Price Discovery in the Machine Learning Age[1]

Agostino Capponi Shihao Yu

Current version: March 15, 2024

[1] Agostino Capponi, Columbia University, Department of Industrial Engineering and Operations Research, ac3827@columbia.edu. Shihao Yu, Columbia University, Department of Industrial Engineering and Operations Research, sy3119@columbia.edu. We thank Joel Hasbrouck, Ruizhe Jia, and participants at the Columbia-Bloomberg Machine Learning in Finance Workshop, Markov Decision Process and Reinforcement Learning Workshop for their valuable comments. We acknowledge computing resources from Columbia University's Shared Research Computing Facility project.

Price Discovery in the Machine Learning Age

Abstract

Analyzing high-frequency price discovery in a fast and fragmented market faces big data challenges. To tackle it, we integrate machine learning models with level-three order book message data from multiple exchanges. We show that the gradient-boosted regression tree model (GBRT) outperforms linear models in forecasting short-term price changes by accounting for temporal dependence and interactions between limit order book features. Notably, expanding the order book history from one to 25 events enhances the out-of-sample R^2 of GBRT from 4.34% to 6.32%, a pattern not observed in linear models. Through feature permutation importance, we interpret the prediction results of machine learning models and analyze the price discovery process, showing that GBRT allocates a larger information share to top-of-book features compared to linear models.

1 Introduction

The modern trading environment, particularly evident in the US equities market, is defined by several characteristics. First, trading is extremely fast, with a significant fraction of trades executed in sub-millisecond time intervals (Menkveld 2018). Second, trading is highly fragmented across dozens of different marketplaces such as public exchanges and dark pools (O'Hara and Ye 2011). Given its fast and fragmented nature, trading activities on the exchanges constantly generate a vast amount of market data. For example, the most granular data sold by the exchanges to sophisticated market participants such as high-frequency traders (HFT), the so-called "direct feeds", record every play of the entire limit order book: trade executions, limit order submissions, modifications, and cancellations.

Such an environment poses a unique "big data" challenge to analyzing price discovery. The set of useful signals that can be constructed from the granular market data has expanded considerably, resulting in the high dimensionality of the predictors. In addition, sophisticated algorithmic trading strategies result in complex non-linear effects and interactions between the predictors. Traditional structural econometric models, such as the vector-autoregressive (VAR) model, are limited in their capacity to tackle the above issues.

In response, there is an emerging consensus on the need for machine learning (ML) models that can adequately process "big data" and account for the non-linear effects and interactions among the predictors and can help enhance our understanding of the price discovery process in the modern trading landscape. The objective of our paper is twofold. First, we aim to identify the sources of the superior performance of machine learning models over other linear models in predicting short-term price changes——particularly, their capability to encapsulate temporal dependencies and inter-feature dynamics. Second, we seek to interpret the prediction results of the machine learning models, some of which are referred to as "black box" models, through permutation importance, a model-agnostic approach. With interpretability, we are able to answer economic questions such as information attribution across different levels of the order book and exchanges in the price discovery process.

We obtain direct feeds data from all US public exchanges[1]. Direct feed data provide the most granular record of all limit order book events: trade executions, new limit order submissions, modifications, and cancellations. Our dataset encompasses 30 constituent stocks of the Dow Jones Industrial Average (DJI), offering a representative cross-section of highly liquid securities. The data sample stretches across 54 trading days, capturing a diverse array of market conditions from the years 2017 to 2021.

Based on the order book message data, we build the complete limit order book for each stock, enabling us to construct a rich set of high-frequency order-book features. Specifically, we construct order-book event variables capturing the executions, new order submissions, and order cancellations, at different price levels in the order book. In addition to the order-book event variables, we further construct order-book state variables such as the depth imbalance at various price levels. Last, to capitalize on the comprehensive nature of the direct feeds data—which encompasses order-by-order, or level-3 data—we incorporate novel features such as the length of order queues and the resting time of canceled orders as further features.

We use several popular machine learning models for our prediction exercises including dimension-reduction models such as elastic net, tree-based models such as gradient-boosted regression tree (GBRT), and more advanced neural network models such as the long short-term memory (LSTM). While we leverage various machine learning models, our primary objective extends beyond merely comparing their effectiveness. Instead, we aim to identify the reasons that contribute to the superior performance of certain models over others, with a focus on interpretability.

In addition, we use a simple, agnostic approach to analyze feature importance and information attribution across different levels of the order-book information and across exchanges. Permutation importance quantifies the impact of each feature on a model's prediction accuracy by measuring the performance reduction when the feature's values are randomly shuffled, thereby breaking the association with the target.

[1] The direct feeds data is collected by MayStreet, now part of the London Stock Exchange Group (LSEG).

We show that machine learning models, which can capture non-linearity and feature interactions, perform much better compared to other models. Specifically, we find that the tree-based model, GBRT, performs the best. The average out-of-sample R^2 with an order book history of 25 events is 6.32%. The temporal-aware neural network model, LSTM, follows with an out-of-sample R^2 of 3.14%. Mere dimension-reduction model, Elastic Net, and the plain-vanilla MLP perform better than OLS, but at a much smaller margin.

Our analysis reveals that the ability to account for both state dependence and temporal dependence among features is the source of the superior performance of machine learning models. Incorporating state features from the order book, such as depth imbalance, alongside existing event features, like cancel orders and thus allowing the price impacts of order book actions to depend on order book conditions boosts the out-of-sample R^2 for machine learning models, such as GBRT. Specifically, we observe that R^2 more than doubles, jumping from 2.66% to 5.29%. In contrast, the out-of-sample R^2 for the OLS model remains negligible, and close to zero. Moreover, the inclusion of an extended history of order-book events and states significantly boosts predictive accuracy. Expanding the analysis to include up to 25 historical order book events, as opposed to just a single event, leads to a marked improvement in the out-of-sample R^2 for the GBRT model, elevating it by about 50%.

Our feature importance analysis aligns with established microstructure theory, revealing that top-of-book features—specifically, trade executions, cancellations, and additions at the top of the book—play a critical role in forecasting short-term price movements. Intriguingly, we observe that machine learning models, particularly the GBRT model, allocate greater importance to top-of-book features in comparison to linear models. Moreover, we find that incorporating features derived from the more granular level-three limit-order-book messages data such as the resting timing of order cancellations significantly enhances the accuracy of high-frequency price predictions: the out-of-sample R^2 of the GBRT model increases by roughly 20% when adding level-3 features to existing level-2 features.

Our paper proceeds as follows. We describe our dataset and define variables in Section 3. In

Section 4, we detail our identification strategy used for the empirical analysis. We discuss the results in Section 5 and conclude in Section 6.

2 Literature

Our paper relates to the literature on empirical market microstructure. Traditionally, the empirical microstructure literature has leveraged vector autoregressive (VAR) or vector error correction (VEC) models to analyze and interpret the price discovery process. These models typically incorporate structural functional forms with a relatively small set of features, focusing on in-sample attribution of information shares. While these methods have provided valuable insights, the advent of the machine age has introduced new challenges that necessitate a reevaluation of the analytical frameworks used in this domain.[2]

Our paper is different from Hasbrouck (1991) and Hasbrouck (1995) and later extensions such as Brogaard, Hendershott, and Riordan (2019) and Hagströmer and Menkveld (2023) in several aspects. First, past papers use standard vector auto-regressive and vector error correction models (VARs and VECMs). VARs and VECMs are both linear models and it is difficult for them to capture non-linear and interaction effects. Second, VARs and VECMs can distinguish permanent informational effects from transient behaviors. With the usual transformations, we can obtain random-walk variances and information shares. However, there are no such procedures in machine learning models. For example, impulse responses are not clearly defined for machine learning models. Third, the two key informational measures of interest, permanent price impact, and information share, are essentially calculated in the sample. However, machine learning models are built for out-of-sample prediction purposes. Fourth, the usual way of determining feature importance in a machine learning model is either permutation importance or a global version of Sharpley value. While the two ways are model agnostic, they are target-variable dependent. Unless the target rep-

[2] Several models have been proposed to calculate the information shares. VAR (Hasbrouck 1991); "Information Share" via VECM (Hasbrouck 1995); "Component share" (Harris, McInish, and Wood 2002); "Information Leadership Share" (Putniņš 2013); VAR + VECM (Hagströmer and Menkveld 2023); Applications: spot vs futures (Hasbrouck 2003); Cross-border listings (Eun and Sabherwal 2003); Dark vs Lit trading (Hendershott and Jones 2005)

resents some true information measure, it is hard to translate feature importance to information share.

More broadly, our paper is related to the strand of literature on financial machine learning. Machine learning models have been applied in several markets. Noticeable contributions include Gu, Kelly, and Xiu (2020), which pioneer the use of machine learning models in cross-section asset pricing. Li and Rossi (2020) use machine learning models to select mutual funds based on the characteristics of the underlying stocks. Rossi and Utkus (2020) use machine learning models to explain the cross-sectional variation in the effects of robo-advising on portfolio allocations. Cong, Tang, Wang, and Zhang (2021) propose a deep reinforcement learning approach, integrated with attention-based neural-network models, for optimal portfolio management. Bali, Goyal, Huang, Jiang, and Wen (2020) use machine learning models for corporate bond return prediction. Bryzgalova, Pelger, and Zhu (2019) use decision trees to build cross-sections of asset returns where managed portfolios serve as test assets and building blocks for tradable risk factors. Cao, Jiang, Wang, and Yang (2021) use machine learning models to train an AI analyst who digests various information including corporate disclosures, industry trends, and macroeconomic indicators, and shows that it beats most human analysts. Besides empirical applications of the machine learning models, they have also helped develop new theories. For example, Colliard, Foucault, and Lovo (2022) build an "algorithmic market-makers" using Q-learning algorithms to learn private information and copy with adverse selection. Last, Kelly and Xiu (2023) provides an excellent survey on the state-of-the-art literature on financial machine learning.

All the previously mentioned works primarily focus on predicting prices in the cross section, and thus do not consider the time-series dimension. However, in the context of high-frequency price movements prediction, temporal dependence is a key factor to account for. Our contribution with respect to this stream of literature is to integrate machine learning models suitable for time-series empirical microstructure prediction. Moreover, we identify the sources of the forecast performance of such models and interpret the prediction results to attribute information shares.

3 Data and Variables

3.1 Data source

We use direct feeds from all public exchanges in the US equities market[3]. Direct feeds are the most comprehensive and fastest data feeds available in the US equities market. They are mainly consumed by sophisticated traders such as high-frequency traders (HFTs) for market making or arbitrage. Direct feeds contain full order-book event messages such as every new order submission, cancellation, and execution.

Our sample stock includes 30 constituent stocks of the Dow Jones Index (DJI). Our sample period includes 54 trading days spanning from the year 2017 to 2021. Specifically, we select the first Wednesday of every month across the sample years.

Based on the direct feeds, we re-build the order books for each exchange. In Appendix A, we provide details for the orderbook building procedures. With the order books built, we are able to calculate a series of high-frequency order-book features that potentially have predicting power over short-term price changes. In the empirical analysis below, we include four maker-taker exchanges (Nasdaq, NYSE, Arca, and BATs) and three take-maker exchanges (BX, BATSY, and EDGA).

3.2 Order-book features

There are various features one could construct from order books. We use two major sets of order-book features. The first set captures the order-book actions and the second characterizes the order-book states.

[3] The direct feeds data is collected by MayStreet, a US data company and supplier of SEC's MIDAS. Specifically, we use the following direct feeds: BATS BZX Multicast Pitch, BATS BYZ Multicast Pitch, CHX Book Feed, DirectEdge EDGA Multicast EdgeBook Depth, DirectEdge EDGX EDGX Multicast EdgeBook Depth, Nasdaq BX TotalView-ITCH, Nasdaq TotalView-ITCH, Nasdaq PSX TotalView-ITCH, NYSE MKT OpenBook Ultra, NYSE ARCA ARCABook, ARCA Trades, NYSE OpenBook Ultra, NYSE Trades, National Stock Exchange Multicast Depth of Book.

3.2.1 Order-book event variables

The first set of order-book features represent order-book actions, that is, each order submission, cancellation, and execution as used in (Brogaard, Hendershott, and Riordan 2019). The major difference is that in Brogaard, Hendershott, and Riordan (2019), they only use the direction of the event. So the variables they construct are all dummy variables of +1 or -1. To fully use the potential of machine learning models, we keep the original quantity information, which can be informative. For example, the size of trade or new liquidity added can provide useful information.

In financial market microstructure analysis, several order types play pivotal roles. Trades can either be marketable buy orders, which are aggressive and contribute to upward price movement, or sell orders, causing downward pressure. When traders place limit orders that improve the best bid or ask, they tighten the spread, enhancing the market's liquidity. Conversely, canceling orders that worsen the best bid or ask widens the spread and can harm liquidity.

Additionally, orders that add depth at the current best bid or ask increase the resilience of the BBO, while those that remove depth could potentially make the BBO more vulnerable to price fluctuations. Limit orders outside the BBO (non-BBO-depth) contribute to the overall depth of the market, providing a cushion against large trades that could move the price, while their cancellation can reduce this protective layer, affecting the market's ability to absorb large orders without significant price changes.

The order-book event variables we consider are listed below in the descending order of aggressiveness:

- BBO-Moving Trade: Market(able) buy or sell orders resulting in trades that move the BBO prices. For example, a market(able) buy order that consumes all depth at the best ask will move the best ask upward. So a resulting trade will be counted as the event.
- Non-BBO Moving Trade: Market(able) buy orders or sell orders resulting in trades
- BBO Improving Limit: Limit orders increasing the best bid or decreasing the best ask

- BBO Worsening Cancel: Cancel orders decreasing the best bid or increasing the best ask
- BBO-Depth Add Limit: Limit orders adding depth at the current best bid or at the best ask
- BBO-Depth Remove Cancel: Cancel orders removing depth at the current best bid or at the best ask
- Non-BBO-Depth Add Limit (within best five prices): Limit orders adding depth outside the best prices and within the best five prices.
- Non-BBO-Depth Remove Cancel (within best five prices): Cancel orders removing depth outside the best prices and within the best five prices.
- Non-BBO-Depth Add Limit (beyond best five prices): Limit orders adding depth beyond the best five prices
- Non-BBO-Depth Remove Cancel (beyond within best five prices): Cancel orders removing depth beyond the best five prices

It should be noted that there are potentially complicated interactions between the features and their lagged values resulting from traders' execution strategy. For example, a BBO-improving limit order followed by trade executions can be viewed as interactions between a buy-side execution algorithm and a high-frequency market making.

3.2.2 Order-book "state" variables

The second set of order-book features we consider characterizes the states of the order books. Note that order-book event variables and state variables are closely linked as it is the latter that induces changes in the former. Actually, one order-book event can change multiple state variables. For example, a limit order setting a new best bid will change the cumulative depth at different price levels.

Although the order-book events and states are closely linked, the former is typically not a perfect substitute for the latter. Including the order-book states might help the price prediction as the price impact of the same order-book event depends on market conditions. For example, Kwan, Philip, and Shkilko (2020) estimate the price impacts of different order-book actions conditioning on different order-book states such as the depth imbalance. A buy trade conditioning on a positive depth imbalance will have a different price impact compared to a negative depth imbalance.

- BBO-Depth Imbalance: the difference between depth at the best bid and depth at the best ask
- Non-BBO-Depth Imbalance (within best five prices): the difference between the cumulative depth within the best five bids and cumulative depth within the best five asks
- Non-BBO-Depth Imbalance (beyond best five prices): the difference between the cumulative depth at all bids and cumulative depth at all asks

3.3 Utilizing the granularity of the level-3 data

Order book data offers insights into trading activity with granularity ranging from Level 1 to Level 3. Each level provides different depths of information. Level 1 order book data is the most basic form, offering the highest bid price and the lowest ask price, which represent the top of the order book. This data is crucial for determining the current market price of a security and suffices for most investors making basic buy or sell decisions. However, it lacks information on market depth beyond the best available prices.

Level 2 order book data expands on Level 1 by including all public quotes of market makers and participants. It shows not only the best bid and ask prices but also the quantities available at those and other price levels within the order book. Level 2 data is valuable for understanding market depth and liquidity, identifying potential support and resistance points, and is particularly useful for day traders and those employing sophisticated trading strategies. Despite offering more depth, it doesn't reveal hidden orders or the intentions behind segmented large orders.

Level 3 order book data is the most detailed and is primarily used by market makers. It encompasses everything in Level 2, plus it allows market makers to enter quotes, execute orders, and view other market makers' orders. Level 3 data enables market makers to contribute to market liquidity and manage their trading and quoting strategies in real-time, providing the deepest insight into market dynamics.

To explore the information contained in the level 3 order-book messages data we have, we construct two other features that can only be constructed based on order-level data. Specifically,

- BBO Queue Length Imbalance: the difference between the number of orders at the best bid and the number of orders at the best ask
- Order Cancellation Time: the fill time of a resting limit order either through execution or cancellation

We provide a rationale for including the above two level-3 features. Market-making strategies rely heavily on interpreting the order book to anticipate short-term price movements and manage inventory risk. The lengths of orders queued in the order book play a pivotal role in these strategies by providing real-time insights into supply and demand dynamics. Market makers analyze the depth and distribution of buy and sell orders to identify potential support and resistance levels, which helps them set competitive bid and ask prices. A disproportionate length of orders on one side of the book may signal an impending price shift, prompting market makers to adjust their positions accordingly to capitalize on the movement or hedge against adverse shifts.

Additionally, understanding the order book's structure aids market makers in assessing the liquidity of the asset, crucial for executing large orders with minimal impact on the market price. This insight allows market makers to better gauge the timing for order execution, optimizing their spread capture while maintaining sufficient liquidity to fulfill their role in the market. In essence, the detailed information from the order book, including order lengths, is instrumental in informing market-making strategies, enabling these participants to provide liquidity efficiently while seeking to profit from the bid-ask spread in a constantly fluctuating market environment.

The timing of order cancellations in the order book is a crucial indicator for market makers and traders, offering insights into market sentiment, liquidity dynamics, and impending price movements. Rapid changes in cancellation rates can signal shifts in market sentiment or reactions to new information, influencing market makers to adjust their strategies for quote setting and inventory management. Moreover, patterns in order cancellations can hint at future price directions, allowing traders to preemptively position themselves (van Kervel 2015). High cancellation rates may also reflect changes in market liquidity, potentially increasing volatility, or indicate manipulative trading practices like spoofing. Consequently, monitoring order cancellation timing helps in refining trading and market-making strategies, ensuring better alignment with current market conditions and participant behaviors.

For each exchange, we construct the above features. Thus, we have in total $7 \times 15 = 105$ features at one point of time. As we include history in the prediction model, the number of total features, including the contemporaneous features and their lagged values, can become very large. For example, when including a history of past 25 events, the total number of features adds up to 2625. Thus, it naturally calls for ways of reducing the dimension of the feature space and machine learning models are well suited for such a task.

4 Methodology

4.1 The need for machine learning models

Before we go about discussing in detail the data and machine learning models we utilize for high-frequency return prediction, we want to first provide a motivation. In other words, why are machine learning models useful in such a context?

There are three main characteristics of high-frequency price prediction where linear models can fail. First, price impacts of trades and orders are non-linear. For example, Philip (2020) shows that permanent price impacts of trade flows are non-linear, and thus linear models such as VAR cannot capture it. Machine learning models are well-suited for high-frequency price prediction as they can naturally handle non-linear relationships

Second, price impacts of order-book actions highly depend on order-book states. For example, Kwan, Philip, and Shkilko (2020) estimate the price impacts of different order-book actions conditioning on different order-book states such as the depth imbalance.

Third, a sequence of order-book actions can be predictive of future price movements as they represent specific trading or market-making strategies. Take, for example, a scenario where market volatility spikes following a pattern of aggressive order cancellations at the best bid price, coupled with a surge in large hidden sell orders. A neural network can learn to anticipate the resulting price dip by identifying the non-linear interplay between these features. Similarly, a Gradient Boosting model might uncover how specific combinations of order-book imbalances and past trade volumes often precede a significant price move, capturing interactive effects without the need for explicit cross-term specification.

Arguably, all three aspects can be to some extent accounted for under the linear model framework. For example, to induce non-linear price impacts, one can include the transformed trade variables in the return equation. For example, it is a commonly used practice to include the signed squared trade variable or the signed square root of the trade variable to capture the nonlinear price impacts of the trade.

Second, to condition on market states, one can potentially include state variables in the return equation. However, the challenge is that the state variables are typically non-stationary and makes the estimation problematic. Third, to incorporate the temporal dependence, one can use an autoregressive structure with the help of for example vector-autoregressive models (VAR). However, the VAR structure is very restrictive.

4.2 Sketch of machine learning models

In each subsection, we introduce a machine learning model we utilize. Our aim is to provide a brief description so that a reader can understand the essential element of it. We refer readers to Kelly and Xiu (2023) for a more in-depth discussion.

We consider linear models such as elastic net and tree-based models such as gradient-boosted regression tree models. Besides tree-based models, we consider two other machine-learning models that are more suitable for dealing with sequential data. Long Short-Term Memory (LSTM) is motivated by its ability to capture temporal dependencies in sequence data, which is crucial in high-frequency financial markets where the sequence and timing of events can be predictive of future prices. LSTMs are particularly good at learning from long sequences without suffering from the vanishing gradient problem, making them suitable for capturing long-term dependencies. These characteristics make it a powerful tool for modeling time-series data in finance.

4.2.1 Linear models with penalties

Linear models with penalties have become indispensable in statistical modeling and machine learning, particularly for addressing overfitting and multicollinearity issues. The elastic net is a prominent regularization technique that combines the $L1$ penalty of lasso regression with the $L2$ penalty of ridge regression.

The elastic net approach aims to capitalize on both the regularization properties of ridge regression, which handles correlated features effectively and the feature selection properties of lasso regression, which can reduce the coefficients for less influential features to zero. This dual approach allows the elastic net to benefit from the strengths of both methods.

The optimization problem for the elastic net can be expressed as follows:

$$\min_{\beta} \left\{ \frac{1}{2n} ||\mathbf{Y} - \mathbf{X}\beta||_2^2 + \lambda \left(\frac{1-\alpha}{2} ||\beta||_2^2 + \alpha ||\beta||_1 \right) \right\} \tag{1}$$

where $\mathbf{Y}$ represents the vector of observations, $\mathbf{X}$ is the matrix of predictor variables, β is the coefficient vector, n is the number of observations, λ is the regularization parameter, and α is the mixing parameter that balances the ridge and lasso penalties.

Parameters λ and α are typically determined via cross-validation. The parameter λ controls the trade-off between model complexity and coefficient shrinkage, while α manages the balance between lasso and ridge penalties.

The elastic net is especially useful in situations with many correlated predictors or when the number of predictors p exceeds the number of observations n. By combining the group effect among correlated predictors and the selection of individual variables, the elastic net can provide superior predictive performance and more reliable feature selection.

4.2.2 Gradient boosted regression trees

The basic idea of decision trees in the context of machine learning is to create a model that predicts the value of a target variable by learning simple decision rules inferred from data features. A decision tree is built through a process of binary recursive partitioning, where the data is successively split according to certain criteria.

The decision rules are usually in the form of simple if-then-else conditions. The topmost node in a decision tree is known as the root node, which goes on to ask a question, and according to the answer (Yes/No), it splits the tree into branches or sub-trees. Subsequent nodes ask further questions, partitioning the dataset recursively until a prediction can be made for each instance. This process ideally continues until each leaf node is pure, meaning that it contains instances of only one class label or the stopping criteria are met.

One of the primary advantages of decision trees is their transparency and ease of interpretation—they can be visualized graphically, and decisions are easy to understand without requiring statistical knowledge to interpret them. However, decision trees can also be prone to overfitting if they become too complex. Techniques such as tree pruning and setting a maximum depth for trees are commonly used to avoid this issue.

A gradient-boosted regression tree (GBRT) is an ensemble learning model that combines the predictions from multiple decision trees to improve the model's performance. It operates on the principle of boosting, which involves sequentially adding trees, where each new tree attempts to correct the errors of the combined ensemble of the earlier trees. In a gradient-boosted regression tree model, each decision tree is typically a weak learner—meaning it does only slightly better than random guessing—however, when combined, these learners form a strong predictor. The "boosting" process adjusts for the shortcomings of existing trees by giving more weight to the observations that were poorly predicted in previous rounds, thus sharpening the accuracy on more challenging cases.

4.2.3 Long short-term memory (LSTM)

Tree models such as gradient-boosted regression tree and random forests are not built for dealing with sequence or time-series data. The Long Short Term Memory (LSTM) neural network is proposed by Hochreiter and Schmidhuber (1997) and is an advanced variant of Recurrent Neural Networks (RNNs), designed to effectively capture temporal dependencies and address the challenge of long-term information retention in sequential data. The LSTM's architecture is distinguished by its utilization of a complex system of gating mechanisms—namely the input, forget, and output gates—which orchestrate the modulation of information flow within the unit.

These gates function collaboratively to regulate the state of the memory cell, permitting the LSTM to maintain or erase information selectively. This feature is particularly critical in circumventing the vanishing gradient problem that undermines the capacity of traditional RNNs to learn from data points situated at distant intervals in time. Consequently, LSTMs are adept at preserving context over lengthy sequences, which is quintessential for rigorous time-series analysis.

In the context of financial time-series forecasting, the ability to recall and leverage long-span historical data facilitates more accurate prediction models. LSTMs can discern intricate patterns in market dynamics, allowing for the anticipation of future trends based on the temporal progression of variables. This characteristic renders them an invaluable asset in the domain of market microstructure analysis, where the prediction of short-term market fluctuations based on the sequencing of past trades and quotes is paramount. The LSTM's proficiency in handling sequential data extends its applicability to a broad range of time-series forecasting tasks, including but not limited to stock market prediction, economic trend analysis, and algorithmic trading. Bali, Goyal, Huang, Jiang, and Wen (2020) is one of the few papers that use LSTM for return prediction.

4.3 Performance evaluation

In assessing the predictive success of our models, we use the short-term midquote return over the next five, ten, 25 events as our target variables. Notably, both the features and target variables are sampled by the event-clock time, rather than wall clock time. The efficacy of our models is gauged through out-of-sample R^2 values, computed as follows:

$$R^2_{\text{OOS}} = 1 - \frac{\sum_i (Y_i - \widehat{Y}_i)^2}{\sum_i (Y_i - \overline{Y})^2} \tag{2}$$

where $\overline{Y}$ represents the mean of observed values within the out-of-sample dataset. An R^2_{OOS} value exceeding zero signifies that the model's predictive accuracy surpasses that of the naive benchmark, which is the out-of-sample mean, thus confirming the model's utility beyond mere average estimation. Note that R^2_{OOS} can be negative as the fitted values can be arbitrarily bad.

4.4 Model fitting

In machine learning, the training, validation, and testing procedures involve subdividing data to prevent overfitting and ensure model generalizability. The training set is used to fit the model, the validation set to tune hyperparameters and select the best model iteration, and the testing set to assess the final model's performance on unseen data. This process helps in finding a balance between model complexity and predictive power.

For computational considerations, we segment each trading day into thirteen half-hour time intervals. We adopt an inter-day rolling window approach to split the data into training, validation, and testing sets. This method involves using successive intervals across days for different purposes. For example, the interval from 09:30 to 10:00 on one day is used for training, the same interval on the next day for validation, and the following day for testing.

4.5 Hyper-parameter tuning

The selection of hyper-parameters is crucial to the performance of our models. We consider a range of values for each parameter, with the aim of optimizing model performance. The considered hyperparameters include:

- Elastic-net regularization with $\rho = 0.5$ and $\alpha \in \{0.1, 0.01, 0.001, 0.0001\}$.
- Boosted Regression Trees (BRT) with the depth of trees $\in \{1, 2\}$, number of trees $\in \{100, 1000\}$, and learning rate $\in \{0.01, 0.1\}$.
- Multilayer Perceptron (MLP) with units $((32, 16), (32))$.
- LSTM with units $((32, 16), (32))$ for LSTM layers and $((32, 16), (32))$ for MLP layers.

These hyper-parameters are iteratively adjusted based on the model's performance on the validation set to identify the combination that yields the most accurate predictions.

4.6 Implementation details

We detail several implementation choices. First, we use an event clock for our prediction analysis. Event-clock or volume clock is a commonly used sampling method in market microstructure analysis (Easley, López de Prado, and O'Hara 2012). The benefit of adopting an event/volume clock is that volatility is smoothed and return is close to normal. The idea is simple. Volatility and volume come together: when there is a large price change, there is a high level of market activity. So if we sample based on the same number of shares or transactions, we can roughly have the same volatility per sampling bucket.

We use LightGBM[4] to implement the GBRT model, which features leaf-wise tree growth. We implement the LSTM model with TensorFlow and Keras[5].

5 Results

5.1 Prediction performances of machine learning models

Table 1 reports the forecast performance of different models in terms of mean squared error (MSE) for the validation and test dataset, as well as R^2_{OOS} values. The models compared include Ordinary Least Squares (OLS), Elastic Net, Gradient Boosting Regression Tree (GBRT), Multilayer Perceptron (MLP), and Long Short-Term Memory (LSTM).

There are two notable observations. First, all machine learning models perform better compared with the simplest OLS model. For example, the GBRT model exhibits the best out-of-sample prediction performance with an R^2_{OOS} value of about 6%, which is significantly higher the R^2_{OOS} value of about 1.5% of the OLS model.

Second, there is considerable variation in performance within the machine learning models. Two machine learning models, GBRT and LSTM, outperform others including Elastic Net and MLP. Compared with Elastic Net, tree models such as GBRT can allow for non-linear price impacts and the interactions between order-book features, resulting in better prediction performances. In addition, the LSTM model outperforms the plain-vanilla neural network, MLP, by the a large margin. It suggests that it is important to incorporate temporal dependence in the features. They capture long-term temporal dependence in the feature time series which can result from algorithmic trading strategies, e.g., informed traders slice and dice their orders to minimize price impact.

[4] https://lightgbm.readthedocs.io/en/stable/index.html

[5] https://keras.io/.

Table 1. Prediction performance across models. This table reports the mean squared error (MSE) and out-of-sample R^2 (%) for all models. OLS is the ordinary least squares. Elastic Net is the linear model with an elastic net penalty. GBRT is the gradient-boosted regression tree model. MLP is the multi-layer perceptron model, and LSTM is the long short-term memory model. The numbers are averages across all tickers and intraday intervals. Note that the MLP and LSTM models have higher MSE values because the feature values are normalized.

Model	MSE-Validation	MSE-Test	R^2_{OOS}
OLS	0.28	0.24	1.45
Elastic Net	0.28	0.24	1.61
GBRT	0.27	0.23	6.13
MLP	41.98	35.74	1.77
LSTM	41.6	35.16	3.38

5.2 Source of prediction performances of machine learning models

Next, we examine what leads to the superior prediction performances of the machine learning models. There are potentially three different sources. First, machine learning models can capture non-linear price impacts of the order-book events. Second, they can capture state dependence on order-book actions. For example, price discovery can depend on different states of the limit order book as shown in (Kwan, Philip, and Shkilko 2020). Third, they can capture the temporal dependence of the order-book events. For example, a sequence of order-book actions might represent the trading strategy of a particular trader.

Note that although we distinguish the third source from the second source. In some machine models, they are equivalent in terms of implementation. For example, in order to incorporate order-book event history, we include lagged variables of the order-book features in a tree-based model. So the state dependence will be captured by the interactions between the event features and state features. Instead, temporal dependence will be captured by the interactions between the current observations of the features and their lagged values.

5.2.1 Prediction performance from state dependence

As mentioned above, the superior performance of machine learning models can result from them allowing for more flexible state dependence. In order to distinguish the state dependence from temporal dependence, we only include the most recent history, that is, the last order-book event and the last order-book state. So we rule out the possibility for the model to learn any temporal dependence within the features themselves or across features.

We train the model with two sets of order-book features: The first set only includes the order-book event variables; the second set includes the order-book event variables and the order-book state variables. If the state dependence is important for short-term price prediction, we can see an significant increase in the out-of-sample R^2 when including extra information about the order-book conditions. Table 2 reports the results. We select the best-performing model, GBRT, as the representing machine learning model

It shows that across all levels of the order-book features, including order-book state variables in addition to the order-book event variables significantly increases the out-of-sample R^2. For example, when using all features constructed from the level-3 dataset, the out-of-sample R^2 with only order-book event variables included is 2.66%, and it more than doubles to 5.29% when including the state variables, and thus the interactions between the event and state variables. Importantly, adding the order-book state variables does not boost the performance of the OLS model to the same extent. The out-of-sample R^2 remains to be close to zero although they turn positive. But in some cases such as when including all level 3 features, the performance even worsens.

The results show that machine learning models such as the GBRT can capture the "state-dependence", that is, interactions between the order-book states and the order-book events, and help improve its forecasting performance. In contrast, OLS can not achieve the same goal due to its limitations.

5.2.2 Prediction performance from temporal dependence

The second source of prediction performance of machine learning models can come from them being able to capture the temporal dependence of the features. For tree models, we include lagged variables of the features so that during the splitting of the trees, flexible interaction can be realized

Table 2. Prediction performance: state-dependence. This table reports the out-of-sample R^2 (%) for OLS and GBRT with only order-book event variables are included in the model compared with when both the order-book event and state variables are included. OLS is the ordinary least squares. GBRT is the gradient-boosted regression tree model. Level 1 and so on indicate the depth of the order book used when constructing the features. The numbers are averages across all tickers and intraday intervals.

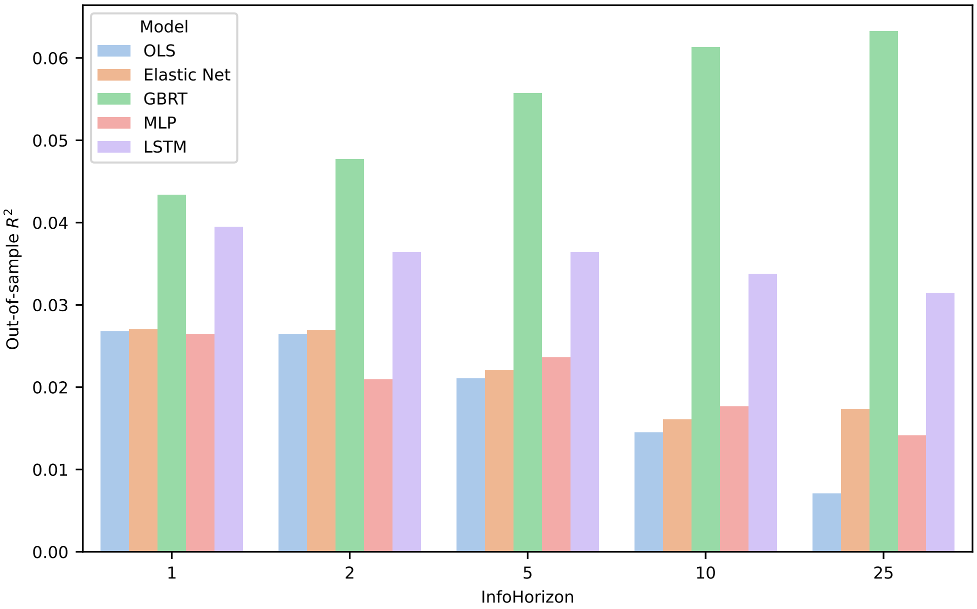
	Feature Type	Order-book Event	Order-book Event + State
Model	Feature Level		
GBRT	Level1	2.46	5.29
	Level2	2.63	5.43
	Level2-Top5	2.42	5.24
	Level3	2.66	5.42
OLS	Level1	-0.74	0.61
	Level2	-0.74	0.38
	Level2-Top5	-0.73	0.41
	Level3	-0.85	-1.1

between the different lags of the same feature or across two different features. Of course, including more lagged values, or a long history can allow for a potentially longer temporal dependence structure.

Figure 1 plots the out-of-sample R^2 for all models with different lengths of history included. It shows that, as we include a longer history which allows for more temporal dependence of the order-book events and states, the out-of-sample R^2 increases significantly. For example, when only one lag is included, the GBRT generates an out-of-sample R^2 of about 4.34. In contrast, when 25 lags are included, the out-of-sample R^2 increases to 6.32%. More importantly, the increase only happens with the GBRT, the best-performing machine learning model. For the OLS model, its out-of-sample R^2 actually decreases significantly as we include more lags in the model.

The results above show the second source of the superior forecasting performance of the machine learning models indeed comes from them being able to capture the "temporal dependence" of the features. Taking stock, machine learning models such as GBRT can effectively use the rich information when including a long history of both the order-book event and state variables and improve its forecasting performance.

Figure 1. Information horizon and forecast performance. This figure plots the out-of-sample R^2 of the different models for different information horizons. Information horizon is measured in order-book event time. OLS is the ordinary least squares. Elastic Net is the linear model with the elastic net penalty. GBRT is the gradient-boosted regression tree model. MLP is the multi-layer perceptron model, and LSTM is the long short-term memory model. The table below shows the detailed breakdown of the out-of-sample R^2.



InfoHorizon	1	2	5	10	25
Model					
OLS	2.68	2.65	2.1	1.45	0.71
Elastic Net	2.7	2.7	2.21	1.61	1.74
GBRT	4.34	4.77	5.57	6.13	6.32
MLP	2.65	2.09	2.36	1.77	1.41
LSTM	3.95	3.64	3.64	3.38	3.14

5.3 Feature importance analysis

In the following section, we use the results from machine learning models to answer a key question in microstructure or finance in general: How does the price discovery happen? For example, given multiple exchanges trading the same security, which exchange plays the most important role in discovering the price? Or are price levels and liquidity deep in the order book important for price discovery? Are the high cost of obtaining the most granular data such as the exchange direct fees justified?

5.3.1 Permutation importance

In order to assess the significance of a feature or a set of features in a predictive model's accuracy, we use a popular technique in machine learning called permutation importance. The permutation importance technique is model-agnostic, meaning it can be applied regardless of the model type. The process involves several steps:

1. We first train a model on the dataset and evaluate it using an appropriate performance metric such as the mean squared error (MSE).

2. For each feature or set of features, we permute its values across all samples.

3. The importance of a feature or a set of features is quantified as the decrease in the model's performance metric due to its permutation. The greater the decrease, the more important the feature is considered.

4. This process is repeated for all features or sets of features in the dataset to assess each one's importance. Features or sets of features can then be ranked based on their importance scores, providing insights into which features most significantly drive the model's predictions.

Advantages of this method include its applicability to any model type, straightforward concept, and robustness to correlations among features. However, it may be computationally expensive, introduce randomness to importance scores, and not fully capture complex interactions between features. Despite these limitations, permutation feature importance remains a valuable tool for feature selection and understanding model behavior.

For example, if we aim to assess the importance of the Trade variable on NYSE, we simply shuffle the values of the the variable across time, so that we make it into a "noise" variable. If the resulting increase in MSE is large, then it is clear that the NYSE trade variable plays an important role in forecasting the high-frequency price. It applies not only to a single feature but also to a set of features. For example, if we aim to assess the importance of a particular level of the order book, for example, the best bid and ask, then we can include all features related to that level: liquidity-adding and -removing orders at the best bid and ask, or best price-improving orders. Then we shuffle all related features simultaneously so that the whole group of features turn into pure "noises".

5.3.2 LOB level importance across different models

Figure 2 shows the feature importance scores for various order book features across three different predictive models: Ordinary Least Squares (OLS), ElasticNet, and Gradient Boosting Regression Tree (GBRT). The features include order book events like trades and cancellations at different levels, with a distinction between those affecting the Best Bid and Offer (BBO) and those within 5 levels from the BBO.

The color intensity on the heatmap represents the magnitude of the feature importance scores. A darker shade indicates a higher importance. In this heatmap, "Trade (BBO-Moving)" exhibits the highest importance in the GBRT model, while "Trade" without the BBO-moving distinction is more influential in the OLS and ElasticNet models. This shows that GBRT assigns a higher weight to the top-of-book features compared to linear models. The relative importance of other features like "Add (BBO)" and "Cancel (> 5Lv1)" appears to be lower, as indicated by the lighter shades.

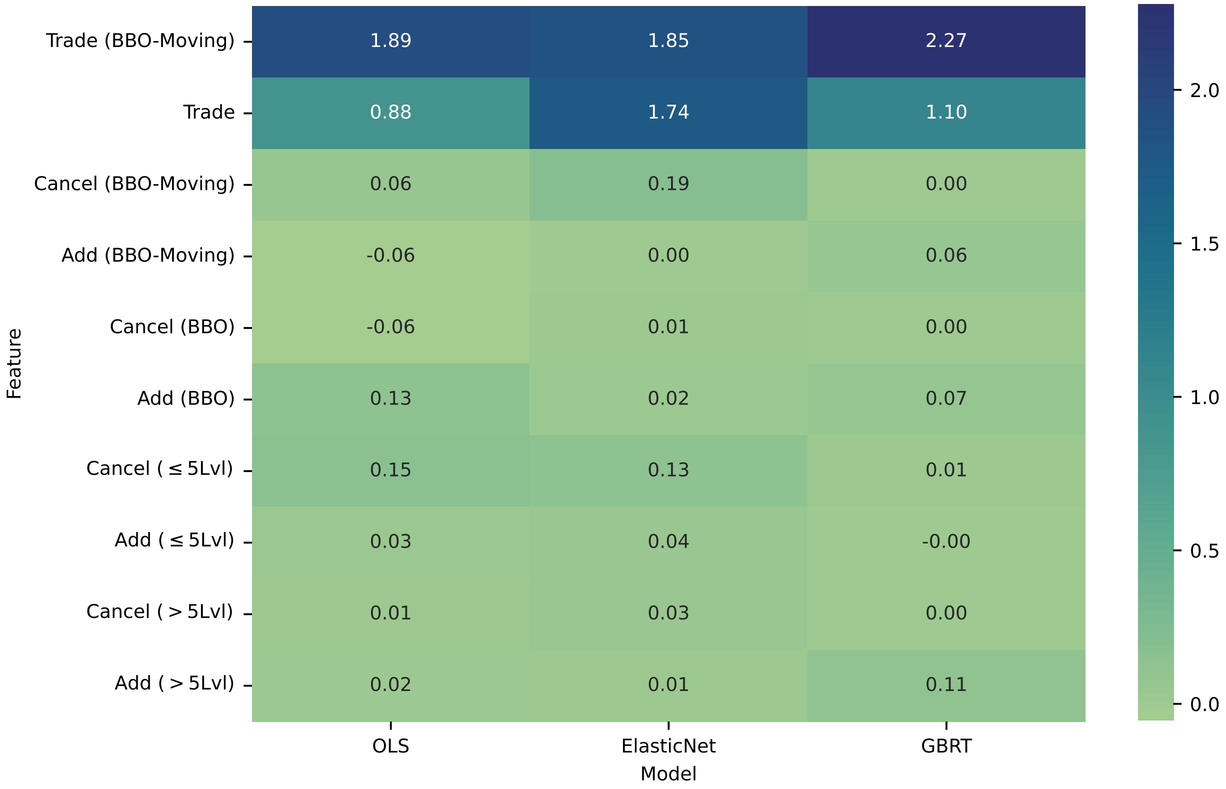
The results suggest that trading events, particularly those moving the BBO, are key predictors of price movements, with their influence varying across different modeling approaches. The GBRT model seems to heavily weigh the BBO-moving trades, possibly due to its ability to capture complex non-linear relationships in the data.

5.3.3 LOB level importance across different information horizons

Figure 3 plots the feature importance scores for a range of order book features across different information horizons including 1, 2, 5, 10 and 25 events respectively. The features include "Trade (BBO-Moving)", "Trade", "Cancel (BBO-Moving)", and "Add (BBO-Moving)" among others, with colors representing the magnitude of importance, darker indicating higher importance.

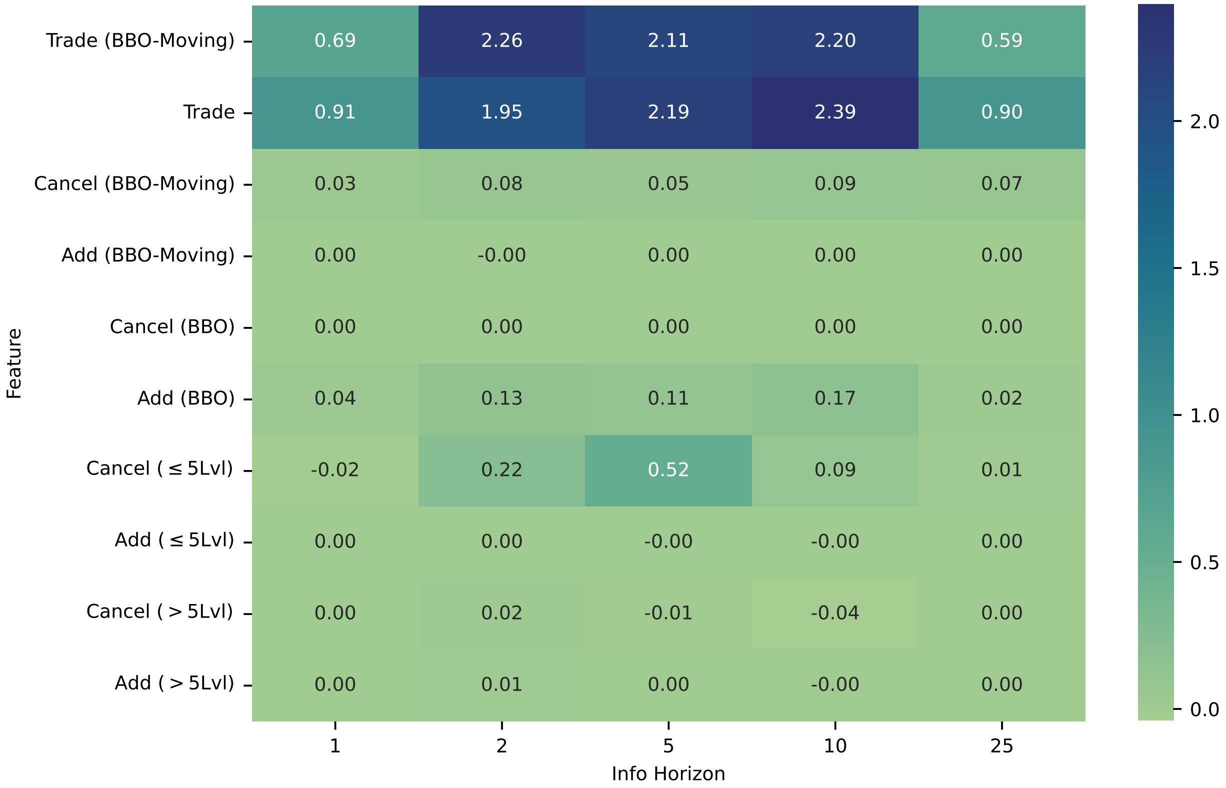
It shows that "Trade" and "Trade (BBO-Moving)" features show significant importance across

Figure 2. LOB level importance. This figure plots the permutation importance of different levels of the order-book. The permutation importance of a level is calculated as the relative increase in out-of-sample MSE when we shuffle the ordering of all variables pertinent to the particular level. For example, the tile in the northwest corner has a value of 1.89, meaning that the MSE will increase by 1.89% if the Trade (BBO-Moving) feature across all exchanges is permuted.



all horizons, with their impact peaking at the 10-events horizon. "Cancel (≤ 5Lv1)" sees a notable peak at the 5-events horizon, suggesting a temporal sweet spot where this feature is particularly predictive. The importance scores generally decline for longer horizons (25 events), except for the "Trade" features, which maintain higher scores, indicating that trading activity, especially that which impacts the BBO, is a consistent predictor over time. Other features, especially cancellations and additions further from the BBO, show less importance, indicating that the model prioritizes events closer to the trade execution price for its predictions. The results show that the relevance of different order book events to price prediction changes with the length of historical data considered.

Figure 3. LOB level importance across different information horizons. This figure plots the permutation importance of different levels of the order-book across different information horizons. The permutation importance of a level is calculated as the relative increase in out-of-sample MSE when we shuffle the ordering of all variables pertinent to the particular level. For example, the tile in the northwest corner has a value of 0.69, meaning that the MSE will increase by 0.69% when we only include one lag for all variables and if the Trade (BBO-Moving) feature across all exchanges is permuted. We use the best-performing GBRT model to calculate the permutation importance.

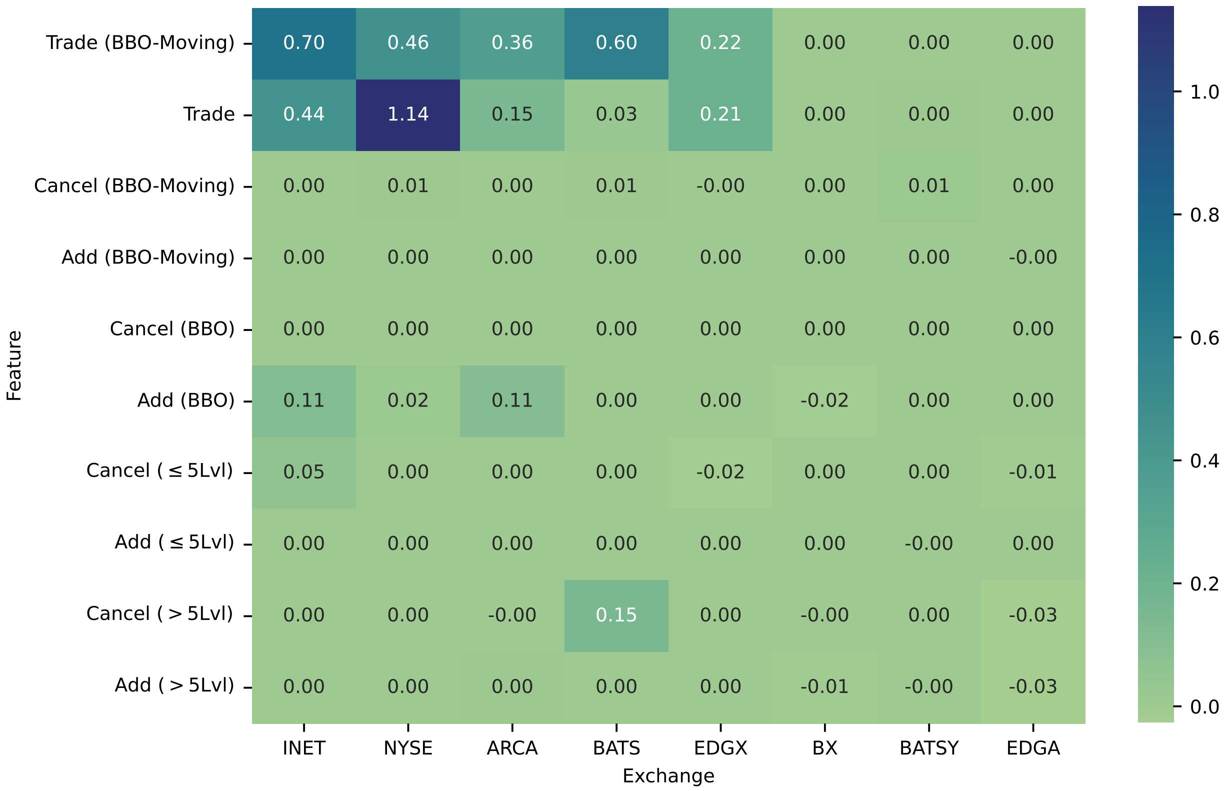


5.3.4 Variable importance

Figure 4 plots the feature importance scores across various features and exchanges. The features, such as “Trade (BBO-Moving)”, “Trade”, “Cancel (BBO-Moving)”, and so on, are indicators used in a model to predict stock price movements. Each cell’s color intensity reflects the importance score, with darker shades representing higher importance.

It appears that “Trade” on the NYSE has the highest score, suggesting it’s a strong predictor for short-term price prediction. “Trade (BBO-Moving)” also shows significant importance across various exchanges but particularly on INET (NASDAQ) and BATS. The general trend indicates that trading activity, especially BBO-moving trades, is a vital indicator of future price changes across different marketplaces. Features related to cancellations and additions, especially beyond

Figure 4. Granular feature importance. This figure plots the permutation importance of all exchange-specific features. The permutation importance of a feature is calculated as the relative increase in out-of-sample MSE when we shuffle the ordering of all variables pertinent to the particular level. For example, the tile in the northwest corner has a value of 0.70, meaning that the MSE will increase by 0.70% when if the Trade (BBO-Moving) feature on INET (NASDAQ) is permuted. We use the best-performing GBRT model to calculate the permutation importance.



the top 5 levels, seem to hold less predictive power.

6 Conclusion

In our paper, we demonstrate that machine learning models can achieve much better performance over traditional econometric models in forecasting short-term price movements in a fast and fragmented market with big data challenges. Among them, the Gradient Boosted Regression Tree (GBRT) model has the highest out-of-sample R^2 of 6.32% with tick-by-tick features constructed from level-three order book message data and based on a sample of highly liquidity US stocks. We further show that the superior performance of the machine learning models comes from their ability to capture temporal dependence and the interactions among features. The inclusion of historical order-book information is shown to be critical, with the GBRT model significantly benefiting from longer information horizons. Finally, through permutation importance analysis, we interpret the prediction results of machine learning models, showing that top-of-book features such as trade executions and order updates are the most influential predictors of short-term price movements, with the GBRT model assigning a higher weight compared to linear models.

A Construction of Order Books from Direct Feeds Data

INET, BX, PSX: Trade messages from the display and hidden orders include details such as price and quantity, with 'side' information absent for hidden orders. Hidden orders are denoted by an order_id of "0" and flagged with "h". Auction messages, flagged as "SOX" or "SCX", lack comprehensive details. Modification messages are back-filled with information from the original add messages.

NYSE: Similar handling of display order trades as INET, BX, PSX, but hidden orders lack both 'side' and order_id, flagged with "h". Auction trades have missing details and are flagged "SOX"/"SCX" for single trades and "n"/"hn" for multiple trades. Modification messages require filling in missing information from the original add messages.

NYSE Arca: Display order trades include complete details, with ISO trades flagged as "s". Hidden order trades include 'side' but lack an order_id. Auction messages are flagged "nSO" or "nSC". Modifications do not include price and side, necessitating a fill-in from add messages.

BATS, BATSY: Trades from display orders follow the standard model, while hidden orders are given a new order_id ending with "?". Auction trades are flagged "nSO" or "nSC".

IEX and AMEX: These exchanges provide quote messages rather than trade messages.

References

Bali, Turan G., Amit Goyal, Dashan Huang, Fuwei Jiang, and Quan Wen (July 24, 2020). *Predicting Corporate Bond Returns: Merton Meets Machine Learning*. URL: https://papers.ssrn.com/abstract=3686164 (visited on 04/06/2023). preprint.

Brogaard, Jonathan, Terrence Hendershott, and Ryan Riordan (Aug. 2019). "Price Discovery without Trading: Evidence from Limit Orders". In: *The Journal of Finance* 74.4, pp. 1621–1658.

Bryzgalova, Svetlana, Markus Pelger, and Jason Zhu (Aug. 25, 2019). *Forest Through the Trees: Building Cross-Sections of Stock Returns*. URL: https://papers.ssrn.com/abstract=3493458 (visited on 04/08/2024). preprint.

Cao, Sean, Wei Jiang, Junbo L. Wang, and Baozhong Yang (May 5, 2021). *From Man vs. Machine to Man Machine: The Art and AI of Stock Analyses*. URL: https://papers.ssrn.com/abstract=3840538 (visited on 04/08/2024). preprint.

Colliard, Jean-Edouard, Thierry Foucault, and Stefano Lovo (2022). "Algorithmic Pricing and Liquidity in Securities Markets". In: *SSRN Electronic Journal*.

Cong, Lin William, Ke Tang, Jingyuan Wang, and Yang Zhang (Aug. 1, 2021). *AlphaPortfolio: Direct Construction Through Deep Reinforcement Learning and Interpretable AI*. URL: https://papers.ssrn.com/abstract=3554486 (visited on 03/15/2023). preprint.

Easley, David, Marcos M. López de Prado, and Maureen O'Hara (May 2012). "Flow Toxicity and Liquidity in a High-frequency World". In: *Review of Financial Studies* 25.5, pp. 1457–1493.

Eun, Cheol S. and Sanjiv Sabherwal (2003). "Cross-Border Listings and Price Discovery: Evidence from U.S.-Listed Canadian Stocks". In: *The Journal of Finance* 58.2, pp. 549–575.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu (May 1, 2020). "Empirical Asset Pricing via Machine Learning". In: *The Review of Financial Studies* 33.5, pp. 2223–2273.

Hagströmer, Björn and Albert J. Menkveld (Feb. 13, 2023). *Trades, Quotes, and Information Shares*. URL: https://papers.ssrn.com/abstract=4356262 (visited on 02/13/2023). preprint.

Harris, F.H.DeB., T.H. McInish, and R.A. Wood (2002). "Security Price Adjustment across Exchanges: An Investigation of Common Factor Components for Dow Stocks". In: *Journal of Financial Markets* 5.3. Cited By :156, pp. 277–308.

Hasbrouck, Joel (Mar. 1991). "Measuring the Information Content of Stock Trades". In: *The Journal of Finance* 46.1, pp. 179–207.

— (Sept. 1995). "One Security, Many Markets: Determining the Contributions to Price Discovery". In: *The Journal of Finance* 50.4, pp. 1175–1199.

Hasbrouck, Joel (Dec. 2003). “Intraday Price Formation in U.S. Equity Index Markets”. In: *The Journal of Finance* 58.6, pp. 2375–2400.

Hendershott, Terrence and Charles M. Jones (2005). “Island Goes Dark: Transparency, Fragmentation, and Regulation”. In: *Review of Financial Studies* 18.3, pp. 743–793.

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). “Long Short-Term Memory”. In: *Neural Computation* 9.8, pp. 1735–1780.

Kelly, Bryan T. and Dacheng Xiu (July 1, 2023). *Financial Machine Learning*. URL: https://papers.ssrn.com/abstract=4501707 (visited on 07/14/2023). preprint.

Kwan, Amy, Richard Philip, and Andriy Shkilko (Oct. 13, 2020). *The Conduits of Price Discovery: A Machine Learning Approach*. URL: https://papers.ssrn.com/abstract=3710491 (visited on 03/01/2023). preprint.

Li, Bin and Alberto G. Rossi (Sept. 27, 2020). *Selecting Mutual Funds from the Stocks They Hold: A Machine Learning Approach*. URL: https://papers.ssrn.com/abstract=3737667 (visited on 04/05/2023). preprint.

Menkveld, Albert J. (Apr. 1, 2018). “High-Frequency Trading as Viewed through an Electron Microscope”. In: *Financial Analysts Journal* 74.2, pp. 24–31.

O’Hara, Maureen and Mao Ye (June 1, 2011). “Is Market Fragmentation Harming Market Quality?” In: *Journal of Financial Economics* 100.3, pp. 459–474.

Philip, R. (Apr. 1, 2020). “Estimating Permanent Price Impact via Machine Learning”. In: *Journal of Econometrics* 215.2, pp. 414–449.

Putniņš, Tālis J. (Sept. 2013). “What Do Price Discovery Metrics Really Measure?” In: *Journal of Empirical Finance* 23, pp. 68–83.

Rossi, Alberto G. and Stephen P. Utkus (Mar. 10, 2020). *Who Benefits from Robo-advising? Evidence from Machine Learning*. URL: https://papers.ssrn.com/abstract=3552671 (visited on 04/04/2023). preprint.

Van Kervel, Vincent (July 1, 2015). “Competition for Order Flow with Fast and Slow Traders”. In: *The Review of Financial Studies* 28.7, pp. 2094–2127.