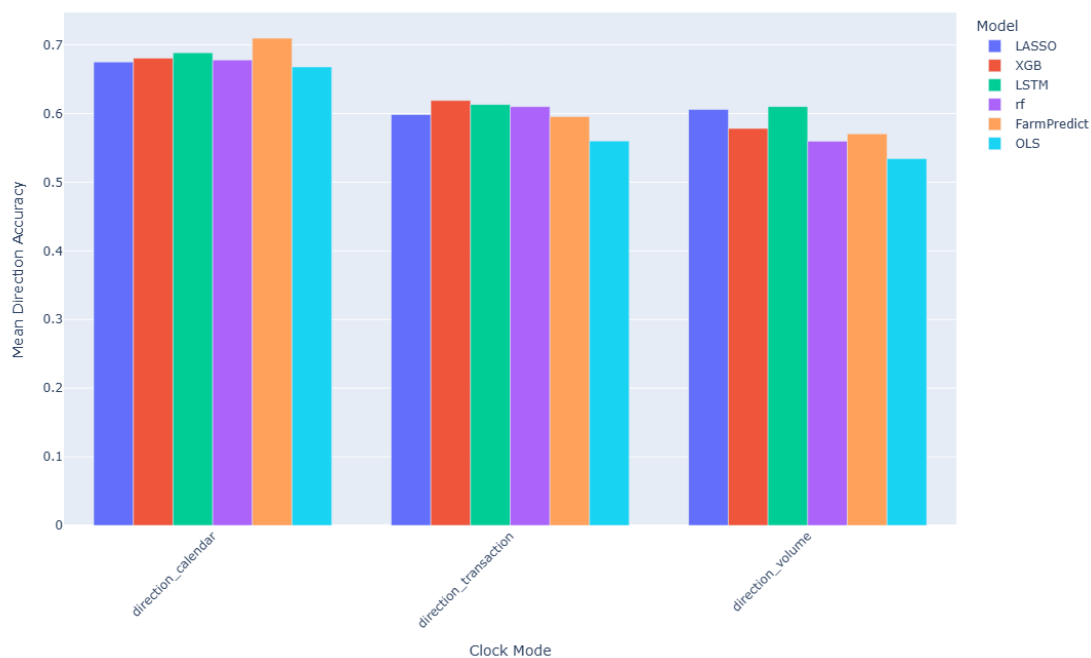
Note: Accuracy of Direction for different levels of delay for Tesla. Left and right panel depict the decay of accuracy for calendar and volume clock modes, respectively. The forecasting horizons are 5 seconds and 5k total shares traded, respectively, while the shaded area indicates the 95% confidence intervals of mean daily accuracy for days 21-23 November.

Figure 30: Peek ahead: Predictability of return as a function of the probability of the directional upcoming signal being correct



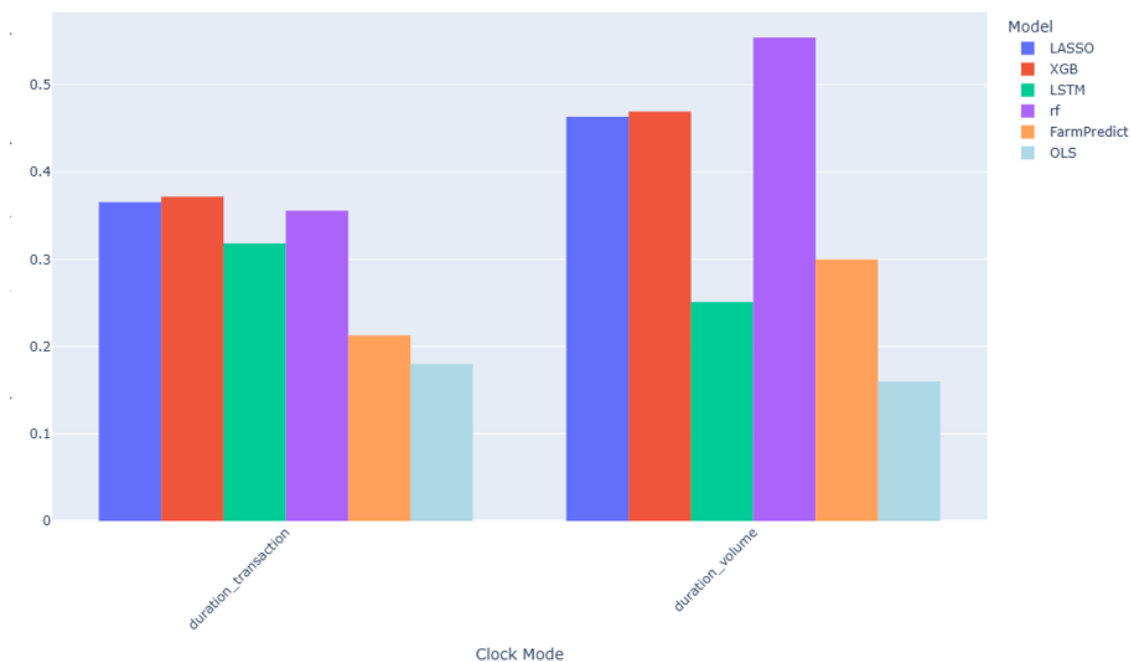
Note: Predictability across different levels of correct signals. The response is 5 second return of Tesla. The blue shaded area depicts the 95% confidence intervals of mean daily predictability for 3 days of 21-23 November, while the grey indicates a rational range of probability of being correct. Red line depicts the average predictability, when no look-ahead signals are used.

Figure 31: Model performance comparison across various machine learning models for price direction



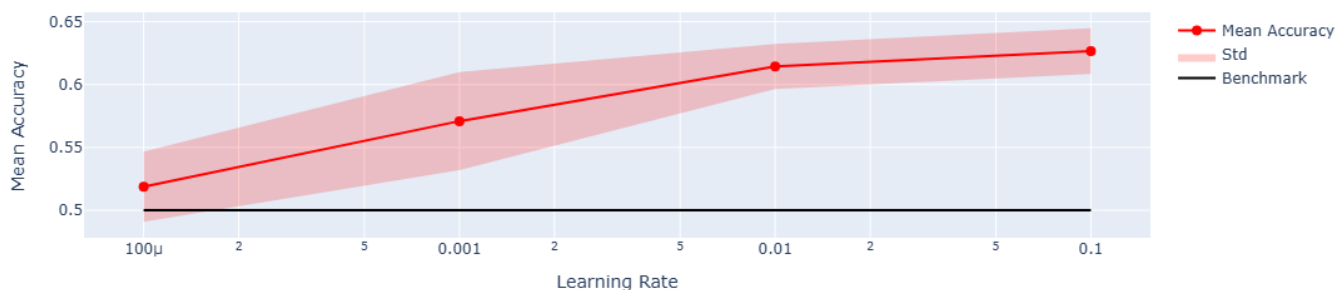
Note: Average accuracy of price direction for all 5 stocks across different machine learning models. Each bar summarizes the mean performance for all 30 testing days, while each group of bars, represents a clock mode, precisely, calendar, transaction and volume, respectively.

Figure 32: Model performance comparison across various machine learning models for trade duration



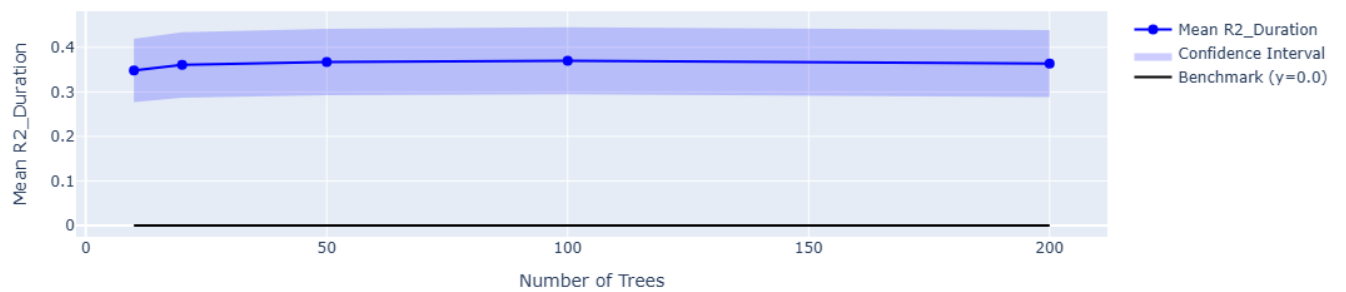
Note: Average out-of-sample R^2 of trade duration for all 5 stocks across different machine learning models. Each bar summarizes the mean performance for the 30 testing days of October 2023 to December 2023. Left group of bars represents results for transaction clock mode, while right group for volume.

Figure 33: Sensitivity of direction predictability performance to the learning rate in XGB



Note: This plot illustrates the accuracy of price direction as a function of the learning rate in XGB model for Tesla across all clock modes. y axis depicts the average accuracy obtained across all 30 days of testing period, for price direction, while x axis the value of the learning rate. The shaded area depict the 95% confidence intervals of the mean accuracy.

Figure 34: Sensitivity of duration predictability performance to the number of trees in random forests



Note: This plot illustrates the predictability of duration as a function of the tree numbers of Random Forest model for Nflx, across all clock modes. y axis depicts the average out-of-sample R^2 obtained across all 30 days of testing period, for trade duration, while x axis the total number of trees. The shaded area depicts the 95% confidence intervals of the mean R^2 .