informs
http://pubsonline.informs.org/journal/ijds

MANAGEMENT SCIENCE
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 2694-4022, EISSN 2694-4030

Submitted to *Management Science*

Supplemental Material for "How and When are High-Frequency Stock Returns Predictable?"

Yacine Ait-Sahalia
Department of Economics, Princeton University and NBER yacine@princeton.edu

Jianqing Fan
International School of Economics and Management, Beijing Capital University of Economics and Business
Department of Operations Research and Financial Engineering, Princeton University jqfan@princeton.edu

Lirong Xue, Xiaonan Zhu
Department of Operations Research and Financial Engineering, Princeton University

A.1. Summary statistics of TAQ data

Summary statistics of our data and response variables are reported in Table A.2. 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 estimated 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.[1] 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 nanosecond-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.

A.2. 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

[1] Detailed definitions can be found in Section A.4.3 below.

Table A.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

sectors 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 A.6 of this supplemental material.

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) 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:

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

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 (Δ_1, Δ_2)

Table A.2 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: $1\text{bp} = 0.0001 = 0.01\%$. The upper panel summarizes the variables on aggregated statistics at daily level across all 101 stocks. It encompasses 101 stocks, 505 trading days from 2019 to 2020 with 50273 samples. The lower panel presents summary statistics of each response variable for each stock and timestamp. The data are downsampled so that each pair (day,stock) contributes a comparable amount of data in calculating the statistics. In the analysis, downsampling is employed solely to accelerate model fitting, with no impact on prediction target and features. Specifically, we first compute the features and targets using the complete set of transactions and quotes data. During model training, for each rolling-widow, if the training dataset exceeds 10^6 samples, due to the repetitive nature of quotes updates, we randomly select 10^6 samples to fit into the model to enhance computational efficiency. Besides computational efficiency, this was also done to avoid specific combinations of (day,stock) having too high a weight in the training. Overall, the downsampling results in removing 3.3% of the data, which breaks down to 4.90% on high-volatility days (up to a maximum of 6.73% during the Covid period) and 0.51% on low-volatility days; 4.22% from high-liquidity stocks and 1.33% from low-liquidity stocks. Again, this affects only the training of the model, not the targets and features used in the predictions.

$\in \{(0, 100), (100, 200), (200, 400), \ldots, (12800, 25600)\}$, with units in number of shares, where 1 lot consists of 100 shares.

We consider 15 main predictors, each being implemented over the 9 time spans so that $9 \times 15 = 135$ variables are used for each of the 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 15 predictors can be grouped into 3 categories, which we now describe.

Volume and duration. The first group of predictors is 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:

$$\text{Breadth}(T, \Delta_1, \Delta_2, M) = \left|\mathbf{D}^{\text{txn}} \cap \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, \text{M})\right|. \tag{2}$$

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

$$\text{Immediacy}(T, \Delta_1, \Delta_2, M) = \frac{\Delta_2 - \Delta_1}{\text{Breadth}(T, \Delta_1, \Delta_2, M)} \tag{3}$$

3. *VolumeAll* is the total number of shares transacted in the interval:[2]

$$\text{VolumeAll}(T, \Delta_1, \Delta_2, M) = \sum_{t \in \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, \text{M})} V_t. \tag{4}$$

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

$$\text{VolumeAvg}(T, \Delta_1, \Delta_2, M) = \frac{\text{VolumeAll}(T, \Delta_1, \Delta_2, M)}{\text{Breadth}(T, \Delta_1, \Delta_2, M)}. \tag{5}$$

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

$$\text{VolumeMax}(T, \Delta_1, \Delta_2, M) = \max\left\{V_t : t \in \text{Int}^{\text{back}}(T, \Delta_1, \Delta_2, \text{M})\right\}. \tag{6}$$

Return and imbalance. The second group of predictors is related to the stock's recent trading asymmetry. For example, if a majority of trades are buy trades that hit the limit sell, or 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

$$\text{Lambda}(T, \Delta_1, \Delta_2, M) = \frac{P_{\max(\mathbf{I})} - P_{\min(\mathbf{I})}}{\text{VolumeAll}(T, \Delta_1, \Delta_2, M)}. \tag{7}$$

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

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

$$\text{LobImbalance}(T,\Delta_1,\Delta_2,M) = \text{Average}\left[\frac{S_t^a - S_t^b}{S_t^a + S_t^b} : t \in \text{Int}^{\text{back}}(T,\Delta_1,\Delta_2,\text{M})\right]. \tag{8}$$

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

$$\text{TxnImbalance}(T,\Delta_1,\Delta_2,M) = \frac{\sum_{t\in\mathbf{D}^{\text{txn}}\cap\text{Int}^{\text{back}}(T,\Delta_1,\Delta_2,\text{M})}\left(V_t\cdot\text{Dir}_t^{\text{LR}}\right)}{\text{VolumeAll}(T,\Delta_1,\Delta_2,M)}. \tag{9}$$

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[3]

$$\text{PastReturn}(T,\Delta_1,\Delta_2,M) = 1 - \text{Average}\left[P_t^{\text{txn}} : t\in\mathbf{I}\right]/P_{\max(\mathbf{I})}. \tag{10}$$

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).

$$\text{Turnover}(T,\Delta_1,\Delta_2,M) = \frac{\text{VolumeAll}(T,\Delta_1,\Delta_2,M)}{S}. \tag{11}$$

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

$$\begin{aligned}\text{AutoCov}(T,\Delta_1,\Delta_2,M) = \text{Average}\Bigg[&\log\left(\frac{P_t^{\text{txn}}}{P_{Lt}^{\text{txn}}}\right)\log\left(\frac{P_{Lt}^{\text{txn}}}{P_{L(Lt)}^{\text{txn}}}\right) : \\ &t\in\mathbf{D}^{\text{txn}}\cap\text{Int}^{\text{back}}(T,\Delta_1,\Delta_2,\text{M})\Bigg].\end{aligned} \tag{12}$$

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

$$\text{QuotedSpread}(T,\Delta_1,\Delta_2,M) = \text{Average}\left[\frac{P_t^a - P_t^b}{P_t} : t\in\text{Int}^{\text{back}}(T,\Delta_1,\Delta_2,\text{M})\right]. \tag{13}$$

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

$$\text{EffectiveSpread}(T,\Delta_1,\Delta_2,M) = \frac{\sum_{t\in\mathbf{D}^{\text{txn}}\cap\text{Int}^{\text{back}}(T,\Delta_1,\Delta_2,\text{M})}\left[\log\left(\frac{P_t^{\text{txn}}}{P_t}\right)\cdot\text{Dir}_t^{\text{LR}}\cdot V_t\cdot P_t^{\text{txn}}\right]}{\sum_{t\in\mathbf{D}^{\text{txn}}\cap\text{Int}^{\text{back}}(T,\Delta_1,\Delta_2,\text{M})}\left(V_t\cdot P_t^{\text{txn}}\right)}. \tag{14}$$

5. *RealizedVolatility* is the local variance of transaction returns over the lookback interval:

$$\text{RV}(T,\Delta_1,\Delta_2,M) = \text{Average}\left[\left(\log P_t^{\text{txn}} - \log P_{Lt}^{\text{txn}}\right)^2 : t\in\text{Int}^{\text{back}}(T,\Delta_1,\Delta_2,\text{M})\right]. \tag{15}$$

6. *TSRV* (Two Scale Realized Volatility, see Zhang et al. (2005)) is a noise-robust volatility estimator obtained by computing a set of $\text{RV}(T,\Delta_1,\Delta_2,M)$ at a lower sampling frequency (replacing Lt with a time further lagged), with different starting points to cover the lookback interval, and averaging RV over that set.

[3]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.

Table A.3 Regression of predictability on liquidity measures

	5s return	5s direction	10 txn duration
Traded Volume (log)	-0.019^{***}	-0.002^{***}	0.057^{***}
	(0.001)	(0.0003)	(0.001)
Proportional Spread	0.005^{***}	0.005^{***}	0.003^{***}
	(0.0003)	(0.0002)	(0.0004)
Symbol fixed effect	YES	YES	YES
Date fixed effect	YES	YES	YES
Adjusted R^2	0.706	0.691	0.228

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

A.3. Results on Average daily performance

Here we summarize the distribution of out-of-sample R^2 for each given day across 100 stocks. The results are in Figure A.1.

A.4. Additional Results on Determinants of Predictability

Here, we furnish additional results on the marginal contributions of trading liquidity, volatility, jumps, and asset pricing characteristics of individual stocks to the predictability of their returns, with and without adjusting date and symbol effects, depicted respectively in the tables and figures below. In addition, we also examine the impact of the market environment on the predictability of stock returns.

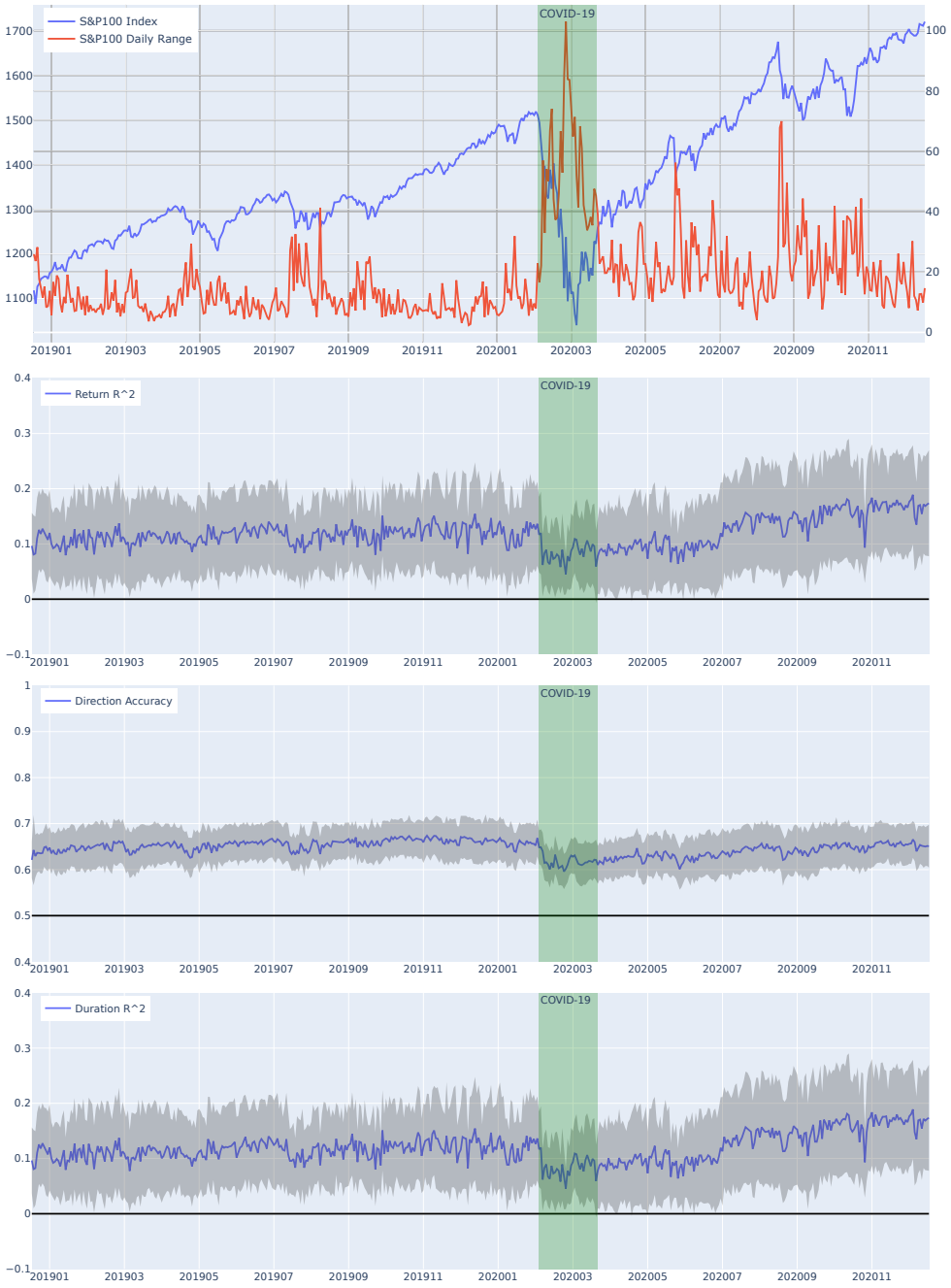
A.4.1. Stock Trading Liquidity

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 A.2 shows the stock-by-stock results, with daily predictability 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 A.3, 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 impact the overall predictability. On the other hand, for stocks where there is relatively little liquidity available at the best bid and ask, traders

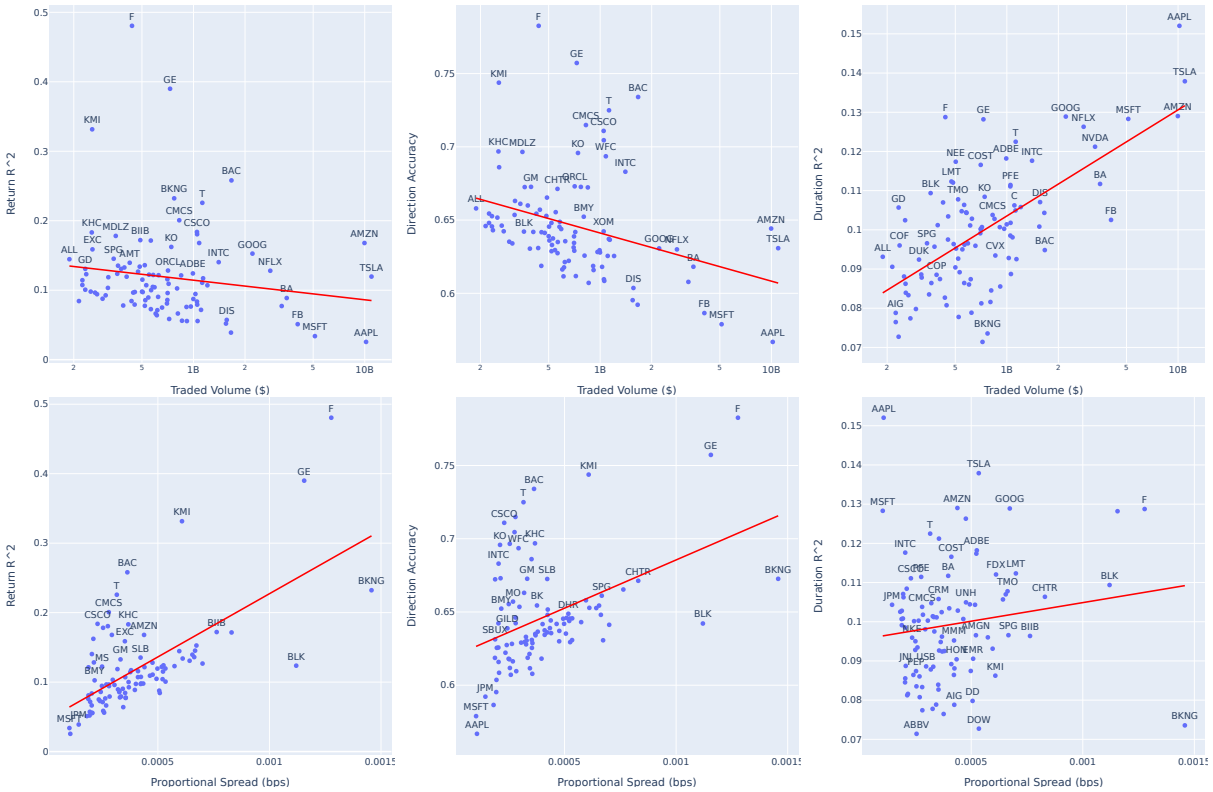
Figure A.1 Average daily performance of RF for transaction return, direction and duration predictions



Note: The top panel shows the S&P 100 index along with its daily range as a proxy for its variability. The bottom three panels depict daily average out-of-sample R^2s (blue curve) and their associated standard deviations (shaded bars), over stocks, for predicting 5-second returns, their signs, and 10-trade duration. RF is used for prediction. The horizontal black lines indicate the performance resulting from using the in-sample average (second and fourth panel) or random guesses (third panel).

are likely to break up a large order over many exchanges simultaneously to capture all available liquidity that is spread out over multiple exchanges, leading to several transactions. By contrast, the same trader behavior in stocks with plenty of liquidity available (perhaps due to a low nominal share price and a binding

Figure A.2 Prediction performance as a function of liquidity: Transaction volume and bid-ask spread



Note: Each point represents a security's average daily performance and average daily dollar traded volume / proportional spread in 2019 and 2020. The same response variables as in Figure 13 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.

minimum tick) may lead to a single trade on one exchange. 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.

A.4.2. 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 of 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 A.4. 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

Table A.4 Regression of predictability on volatility and jump measures

	5s return		5s direction		10 txn duration	
Volatility	-0.019^{***}	-0.016^{***}	-0.001^{**}	-0.0002	0.043^{***}	0.032^{***}
	(0.001)	(0.001)	(0.0003)	(0.0003)	(0.001)	(0.001)
Jump 3-4%		-0.014^{***}		-0.0003		0.026^{***}
		(0.001)		(0.001)		(0.002)
Jump 4-5%		-0.014^{***}		-0.002^{*}		0.051^{***}
		(0.001)		(0.001)		(0.002)
Jump 5%+		-0.022^{***}		-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	0.704	0.707	0.684	0.685	0.193	0.220

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

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 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.

A.4.3. 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 $\mathbf{R}_{\text{x}}$. 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

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

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

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

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 A.5 Regression of predictability on asset pricing characteristics

	5s return	5s direction	10 txn duration
Beta with SPY	-0.007^{***}	-0.001^{**}	0.016^{***}
	(0.001)	(0.0003)	(0.001)
R^2 with SPY	-0.008^{***}	-0.012^{***}	-0.035^{***}
	(0.001)	(0.0005)	(0.001)
Idiosyncratic Volatility	-0.011^{***}	-0.001^{***}	0.016^{***}
	(0.001)	(0.0003)	(0.001)
Symbol fixed effect	YES	YES	YES
Date fixed effect	YES	YES	YES
Adjusted R^2	0.708	0.698	0.214

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

Table A.6 Regression of predictability on aggregate market conditions

	5s return		5s direction		10 txn duration	
VIX	-0.007^{***}		-0.012^{***}		-0.003^{***}	
	(0.0002)		(0.0001)		(0.0004)	
S&P500 Return		0.003^{***}		0.002^{***}		0.002^{***}
		(0.0002)		(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	0.616	0.611	0.644	0.588	0.041	0.040

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

Table A.5 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 A.4.2. 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 forecast the returns of SPY separately, which would help in this case.) And stocks with higher R^2 relative to the market portfolio tend to be more liquid, ceteris paribus.

A.4.4. 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 A.6. 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.

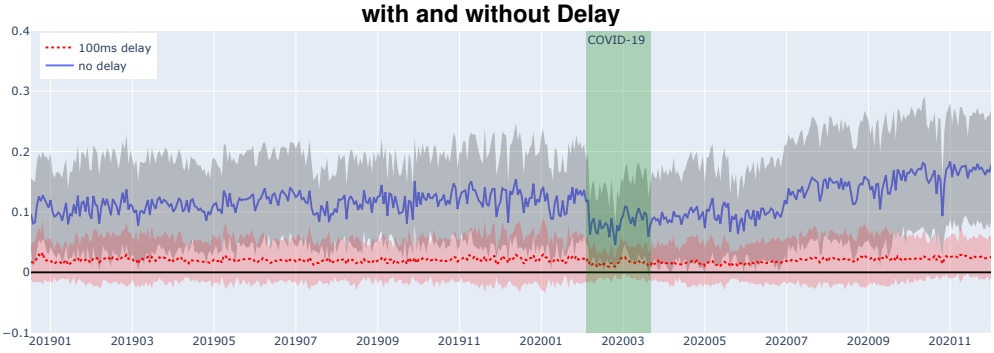
Table A.7 Determinants of predictability for duration: Panel regression of out-of-sample R^2

	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)	0.007^{**}	0.005	0.010	0.004	-0.003	-0.004	-0.015^{*}	-0.022^{**}
	(0.003)	(0.004)	(0.008)	(0.008)	(0.004)	(0.004)	(0.009)	(0.009)
Daily Return (log)	-0.004^{***}	-0.004^{***}	-0.002^{**}	-0.002^{*}	-0.002^{***}	-0.002^{***}	-0.002^{*}	-0.002^{*}
	(0.0004)	(0.0004)	(0.001)	(0.001)	(0.0005)	(0.0005)	(0.001)	(0.001)
Traded Volume (log)	0.034^{***}	0.041^{***}	0.003	0.018^{***}	0.036^{***}	0.040^{***}	0.009^{***}	0.019^{***}
	(0.001)	(0.001)	(0.003)	(0.002)	(0.001)	(0.001)	(0.003)	(0.003)
Total Market Cap (log)	0.0003	-0.003	0.022^{***}	0.008	0.003	0.0002	0.024^{***}	0.017^{**}
	(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^{***}	-0.001^{***}	-0.004^{***}	-0.003^{**}	-0.010^{***}
	(0.0004)	(0.0004)	(0.001)	(0.001)	(0.001)	(0.0004)	(0.001)	(0.001)
Volatility	0.006^{***}	-0.004^{***}	0.011^{***}	0.005^{**}	0.007^{***}	0.008^{***}	0.018^{***}	0.015^{***}
	(0.001)	(0.001)	(0.003)	(0.002)	(0.002)	(0.001)	(0.003)	(0.002)
Beta with SPY	-0.003^{***}	-0.0002	0.004	0.015^{***}	-0.003^{**}	-0.0002	0.004	0.012^{***}
	(0.001)	(0.001)	(0.003)	(0.001)	(0.001)	(0.001)	(0.003)	(0.002)
R^2 with SPY	-0.001	-0.002^{**}	-0.007	-0.019^{***}	-0.005^{**}	-0.010^{***}	-0.018^{***}	-0.030^{***}
	(0.002)	(0.001)	(0.004)	(0.002)	(0.002)	(0.001)	(0.005)	(0.002)
Idiosyncratic Volatility	0.020^{***}	0.020^{***}	0.023^{***}	0.012^{***}	0.016^{***}	0.010^{***}	0.009^{**}	0.0004
	(0.002)	(0.001)	(0.004)	(0.003)	(0.002)	(0.001)	(0.004)	(0.003)
Jump 3-4 %	0.017^{***}	0.016^{***}	0.022^{***}	0.016^{***}	0.011^{***}	0.009^{***}	0.003	-0.005
	(0.002)	(0.002)	(0.004)	(0.004)	(0.002)	(0.002)	(0.004)	(0.004)
Jump 4-5 %	0.038^{***}	0.035^{***}	0.052^{***}	0.044^{***}	0.031^{***}	0.026^{***}	0.038^{***}	0.025^{***}
	(0.002)	(0.002)	(0.005)	(0.005)	(0.002)	(0.002)	(0.006)	(0.005)
Jump 5 %+	0.060^{***}	0.040^{***}	0.063^{***}	0.038^{***}	0.048^{***}	0.035^{***}	0.048^{***}	0.024^{***}
	(0.002)	(0.002)	(0.005)	(0.005)	(0.002)	(0.002)	(0.005)	(0.005)
S&P500 Return		0.004^{***}		0.002		-0.006^{***}		-0.012^{***}
		(0.0004)		(0.001)		(0.001)		(0.001)
VIX		-0.023^{***}		-0.023^{***}		-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	0.266	0.182	0.164	0.099	0.216	0.173	0.126	0.090

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 control each i and dates control each t. Data dates fixed effects include fixed effects on dates when important market data are released, which include 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.

Figure A.3 Average daily performance of RF for transaction return predictions across S&P 100 stocks: with and without Delay



Note: Daily average out-of-sample R^2s (curves) and their associated standard deviations (shaded bars), over stocks, for predicting 5-second returns. RF is used for prediction. The blue curve with gray shade is the result without delay (same as the second subfigure in Figure A.1), and the red dotted curve with red shade is the result with a 100 ms delay. The horizontal black lines indicate the performance resulting from using the in-sample average.

A.5. Additional Results on the Impact of a Delay

Beyond the results presented in Section 4.2, we further examine how a delay affects predictability over time and across the cross-section. First, we assess whether the impact of a delay results in a uniform shift in predictability over time or varies across specific days. Figure A.3 replicates the time series results from the second subplot of Figure A.1 with a (relatively large) 100 ms delay. The results show that a delay consistently reduces prediction accuracy in the time series, as evidenced by the lower red dotted curve compared to the no-delay blue curve.

To further analyze this effect, Table A.8 examines the impact of a delay on days with different volatility levels. The daily volatility for a given stock is computed as the standard deviation of its mid-price returns at 15-second intervals, consistent with Section A.4.2. High-volatility days are defined as those with volatility above the stock's median daily volatility, while low-volatility days are those that fall below this threshold. We find that the impact of a delay is generally more pronounced on low-volatility days. A possible explanation is that high-volatility days tend to exhibit a more persistent signal-to-noise ratio than low-volatility days. Consequently, the noise introduced by a 100 ms delay has a greater effect on low-volatility days, as predictive signals become stale.

Additionally, we examine how this effect varies across stocks with different liquidity levels, with the results shown in Table A.9. Stocks are categorized by liquidity using both traded volume and proportional spread as measures. Stocks with liquidity measures above the median are classified as having high liquidity, while those below the median are considered having low liquidity. The delay appears to have a slightly larger impact on low-liquidity stocks. A possible reason is similar to the volatility case: stocks with lower liquidity exhibit a less persistent signal-to-noise ratio, making them more susceptible to the noise introduced by delayed updates.

Table A.8 Impact of a delay on the predictability over time: Out-of-sample R^2 as a function of day's volatility

	Full Sample	High Volatility Days	Low Volatility Days
100 ms delay	0.020 (0.038)	0.022 (0.040)	0.018 (0.034)
No delay	0.120 (0.085)	0.110 (0.086)	0.128 (0.083)

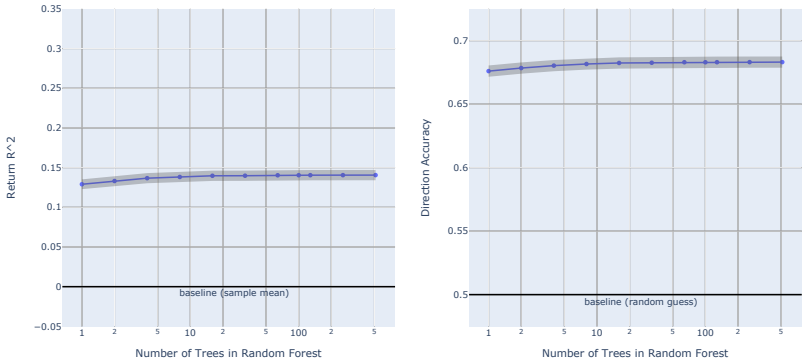
Note: Average (over days and stocks) out-of-sample R^2 (and standard deviations) of 5-second return predictions with 100 ms delay (first row) or without delay (second row). Results are reported for the full sample, high-volatility days, and low-volatility days, respectively.

Table A.9 Impact of delay on the predictability in the cross-section: Out-of-sample R^2 as a function of stocks' liquidity

		Full Sample	High Liquidity Stocks	Low Liquidity Stocks
traded volume	100 ms delay	0.020 (0.032)	0.021 (0.030)	0.019 (0.034)
	no delay	0.119 (0.066)	0.113 (0.065)	0.125 (0.065)
proportional spread	100 ms delay	0.020 (0.032)	0.013 (0.014)	0.026 (0.043)
	no delay	0.119 (0.066)	0.097 (0.045)	0.140 (0.076)

Note: Average out-of-sample R^2 (along with standard deviations) for 5-second return predictions with a 100 ms delay, computed across days and stocks. Results are reported for the full sample, high-liquidity stocks, and low-liquidity stocks. Stock liquidity is assessed using two distinct measures: traded volume and proportional spread.

Figure A.4 Sensitivity of predictability performance to the number of trees in random forests



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

A.6. Additional Robustness Checks

A.6.1. 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 A.4, we can see that the improvement from increasing the number of trees is limited and we observe almost no difference after 16 trees. An

Table A.10 Predictability of returns using subtypes of data: Transactions-only vs. quotes-only data

		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.

important reason for this is that the performance is averaged over 505 days which is already quite stable. Day-by-day performance however benefits from using additional trees. When more trees are used, tuning selects a deeper depth for each tree as hyper-parameter.

A.6.2. 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[4]. 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 affect 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 A.10. We can see that the predictability at transaction-only timestamps is significantly higher than those on quote update-only timestamps. This is consistent with the variable selection findings from LASSO in Section 3.1 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.

A.6.3. 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

[4]Recall that the return at a quote timestamp is defined as 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.

Table A.11 Predictability 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 (total number of stocks) $\times$ (number of spans). The number of spans is chosen so that the total number of predictors is approximately the same, 9 or 10.

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 synchronized. 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 ($5 \times 2 = 10$ 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.[5] We compute predictability results using a calendar clock as it makes the most sense when predictors from multiple 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 improves 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 A.11.

A.7. Lasso and Random Forests

We now briefly describe the two main methods that we use to predict stock returns and durations: LASSO and random forests (RF).

[5]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 $(4+1)\times 2=10$ predictors for each crafted feature.

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 135, consisting of 9 time spans for each of the 15 predictor variables. The algorithms can then construct to combine these variables for predicting the response variable Y, as well as select the most informative subsets of variables.

A.7.1. 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 of 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

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \left\{ \frac{1}{n} \sum_i \left(\widetilde{Y}_i - \boldsymbol{\beta}^T \widetilde{\mathbf{X}}_i \right)^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}. \tag{16}$$

This optimization problem does not admit an analytic solution, but can easily be solved using convex optimization algorithms, coordinate descent methods, least angle regression, among others.[6] For new data $\mathbf{X}_{\mathrm{new}}$, the model will predict its associated response as

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

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. [7]

On the other hand, being a least-squares method, LASSO is not robust to heavy-tailed data. As shown in Table A.2, the response variables we employ have heavy tails with large kurtosis. To mitigate this problem, we clip our training responses at their 5^{th} and 95^{th} percentile to reduce the influence of outliers.[8]

[6] We use the coordinate descent algorithm implemented in the *Scikit-learn* software in *Python* to solve (16).

[7] The model selection property has been established for LASSO under the so-called irrepresentable conditions by Zhao and Yu (2006) while 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 a more comprehensive account on high-dimensional model selection and prediction.

[8] Such a method is referred to as Winsorization in statistics. The theoretical foundations for employing 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.

A.7.2. 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 consider ensemble learning trees (see, e.g., Hastie et al. (2009), Murphy (2014), Fan et al. (2020b)). A random forest (RF) 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.

RF is fitted via iteratively growing i.i.d. regression trees by drawing bootstrap samples from the dataset. 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:

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

The rest of the tree is grown recursively in this fashion. 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 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

$$\widehat{f}_j(\mathbf{X}_{\text{new}}) = \sum_{k \in \mathcal{I}_j} \widehat{\beta}_{j,k} \mathbb{1}_{\left\{\mathbf{X}_{\text{new}} \in \mathcal{R}_{j,k}\right\}}, \tag{17}$$

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.

A.7.3. Algorithm Tuning and Testing

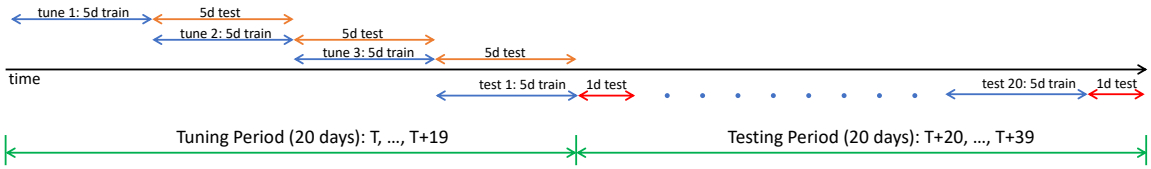
Each model we employ has a large number of parameters and is tuned and tested on a rolling window basis. A new model is fitted for every testing day using data from the past 5 days, so only the most recent information is included. Hyper-parameters of each method are tuned every month (20 trading days).

Tuning hyper-parameters 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 the reduced set of hyper-parameters that matter 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 RF, 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 tests 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 hyperparameters 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 A.5. 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 follows:

Figure A.5 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.

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 R^2, that is $\frac{1}{15}\sum_{t=T+5}^{T+19} R^2_t$, and fix the hyper-parameters for the predictions in the next step.
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 $1-4$.

References

Bickel PJ, Ritov Y, Tsybakov AB (2009) Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics* 37(4):1705–1732.

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

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

Fan J, Li R, Zhang CH, 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*. Springer Series in Statistics (New York: Springer-Verlag), second edition.

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

Lee CM, Ready MJ (1991) Inferring trade direction from intraday data. *The Journal of Finance* 46:733–746.

Murphy KP (2014) *Machine Learning: A Probabilistic Perspective* (MIT Press).

Zhang L, Mykland PA, Aït-Sahalia Y (2005) A tale of two time scales: Determining integrated volatility with noisy high-frequency data. *Journal of the American Statistical Association* 100:1394–1411.

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