How and When are High-Frequency Stock Returns Predictable? [*]

Yacine Aït-Sahalia[†]
Department of Economics
Princeton University and NBER

Jianqing Fan[‡]
Department of ORFE
Princeton University

Lirong Xue
Department of ORFE
Princeton University

Yifeng Zhou
Department of ORFE
Princeton University

This Version: April 27, 2022

Abstract

This paper studies the predictability of ultra high-frequency stock returns and durations to relevant price, volume and transactions events, using machine learning methods. We find that, contrary to low frequency and long horizon returns, where predictability is rare and inconsistent, predictability in high frequency returns and durations is large, systematic and pervasive over short horizons. We identify the relevant predictors and examine what determines the variation in predictability across different stock's own characteristics and market environments. Next, we compute how the predictability improves with the timeliness of the data on a scale of milliseconds, providing a valuation of each millisecond gained. Finally, we simulate the impact of getting an (imperfect) peek at the incoming order flow, a look ahead ability that is often attributed to the fastest high frequency traders, in terms of improving the predictability of the following returns and durations.

Keywords: high-frequency returns, durations, predictability, machine learning, random forests, LASSO, penalized regression.

[*] We are grateful to seminar and conference participants at Columbia University, SoFiE 2021 Summer School, the Econometric Society, and the 2021 ICSA Applied Statistics Symposium, for comments and suggestions.

[†] Address: JRR Building, Princeton, NJ 08544, USA. E-mail address: yacine@princeton.edu.

[‡] Address: Sherred Hall, Princeton, NJ 08544, USA. E-mail address: jqfan@princeton.edu

1. Introduction

Low frequency predictability of asset returns over medium to long horizons has been extensively studied and hotly debated in the literature: classical examples of the two sides of the debate are Fama (1970) and Malkiel (1973) vs. Lo and MacKinlay (2002). To the extent that such predictability is present in the data, the empirical evidence suggests that it is overall relatively small and difficult to pin down, and depends heavily upon the stocks or sectors studied, the predictor variables included, the horizon and time periods considered as well as the methodology employed. Given low signal-to-noise ratios, weak and persistent predictors, and instability of the predictive relations (see, e.g., Timmermann (2018)), it is not surprising that achieving consistent low frequency returns predictability is challenging.

By contrast, we show in this paper that predictability of returns and durations is systematic and pervasive at high frequency over ultra short horizons. Empirically, we find that almost every quantity of interest is strongly and consistently predictable over ultra short horizons, across all stocks and periods in our sample. Specifically, price moves over short periods of time, their direction, magnitude and momentum, and the duration between successive arrivals of both orders and transactions, can be predicted. We show this using machine learning algorithms designed to extract predictability from high dimensional data.

The question of predictability at low frequency has theoretical implications for whether financial markets are informationally efficient, as well as practical implications for asset allocation strategies. Predictability at high frequency, on the other hand, has theoretical implications for the design, operation and regulation of financial markets, as well as practical implications for trading and execution strategies. Low frequency predictability, if and when it is identified, should not be expected to last due to competitive pressures in the asset management industry and investors' learning. By contrast, the technological costs and barriers to entry into high frequency trading and the short shelf life of ultra short horizon predictability make it more likely to persist.

Indirect evidence in favor of the presence of high frequency predictability comes in the form of the growth and large and consistent profitability[1] of high frequency trading firms over the past fifiteen years. It is implausible that such extraordinary profitability can be achieved simply by collecting compensation for intermediation services or trading profits, or as a reward for speed alone, in the absence of some ability to predict the short term direction of the order flow, price movements, or both (see Baron et al. (2019) and Aït-Sahalia and Brunetti (2020)).

[1] For example, the high-frequency trading firm Virtu Financial reported as part of its IPO filing in 2014 that it had experienced only 1 day of trading losses in 1,238 days.

On the other hand, direct evidence in the literature regarding the extent, scope and pervasiveness of high frequency predictability is less developed than at low frequency. Huang and Stoll (1994) use a two-equation econometric model of quote revisions and transaction returns to make short horizon returns predictions. Alvim et al. (2010) shows that the daily trading volume is predictable using high-frequency data. Zheng et al. (2013), Kercheval and Zhang (2015), Tsantekidis et al. (2017) and Ntakaris et al. (2018) show that information obtained from the limit order book can predict the direction of the next price move and crossings. Panayi et al. (2018) show that liquidity demand throughout the trading day is predictable and far from uniformly distributed. Knoll et al. (2019) show that trading signals extracted from Twitter data can predict returns. Sirignano (2019) shows that the liquidity sitting deep into the order book can be predicted. Chinco et al. (2019) show that one-minute-ahead returns can be forecast using the entire cross-section of lagged returns and identify as predictors stocks with news about fundamentals.

Closest to this paper is Easley et al. (2021), who show that some microstructure-based measures are useful for predicting six variables that are indicative of different aspects of market quality: the bid-ask spread, realized volatility, normality, skewness, kurtosis and serial correlation of returns. Different from Easley et al. (2021), we examine the extent to which direct aspects of the next transaction or group of transactions (such as their direction, size and price) as well as the duration to the next event (such as the next price change, or the next volume traded, or the next number of transactions) can be predicted. Forecasts of price direction over very short periods of time are direct inputs into directional positions either for an aggressive trading strategy or for passive market making. Duration variables are important inputs to order placement and cancellation decisions, especially for liquidity providers whose limit orders may be queued in the book and require an estimate of the time before they reach the front of the line. This is a situation where predictability translates directly into profitability: see, e.g., Aït-Sahalia and Sağlam (2021) for a model of market making at high frequency where the market maker is able to imperfectly predict the direction and aggressivity of incoming orders, and Dixon (2018) for a method to assign the expected profit and loss of a market maker order under execution constraints to correct and incorrect predictions.

Besides quantifying the predictability that is present in the data, and how fast it dissipates, we seek to determine which predictor variables are most informative for the next trade and durations. We use only widely available trade and quotes data. For both directional traders and market makers, one of the main source of data that is relevant for the decision to send in a trade order, place or withdraw quotes on these very short time scales – where fundamental information about the asset being traded is unlikely to have changed – are variables that can be exclusively inferred from recent trades, as well as the current state and recent evolution of the limit order book. By construction, the information set we base our predictions on is a subset of the information set available to market participants, who might conceivably be able in real time to extract further information from observing the asynchronously incoming order flow on a variety of linked exchanges where they may have posted orders. As a result, the amount of predictability we identify using merely transactions and quotes data, although already quite strong, is a lower bound on the amount that is in principle achievable by a fast and technologically up to date market participant.

The types of machine-leaning methods we employ in this context are different from probabilistic or statistical approaches that model the limit order book (see e.g., Cont et al. (2010), Roşu (2009)), which first often rely on specific functional form assumptions, and second are not primarily designed to produce out-of-sample forecasts. In any event, one of the findings in this paper is that variables derived from the state and evolution of the limit order book tend to be less useful for predicting future returns than variables derived from the transactions record. On the other hand, machine learning methods are not designed to estimate a model in an econometric setting, but are well suited to generating predictions (see, e.g., Mullainathan and Spiess (2017)). Among the machine learning methods we consider, such as neural networks and random forests, the specific method employed generally makes little difference to the outcome provided the methods are trained to the same data and fine-tuned to a comparable degree.

More specifically, we study the predictability over ultra short horizons of transaction returns, directions, and durations to specific price, volume or transactions events. Each problem is tested with different machine learning methods over different time horizons and clocks, including calendar, trade, and volume clocks. We use the complete transactions and quote data for the 101 stocks that were constituents of the S&P 100 index over the January 2019 - December 2020 period. In typical machine learning fashion, we construct large numbers of data features using microstructure and other measures, use various lags, as well as combinations thereof. We find that out of sample predictability exists universally in all the stocks examined, and all the time periods, including the highly volatile environment of March and April 2020. With minimal algorithm tuning, for the median stock in the sample, a 10.5% out-of-sample R^2 for predicting 5-second returns can be achieved using merely past trade and quote data and an accuracy of 64% for predicting the direction of the next trade. When predicting the duration till the next 10 trades, the median out-of-sample R^2 is 9.8%. For return and direction predictions, important predictors include the imbalance in the limit order book, recent transaction imbalance, and the past trade returns, while statistics derived from recent trade volume are most effective for duration predictions.

Second, we investigate whether any differences in predictability can be explained by stock-level characteristics and the market environment. Determinants of predictability, including stocks' own characteristics combined with cross-sectional and time-series factors, are studied using panel regressions. Returns and trade direction are found to be more predictable for stocks that have smaller nominal share prices, that are less liquid, less volatile, and less correlated with the aggregate market. These aspects, together with fixed effects for dates and stocks, explain nearly three quarters of the variability of 5-second return R^2s. By contrast, predictability for durations is higher under liquid and volatile conditions.

Third, we examine how long high frequency predictability lasts for a typical stock in the sample. We find that predictability of returns vanishes to 0 (meaning that a binary directional prediction becomes no better than a coin toss) in just five minutes, and approximately $2,000$ transactions or $2,000$ lots transacted.

Fourth, we quantify the importance of the timeliness of the data. We show that approximately 80% of the overall predictability is achieved by relying on the most recent 10 milliseconds, 10 transactions or 10 lots transacted. We also show that introducing a small artificial delay in processing the data, in the form of a lag, decreases sharply the accuracy of the predictions. We map the decline as a function of the delay. Using this method, we can assign a value to each millisecond of delay, and rationalize the vast investments and sometimes extreme steps undertaken by high frequency market participants to lower their latency.

Fifth, we simulate the effect that acquiring some signal on the direction of the order flow would have for the accuracy of the predictions. The idea here is to model a high frequency trader who, through order placement in different exchanges, is able to (imperfectly) infer the direction of the next trade as suggested in, e.g., Lewis (2015). We ask how much such information would improve forecasts of the next price move, as a function of the accuracy of the directional signal. Such ability to "look ahead" at the incoming flow, even limited to an imperfect sign prediction, is able to boost 5-second return R^2 from 14.0% up to 27.1% and the price direction accuracy from 68.3% up to 79.0%. Whether it is realistic or not to endow some high frequency traders with such an ability, it is clearly valuable.

Finally, we check the robustness of the results to using different machine learning methods and tuning algorithms, determine how predictability varies at different times of day, the relative value of the two subtypes of data (trades and quotes), as well as the incremental value of supplementing the data on a given stock with data from other (correlated) stocks for the different prediction objectives.

The paper is organized as follows. We start in Section 2 by describing the different prediction problems and their corresponding response variables, the construction of the predictor variables, and the criteria we employ to evaluate model performance. Section 3 introduces the data and machine learning approaches we use. Section 4 presents the stock-level results. In Section 5, we turn to the question of determining what explains any differences across stocks and time in the level of predictability we found: we examine different stock characteristics and market conditions, such as differences in aggregate volatility. In Section 6, we study how the timeliness of the data and the ability to look ahead at the sign of the order flow change the predictability results. Section 7 reports the results of a series of robustness checks. Finally, Section 8 concludes this paper.

2. Response and Predictor Variables

The raw data take the form of the complete record of transactions and quote updates, along with their calendar timestamp. In the following, we define more precisely the variables of interest that we construct from the raw data, first the response variables and then the predictors.

2.1 Transactions and Quotes Data

We use only standard, widely-available data to construct our variables. Namely, all the variables are derived from the NYSE's Trade and Quote (TAQ) database for two full years 2019 and 2020. TAQ contains consolidated intraday trades and level-1 quotes (best bids and offers on the market) of all securities listed on the New York Stock Exchange (NYSE), Nasdaq Stock Market (NASDAQ) and American Stock Exchange (AMEX). We restrict attention to the 101 stocks that were constituents of the S&P 100 index on December 31, 2020.[2] Although heavily weighted towards more liquid stocks compared to the full stock universe, this provides a wide sample of stocks across all industry groups. A brief summary of the size of the dataset we use is shown in Table 1.

Tables 2 shows an example of transaction data for Intel Corporation. For a given each date and ticker symbol, a row in the transactions data reports a single transaction. It contains a timestamp of the transaction and its associated price, size and trading direction. for all the stocks we consider, quote and trade prices are reported as multiples of \$0.01.[3] The timestamp is measured in nanoseconds. We follow the usual Lee and Ready (1991) algorithm to infer order direction from the sequence of the trades, and indicate the trade direction as $+1$ if it is a buy-initiated trade and -1 if it is a sell-initiated trade.

[2] There are 101 securities listed in S&P 100 Index as of Dec 31, 2020. We removed GOOGL and only kept GOOG as they are largely similar. Among the remaining 100 securities, DD DOW and RTX do not have full price history throughout 2019 and 2020 due to corporate mergers and spinoffs. We added RTN as the predecessor of RTX before the merger.

[3] The NYSE requires that "the minimum price variation (MPV) for quoting and entry of orders in securities traded on the NYSE Arca Marketplace is \$0.01, with the exception of securities that are priced less than \$1.00 for which the MPV for quoting and entry of orders is \$0.0001."

A snapshot of the quote update data is illustrated in Table 3. Each row in the quote data corresponds to the NBBO at a certain timestamp. The third line of quote update is likely caused by the fourth transaction shown in Table 2. O'Hara et al. (2014) report possible issues with the lack of records of odd-lot trades when TAQ only recorded round-lot trades; TAQ started to include odd-lot trades since 2014, as we can see in Table 3. The quotes are still round-lot but this should have only a minimal impact on our response variables.

The transaction and quote update data are then merged based on their timestamps. The best bid and ask information of trades are determined by the most recent quote information as of the time of the transaction. We remove all observations outside normal trading hours so every timestamp is within 9:30:00 and 16:00:00 (strictly). Both transaction data and quote update data are used in tuning, training and testing of the algorithms.

Summary statistics of our data and response variables are reported in Table 4. The table is divided into two parts. The upper panel contains a summary at the daily level (505 trading days) aggregated across all 101 stocks, with each stock contributing one observation each day it is available. The total market capitalization, nominal stock price, and daily returns are all based on daily closing prices. The daily beta of the stock is estimating by regressing the stock's 15 second returns on the 15 second returns of SPY (an ETF tracking the S&P500 index) and R^2 is the coefficient of determination in this regression.[4] The turnover rate is the ratio of traded volume multiplied by the daily closing price over market capitalization. The lower panel of the table contains summary statistics for the response variables computed over all nanonsecond-level timestamps and all stocks. We downsample the data so that each stock and day are approximately equally represented. For each pair of stock and date, $1,000$ response variables are sampled and the data from all $50,273$ such pairs are merged to compute the summary statistics. We can observe that the high-frequency returns are largely symmetrically distributed but with very heavy tails (large kurtosis). The duration variables are both skewed and heavy-tailed.

2.2 Response Variables

The response or dependent variables we investigate are transaction returns and direction, and the duration to relevant market events such as the arrival of the next group of trades, the next volume quantity, or the time to the next price change.

[4] Detailed definitions can be found in Section 5.4.

2.2.1 Time clocks

Duration quantities can be measured using three different time clocks: a calendar clock (corresponding to the usual measurement of time), a transaction clock (where time is measured in terms of the number of transactions that have taken place, irrespectively of their size) and a volume clock (where time is measured in terms of the total volume transacted, irrespectively of the number of transactions). We examine predictability in all three clocks.

The variables that follow are all defined for each stock individually. All the trades and quotes for a given stock are uniquely labeled by the timestamp of their occurrence in calendar time $t \in \mathbb{R}^+$. Here t represents the number of seconds since the beginning of the day, with precision up to a nanosecond (10^{-9}). For example, $t = 35493.132273873$ represents calendar time 9:51:33.132273873. We denote the number of shares traded at t as V_t, with $V_t = 0$ indicating no trade at t. Thus, $\mathbb{1}_{\{V_t>0\}}$ indicates whether a transaction has taken place at time t. The calendar time interval between two timestamps $T_1, T_2 \in \mathbb{R}$ can be defined as the following half open half closed interval

$$\operatorname{Int}(T_1, T_2) = \{t \in \mathbb{R} : T_1 < t \leq T_2\}. \tag{2.1}$$

For convenience and brevity of exposition, we extend the notion of interval to work for any of the three time clocks, in order to use a unified interval definition when considering each response variables. With a starting timestamp T, a span $\Delta > 0$ and a clock mode $M \in \{\text{calendar}, \text{transaction}, \text{volume}\}$, we define the forward looking time interval as

$$\operatorname{Int}^{\text{forward}}(T, \Delta, \text{M}) = \begin{cases} \operatorname{Int}(T, T+\Delta) & \text{if } M = \text{calendar} \\ \left\{t > T : \left(\sum_{s\in\operatorname{Int}(T,t)} \mathbb{1}_{\{V_s>0\}}\right) \leq \Delta\right\} & \text{if } M = \text{transaction} \\ \left\{t > T : \left(\sum_{s\in\operatorname{Int}(T,t)} V_s\right) \leq \Delta\right\} & \text{if } M = \text{volume} \end{cases}. \tag{2.2}$$

Under the calendar clock, Δ measures the target time horizon over which the prediction takes place. Under the transaction clock, Δ measures the target number of transactions: the interval contains all the time stamps after T such that the total number of trades after T is no more than Δ. Under the volume clock, Δ measures the target volume: the interval contains all the time stamps such that the total volume traded after T is no larger than Δ. The interval is defined as a set of consecutive timestamps and will be used to define relevant quantities such as the average return and duration over that interval. The interval does not contain the starting timestamp T.

Let $\mathbf{D}^{\text{txn}}$ denote the set of all record timestamps $t \in \mathbb{R}^+$ corresponding to trade transactions and $\mathbf{D}^{\text{qt}}$ its quote counterpart. Set $\mathbf{D} = \mathbf{D}^{\text{txn}} \cup \mathbf{D}^{\text{qt}}$. The National Best Bid and Offer (NBBO) prices are indexed by $t \in \mathbf{D}$ and denoted as $\left(P_t^b, P_t^a\right)$ where P_t^b is the best bid price and P_t^a the best ask price. The mid-price is their simple average

$$P_t = \frac{P_t^b + P_t^a}{2}. \tag{2.3}$$

Denote by P_t^{txn} the transacted price if $t \in \mathbf{D}^{\text{txn}}$. The best bid and ask sizes are denoted S_t^b and S_t^a respectively for the record indexed by t.

2.2.2 Transaction return

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

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

This quantity measures the average return from transactions in a forward window, typically over a very short horizon (Δ small). Compared with the return from a single transaction or at a fixed point in time, this definition leads to a less volatile response variable and is more indicative of the aggregated trading behavior over the span Δ instead of the information at an arbitrary point in the near future. From the perspective of a market maker with quotes in the limit order book, predicting (2.4) is more relevant since, unless the quotes are at the front of the queue, the market maker cannot be certain of the exact execution timeframe, but can reasonably infer that it will take place over a short horizon depending upon the current state of the book and market activity. cannot be certain that their quotes will be the next ones.

2.2.3 Price direction

Next, we are interested in predicting whether the next (set of) price movements will be up or down. Let $\bar{R}(\Delta, M)$ be the average past transaction return of the stock. At time $T \in \mathbf{D}$, with horizon Δ and time clock M, the trade direction is defined as

$$\text{Direction}(T, \Delta, M) = \mathbb{1}_{\left\{\text{Return}(T, \Delta, M) \ > \ \bar{R}(\Delta, M)\right\}}. \tag{2.5}$$

This variable further normalizes the transaction return by making it a binary variable, regularizing outliers and tail behavior, and potentially facilitating the prediction, but at the cost of losing the information contained in the magnitude in the return. Nevertheless, direction is by itself a very important variable and an accurate prediction can lead to substantial profitability. Here the return in the recent past is employed to normalize any local short term trend that may be present. However, the normalizing number $\bar{R}(\Delta, M)$ is very close to zero since the time horizon is very short.

2.2.4 Transaction duration

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

$$\text{Duration}(T, \Delta, M) = \text{argmax}_{t \in \mathbf{D}} \left\{ t \in \text{Int}^{\text{forward}}(T, \Delta, \text{M}) \right\} - T. \tag{2.6}$$

This measures the amount of (calendar) time it takes to record either Δ transactions or Δ shares: by definition, Duration(T, Δ,calendar) = Δ so the measurement is only relevant when M is either transaction or volume. The transaction duration variable measures trading intensity; predicting it is an input for order execution strategies as well as quote placement or cancellation strategies. For example, a trader seeking to place an order for a quantity that is not small relative to Δ might place a partial order and keep the rest in reserve. Predicting duration could help setting the rest of that trader's execution schedule. Similarly, market makers who do not wish to see their quotes hit (for inventory or other reasons) might want to cancel them before they rise near the front of the queue; predicting duration helps determine how fast quotes need to be cancelled.

2.3 Predictor Variables

To examine how well the response variables just described can be predicted, we consider a large number of predictor (or independent) variables representing a broad range of features indicative of the short-term trading environment for a particular stock. For now, we use only a given stock's variables to make predictions about that stock. In practice, it is plausible that predictions could be improved even further by exploiting sector or industry-level correlation patterns to make predictions about a stock from preceding observations about another stock with correlated patterns, in what is often called " statistical arbitrage." We quantify the incremental predictability for a given stock that can be achieved by using correlated stocks' data in Section 7.5.

Just like the response variables, the predictor variables we use can all be constructed using exclusively the complete record of time-stamped transactions and quote updates. They take the form of (possibly nonlinear) transformations of the transactions and quote data recorded before the time T at which the prediction takes place.

For each variable, an interesting question lies in determining the length of the lookback window that precedes T. We use a large set of disjoint such windows covering the most recent records (on a scale of seconds) all the way to far back in time (on a scale of hours) so as not to prejudge the results in favor of a short lookback window. Furthermore, there is no reason to assume that the length of the most informative lookback window should be the same for each predictor variable. And, in principle, predictors derived under one time clock can be useful for predicting a response variable measured under a different time clock. So there is a wide range of possible combinations, making machine learning algorithms (as opposed to traditional forecasting methods) well suited to the problem.

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

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

For each timestamp T and clock mode M, we set the lookback spans (Δ_1, Δ_2) to create different features at T with multiple disjoint intervals. For the calendar clock (M = calendar), nine look-back windows are used: $(\Delta_1, \Delta_2) \in \{(0, .1), (.1, .2), (.2, .4), \ldots, (12.8, 25.6)\}$ with number of seconds as the unit; for the transaction clock (M =transaction), the 9 spans are $(\Delta_1, \Delta_2) \in \{(0, 1), (1, 2), (2, 4), \ldots, (128, 256)\}$ using number of transactions as a unit. Similarly, the spans for the volume clock (M =volume) are set as $(\Delta_1, \Delta_2) \in \{(0, 100), (100, 200), (200, 400), \ldots, (12800, 25600)\}$, with units in number of shares, where 1 lot consists of 100 shares.

We consider 13 main predictors, each being implemented over the 9 time spans and 3 time clocks. Furthermore, the predictors are allowed to interact in multiple nonlinear fashion, selected by the machine as part of the prediction algorithm. The 13 predictors can be grouped into 3 categories, which we now describe.

Volume and duration The first group of predictors are related to a stock's trading intensity. One might expect for example that a higher than normal occurrence of block trades or very frequent transactions may persist over short horizons and therefore have predictive power.

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

$$\text{Breadth}(T, \Delta_1, \Delta_2, M) = \left| \mathbf{D}^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, \text{M}) \right|. \tag{2.8}$$

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

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

3. *VolumeAll* is the total number of shares transacted in the interval:[5]

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

4. *VolumeAvg* is the average number of shares transacted for each transaction in the interval:

$$\text{VolumeAvg}(T, \Delta_1, \Delta_2, M) = \frac{\text{VolumeAll}(T, \Delta_1, \Delta_2, M)}{\text{Breadth}(T, \Delta_1, \Delta_2, M)}. \tag{2.11}$$

5. *VolumeMax* is the maximum number of shares transacted in one transaction in the interval:

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

Return and imbalance The second group of predictors are related to the stock's recent trading asymmetry. For example, if a majority of trades are buy trades that hit the limit sell, or the bid is dominating the ask in the level 1 quotes, then we might expect to see an upward pressure on the price. It is natural to expect that an element in predicting future returns and durations will be characteristics of the current limit order book (LOB), including any imbalances; such imbalances are known to be indicative of future price movements (see, e.g., Cont et al. (2014) and Kercheval and Zhang (2015)). We define the following variables:

1. *Lambda* is the price change in the interval relative to total volume. Let $\mathbf{I} = \mathbf{D}^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, \text{M})$, then

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

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

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

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

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

4. *PastReturn* is the past return in the interval. It is defined similarly to the transaction return response, except over a lookback window. Let $\mathbf{I} = \mathbf{D}^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, \text{M})$, then[6]

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

[5] The transaction volume is set to $V_t = 0$ for non-trading events. Thus, the definition is identical to $\sum_{t\mathbf{D}^{\text{txn}} \cap \in \text{Int}^{\text{back}}(T,\Delta_1,\Delta_2,\text{M})} V_t$.

[6] The past return could be defined as $P_{\min(\mathbf{I})}$. We use $P_{\max(\mathbf{I})}$ to retain the same denominator as the future returns.

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

1. *Turnover* is the speed of transactions in relation to the stock's total number of shares outstanding (denoted as S).

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

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

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

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

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

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

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

3. Machine Leaning Methods

3.1 Models

We now briefly describe the two main methods that we use to predict stock returns and durations: regularized or penalized linear regression (in the form of least absolute shrinkage and selection operator or LASSO) as a representative parametric method, and random forests (RF) as a representative nonparametric one. In Section 7.1 below, we perform a horse race across a large number of methods, including the two already mentioned and ordinary least squares (OLS), ridge regression, FarmPredict linear regression, and gradient boosted trees (GBT). More details containing these various methods can be found in Hastie et al. (2009) and Fan et al. (2020b).

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 $\mathbf{Y} = (y_1, y_2, \cdots, y_n)^T$. In our data, each $\mathbf{X}_i$ has dimension 117, consisting of 9 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.

3.1.1 Penalized linear regression

One of the simplest methods for predicting response variable Y based on covariates $\mathbf{X}$ is the linear model

$$Y = \boldsymbol{\beta}^T \mathbf{X} + \varepsilon.$$

In the absence of some form of regularization, standard OLS in a large dimensional setting is likely to have poor out of sample predictive power due to in sample overfitting. A standard method to address this issue consists in regularizing the model using a penalty function applied to normalized variables. Penalized least-squares with an L_1 penalty is known as the least absolute shrinkage and selection operator (LASSO). Specifically, let $\bar{\mathbf{X}} = \frac{1}{n}\sum_i \mathbf{X}_i$ and $s_i = \sqrt{\frac{1}{n}\sum_i (x_i - \bar{x}_i)^2} (i = 1, 2, \cdots, p)$ be the sample mean vector and the sample standard deviations of predictor variables. Let $\bar{Y} = \frac{1}{n}\sum_{i=1}^n Y_i$ be the mean of the response variable. Define the centered response $\widetilde{Y_i} = Y_i - \bar{Y}$ and standardized predictors $\widetilde{\mathbf{X}}_i = \mathrm{diag}(s_1^{-1}, s_2^{-1}, \ldots, s_p^{-1})(\mathbf{X}_i - \bar{\mathbf{X}})(i = 1, \cdots, n)$. LASSO then fits the centered response on standardized predictors by solving the following optimization problem

$$\widehat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_i \left(\widetilde{Y_i} - \boldsymbol{\beta}^T \widetilde{\mathbf{X}}_i \right)^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}. \tag{3.1}$$

This optimization problem does not admit an analytic solution, but can easily be solved using convex optimization algorithms, coordinate descend methods, least angle regression, among others.[7] For new data $\mathbf{X}_{\text{new}}$, the model will predict its associated response as

$$\widehat{Y}_{\text{new}} = \bar{Y} + \widehat{\boldsymbol{\beta}}^T \widetilde{X}_{\text{new}}, \quad \text{with} \quad \widetilde{X}_{\text{new}} = \mathrm{diag}(s_1^{-1}, s_2^{-1}, \ldots, s_p^{-1})(\mathbf{X}_{\text{new}} - \bar{\mathbf{X}}).$$

LASSO is a simple and useful method that shrinks the coefficients of less useful predictors towards 0. This allows us to rank the relevance of different predictors for each prediction problem. However, being a least-squares method, LASSO is not robust to heavy-tailed data. The model selection property has been established under the irrespresentable conditions by Zhao and Yu (2006) and the prediction risk property of Lasso is established by Bickel et al. (2009). See also Fan et al. (2020a) for model selection consistency for the covariates that admit the factor structure and Chapter 3 of Fan et al. (2020b) for more comprehensive account on high-dimensional model selection and prediction.

[7] We use the coordinate descent algorithm implemented in the *Scikit-learn* software in *Python* to solve (3.1).

As shown in Table 4, the response variables in our studies have heavy tails with very large kurtosises. To mitigate this problem, we clip our training responses at their 5^{th} and 95^{th} percentile to reduce the influence of outliers.[8]

3.1.2 Random forests

Classification and regression trees (CART) are important alternatives to parametric methods like LASSO. CART is a scalable nonparametric learning method that can capture the interactions among predictors and nonlinear relationships. A single tree method is known to be unstable and exhibit less predictive power. This leads us to considering ensemble learning trees (see, e.g., Hastie et al. (2009), Murphy (2014), Fan et al. (2020b)). Specifically, we use random forests. A random forest prediction is an ensemble estimator of many individual randomized decision trees. By averaging the outcomes of many i.i.d sampled decision trees via bootstrap samples, the variance of prediction is reduced and the prediction result becomes more stable.

Random forest is fitted via iteratively growing i.i.d regression trees by drawing bootstrap samples from a given data set. Then, a decision tree is built by greedily and recursively minimizing the mean squared error (MSE) loss: At each split, a random subset of predictors of size s are considered as the candidate variables for data partition. This makes grown trees from bootstrap samples more independent. Specifically, let $\mathbf{N}$ be a bootstrap sample that is used for growing a regression tree. The method first seeks a variable and a location to split the data $\mathbf{N}$. Instead of considering all features, it randomly selects s features with index set $\mathbf{S}$ as candidates. For each feature $k \in \mathbf{S}$ at value x, the method divides the data into two subsets

$$\mathbf{N}^{+}(k,x) = \{i \in \mathbf{N} : X_{i,k} > x\} \quad \text{and} \quad \mathbf{N}^{-}(k,x) = \{i \in \mathbf{N} : X_{i,k} \leq x\},$$

where $X_{i,k}$ is the k^{th} component of the i^{th} sample. Let $\bar{Y}^{+}$ be the average of Y_i in $\mathbf{N}^{+}(k,x)$ and $\bar{Y}^{-}$ be the average of Y_i in $\mathbf{N}^{-}(k,x)$. The splitting node (k,x) is chosen to minimize the mean square errors:

$$\widehat{k}, \widehat{x} = \text{argmin}_{k,x} \left\{ \sum_{i \in \mathbf{N}^{+}(k,x)} (Y_i - \bar{Y}^{+}_{k,x})^2 + \sum_{i \in \mathbf{N}^{-}(k,x)} (Y_i - \bar{Y}^{-}_{k,x})^2 \right\}$$

The rest of the tree is grown recursively. Both subsets of data $\mathbf{N}^{+}(\widehat{k},\widehat{x})$ and $\mathbf{N}^{-}(\widehat{k},\widehat{x})$ are partitioned further using a similar method: randomly choosing a subset of variables of size s as the candidate

[8] Such a method is referred to as Winsorization in statistics. The theoretical foundation of such a method in high-dimensional statistics can be found in Fan et al. (2021). We also tested other clipping thresholds like 1^{st} and 99^{th} percentile as well as 10^{th} and 90^{th} percentiles, and find that the final results are similar. Larger clipping tends to produce more stable results while lower clipping is more likely to overfit.

set to partition the data, selecting a variable and a value to optimally split the data. Recursively repeating this splitting process in each subset will yield a regression tree. The stopping criteria include the maximum rounds of splitting (tree depth) and minimum number of data points in each division (leaf size). Iterating the entire process multiple times yields a family of regression trees.

The above process for growing a regression tree divides the sample space into several rectangles and assigns a constant prediction value for each rectangle. Usually the constant is the training sample mean of Y_i in that node of the tree. Specifically, letting $\{\mathcal{R}_{j,k} : k \in \mathcal{I}_j\}$ denote the disjoint partitions of the j-th regression tree, the prediction function for the j^{th} regression tree is

$$\widehat{f}_j(\mathbf{X}_{\text{new}}) = \sum_{k \in \mathcal{I}_j} \widehat{\beta}_{j,k} \mathbb{1}_{\{\mathbf{X}_{\text{new}} \in \mathcal{R}_{j,k}\}}, \tag{3.2}$$

where $\widehat{\beta}_{j,k}$ is the sample mean of the responses in training data falling in the partition $\mathcal{R}_{j,k}$. In other words, a new predictor $\mathbf{X}_{\text{new}}$ must fall in one of the partitions and the partition average response is used as the prediction by the j^{th} tree. The prediction of a random forest with M trees is the average of their predictions:

$$\widehat{Y}_{\text{new}} = \frac{1}{M} \sum_{j=1}^{M} \widehat{f}_j(\mathbf{X}_{\text{new}}).$$

Random forests are improvements over the bagging method where a number of trees are trained independently with bootstrapped samples, but without the process of the random selection of candidate variables at each node. Predictions from bagging are often highly correlated as the bootstrap samples from the same data set tend to overlap. Random splitting in random forests increases the independence of resulting trees and hence reduces the dependence of the prediction. Therefore, the predictions by random forests often end up with a smaller variance than those based on bagging or a single tree, yielding better predictions.

3.2 Measuring Prediction Accuracy

We now describe how we measure and compare prediction results produced by different methods. For robustness, we use different criteria for the same prediction problem, emphasizing different aspects of the quality of the prediction made. For the prediction of stock returns, we employ both the out-of-sample coefficient of determination R^2 and the directional accuracy as measures of fit. R^2 measures how precisely targets are predicted in a normalized fashion while the directional accuracy focuses on whether the predicted values are on the correct side of the target. Both metrics are applied to the testing dataset. Typically, R^2 will be used when assessing the performance of trade returns and duration predictions, as well as in tuning model hyper-parameters while accuracy will be used for evaluating trade direction predictions.

Out-of-sample R^2 is a widely used criterion for evaluating regression models. It measures the normalized prediction error in the target $\mathbf{Y}$ in comparison with a trivial prediction such as the sample average. For given targets $\mathbf{Y}$ and predictions $\widehat{\mathbf{Y}}$, the out-of-sample R^2 is defined as[9]

$$R^2(\mathbf{Y}, \widehat{\mathbf{Y}}) = 1 - \frac{\sum_i (Y_i - \widehat{Y}_i)^2}{\sum_i (Y_i - \frac{1}{n}\sum_i Y_i)^2}. \tag{3.3}$$

One could use the in-sample average as the trivial predictor. Here, we use out-of-sample average as the baseline forecaster. This is a more stringent (or more difficult to beat) criterion than the in-sample average, as the out-of-sample average is not available during the training. Getting a $R^2 > 0$ means the model produces meaningful results and outperforms the future mean. R^2 computes the relative prediction by using a prediction method with that of a sample average predictor. The measure outputs a value $R^2 \in (-\infty, 1]$ and the larger the score the better the model performance. When $R^2 > 0$, the corresponding method outperforms the sample average estimator. In the limit, $R^2 = 1$ means that the target can be perfectly predicted.

Note that out-of-sample R^2 aggregates the squared errors for testing data with the same weight, which can be influenced by outliers of the prediction errors. Unfortunately, prediction errors follow heavy-tailed distributions, as stocks prices jump frequently, creating large outliers in returns. And, for example, the trading volume varies substantially over time, with disproportionately large orders from time to time. This also means we can experience outliers in the duration variable. These considerations lead us to consider a more robust measurement, the sign accuracy, that is less sensitive to outliers.

We use sign accuracy to measure the results of directional predictions. Let $\mathbf{Y}$ be the target and $\widehat{\mathbf{Y}}$ be the prediction, and consider the accuracy measure:

$$\text{Accuracy}(\mathbf{Y}, \widehat{\mathbf{Y}}) = \frac{1}{n}\sum_i \mathbb{1}_{\left\{\widehat{Y}_i \ \cdot \ Y_i > 0\right\}}. \tag{3.4}$$

This measure calculates the proportion of predicted $\widehat{Y}_i$ that have the same sign (or direction) as the true Y_i. For example, when predicting the sign of an upcoming transaction prediction, let $\mathbf{R}$ denote the vector of actual returns, $\widehat{\mathbf{R}}$ the vector of predicted returns and $\bar{R}$ the average return in training data (usually very close to zero). Then the accuracy of direction prediction is defined as

$$\text{Accuracy}(\mathbf{R}, \widehat{\mathbf{R}}, \bar{R}) = \frac{1}{n}\sum_i \mathbb{1}_{\left\{(\widehat{R}_i - \bar{R})(R_i - \bar{R}) \ > \ 0\right\}}. \tag{3.5}$$

[9] One can use the in-sample average as the trivial predictor. Here, we use out-of-sample average as forecaster. This is a more stringent (or more difficult to beat) criterion than the in-sample average, as the out-of-sample average is not available during the training. Getting a $R^2 > 0$ means the model produces meaningful results and outperforms the future mean.

3.3 Algorithm Tuning and Testing

Each model we employ has a large number of parameters, and is tuned and tested on a rolling window basis. A new model is fitted for every testing day using data from the past 5 days, so only the most recent information is included. Hyper-parameters of each method are also tuned every month (20 trading days) to keep up them to date.

Tuning hyper-parameters Indeed, hyper-parameters controlling each method need to be tuned periodically in order to accommodate possible structural changes over time. For computational efficiency, we tune only on the reduced set of hyper-parameters that matters most to each model. We determine this iteratively. We start with a single stock, INTC (Intel Inc.), to find a range for each parameter. Then we optimize (i.e., tune) over that range separately for each stock. For example, in LASSO we tune the ℓ_1-penalty term λ from 10^6 to 10^{-8}. For random forests, we fix the total number of trees at 100, fix the size of each subsample at $100,0000$ (in order to speed up computations) when fitting a single tree. The regression trees are then grown greedily by minimizing the total mean squared error. The depth of each tree is tuned from a range of 3 to 7.

We have also experimented with other choices of hyper-parameters such as the number of days in training a model, a wider range of values of λ, and more depth of trees. The results we obtain are quite robust.

Tuning, training and testing windows The rolling window structure is set to ensure that all testings are done with the most up to date model and data. As noted, the models used each testing day are fitted with the most recent 5 trading day's data and all hyper parameters are re-tuned every 20 testing days. All experiments are conducted on a two-layer rolling window basis, corresponding to the training and tuning. An example of one such window is shown in Figure 1. The length of each training window is also set at 5 trading days. The outer layer of rolling window consists of 40 trading days where the first 20 days are used only for tuning hyper-parameters while the next 20 days are used for testing. More specifically, for the current time T that represents a window of 40 trading days $\{T, T+1, \cdots, T+39\}$, the procedure is implemented as follow:

1. *Learning*: For each combination of hyper-parameters and $t = T, T+5, T+10$, fit a model with data from day t to day $t+4$ (5 trading days), evaluate it on the next 5-day interval $[t+5, t+9]$ and calculate the out-of-sample R^2 for each testing day, which results in $R^2_{t+5}, \cdots, R^2_{t+9}$.

2. *Tuning*: Choose the combination of hyper-parameters that produces the largest average $\frac{1}{15}\sum_{t=T+5}^{T+19} R^2_t$ and fix the hyper-parameters for the predictions in step 3.

3. *Predicting*: For each $t = T + 20, \ldots, T + 39$, fit a model with data from $(t - 5, \ldots, t - 1)$ and use it to predict on day t. Save each prediction result.

4. Roll forward the entire window by 20 trading days, namely, $T \to T + 20$ and repeat steps $1 - 4$.

4. Predictability Results for Individual Stocks

In this section, we report the predictability results for all S&P 100 stocks over the 505 trading days in the 2019-2020 period. As described in Section 2.3, we use 13 variables defined there over 9 different time windows, resulting in 117 variables, that the machine learning algorithms are then free to combine and make interact as they choose. The variables that have important predictive power for specific objectives are also identified by LASSO.

4.1 Transaction Return Prediction

We start by examining our ability to predict future stock returns over horizons of 5 and 30 seconds in calendar time, 10 and 200 transactions and 1,000 and 20,000 shares, using LASSO and random forests. The prediction performance for individual stocks, as measured by out-of-sample R^2 averaged over the 505 days in the sample, is reported in Figure 2 in the form of boxplots summarizing the distribution of R^2 over the S&P 100 stocks. Recall from (3.3) that $R^2 = 0$ is the benchmark where predicability using an algorithm is no better than using the sample short horizon average return to predict the future return.

The main result here is that the algorithms, even with minimal tuning, largely outperform the benchmark. The median out-of-sample R^2 for predicting 5-second returns is approximately 10% and furthermore $R^2 > 0$ for every single stock, including the least predictable. The prediction results are slightly better using RF than LASSO. In addition, as expected, 30-second returns are harder to predict than 5-second ones, with a median R^2 of approximately 4%, but the returns of every single stock remain predictable. The results in trade and volume clocks are similar and consistent: strong predictability over the shorter horizon that gets weaker as the horizon increases.

We next seek to determine which variables are most responsible for the predictability. For this purpose, we use LASSO variable selection and measure the importance of a variable in two ways: First, the frequency with which a variable is being selected (across stocks and days) and, second, the size of the average regression coefficients over all LASSO regressions across stocks and days (recall that variables are standardized before LASSO is applied). Figure 3 shows the results for predicting 5-second returns. Both measures give reasonably consistent results concerning the ranking of variables.

The top two predictors are the transaction imbalance (TxnImbalance) and the past short horizon average return (PastReturn). Interestingly, both are derived from the transactions rather than the quotes data, and capture some form of trade momentum. Both coefficients have the expected sign: for example, if there are many buy trades happening in a the short look-back window, this trend is likely to persist for a short while and push the price upward. The next source of signal from the LASSO ranking comes from the quote data, in the form of the imbalance between the bid and ask side in the limit order book (LobImbalance). The coefficient also has the expected sign, namely that a bid size dominating the ask size predicts upward pressure on the price.

Another interesting observation is that the most informative predictors are constructed by using the most recent past data. The strongest signals always come from the most recent window, as we can see from the magnitude of the coefficients for the top three predictors, which are constructed over the most recent 0.1 second (below, we will explore how fast the predictability deteriorates when the algorithms are precluded from exploiting the most recent past data.)

The consistency of the selection of each variable is examined in Figure 4, where we plot the frequency with which a given variable is selected by the LASSO model, for the three time clocks and two time spans. Here, we aggregate the presence of a variable in the LASSO model at the group level (among a variable measured in 9 different time windows, as long as one of them is in the LASSO model, that group of variables is counted). The results reveal that variables in the group TxnImbalance, PastReturn, LobImbalance are almost always used, whereas the variables VolumeAll, VolumeAvg, VolumeMax are consistently not predictive in different models.

4.2 Price Direction Prediction

We now study the predictability of the return direction: is the price going up or down over the very short term? This basically ignores the magnitude of the return prediction and counts only the sign of the prediction, using the same LASSO and RF trained models. For reasons already discussed, this assessment of the predictions is less sensitive to outliers. Figure 5 shows the results in the form of the percentage of time the algorithm predicts the sign of the upcoming return correctly. The benchmark from a coin toss is 50%. The prediction accuracy is approximately 64% for returns over short horizons – the next 5 seconds, 10 trades or 1,000 shares transacted. Similar to the patterns in return predictions, the accuracy decreases as the time horizon increases. However, compared to Figure 2, there are some differences shown in the figure. First, LASSO has substantially the same performance as random forests. Second, the result is more robust (or less dispersed) as can be seen from the interquartile range and fewer scattered outliers in the boxplots.

Overall, the conclusion remains that stock returns over short horizons are highly predictable using these algorithms. Given the ultra short horizons over which it applies, such predictability can in principle translate into profitabilty since a high frequency trader could trade thousands a time a day, replicated over a whole array of stocks, while achieving an expected success rate of 60%. By the law of large numbers, as the number of such trades increases, the fast trader is likely to come increasingly close to achieving this success rate in reality, resulting in consistent profitability such as that reported in footnote 1.

4.3 Transaction Duration Prediction

We now examine the predictability of trading durations, specifically targetting the amount of time necessary for a certain number of transactions to take place, or a certain volume to be traded. This variable is an important input into many execution strategies as well as price impact models. The methods we use are the same as those used for predicting returns. Figure 6 reports the out-of-sample R^2. Durations are predicted even more accurately than returns, and the predictability actually improves for longer durations (time to wait for 200 transactions to occur or 20,000 shares to change hands vs. time for 10 transactions or 1,000 shares), where we achieve out-of-sample R^2 above 10% with once again $R^2 > 0$ for every single stock.

The predictor variables identified by LASSO as important for duration prediction are however very different than those identified for predicting returns. The results are in Figure 7. The top 20 most significant features are all derived from two families of variables: VolumeMax and VolumeAll. The estimated regression coefficients of all variables of one family have the same signs, and the two families have opposite signs. The magnitude of the coefficient is proportional to the window length of the feature. A large value of VolumeAll indicates that trading activity is intense in the (very) recent past. This is likely continue for a brief moment at least, as a result the duration for a fixed number of shares is more likely to be predicted to be short, hence the negative value of the LASSO coefficients on VolumeAll. On the other hand, a large value of VolumeMax variable indicates that large size (perhaps block) trades are taking place at the moment. These large(r) trades may have more market impact, but tend to be isolated and the LASSO model predicts that they are followed by longer than usual durations resulting in a positive coefficient for VolumeMax. So the two types of volume measurement play two very different roles in terms of forecasting durations. While the two volume variables are consistently the most important, the LASSO models for duration prediction use almost all other variables every time, so the grouped feature usage plots are omitted.

4.4 Prediction Consistency Over Time

To summarize, with little algorithm tuning, we can achieve out-of-sample R^2 for predicting returns and durations between 8 and 10% averaged across stocks and days. Is this performance achieved consistently over time, or is the average out-of-sample R^2 the product of having a few days where accurate predictions compensate for poor predictions the rest of the time? To answer this question, we report the time series of day-by-day out-of-sample R^2, along with the associated standard deviation computed across all stocks in Figure 8. We also report there the times series of results for the direction accuracy measure, and the out-of-sample R^2 for predicting duration. As the figure shows, the results are consistent across the sample, and the predictability measures are significantly positive. The green shaded area on the graph indicates the roughly two month period during the Spring of 2020 when the aggregate stock market dropped precipitously due to the advent of Covid-19. This period is associated with increased volatility, and resulted in a slight decrease in predictability of high frequency returns and more volatility in duration predictions.

Overall, the predictability appears to be quite stable over time. It is useful nevertheless – and this is the object of the next Section – to examine in more detail what factors affect the variation in the amount of predictability both cross-sectionally across stocks and in the time series across different aggregate market environments.

5. Cross-Sectional and Time-Series Determinants of Predictability

In this Section, we study how the predictability we are able to achieve for returns and durations varies across stocks and days, and what variables explain this variation.

5.1 Nominal Share Price Level

The minimum price increment \$0.01 that is necessary to record a non-zero return means that a non-zero return is a larger event for a stock with lower nominal price per share levels, so we should expect such an event to be easier to predict. Stocks with small nominal share prices are traded with tick sizes as large as 5bps or 10bps. And since the bid-ask spread is wide and the gap between the best and second bid / ask prices are also wide, a larger proportion of orders are placed at the best bid and ask, making the estimation of buy and sell pressure from either side more accurate using the level-1 quote data, resulting in better predictions of the short-horizon return and trade direction.

Figure 9 shows a scatterplot of each stock's average daily predictability against its average daily closing share price. The figure shows that the three stocks with the highest predictability in returns, which are Ford (F), General Electric (GE) and Kinder Morgan Inc. (KMI), are also the ones with the lowest nominal share prices. The average daily closing prices in 2019-2020 for F, GE, and KMI are \$8.10, \$9.00 and \$17.60, respectively. We record a remarkably high average daily out-of-sample R^2 of 48%, 39% and 33% when predicting their 5-second returns. By contrast, the majority of stocks in the S&P100 index have a share price around or above \$100 and an average daily out-of-sample R^2 between 5 and 15%..And there is a clear negative relationship between share prices and predictability in return and trade direction.

On the other hand, when predicting duration to the next 10 transactions, predictability increases for stocks with larger nominal share prices. A likely explanation is the larger nominal share price tends to correlate with more liquid trading, which make durations easier to anticipate. Table 5 shows the result of a panel regression of predictability on log nominal share prices across all days and stocks, with days and stocks fixed effects. Each explanatory variable is normalized prior to the panel regression. The regression results confirm that nominal share price is negatively associated with return or direction predictions but positively associated with predictions of the duration of transactions, ever after controlling for additional confounding factors.

5.2 Stock Trading Liquidity

We then examine the effect of the liquidity of individual stocks on their individual predictability. We use two separate measures of liquidity: total traded dollar volume and percentage spread. For a given stock and date, the total traded volume is calculated as the total number of shares traded in a day times the closing price. The percentage spread is calculated as the average of the stock's bid-ask spread divided by its midprice, sampled every 15 seconds throughout the day. It is natural to expect that the markets for more liquid stocks are more efficient in terms of price discovery, with past signals being incorporated faster into prices, which should make the prediction of returns (i.e., future prices)more difficult.

Figure 10 shows the stock-by-stock results, with daily predicatability averaged over the full sample. A clear pattern emerges: better liquidity, in the form of higher volume or lower spreads, leads to weaker predictability in return and trade direction, whereas durations are easier to predict when liquidity is higher. The univariate relationships are confirmed by multivariate panel regressions in Table 6, with fixed effects and controls. For the same reason, market capitalization of the stock is also found to be negatively correlated with predictability in most cases, likely due to its positive relationship with liquidity.

One reason durations are more difficult to predict for less liquid stocks is that there are larger outliers in durations, in the forms of (relatively) long gaps in trading, that negatively impacts the overall predictability. But we note that even for the least predictable stock, durations remain quite predictable with out-of-sample R^2 above 7% despite a limited training set we are providing to the algorithms and no attempt at tuning the algorithms for this specific problem.

5.3 Stock-Level Volatility and Jumps

It is natural to expect that cross-sectional differences in volatility and jump intensity might affect predictability. If the trading conditions of a stock change more rapidly, a model fitted with past data and patterns is less likely to predict well out-of-sample. For a given stock and day, its volatility on that day is calculated as the standard deviation of its mid-price returns computed at 15 second intervals. Every full trading day is 6.5 hours from 9:30 to 16:00 and is divided into 1560 disjoint 15 second intervals. The jump indicators are binary dummies, indicating whether the absolute values securities' daily close-to-close returns fall into the range of $3-4\%$, $4-5\%$ or greater than 5%.

Panel regression results are shown in Table 7. We regress the out-of-sample R^2 or accuracy measure for the three prediction problems on either volatility only or both volatility and jump indicators. The results show that volatility has a negative impact on the predictability of returns but a positive impact on the predictability of duration, with or without the presence of the jump indicators. Without jump adjustment, one standard deviation increase in volatility reduces, on average, the out-of-sample R^2 for 5-second return prediction by 2%. By comparison, volatility has smaller (still negative, but insignificant) impact on the directional accuracy. The presence of jumps (of any of a the three sizes) has a significant negative impact on the predictability. Durations, on the other hand, are more predictable for assets with higher volatility and jumps; higher price variability translates into more trading activity, shorter durations, which are more easily predictable.

5.4 Asset Pricing Characteristics

Next, we examine whether asset pricing characteristics of individual stocks, such as their betas, daily R^2s and daily idiosyncratic volatilities, affect the predictability. We first estimate these quantities using a standard first-pass regression. For a given day, let the vector of 15-second mid-price to mid-price returns of a stock x be $\mathbf{R}_{\mathrm{x}}$. We use SPY (the most liquid exchange traded fund tracking the S&P500 Index).as a proxy for the market portfolio. Then the daily beta of stock x can be calculated as

$$\mathrm{Beta}(\mathrm{x},\mathrm{SPY}) = \mathrm{Cov}(\mathbf{R}_{\mathrm{x}},\mathbf{R}_{\mathrm{SPY}})/\mathrm{Var}(\mathbf{R}_{\mathrm{SPY}}).$$

Its associated R^2 relative to the market portfolio can be calculated as

$$R^2(\mathrm{x},\mathrm{SPY}) = \mathrm{Corr}^2(\mathbf{R}_{\mathrm{x}},\mathbf{R}_{\mathrm{SPY}}) = \frac{\mathrm{Cov}^2(\mathbf{R}_{\mathrm{x}},\mathbf{R}_{\mathrm{SPY}})}{\mathrm{Var}(\mathbf{R}_{\mathrm{x}})\mathrm{Var}(\mathbf{R}_{\mathrm{SPY}})},$$

which measures the percentage of stock x's movements that can be explained by market movements. Stock x's own idiosyncratic volatility on that day is calculated as the standard deviation of the residuals of the regression, that is

$$\text{Idiosyncratic Volatility} = \sqrt{1 - \text{Corr}^2(\mathbf{R}_\text{x}, \mathbf{R}_\text{SPY})}\ \text{SD}(\mathbf{R}_\text{x}).$$

.

Table 8 summarizes the result of the panel regression. Both Beta and R^2 relative to the market portfolio are negatively correlated with return and trade direction predictability, while idiosyncratic volatility follows the same pattern as total volatility in Section 5.3. This is consistent with the earlier finding that the returns of more volatile and more liquid stocks are relatively harder to predict. The former is the case for assets with higher betas, which are more volatile due to higher exposure to systematic risk. (Note that we do not attempt to separately forecast the returns of SPY, which would help in this case.) And stocks with higher R^2 relative to the market portfolio tend to be more liquid, ceteris paribus.

5.5 Market-Wide Environment

Turning to the time series determinant of predictability, we now study whether the market portfolio return and its volatility can affect predictability over time. The close to close returns (proportional change) of the S&P500 index is used to proxy for market returns. The CBOE VIX index is used to proxy for aggregate market volatility.

The panel regression results are shown in Table 9. We include fixed effects for two sets of dates that are expected to have impact on the equity markets. Those are dates when important economic data are released. The first set contains all the dates of the Federal Open Market Committee (FOMC) meetings, when monetary policy decisions are announced. The second set is all dates when the US Bureau of Labor Statistics releases employment data. A fixed effect will be estimated for every one of these dates: There are 14 FOMC dates and 24 employment data dates from 2019 to 2020, resulting in 38 variables.

As expected, VIX, the market volatility measure, has a negative effect on return and direction predictions. This is in line with the expectation: larger market volatility makes things harder to predict. We also find that days when the market return is positive are more easily predictable than when it is negative, a finding consistent with the classical "leverage effect". Indeed, the correlation between market returns and volatility is around -0.7.

5.6 Multivariate Stock-Level Predictability

Finally, we consider all these determinants of predictability together in a comprehensive multivariate panel regression analysis, including stock level fixed effects and date effects. The results for return and duration predictions are shown below and the results for direction prediction can be found in Appendix.

Table 10 shows the regression results for the stock return predictions. We include two regressions: in the first, fixed effects control for all dates while in the second fixed effects control only for the dates with material economic data releases. The impact of most of the explanatory variables is similar to previously identified univariate effect. Also, the results are mostly consistent across different lengths of windows and different time clocks. As a quick summary, the returns and trade directions of less liquid and less volatile securities with smaller nominal share prices and less correlated with the market are on average easier to predict.

Table 11 gives the regression results for the duration predictions with two regressions corresponding to different sets of fixed effects for each problem, as in the previous table. Both liquidity measures and volatility measures have clear positive relationships with duration predictability, while share prices, market capitalization, beta, R^2 relative to the market are negatively related to duration predictability.

6. The Value of a Millisecond

There is a large amount of evidence that some high-frequency trading firms go to extreme lengths to reduce the latency of their interactions with stock exchanges: this includes physically locating as close as possible to the exchange servers, investing in dedicated "over the horizon" or other "straight line" transmission technologies between major financial centers, etc. The most apparent aim is to be able to quickly send and receive messages with the exchange for the purpose of placing or cancelling orders. But in order to decide what these orders should be, a necessary condition is to have up-to-date data coming from the exchanges. Without data, no predictions are possible, and without predictions trading reduces to a sequence of coin tosses.

In this Section, we explore how the amount of predictability varies with the timeliness of the data. We answer several questions related to this. First, how fast does the (fairly large) preditability identified above disappear? Second, how costly, in terms of predictability, would a delay or lag in acquiring and processing the data be? Third, is there additional predictability to be obtained from being able to look ahead, however briefly and imperfectly, at the incoming order flow?

6.1 The Predictability Lifespan

We have quantified in previous sections the predictability of returns, trade direction and durations. Such predictability, especially in returns, should be very short-lived under a competitive and efficient market. Especially since all the data we use are easily and publicly available, any systematic predictability that relies on it should be traded away quickly by others with similar data and models. This is indeed what we find: the predictability is very short lived.

More specifically, for predicting Return(T, Δ, M), we consider horizons Δ =1, 3, 5, 15, 30, 60, 120, 300, 600, 1800 seconds in the calendar clock, Δ =1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000 transactions in the transaction clock and Δ = 0.1K, 0.2K, 0.5K, 1K, 2K, 5K, 10K, 20K, 50K, 100K, 200K, 500K traded volume in the volume clock. The Δs for duration and direction predictions are set similarly. The predictor variables in each time clock are kept the same as the original experiment, and the algorithms are retuned separately for each horizon.

Figure 11 shows the results for each return and direction prediction. We use Intel (INTC) as a typical illustrative example, whose price, traded volume, market capitalization, returns and predictability are all within the middle range of the S&P 100 stocks examined ealier. The x-axis represents different horizons (Δ) and is plotted in log-scale. The shaded area around each point is the 95% confidence interval of INTC's daily predictability, measured in terms of out-of-sample R^2, over 505 days. In each of the tests, predictability decreases as the span increases. In returns predictions, the out-of-sample R^2 may even drop below 0, indicating that the model is unable to predict returns better than the local sample average when the window is long enough (recall that we use the infeasible future sample-average returns as the benchmark to beat, which is challenging.) We find that returns are only predictable over the next 3 minutes, 2,000 transactions or 500K volume. The predictability reaches a plateau in calendar time before 5 seconds before decreasing. In transaction and volume clocks, the predictability first increases and peaks around 20 transactions or 20K in volume. This is possibly due to a bias variance trade-off, since the average return in the benchmark becomes less noisy as more transactions are included, while additional returns included in the averaging become less and less predictable.

The results for duration predictions are shown in Figure 12. The transaction clock duration is more predictable over a longer horizon than a shorter one. This makes sense as the number of transactions becomes more evenly distributed when viewed over longer horizons. By contrast, predictability for volume clock duration is largely stable and slowly decreases as the span increases.

6.2 The Impact of Delays in Acquiring or Processing Data

In all the results above, we have studied the predictability of future returns with the benefit of the most up-to-date data. However, delays in data transmission and then computation are inevitable in real-world situations. What is the impact of such delays on the amount of predictability that can be extracted from the past data? News reports have indicated that high-frequency trading firms spent hundreds of millions of dollars building dedicated microwave towers, just to cut the latency of one-way communication between New York and Chicago by about half a millisecond closer to the theoretical minimum implied by the speed of light. So it is natural to expect that each millisecond is very valuable, or conversely every delay is costly.

A delay can appear in various forms. The newest transaction messages and quote updates sent from exchange servers might take some time to reach the system of a trading algorithm. The system needs time to process the data, makes decisions based on new information and send orders based on them. These are delays in calendar clock. In addition, a limit order might not be on the top of the limit order book when received by the exchange and will need to wait for a few transactions before being executed. Such delays can be modeled in a transaction or volume clock.

We now quantify the predictability decay when a delay in data availability and/or processing is incorporated. More specifically, letting δ denote the delay at time T, we now predict the delayed version of returns as $\text{Return}(T + \delta, \Delta, M)$. All the predictor variables continue to be calculated at time T. Depending on the clock mode, the delay can be a few milliseconds, several transactions or a few lots of traded volume. $\delta = 0$ corresponds to no delay, and reduces to the previous results. We focus attention to the predictions of returns and trade direction for INTC at horizons Δ of 5 seconds, 10 trades or 10 lots traded volume.

The results are shown in Figure 14. We use the same time clock for each delay and prediction problem. The delays δ are 0s, 0.1ms, 0.3ms, 1ms, 3ms, 10ms, 30ms, 0.1s, 0.3s, 1s, 3s and 5s for the calendar clock, 0, 1, 3, 5, 10, 30, 150 trades for the transaction clock, and 0, 1, 3, 5, 10, 30, 150 lots for the volume clock. We find that the predictability decreases monotonically and sharply as the delay increases. Especially for the the calendar clock case, the average out-of-sample R^2 drops from 14% to 2.5% after a delay of 10ms (1% of a second), demonstrating the extreme value of the timeliness of the data on a scale of a few milliseconds. Similar sharp decreases also appear in other problems and other time clocks. The results suggest that the majority of the predictability has its source in the most recent few milliseconds of price and quote movements. Such results provide strong evidence on the value of timely data and partly explains why data latency is so highly valued by high-frequency market participants.

6.3 Peeking into the Future: The Value of Signals on the Direction of the Incoming Order Flow

We just showed that delays in acquiring or processing data are very costly. How about the opposite? What if a trader were able to acquire advance information about some limited characteristics of the order flow, such as only the sign of an incoming order, most likely imperfectly, and with very minimal time to react to it. Such knowledge about the direction of the order flow may come from different sources. For example, the advantage may come from the trader's use of deeper limit order book data, from data on transactions on futures or other related securities, from the trader's ability to interact with the market using quotes posted on other exchanges, having a direct feed to the exchanges that is faster than the publicly available one, etc. Lewis (2015) describes many possibilities. Whether all these explanations are realistic or not, it is interesting to compute how much information of this type might help predictions .

We model this situation as follows. In addition to all the previous predictors, we now add a binary predictor that serves as a noisy estimator of the direction of the incoming order flow, averaged over a span of length Δ. Let $\mathrm{Dir}_t^{\mathrm{LR}}$ be the binary trading direction signed using the Lee and Ready (1991) algorithm at time t and X be a Bernoulli random variable with $\mathbb{P}(X = 1) = p$. $(1 - p)$ represents the probability that the signal is actually correct. A signal of this type is an input to the optimal trading strategy by the market maker in the theoretical model of Aït-Sahalia and Sağlam (2021). The advance signal at time T is the noisy version of average direction of future order flows, defined as

$$\mathrm{FlowDir}(T, \Delta, M, p) = \mathrm{sign}(2X - 1) \cdot \mathrm{sign}\Big(\sum_{t \in \mathbf{D}^{\mathrm{txn}} \cap \mathrm{Int}^{\mathrm{forward}}(T,\Delta,\mathrm{M})} \mathrm{Dir}_t^{\mathrm{LR}} \Big). \tag{6.1}$$

This variable flips randomly the average direction of future trades with probability $(1 - p)$. Hence, the predictor is noiseless at $p = 0$ and turns into pure random noise when $p = 0.5$. Note that *FlowDir* is based solely on the sign of the sum of binary trade directions, without using any knowledge of price or volume information.

We focus on a peeking-ahead span of $\Delta = 5$ seconds in $M =$ calendar clock, the same as the horizon for the returns predictions. To let the algorithms make full use of such information, we allow its interactions with every other predictor already available. This doubles the total number of predictors to 234 from the previous 117. Figure 15 reports the results for various values of p for predicting INTC's 5-second returns and their associated directions. The shaded area indicates the 95% confidence intervals of the mean over 505 days. The results show that including the sign of the average future transaction direction can boost the return predictability from 14% up to 27% and directional accuracy from 68% up to 79%. And, as expected, the predictability decreases monotonically as signals become less informative (p increases). Again, we take no stand on whether traders have in reality the ability to infer the direction of the incoming order flow. What is clear, however, is that such ability is (or would be) very valuable.

7. Robustness Checks

We conclude the analysis with a series of robustness checks to understand how the results might change with respect to some modeling details. First, we examine the impact of the choice of algorithm by including a wider range of prediction methods, such as ridge regression and neural networks. Second, we compute how much additional predictability can be coaxed out of the random forests by increasing the number of trees. Third, we quantify the loss of predictability that would result from limiting the predictors to be statistics derived from the transactions data alone, or the quote data alone, as opposed to being able to use and combine both. Fourth, we check whether there are any intraday seasonality patterns in the amount of predictability at different times of the day.

7.1 Comparison and Consistency of Results Across Prediction Methods

We start by checking the robustness of the prediction results by conducting a horse race against a wider range of prediction methods. For predicting returns and their signs, we include five additional prediction methods: (i) linear methods in the form of ordinary least square linear regression (OLS) and ridge regression; (ii) FarmPredict, which first decomposes observed data into factors and idiosyncratic components and uses them as new predictors in a statistical machine learning method (Fan et al., 2020a); in addition to using the learned factors, we apply LASSO on the idiosyncratic components to potentially enhance the predictive power; (iii) nonlinear methods: gradient boosted trees (Fan et al., 2020b), and feed-forward neural networks using one or two hidden layers.

Results are shown in Figure 16. In a nutshell, all methods have very similar performance, except OLS. The improvements from a nonlinear method like random forest or gradient boosting trees are limited for most cases when compared with LASSO. The poor performance of OLS for several tests is due to heavy overfitting, with predictions that are sensitive to noise and result in very far-fetched predictions on several days.

7.2 Fine-tuning the Number of Trees in a Random Forest

Random forests represent the base nonparametric method we used. In the analysis throughout the paper, we fixed the number of trees in each forest at 100. Since each tree is independently and identically distributed, varying the number of trees should only affect the variance of predictions.

We report results where the number of trees ranges from 1, 2, 4 to 512. The same procedure for tuning the algorithm is used in each case, as we did for the base case of 100 trees. From the results in Figure 17, we can see that the improvement from increasing the number of trees is limited and we observe almost no difference after 16 trees. An important reason for this is that the performance is averaged over 505 days which is already quite stable. Day-by-day performance hower benefits from using additional trees. When more trees are used, tuning selects a deeper depth for each tree as hyper-parameter.

7.3 Predictability Using Only Subtypes of Data: Trades vs. Quotes

We allowed the algorithms so far to use the merged transaction data and quote update data in all the training and testing. This involves two aspects. First, at each time t both data are used in calculating the predictor variables. Specifically, *LobImbalance* and *QuotedSpread* are the only two variables that utilize both data, while other variables are calculated with transaction data only. These two variables can also be calculated with just transaction data. Second, the timestamps t at which we calculate predictor and response variables may come from either a transaction or a quote update event[10]. For INTC in 2019 and 2020, the daily averages of transactions and quote updates are 135K and 630K, respectively. The gap is larger for more liquid assets.

How does limiting the algorithms to using only trade or quote data affects returns predictions? We fit the algorithm with predictors calculated at only timestamps from either trade, quote or both. To mimic the situation where only transaction data are available, we fit models with only variables calculated with trade data on trade timestamps. Results are presented in Table 12. We can see that the predictability at transaction-only timestamps are significantly higher than those on quote update-only timestamps. This is consistent with the variable selection findings from LASSO in Section 4.1 which isolated transactions-derived variables as the most important predictors of future returns. These differences might also be related to the different temporal distributions of timestamps in each group, or the fact that transaction data are inherently less noisy compared to quotes. And models fitted and tested with the same subtypes of data performed slightly better than models where the subtypes are mixed.

[10] Recall that the return at a quote timestamp is defined as the the average transaction price in a time window divided by the mid-quote price at the timestamp. The return at a transaction timestamp is defined similarly except that the mid-quote price at the previous quote timestamp is used as the denominator. Hence, the results depend on whether the timestamps are used or not.

7.4 Incremental Predictability Using Additional Data From Correlated Stocks

The data employed so far to predict a given stock's future returns and durations were derived exclusively from observations on that stock's own past transactions and quotes. We now examine whether additional predictability can be achieved by adding data derived from other stocks. Indeed, correlated stocks do tend to move together but, on a millisecond timescale, correlated moves in different stocks are never perfectly synchrone. Whenever a stock might slightly lead another, data on that stock's transactions and quotes can potentially help predict the laggard stock's moves.

Continuing with INTC as an example, we add to the set of explanatory variables for INTC predictors that are derived from the four stocks most highly correlated with INTC in terms of daily close to close returns from January 2019 to December 2020, among all stocks in the S&P 100. The four stocks selected are TXN, NVDA, MSFT, and QCOM in decreasing correlation order. We then measure the gains in predictability for INTC, if any, by aligning on the basis of their respective timestamps signals from INTC with those from these four stocks.

When using only one additional stock, to keep the number of variables approximately the same, we use the nearest five time spans for each stock ($5 * 2 = 10$ predictors for each crafted feature), instead of the default nine time spans. Similarly, when recruiting two (resp. four) additional stocks, we use three (resp. two) spans from each in order to keep the number of predictors approximately the same.[11] We compute predictability results using a calendar clock as it makes the most sense when predictors from multiples stocks are merged together.

We find that the use of additional stock information leads only to small improvements. The out-of-sample R^2 for predicting 5-second INTC returns improves from 18.07% to 18.26% when TXN is added, and further improved to 18.30% when all five stocks are used. The out-of-sample R^2 for longer 30 second return improves from 6.40% to 6.62% when including TXN, then 6.78% when NVDA is added and dropped to 6.57% when using 4. The results are shown in Table 13.

7.5 Intraday Seasonality: Predictability Across Different Trading Hours

Finally, we examine whether intraday seasonality plays a role in the predictabiliy over a trading session. It is well documented that volatility is typically higher at the beginning and the end of trading days and lower in the middle, and spreads tend to decrease throughout the day. In light of these stylized fact, we divide the trading day into three disjoint time intervals, a 9:30-10:00 opening session, 10:00-15:30 midday session and 15:30-16:00 closing session. For each fraction of the day, we

[11] With two additional stocks on 3 time spans, we have $(2 + 1) * 3 = 9$ predictors for each crafted feature. Similarly, with four additional stocks measured on 2 time spans, we have $(4 + 1) * 2 = 10$ predictors for each crafted feature.

tune, fit and test models using data only from the the same time intervals in past days using the same procedure as before.

The results are shown in Figure 18. Returns tend to be easier to predict in the middle of the day rather than at the beginning of the day, possibly owing to the large moves typically concentrated during the overnight hours, while durations are similarly predictable. Interestingly, predictability at the end of day is higher for all of the prediction problems we considered, despite the higher volatility, suggesting that the trading patterns at the end of the day are more consistent across days.

8. Conclusion

We studied three basic prediction problems over ultra short high-frequency horizons, namely predictions of returns, transaction direction and durations. We examined them in three time clocks using primarily LASSO and random forests, and relying on the complete transaction and quote update data of the S&P 100 stocks over two full years from 2019 to 2020.

We quantified the predictability and found that fairly large amounts of predictability exist universally in every stock and does so consistently over time, a situation very different from long horizon low frequency predictability. We also isolated the variables that were more reponsible for driving this predictability: they include limit order book imbalances, transaction imbalances and past returns for return and direction predictions, and volume statistics for duration predictions.

We the investigated how the predictability depends on stock characteristics and market environments: for return or direction predictions, securities with less liquid, less volatile, smaller nominal share prices, and those weakly correlated with the market are more predictable. By comparison, predictability for duration is higher under liquid and volatile conditions.

We examined next the value of the timeliness of the data and found that predictability of returns and trading directions vanishes quickly as soon as the time span is larger than a few minutes, a few thousands trades or a few thousands lots traded. In addition, the majority of the predictability lies in the most up-to-date few milliseconds, or a few trades, and decreases quickly when a small delay is introduced. We also simulated the possible ability of high-frequency traders in terms of making short-term imperfect predictions on the sign of the incoming order flow. Such ability turned out to be very useful in prediction, almost doubling the out-of-sample R^2 in 5-second return predictions.

Finally, we found that other machine learning algorithms, such as ridge regression, gradient boosted trees or neural networks produce substantially the same results. OLS however does not identify any preditability, highlighting the value of the penalization for overfitting that characterizes most machine learning algorithms. Fine-tuning a given algorithm, such as a increasing the number of trees in random forests, reaches its limits beyond a certain point. If one has to choose only one subtype of data, transactions data are more valuable that quote data in terms of predicting returns. Incorporating data from additional correlated stocks do not have a large impact on the predictability of one stock: its own transactions and quotes are by far the most informative. Finally, there exists some intraday variability in the amount of predictability, with the end of the day subperiod being the most predictable.

References

Aït-Sahalia, Y., Brunetti, C., 2020. High frequency traders and the price process. Journal of Econometrics 217, 20–45.

Aït-Sahalia, Y., Sağlam, M., 2021. High frequency market making: The role of speed. Tech. rep., Princeton University.

Alvim, L. G., dos Santos, C. N., Milidiu, R. L., 2010. Daily volume forecasting using high frequency predictors. Tech. rep., Pontificia Universidade Catolica do Rio de Janeiro.

Baron, M., Brogaard, J. A., Hagströmer, B., Kirilenko, A., 2019. Risk and return in high frequency trading. Journal of Financial and Quantitative Analysis 54, 993–1024.

Bickel, P. J., Ritov, Y., Tsybakov, A. B., 2009. Simultaneous analysis of lasso and dantzig selector. The Annals of Statistics 37 (4), 1705–1732.

Chinco, A., Clark-Joseph, A. D., Ye, M., 2019. Sparse signals in the cross-section of returns. The Journal of Finance 74, 449–492.

Cont, R., Kukanov, A., Stoikov, S., 2014. The price impact of order book events. Journal of Financial Econometrics 12, 47–88.

Cont, R., Stoikov, S., Talreja, R., 2010. A stochastic model for order book dynamics. Operations Research 58, 549–563.

Dixon, M. F., 2018. A high-frequency trade execution model for supervised learning. High Frequency 1, 35–52.

Easley, D., de Prado, M. L., O'Hara, M., Zhang, Z., 2021. Microstructure in the machine age. Review of Financial Studies 34, 3316–3363.

Fama, E. F., 1970. Efficient capital markets: A review of theory and empirical work. The Journal of Finance 25, 34–105.

Fan, J., Ke, Y., Wang, K., 2020a. Factor-adjusted regularized model selection. Journal of Econometrics 216, 71–85.

Fan, J., Li, R., Zhang, C.-H., Zou, H., 2020b. Statistical Foundations of Data Science. Chapman and Hall/CRC.

Fan, J., Wang, W., Zhu, Z., 2021. A shrinkage principle for heavy-tailed data: High-dimensional robust low-rank matrix recovery. Annals of Statistics 49, 1239–1266.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition. Springer Series in Statistics. Springer-Verlag, New York.

Huang, R. D., Stoll, H. R., 1994. Market microstructure and stock return predictions. Review of Financial Studies 7, 179–213.

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

Knoll, J., Stübinger, J., Grottke, M., 2019. Exploiting social media with higher-order factorization machines: Statistical arbitrage on high-frequency data of the S&P 500. Quantitative Finance 19, 571–585.

Lee, C. M., Ready, M. J., 1991. Inferring trade direction from intraday data. The Journal of Finance 46, 733–746.

Lewis, M., 2015. Flash Boys: A Wall Street Revolt. W. W. Norton & Company.

Lo, A. W., MacKinlay, A. C., 2002. A Non-Random Walk Down Wall Street. Princeton University Press.

Malkiel, B. G., 1973. A Random Walk Down Wall Street. W. W. Norton & Company.

Mullainathan, S., Spiess, J., 2017. Machine learning: An applied econometric approach. Journal of Economic Perspectives 31, 87–106.

Murphy, K. P., 2014. Machine Learning: A Probabilistic Perspective. MIT Press.

Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., Iosifidis, A., 2018. Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. Journal of Forecasting 37, 852–866.

O'Hara, M., Yao, C., Ye, M., 2014. What's not there: Odd lots and market data. The Journal of Finance 69, 2199–2236.

Panayi, E., Peters, G. W., Danielsson, J., Zigrand, J.-P., 2018. Designating market maker behaviour in limit order book markets. Econometrics and Statistics 5, 20–44.

Roşu, I., 2009. A dynamic model of the limit oder book. Review of Financial Studies 22, 4601–4641.

Sirignano, J. A., 2019. Deep learning for limit order books. Quantitative Finance 19, 549–570.

Timmermann, A., 2018. Forecasting methods in finance. Annual Review of Financial Economics 10, 449–479.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A., 2017. Forecasting stock prices from the limit order book using convolutional neural networks. In: 2017 IEEE 19th Conference on Business Informatics (CBI). Vol. 1. IEEE, pp. 7–12.

Zhao, P., Yu, B., 2006. On model selection consistency of lasso. The Journal of Machine Learning Research 7, 2541–2563.

Zheng, B., Moulines, E., Abergel, F., 2013. Price jump prediction in a limit order book. Journal of Mathematical Finance 3, 242–255.

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

Securities included	all companies in S&P100 on 2020.12.31
Number of different security symbols	101
Date range of data	2019.1.1 to 2020.12.31
Number of trading days included	505
Number of symbols available on all days	96
Number of available (symbol, date) pairs	50,273
Total disk size	2.3 Terabytes

Table 2: Examples of trade data: INTC on Jan. 3rd, 2019

Time	Price	Size	Direction (Lee-Ready)
10 : 07 : 48.956770900	45.18	100	-1
10 : 07 : 48.956773554	45.18	300	-1
10 : 07 : 48.956916983	45.18	100	-1
10 : 07 : 48.956971093	45.18	100	$+1$
10 : 07 : 48.957830128	45.18	66	$+1$

Table 3: Example of quote data: INTC on Jan. 3rd, 2019

Time	Best Bid Price	Best Bid Size	Best Ask Price	Best Ask Size
10 : 07 : 48.956906761	45.18	100	45.19	4800
10 : 07 : 48.956921135	45.18	100	45.19	4700
10 : 07 : 48.956970663	45.17	1600	45.19	4700
10 : 07 : 48.956980355	45.17	1600	45.19	4100
10 : 07 : 48.956991775	45.17	1600	45.19	4000

Table 4: Summary statistics: TAQ data

Data	mean	std	skewness	kurtosis	10%	25%	50%	75%	90%
# data rows	473K	217K	0.7	-0.3	224K	296K	421K	625K	783K
# transactions	109K	66K	1.7	3.0	51K	61K	84K	140K	195K
# quote updates	364K	165K	0.6	-0.4	170K	231K	325K	478K	609K
Traded Volume (# share)	10.6M	16.5M	5.5	51.5	1.7M	3.0M	5.5M	11.1M	23.2M
Traded Volume ($)	1121M	2488M	12.2	381.2	225M	338M	561M	972M	1885M
Total Market Capitalization ($)	169B	217B	4.6	26.9	41B	66B	111B	201B	304B
Nominal Stock Price ($)	185.7	320.3	5.3	33.5	37.5	54.3	106.4	190.5	317.8
Daily Return	0.0	0.0	0.1	16.7	-0.0	-0.0	0.0	0.0	0.0
Daily Beta with SPY	0.9	0.3	0.9	5.1	0.5	0.6	0.8	1.0	1.3
Daily R^2 with SPY	0.4	0.2	0.0	-0.6	0.2	0.3	0.4	0.6	0.7
Turnover Rate	71bp	96bp	9.2	136.0	27bp	35bp	49bp	74bp	117bp
5s returns	0.0bp	2.3bp	0.6	97.4	-2.2bp	-0.9bp	0.0bp	0.9bp	2.1bp
30s returns	0.0bp	4.4bp	0.1	39.8	-4.0bp	-1.7bp	0.0bp	1.7bp	3.9bp
10trds returns	0.0bp	1.9bp	1.4	195.0	-1.8bp	-0.9bp	0.0bp	0.9bp	1.8bp
200trds returns	0.0bp	5.9bp	0.1	13.6	-6.1bp	-2.7bp	0.0bp	2.7bp	6.1bp
10lots vol returns	0.0bp	2.1bp	0.9	109.1	-2.1bp	-0.9bp	0.0bp	0.9bp	2.1bp
200lots vol returns	0.0bp	7.0bp	0.0	18.4	-6.8bp	-2.9bp	0.0bp	2.9bp	6.7bp
10trds duration (seconds)	7.7	11.0	19.6	3990.8	0.0	1.1	4.2	10.3	19.5
200trds duration (seconds)	126.3	137.5	24.6	1833.8	21.6	46.6	94.9	169.7	266.1
10lots vol duration (seconds)	13.0	68.1	231.1	63390.9	0.1	1.4	5.6	15.0	31.5
200lots vol duration (seconds)	196.8	305.3	7.7	199.0	16.4	43.1	108.0	234.9	446.6

Note: 1bp = 0.0001 = 0.01%. The upper panel summarizes the variables on aggregated statistics at daily level across all 101 stocks. It encompass 101 stocks, 505 trading days from 2019 to 2020 with 50273 samples. The lower panel presents summary statistics of each response variable for every symbol and timestamp. The data are down sampled so that each (symbol, date) has similar amount of data in calculating the statistics.

Table 5: Regression of predictability on nominal share price level

	5s return	5s direction	10 txn duration
Nominal Share Price (log)	-0.025^{***}	-0.014^{***}	0.008^{***}
	(0.001)	(0.001)	(0.002)
Symbol fixed effect	YES	YES	YES
Date fixed effect	YES	YES	YES
Adjusted R^2	0.710	0.695	0.146

Notation: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Table 6: Regression of predictability on liquidity measures

	5s return	5s direction	10 txn duration
Traded Volume (log)	-0.019^{***}	-0.003^{***}	0.057^{***}
	(0.001)	(0.0003)	(0.001)
Proportional Spread	0.006^{***}	0.005^{***}	0.003^{***}
	(0.0003)	(0.0002)	(0.0005)
Symbol fixed effect	YES	YES	YES
Date fixed effect	YES	YES	YES
Adjusted R^2	0.718	0.699	0.232

Notation: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Table 7: Regression of predictability on volatility and jump measures

	5s return		5s direction		10 txn duration	
Volatility	-0.020^{***}	-0.016^{***}	-0.001^{***}	-0.0004	0.044^{***}	0.034^{***}
	(0.001)	(0.001)	(0.0003)	(0.0003)	(0.001)	(0.001)
Jump 3-4%		-0.013^{***}		-0.00003		0.026^{***}
		(0.001)		(0.001)		(0.002)
Jump 4-5%		-0.014^{***}		-0.002^{**}		0.050^{***}
		(0.002)		(0.001)		(0.002)
Jump 5%+		-0.023^{***}		-0.004^{***}		0.074^{***}
		(0.001)		(0.001)		(0.002)
Symbol fixed effect	YES	YES	YES	YES	YES	YES
Date fixed effect	YES	YES	YES	YES	YES	YES
Adjusted R^2	0.715	0.718	0.692	0.692	0.198	0.223

Notation: *p<0.1; **p<0.05; ***p<0.01

Table 8: Regression of predictability on asset pricing characteristics

	5s return	5s direction	10 txn duration
Beta with SPY	-0.006^{***}	-0.001^{**}	0.016^{***}
	(0.001)	(0.0003)	(0.001)
R^2 with SPY	-0.010^{***}	-0.012^{***}	-0.035^{***}
	(0.001)	(0.0005)	(0.001)
Idiosyncratic Volatility	-0.012^{***}	-0.001^{***}	0.017^{***}
	(0.001)	(0.0004)	(0.001)
Symbol fixed effect	YES	YES	YES
Date fixed effect	YES	YES	YES
Adjusted R^2	0.719	0.705	0.217

Notation: *p<0.1; **p<0.05; ***p<0.01

Table 9: Regression of predictability on aggregate market conditions

	5s return		5s direction		10 txn duration	
VIX	-0.007^{***}		-0.012^{***}		-0.003^{***}	
	(0.0003)		(0.0001)		(0.0004)	
S&P500 Return		0.003^{***}		0.002^{***}		0.002^{***}
		(0.0003)		(0.0001)		(0.0004)
Symbol fixed effect	YES	YES	YES	YES	YES	YES
Date fixed effect	NO	NO	NO	NO	NO	NO
Data Date fixed effect	YES	YES	YES	YES	YES	YES
Adjusted R^2	0.630	0.625	0.653	0.597	0.042	0.041

Notation: *p<0.1; **p<0.05; ***p<0.01

Table 10: Multivariate panel regression of out-of-sample R^2 for predicting returns on determinants of predictability

	Dependent variable: out-of-sample R^2s for Return Predictions											
	5 sec	5 sec	30 sec	30 sec	10 txn	10 txn	200 txn	200 txn	1K vol	1K vol	20K vol	20K vol
Nominal Share Price (log)	-0.004	-0.010^{***}	-0.0002	-0.002	-0.022^{***}	-0.030^{***}	-0.025^{***}	-0.026^{***}	-0.036^{***}	-0.041^{***}	-0.040^{***}	-0.042^{***}
	(0.002)	(0.002)	(0.001)	(0.001)	(0.003)	(0.003)	(0.002)	(0.002)	(0.002)	(0.003)	(0.002)	(0.002)
Daily Return (log)	0.0002	0.001^{*}	0.001^{***}	0.001^{***}	-0.002^{***}	-0.001^{***}	-0.0002	-0.00005	-0.001^{***}	-0.001^{**}	-0.001^{*}	-0.0005
	(0.0003)	(0.0003)	(0.0002)	(0.0002)	(0.0003)	(0.0004)	(0.0002)	(0.0002)	(0.0003)	(0.0004)	(0.0003)	(0.0003)
Traded Volume (log)	-0.002^{**}	-0.004^{***}	-0.007^{***}	-0.005^{***}	0.026^{***}	0.022^{***}	0.019^{***}	0.017^{***}	0.027^{***}	0.025^{***}	0.018^{***}	0.017^{***}
	(0.001)	(0.001)	(0.0005)	(0.0004)	(0.001)	(0.001)	(0.001)	(0.0005)	(0.001)	(0.001)	(0.001)	(0.001)
Total Market Cap (log)	-0.030^{***}	-0.005^{**}	-0.007^{***}	0.0004	-0.017^{***}	0.013^{***}	-0.005^{***}	0.005^{***}	-0.014^{***}	0.009^{***}	-0.006^{***}	0.0002
	(0.002)	(0.002)	(0.001)	(0.001)	(0.002)	(0.002)	(0.001)	(0.001)	(0.002)	(0.002)	(0.002)	(0.002)
Proportional Spread	0.009^{***}	0.006^{***}	0.006^{***}	0.004^{***}	0.009^{***}	0.005^{***}	0.005^{***}	0.005^{***}	0.008^{***}	0.006^{***}	0.005^{***}	0.005^{***}
	(0.0003)	(0.0003)	(0.0002)	(0.0002)	(0.0004)	(0.0003)	(0.0002)	(0.0002)	(0.0004)	(0.0003)	(0.0003)	(0.0003)
Volatility	-0.010^{***}	-0.014^{***}	-0.005^{***}	-0.002^{***}	-0.009^{***}	-0.017^{***}	-0.002^{***}	-0.007^{***}	-0.009^{***}	-0.010^{***}	-0.007^{***}	-0.008^{***}
	(0.001)	(0.001)	(0.001)	(0.0004)	(0.001)	(0.001)	(0.001)	(0.0005)	(0.001)	(0.001)	(0.001)	(0.001)
Beta with SPY	0.014^{***}	0.001	0.006^{***}	-0.0005	0.014^{***}	0.002^{***}	0.005^{***}	0.001^{***}	0.014^{***}	0.002^{***}	0.003^{***}	-0.001
	(0.001)	(0.0005)	(0.001)	(0.0003)	(0.001)	(0.001)	(0.001)	(0.0003)	(0.001)	(0.001)	(0.001)	(0.0005)
R^2 with SPY	-0.043^{***}	-0.020^{***}	-0.026^{***}	-0.014^{***}	-0.052^{***}	-0.030^{***}	-0.018^{***}	-0.009^{***}	-0.051^{***}	-0.029^{***}	-0.013^{***}	-0.007^{***}
	(0.001)	(0.0005)	(0.001)	(0.0003)	(0.002)	(0.001)	(0.001)	(0.0003)	(0.001)	(0.001)	(0.001)	(0.0005)
Idiosyncratic Volatility	-0.034^{***}	-0.012^{***}	-0.021^{***}	-0.014^{***}	-0.030^{***}	-0.006^{***}	-0.014^{***}	-0.004^{***}	-0.027^{***}	-0.009^{***}	-0.006^{***}	-0.001
	(0.001)	(0.001)	(0.001)	(0.0005)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Jump 3-4%	-0.012^{***}	-0.016^{***}	-0.003^{***}	-0.004^{***}	-0.015^{***}	-0.018^{***}	-0.002^{**}	-0.002^{***}	-0.013^{***}	-0.015^{***}	-0.003^{**}	-0.004^{***}
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Jump 4-5%	-0.013^{***}	-0.021^{***}	-0.002^{**}	-0.004^{***}	-0.019^{***}	-0.028^{***}	-0.002^{*}	-0.003^{***}	-0.018^{***}	-0.024^{***}	-0.002	-0.004^{***}
	(0.001)	(0.002)	(0.001)	(0.001)	(0.002)	(0.002)	(0.001)	(0.001)	(0.002)	(0.002)	(0.002)	(0.002)
Jump 5%+	-0.022^{***}	-0.032^{***}	-0.008^{***}	-0.011^{***}	-0.031^{***}	-0.042^{***}	-0.004^{***}	-0.007^{***}	-0.029^{***}	-0.036^{***}	-0.005^{***}	-0.008^{***}
	(0.001)	(0.001)	(0.001)	(0.001)	(0.002)	(0.002)	(0.001)	(0.001)	(0.002)	(0.002)	(0.002)	(0.001)
S&P500 Return		0.0004		-0.0002		0.002^{***}		0.0004^{**}		0.002^{***}		0.0001
		(0.0003)		(0.0002)		(0.0004)		(0.0002)		(0.0004)		(0.0003)
VIX		0.022^{***}		0.010^{***}		0.023^{***}		0.009^{***}		0.017^{***}		0.005^{***}
		(0.001)		(0.0003)		(0.001)		(0.0004)		(0.001)		(0.001)
Symbol fixed effect	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Date fixed effect	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO
Data Date fixed effect	NO	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO	YES
Adjusted R^2	0.740	0.681	0.750	0.727	0.765	0.713	0.776	0.764	0.702	0.658	0.764	0.760

Notation: *p<0.1; **p<0.05; ***p<0.01

Note: The response variables are out-of-sample $R^2_{i,t}$s of return predictions for each security i and date t. Fixed effects for symbols controls each i and dates control each t. Data dates fixed effects include fixed effects on dates when important market data are released, which includes FOMC release dates and monthly unemployment data release dates. We regressed over all securities in S&P100 at the end of 2020. Each explanatory variables are standardized before regression, except the three jump indicators.

Table 11: Multivariate panel regression of out-of-sample R^2 for predicting duration on determinants of predictability

	Dependent variable: R^2s of Duration Predictions							
	10 txn	10 txn	200 txn	200 txn	1K vol	1K vol	20K vol	20K vol
Nominal Share Price (log)	0.012^{***}	0.010^{***}	0.026^{***}	0.021^{***}	-0.005	-0.006	-0.010	-0.016^{*}
	(0.003)	(0.003)	(0.008)	(0.008)	(0.004)	(0.004)	(0.009)	(0.009)
Daily Return (log)	-0.004^{***}	-0.004^{***}	-0.003^{**}	-0.002^{*}	-0.002^{***}	-0.002^{***}	-0.002^{**}	-0.002^{*}
	(0.0004)	(0.0005)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Traded Volume (log)	0.034^{***}	0.041^{***}	0.003	0.019^{***}	0.037^{***}	0.040^{***}	0.012^{***}	0.022^{***}
	(0.001)	(0.001)	(0.003)	(0.002)	(0.001)	(0.001)	(0.003)	(0.003)
Total Market Cap (log)	-0.006^{**}	-0.008^{***}	0.014^{**}	-0.002	0.001	-0.0003	0.020^{***}	0.012
	(0.003)	(0.003)	(0.007)	(0.007)	(0.003)	(0.003)	(0.007)	(0.007)
Proportional Spread	-0.001^{**}	-0.005^{***}	-0.008^{***}	-0.014^{***}	-0.002^{***}	-0.004^{***}	-0.002^{*}	-0.009^{***}
	(0.0005)	(0.0004)	(0.001)	(0.001)	(0.001)	(0.0005)	(0.001)	(0.001)
Volatility	0.008^{***}	-0.003^{***}	0.013^{***}	0.006^{**}	0.011^{***}	0.009^{***}	0.021^{***}	0.013^{***}
	(0.001)	(0.001)	(0.003)	(0.002)	(0.002)	(0.001)	(0.004)	(0.003)
Beta with SPY	-0.003^{**}	0.001^{*}	0.006^{**}	0.016^{***}	-0.004^{***}	0.0004	0.003	0.013^{***}
	(0.001)	(0.001)	(0.003)	(0.002)	(0.001)	(0.001)	(0.003)	(0.002)
R^2 with SPY	-0.001	-0.003^{***}	-0.008^{*}	-0.022^{***}	-0.003	-0.010^{***}	-0.017^{***}	-0.031^{***}
	(0.002)	(0.001)	(0.005)	(0.002)	(0.002)	(0.001)	(0.005)	(0.002)
Idiosyncratic Volatility	0.019^{***}	0.019^{***}	0.022^{***}	0.010^{***}	0.017^{***}	0.010^{***}	0.011^{**}	0.0003
	(0.002)	(0.001)	(0.004)	(0.003)	(0.002)	(0.001)	(0.005)	(0.003)
Jump 3-4%	0.016^{***}	0.016^{***}	0.020^{***}	0.016^{***}	0.009^{***}	0.008^{***}	0.00002	-0.007
	(0.002)	(0.002)	(0.004)	(0.004)	(0.002)	(0.002)	(0.004)	(0.004)
Jump 4-5%	0.037^{***}	0.033^{***}	0.049^{***}	0.042^{***}	0.030^{***}	0.025^{***}	0.034^{***}	0.022^{***}
	(0.002)	(0.002)	(0.005)	(0.005)	(0.003)	(0.003)	(0.006)	(0.006)
Jump 5%+	0.059^{***}	0.039^{***}	0.062^{***}	0.039^{***}	0.045^{***}	0.033^{***}	0.042^{***}	0.019^{***}
	(0.002)	(0.002)	(0.005)	(0.005)	(0.003)	(0.002)	(0.006)	(0.005)
S&P500 Return		0.004^{***}		0.002		-0.006^{***}		-0.012^{***}
		(0.0005)		(0.001)		(0.001)		(0.001)
VIX		-0.023^{***}		-0.022^{***}		-0.027^{***}		-0.029^{***}
		(0.001)		(0.002)		(0.001)		(0.002)
Symbol fixed effect	YES	YES	YES	YES	YES	YES	YES	YES
Date fixed effect	YES	NO	YES	NO	YES	NO	YES	NO
Data Date fixed effect	NO	YES	NO	YES	NO	YES	NO	YES
Adjusted R^2	0.270	0.185	0.172	0.107	0.227	0.183	0.133	0.097

Notation: *p<0.1; **p<0.05; ***p<0.01

Note: The response variables are out of sample $R^2_{i,t}$s of duration predictions for each security i and date t. Fixed effects for symbols controls each i and dates control each t. Data dates fixed effects include fixed effects on dates when important market data are released, which includes FOMC release dates and monthly unemployment data release dates. We regressed over all securities in S&P100 at the end of 2020. Each explanatory variables are standardized before regression, except the three jump indicators.

Table 12: Predictabily of returns using subtypes of data: Transactions-only vs. quotes-only

		Testing Timestamps		
Training Timestamps	Used Data	Trade's	Quote's	Both
Trade's	Trade	18.0% (8.2%)	12.4% (7.1%)	13.4% (7.3%)
Quote's	Both	16.8% (8.1%)	13.3% (7.4%)	13.9% (7.4%)
Both	Both	17.3% (8.1%)	13.3% (7.4%)	14.0% (7.5%)

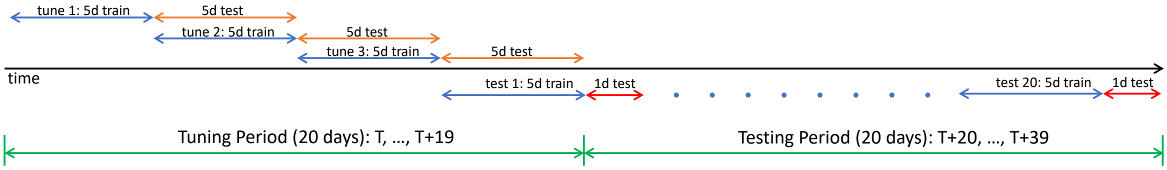
Note: Average out-of-sample R^2 (and standard deviations) of 5 second return predictions with trade or quote data. Each row uses timestamps and data from a specific subtype of data to calculate all the predictor variables in both fitting and predictions. Each column uses a different group of timestamps in testing.

Table 13: Predictabily using data from additional stocks

Total #stocks	Additional symbols	#spans	5s return R^2	30s return R^2
1		9	18.07% (18.50%)	6.40% (6.45%)
2	TXN	5	18.26% (18.79%)	6.62% (6.46%)
3	TXN, NVDA	3	18.30% (18.79%)	6.78% (6.56%)
5	TXN, NVDA, MSFT, QCOM	2	18.30% (18.69%)	6.57% (6.42%)

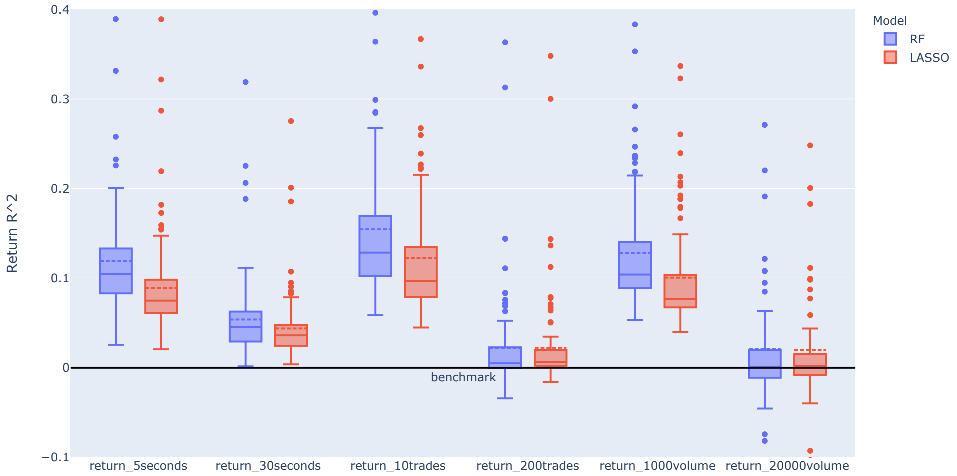
Note: Average out-of-sample R^2 (and standard deviations) of 5-second return predictions and 30-second return predictions. The signals from mostly correlated stocks are merged into INTC as of the newest signal at or before each data point in INTC based on their respective timestamps. For each crafted feature, the number of predictors with merged stocks is (tota number of stocks) * (number of span). The number of spans is chosen so that the total number of predictors are approximately the same, 9 or 10.

Figure 1: Algorithm tuning and testing rolling windows



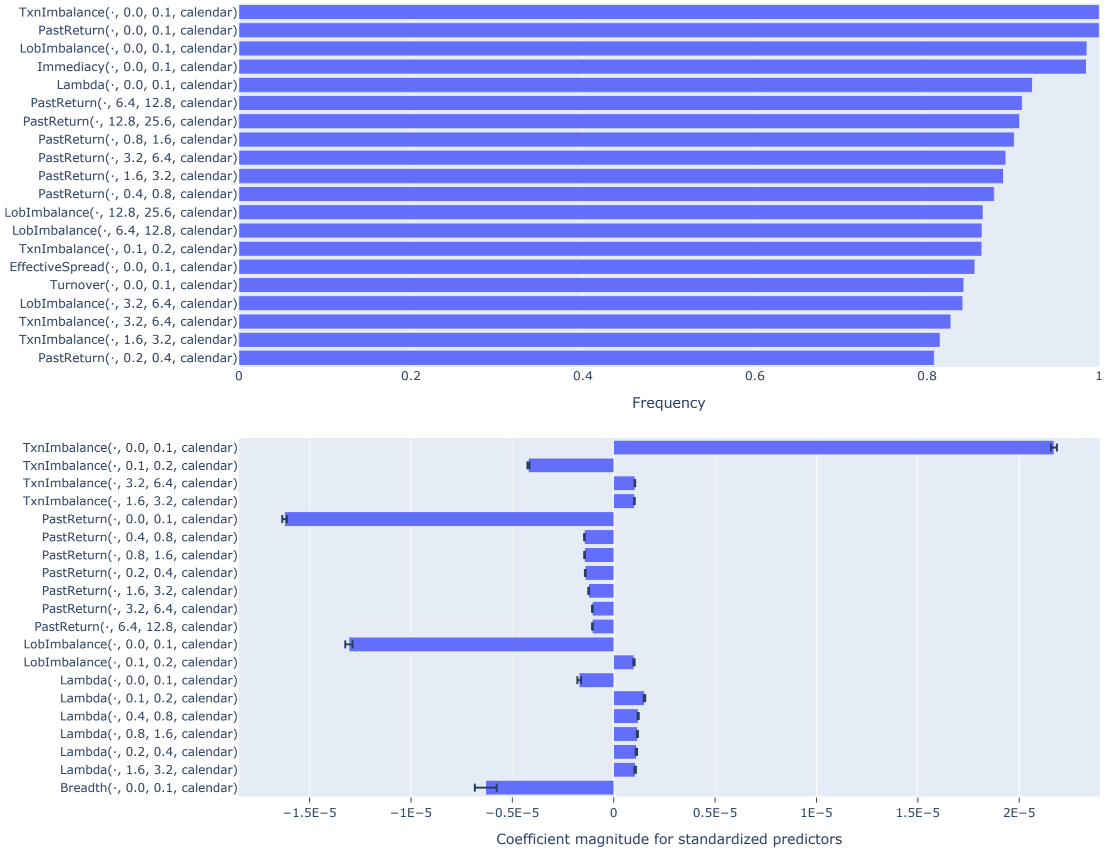
Note: The tuning parameters are optimized once every 20 trading days. For each given set of tuning parameters, we use the past 5 trading data to train the model and the next five days data to compute the testing errors. This is done once every 5 days (illustrated above the black line). The testing errors in the last 15 days (orange color) are used to choose the optimal set of tuning parameters. With the selected tuning parameters, we use past 5 days data to predict the next day target (colored red) in a rolling window manner (indicated above the green line) for the next twenty-days. The cycle process repeats once every 20 days.

Figure 2: Distribution of average out-of-sample R^2 when predicting individual stock returns



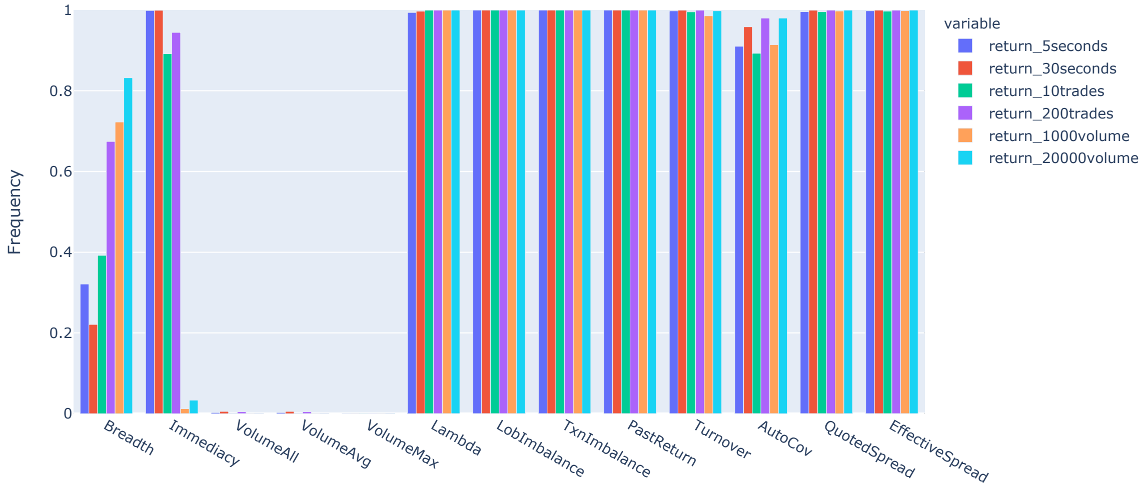
Note: The x-axis shows the look-forward horizon of each return predictions in 3 different clocks. Each box plot summarizes the distribution of the average out-of-sample R^2, which measures overall daily performance of a security in 2019 and 2020, across 101 stocks. The dashed line represents the average performance and the solid bar in the box indicates the median performance. The black horizontal line represents the performance of by using the sample mean of the testing data, which already peaks into the future.

Figure 3: Top 20 explanatory variables selected by LASSO for predicting 5-second returns



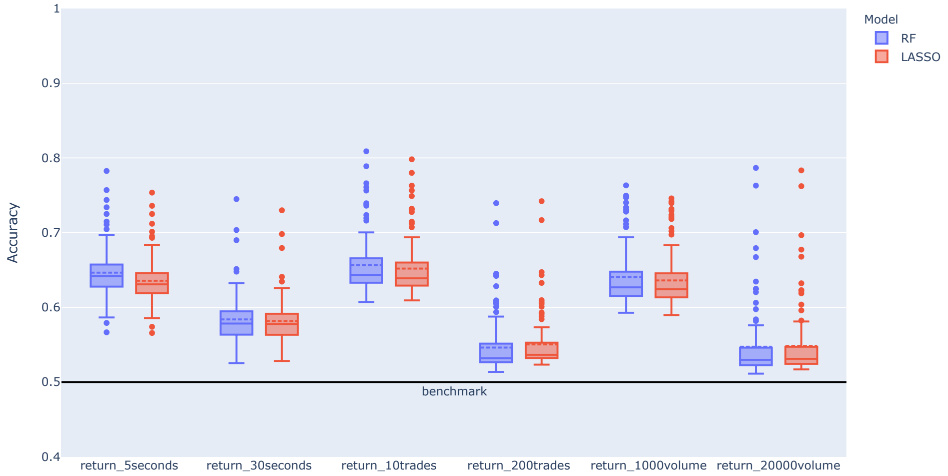
Note: The top panel describes the frequency of mostly used predictor variables. A variable is marked as used if its regression coefficient from LASSO is not zero. Frequencies are calculated over each test of 505 days and over 101 securities. The lower panel shows the average coefficients across all tests with error bars indicating their 95% confidence intervals. Coefficients with largest 20 absolute average values are shown and sorted by the variables of the same kind. The y-axis shows the variables selected. The values in bracket defines a past interval. For example, (., 0.8, 1.6, calendar) includes all the data from past 1.6 seconds to past 0.8 seconds in calendar clock.

Figure 4: Frequency of variable groups selected by LASSO for predicting returns over different horizons and time clocks



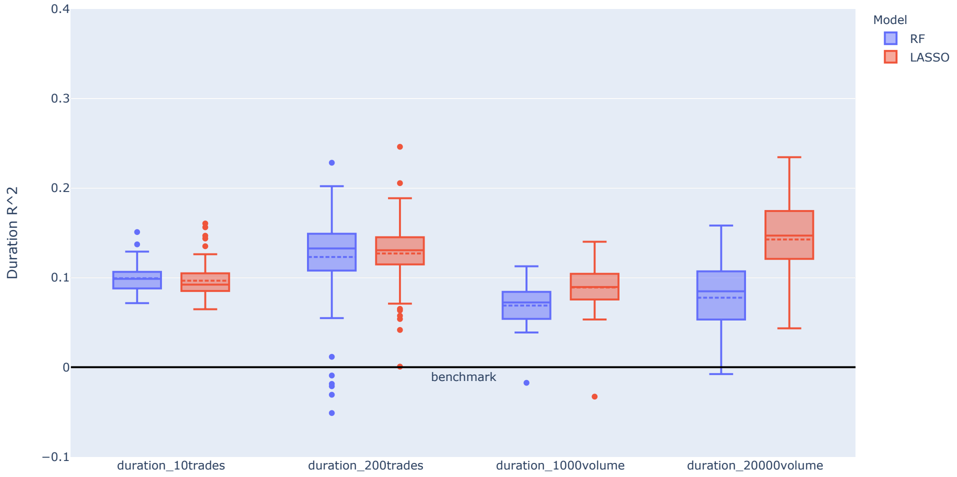
Note: Each variable in x-axis is measured in 9 different time windows. A group of variables is counted as selected if at least one of variables in the group has non-zero LASSO coefficient. For a given past interval, Breadth measures the amount of transactions and Immediacy indicates the average interval between transactions. VolumeAll, VolumeAvg and VolumeMax are related to the total, average and maximum traded volume. Lambda shows the price change in the interval proportional to total volume. LobImbalance and TxnImbalance are the limit order book imbalance and transaction imbalance for the interval. Turnover is the turnover rate and AutoCov is the auto-covariances of returns between consecutive transactions. EffectiveSpread is the dollar weighted spread on transactions. Detailed definitions of each variable can be found in Section 2.3.

Figure 5: Distribution of average out-of-sample directional accuracy when predicting individual stock returns



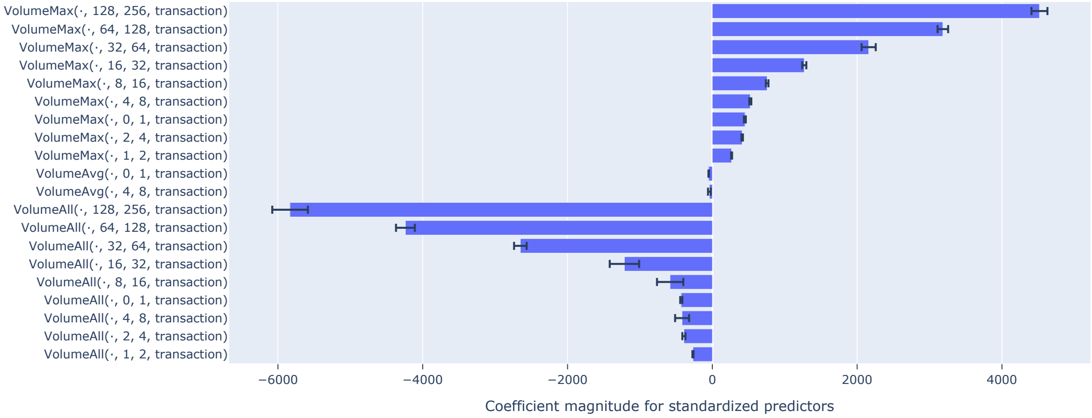
Note: The x-axis shows the look-forward time horizons of returns whose directions to be predicted. Each box plot is summarized over 101 data points where each one is the average daily performance of a stock aggregated over 505 days in 2019 and 2020. The dashed line represents the average performance and the black horizontal line indicates the performance by using the out-of-sample average.

Figure 6: Distribution of average out-of-sample R^2 when predicting individual stock durations



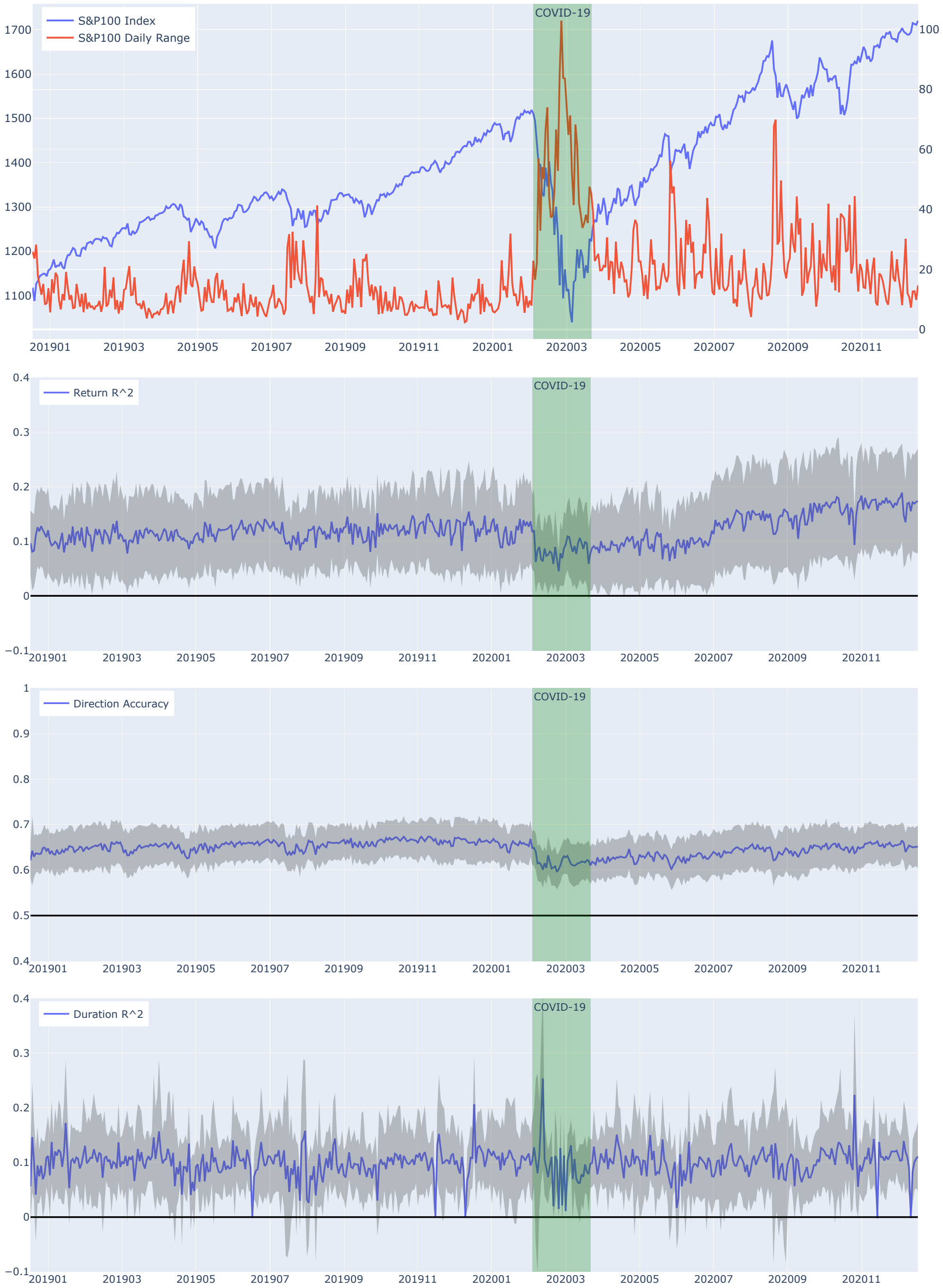
Note: Similar captions to those in Figure 1 are used.

Figure 7: Top explanatory variables selected by LASSO when predicting duration



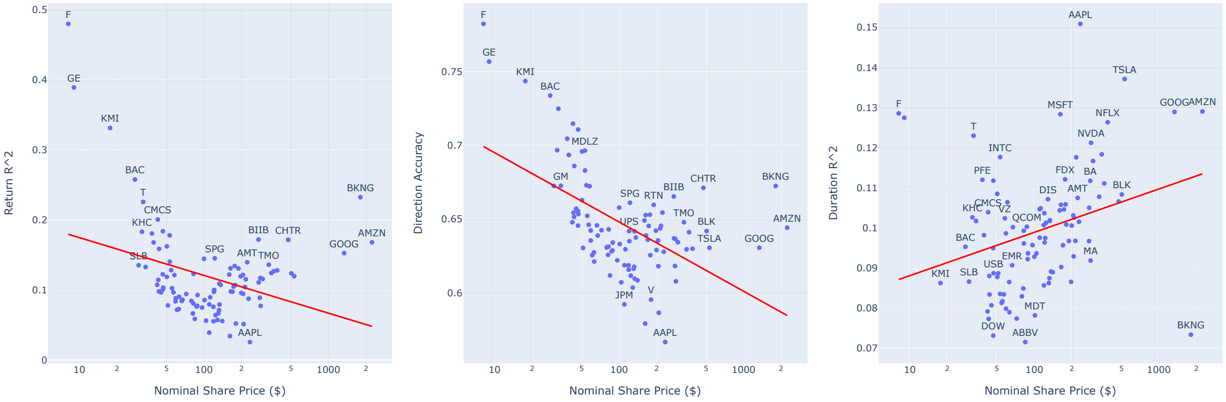
Note: All the explanatory variables are standardized before fitting LASSO and the variables with largest 20 absolute LASSO coefficients are displayed in this plot. VolumeAll and VolumeMax measures the total traded volume and maximum single trade in a past interval. The values in bracket defines the past interval. For example, (., 32, 64, transaction) includes all the data after the past 64th transaction and before or at the past 32th transaction.

Figure 8: Average daily performance of trade return, direction and duration predictions across S&P 100 stocks



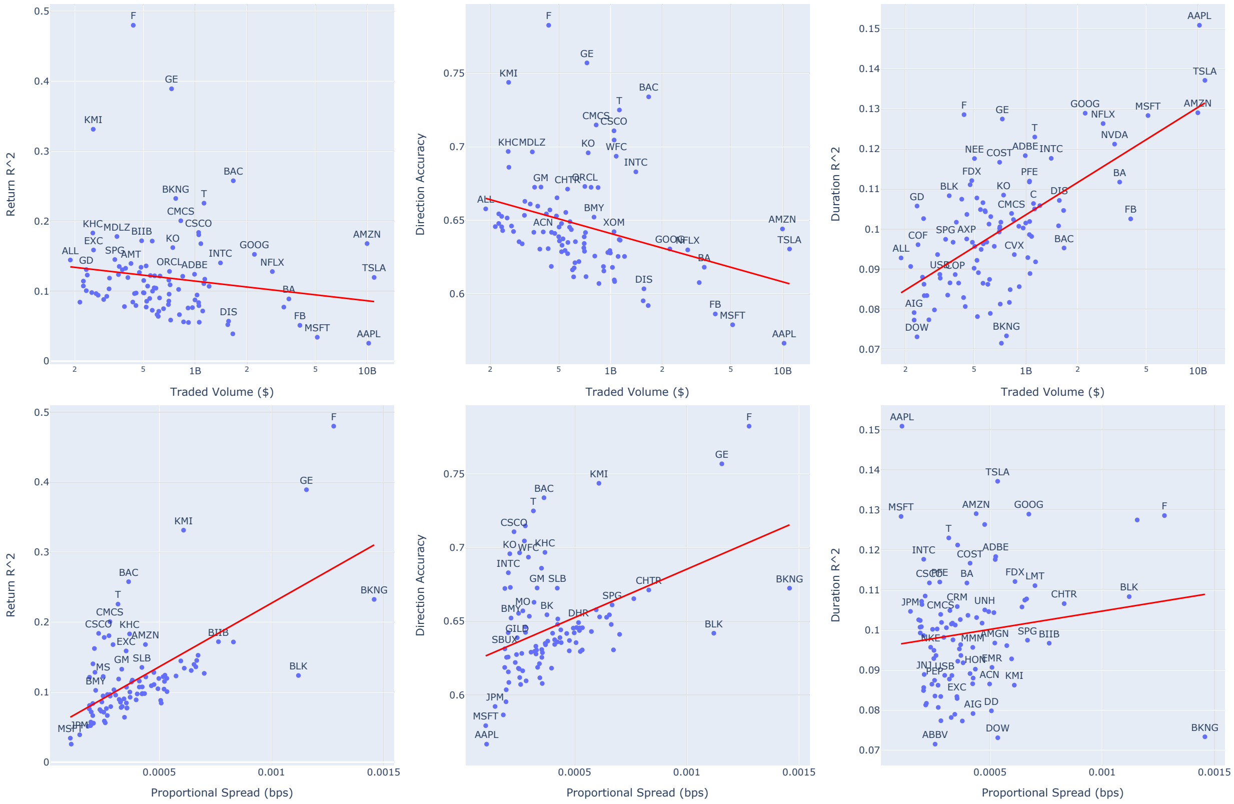
Note: Top panel shows the S&P 100 index along with its daily range, which indicates intraday volatility. The bottom three panels depicts daily average out-of-sample R^2s (blue curve) and their associated standard deviations (shaded bars), over 101 stocks, for predicting 5-second returns, their signs, and 10-trade duration. Random forests are used for prediction. Black lines indicate the performance by using out-of-sample average (second and fourth panel) or random guesses (third panel).

Figure 9: Prediction performance as a function of nominal share prices



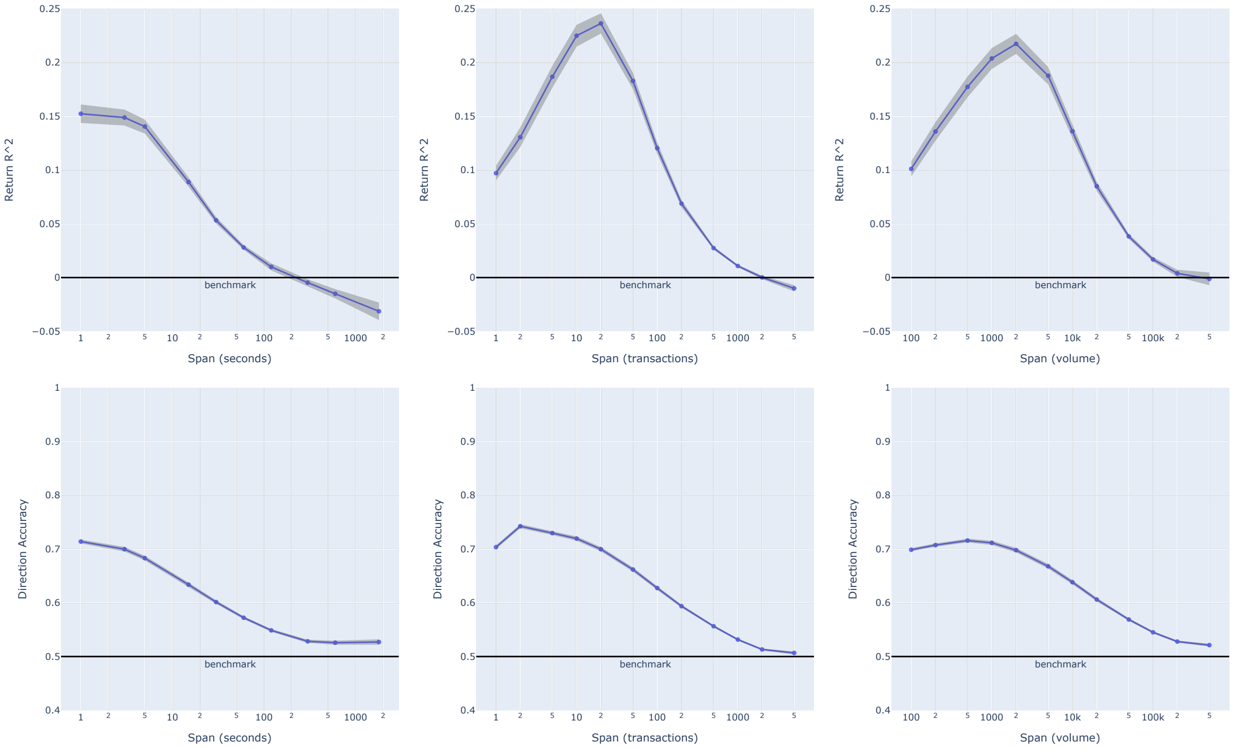
Note: Each point represents a security's average daily nominal share prices in 2019 and 2020 against its average daily performances, which are the out-of-sample R^2 for predicting 5-second returns (left panel), their associated directional accuracy (middle panel), and out-of-sample R^2 for predicting duration in 10 trades (right panel). The red lines are the OLS fits of data.

Figure 10: Prediction performance as a function of liquidity measures: Transaction volume and bid-ask spread



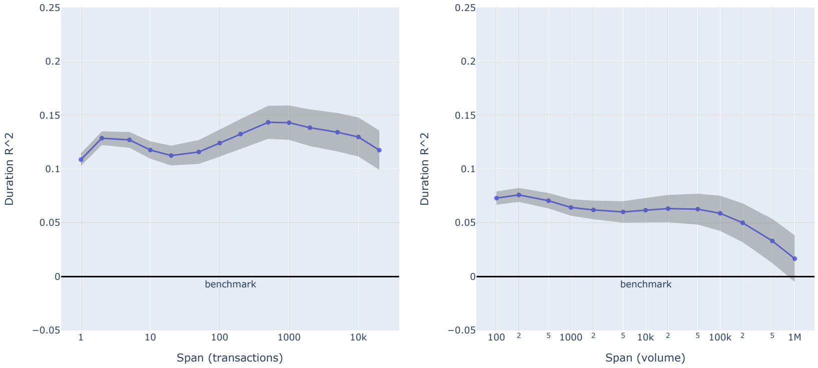
Note: Each point represents a security's average daily performances and average daily dollar traded volume / proportional spread in 2019 and 2020. The same response variables as in Figure 9 are used. The red lines are the OLS fits of data. The proportional spread each day is calculated as the security's average bid-ask spreads divided by mid prices, sampled every 15 seconds.

Figure 11: Predictabilty lifespan: Returns prediction performance as a function of the time horizon



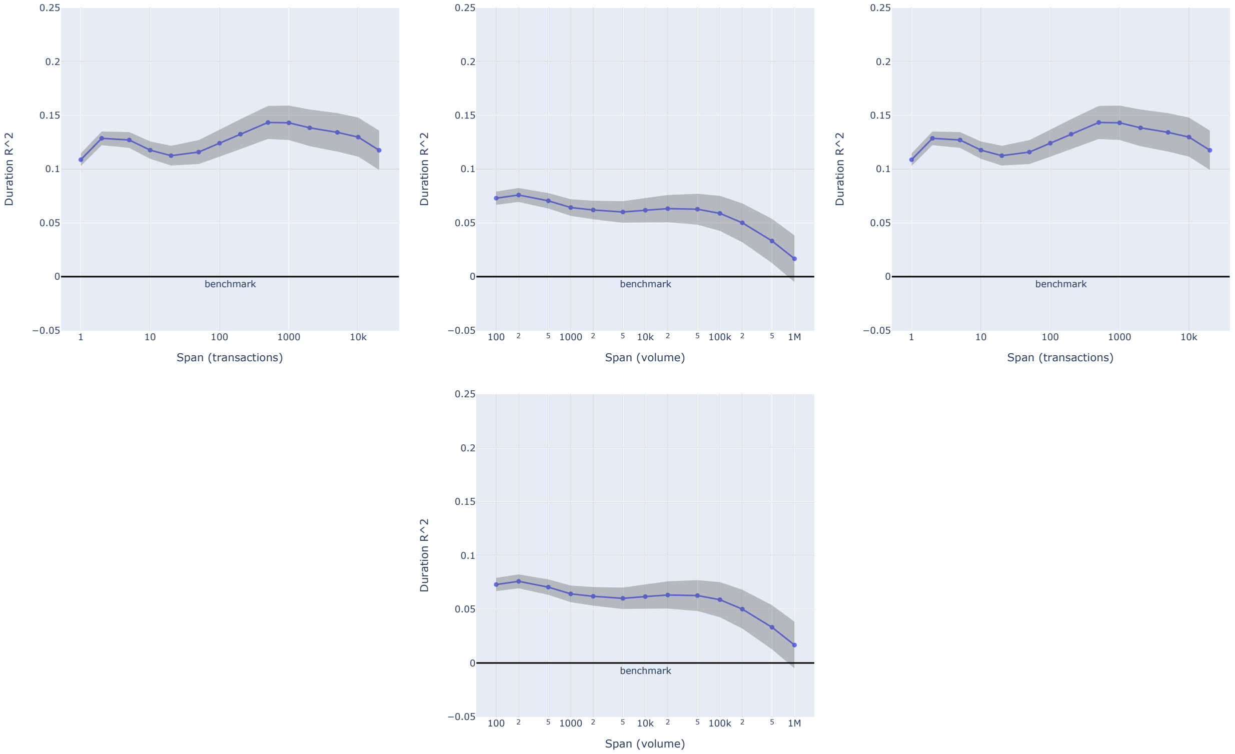
Note: The shaded areas depict 95% confidence intervals of the mean out-of-sample R^2 or accuracy over 505 days for INTC. Upper panels are the results for return predictions in calendar, transaction and volume clocks respectively. The lower panels are their corresponding results for direction predictions.

Figure 12: Predictability lifespan: Duration prediction performance as a function of the time horizon



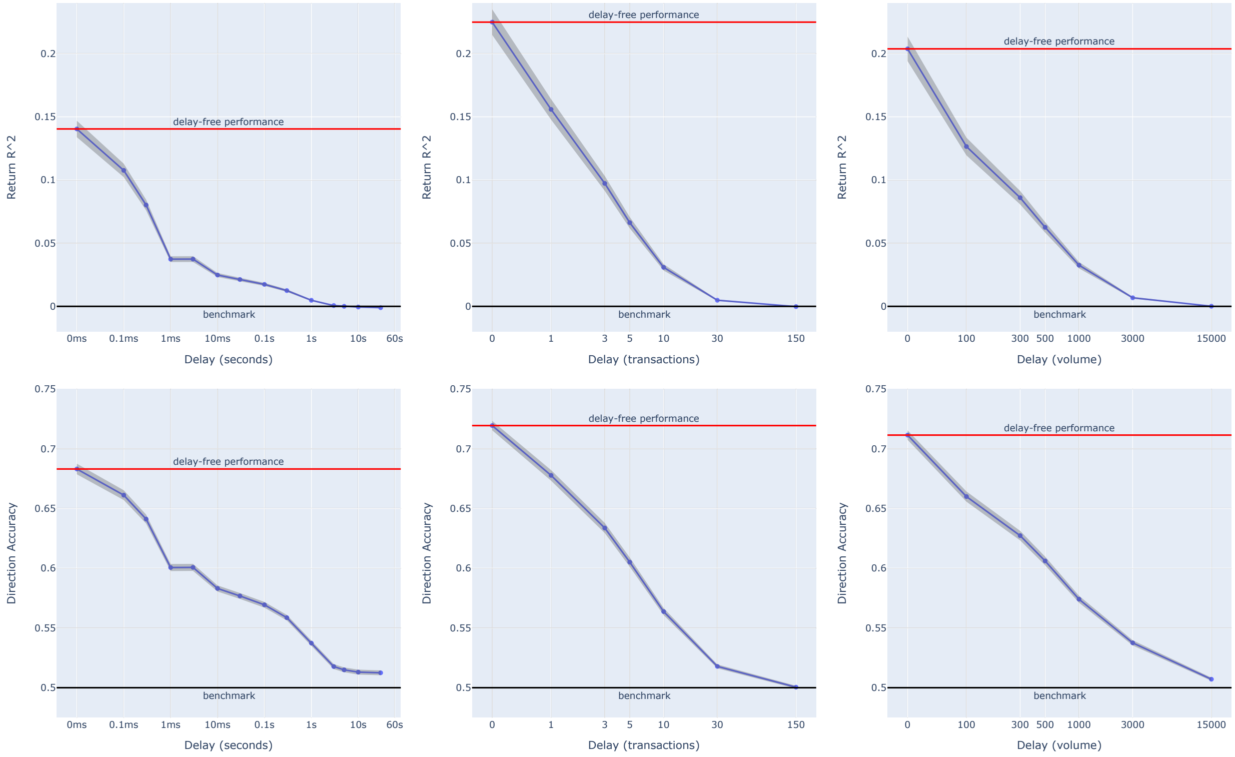
Note: Left and right panel are predictability of duration in transaction clock and volume clock respectively. The shaded area is 95% confidence intervals of the mean out-of-sample R^2 or accuracy (over 505 days).

Figure 13: Predictability lifespan: Duration prediction performance as a function of the time horizon



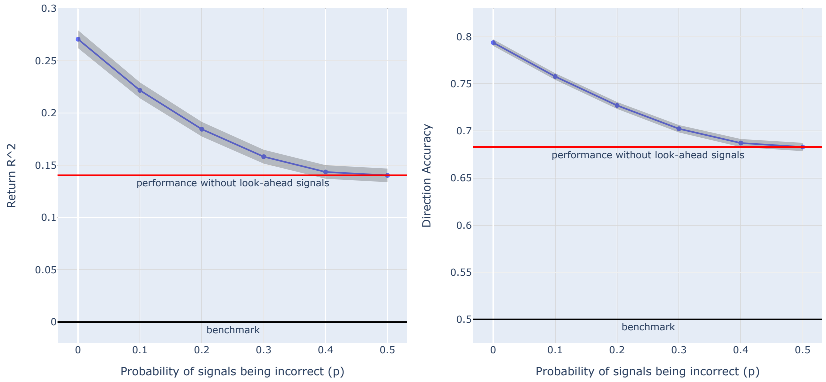
Note: Left and right panel are predictability of duration in transaction clock and volume clock respectively. The shaded area is 95% confidence intervals of the mean out-of-sample R^2 or accuracy (over 505 days).

Figure 14: The cost of data delays: Returns predictability as a function of lags in data acquisition and exploitation



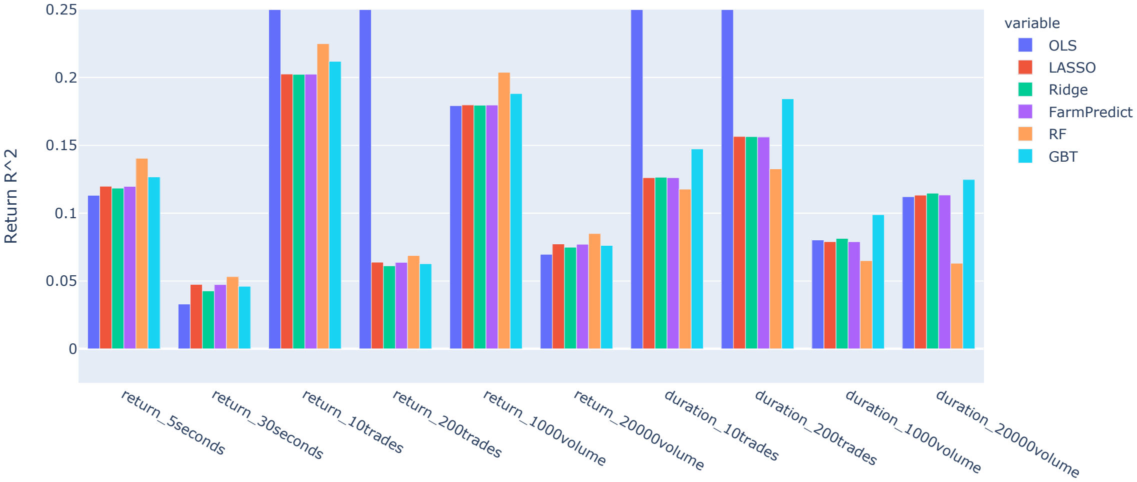
Note: Predictability of returns at different delay levels for INTC. The upper panel shows the decay of out-of-sample R^2 in return predictions with respect to delays in calendar, transaction, and volume clocks. The lower panel demonstrates the decayed accuracy for direction predictions. The shaded area indicates the 95% confidence intervals of mean daily predictability, measured in out-of-sample R^2.

Figure 15: Peeking at the order flow: Predictability as a function of the accuracy of the directional signal on the incoming order flow



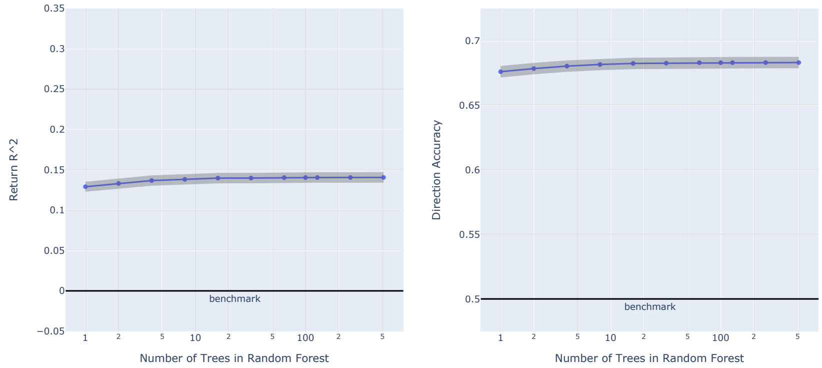
Note: Predictability at different noise level. The responses are 5 second returns and directions of INTC. The shaded area is the 95% confidence intervals of mean daily predictability. The red line depicts the average predictability in our main result, where no look-ahead signals were used.

Figure 16: Comparison of predictability performance using different machine learning methods



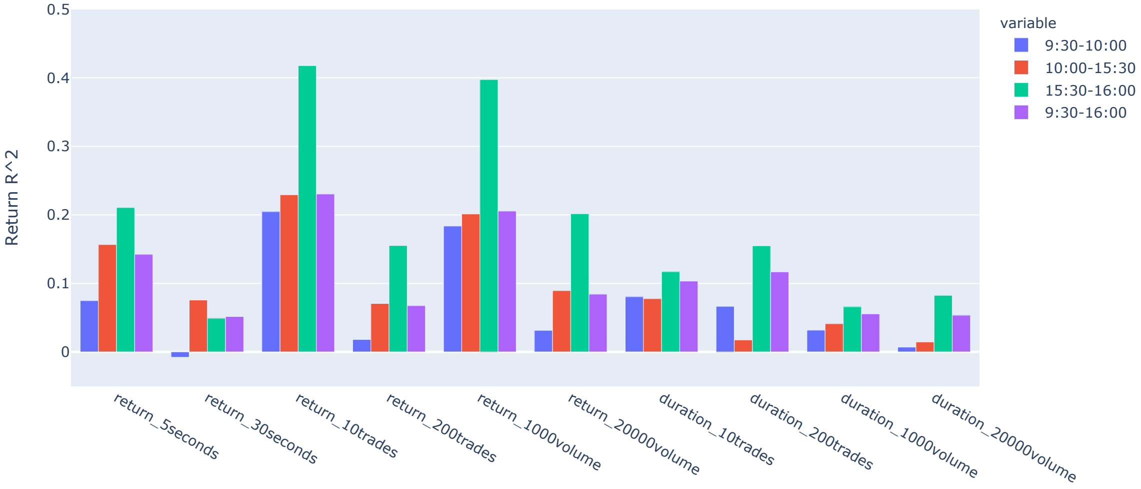
Note: Average performance for INTC's return and duration predictions using OLS, LASSO, ridge regression, FarmPredict with Lasso, Random Forest (RF), and Gradient Boosted Trees (GBT). Each bar summarizes the mean performance over 505 days from 2019 to 2020.

Figure 17: Sensitivity of predictability performance to the number of trees in random forests



Note: Left and right panel are performance in return and direction predictions respectively. The shaded area depict 95% confidence intervals of the mean out-of-sample R^2 or accuracy over 505 days. Black lines are the benchmark performances for each problem. The benchmark for return R^2 is the sample mean of the test sample, and for accuracy is random guess.

Figure 18: Intraday seasonality: Predictability across different phases of the trading day



Note: Each bar is the average performance (out-of-sample R^2) of random forest in the problem. Each model is fitted and tested with data from the same specified range of trading hour.