Inferring Trade Directions in Fast Markets

This Version: May 25, 2019

Simon Jurkatis*,[a]

[a] *Bank of England*

Abstract

The classical trade initiator label that identifies the liquidity demander (buyer or seller) in financial markets transaction data remains high in demand in research on financial topics. Obtaining that label via so-called trade classification algorithms has been difficult recently due to an increase in the frequency of quote changes and historically due to lags between reported trades and quotes. This paper proposes a new method and demonstrates that it outperforms the established algorithms. In particular, using an ideal dataset of stock market transactions I impose various deficiencies to simulate the characteristic problems of usual data records and find that the new method: (i) reduces misclassification rates by up to half, (ii) considerably reduces the bias in the estimation of the order imbalance, particularly for extreme order imbalances, and (iii) improves the precision of high-frequency liquidity measures. This paper also provides some indirect evidence that applications to the popular DTAQ data would benefit from the use of the new method.

Keywords: Trade Classification Algorithm, Trade Initiator, Liquidity, Order Imbalance, Informed Trading, Limit Order Book, Market Microstructure
JEL Classification: C8, G14, G17, G19

*E-mail: simon.jurkatis@bankofengland.co.uk. Tel.: +44 203 461 8588. I would like to thank Bidisha Chakrabarty, Emanuel Gasteiger, Dieter Nautz, Georg Weizsäcker and the participants at the Seminars at Freie Universität Berlin, the CFE-CMStatistic Conference, the RES PhD Meeting and at the Bank of England for their valuable comments.

1 Introduction

Recently, Easley et al. (2012, 2016) proposed a new mechanism to classify financial markets transactions into the trading volume of informed and uninformed traders. Their method is motivated by two observations of modern financial markets: first, the classical classification algorithms, which derive the liquidity demanding party—traditionally assumed to be the informed one—for each trade, do not provide accurate results due to the effect of today's fast markets on the data records, and second, the liquidity demander is not a uniformly valid indicator for the informed trader in the first place.

Despite the second observation, the trade direction of the liquidity demanding side of the order flow remains a popular indicator of informed trading (see, e.g., Bernile et al., 2016; Chordia et al., 2017; Hu, 2014, 2017; Muravyev, 2016) the appropriateness of which is context specific but particularly sensible when informed traders demand immediacy for their transactions in order to gain most from their informational advantage. In these cases, studies rely on the classical classification algorithms, most prominently the Lee and Ready (1991) algorithm, to obtain the indicator of the liquidity demanding side of the transaction, the trade initiator, as do traditional measures of market liquidity (Huang and Stoll, 1996; Fong et al., 2017). Notwithstanding the challenges posed by today's fast markets, however, little attempts have been made to adjust or design new algorithms to obtain the trade initiator for more recent data.

In this paper, I propose a new algorithm to classify transactions into the orders of liquidity demanders and suppliers and show that it outperforms the traditional alternatives, particularly under the challenging conditions of fast markets. The proposed algorithm also provides a remedy to the frequently reported deteriorating effect of time-delays between recorded trades and quotes on the classification accuracy when studying historical data.

The established methods, most notably the algorithms of Lee and Ready (1991) (LR), Ellis et al. (2000) (EMO) and Chakrabarty et al. (2007) (CLNV), classify trades based on the proximity of the transaction price to the quotes in effect at the time of the trade. This is problematic due to the increased frequency of order submission and cancellation. For example, the MTAQ dataset, the consolidated tape of U.S. stock exchanges and the most popular intra-day dataset for research in equities (Holden and Jacobsen, 2014), is timestamped to seconds.[1] Angel et al. (2015) record an

[1]The same is true for other equity and non-equity databases. For example, the equity trans-

average of almost 700 quote changes per minute (i.e. more than 11 per second) for all stocks in the MTAQ dataset at the peak in 2012.[2] With several quote changes taking place at the time of the trade, it is not clear which quotes to select for the decision rule of the algorithm and the wrong choice impedes its accuracy.

Though the MTAQ remains a popular database (see, e.g., Hu, 2014; Bernile et al., 2016; Chordia et al., 2016, 2017), the Daily Trade and Quote data (DTAQ) provides a now common alternative that is timestamped to the millisecond.[3] Still, with order submission and cancellation rates of microseconds or faster, even data timestamped to milliseconds will not be sufficiently precise (O'Hara, 2015).

Using data from the DTAQ Chakrabarty et al. (2015) find an accuracy of the LR algorithm of around 85% even though trades are aggregated over short intervals such that false classifications can cancel each other out. The increase in trading frequency is also made responsible for the poor performance of the LR algorithm found by Panayides et al. (2014) in a study using data from Euronext that is timestamped to seconds. In their case, the LR algorithm classifies only 79% of trades correctly.

Older studies analyzing the accuracy of the LR algorithm, as well as the alternative EMO and CLNV algorithms, find classification accuracies ranging from 75% to 93% (see, e.g., Theissen, 2001; Finucane, 2000; Chakrabarty et al., 2007; Odders-White, 2000; Ellis et al., 2000; Lee and Radhakrishna, 2000). The concern in many of these studies, however, has been more that of a time-delay between the reported quotes and trades. The effect of the report delay is the same as that of insufficiently precise timestamps: the true trade-quote correspondence is unknown to the researcher, impeding the accuracy whenever the wrong quote is used in classifying a trade.

Although technical advances have decreased the potential for report delays, there is still disagreement by how much quote times should be lagged in the MTAQ.[4]

action data of the German Financial Supervisory Authority studied in Kremer and Nautz (2013), which contain all trades conducted on German exchanges, and the futures transaction data studied in Bernile et al. (2016) are timestamped to seconds.

[2]This number includes infrequently traded stocks and intra-day periods of low traffic. The number can be expected to be much higher if one would condition on seconds at which trades occurred.

[3]From August 2015 onwards DTAQ data is timestamped to microseconds, from October 2016 onwards to nanoseconds for NASDAQ trades.

[4]Chakrabarty et al. (2012) recommend to lag quotes by 1 second, Henker and Wang (2006) recommend to use the last quote from the second before the trade, Piwowar and Wei (2006) and Vergote (2005) find optimal delay times for quotes between 1 and 2 seconds, Peterson and Sirri (2003) and Bessembinder (2003) recommend a 0 lag for quotes, though they consider only 5 seconds intervals ranging from 0 to 30 seconds. Reviewing all published papers between 2006 and 2011 in the Journal of Finance, Journal of Financial Economics and the Review of Financial Studies, Holden

Moreover, with increased trading frequency and increasingly precise timestamp precisions the problem of even small report delays may again be elevated. In fact, I provide some evidence that the misalignment problem is not fully amended in today's DTAQ data, which has important implications for the application of trade classification algorithms to these data.[5]

The algorithm proposed in this paper takes a new approach to these issues. Instead of selecting a single pair of ask and bid quotes *before* the actual classification step, it matches the transaction to its corresponding quote at the same time as it is classified. The idea is that a trade executed against the ask must leave its footprint on the ask-side, while a trade against the bid must leave its footprint on the bid-side. Finding these footprints is equivalent to *simultaneously* finding the quote corresponding to a trade *and* classifying it.

Propositions of new classification methods in response to the new reality of financial markets are rare. Easley et al.'s (2012) Bulk Volume Classification (BVC) algorithm is a more radical change in method that is not intended to find the trade direction of the liquidity demanding party. Chakrabarty et al. (2015) show that the BVC algorithm is inferior to the LR algorithm at that task, similarly to Panayides et al. (2014), whose results, however, depend on specific modeling choices. Holden and Jacobsen (2014) recommend to interpolate trade and quote times of imprecisely timestamped data and then to apply the traditional algorithms. Their interpolation method has been applied, for example, in Chordia et al. (2017) and Brennan et al. (2018). I find, however, that their method can *strongly* impede the classification accuracy.

The responses to the problem of delayed trade times have been recommendations to lag quotes by a certain amount based on indirect evidence (see, e.g., Lee and Ready, 1991; Bessembinder, 2003; Henker and Wang, 2006; Piwowar and Wei, 2006; Chakrabarty et al., 2012) or by parametric estimation of optimal delay times (Vergote, 2005; Rosenthal, 2012). Changes in the underlying decision rule of the LR algorithm proposed by EMO and CLNV deal with the classification of trades that were executed off the quotes, as does the less popular algorithm of Blais and Protter (2012), rather than dealing with the problem of frequent quote changes or delayed trade times. Similarly, the algorithm of Rosenthal (2012) requires a choice of a single

and Jacobsen (2014) find that in 28 articles using the MTAQ data 7 used the prior-second rule, 3 the same-second rule, 5 the five-second rule and rest did not provide information on the timing-rule used to match quotes and trades.

[5] I elaborate on this issue in Section 5.

pair of ask and bid quotes for its decision rule and is thus subject to the same errors as the LR, EMO and CLNV algorithm.

I use data from NASDAQ's electronic limit order book to evaluate the new algorithm against the LR, EMO and CLNV algorithm. The sample runs from May to July 2011 with a total of over 134 million transactions timestamped to nanoseconds. The data contain the trade direction of the executed standing order in the limit order book. Hence, the liquidity supplying and demanding side for each transaction is known, which allows me to evaluate the ability of the algorithms to recover this information.

The NASDAQ data, of course, do not contain the same number of trades and quote changes as, for example, the consolidated tape and possibly other high-frequency databases.[6] This is, however, not a problem *per se* as we are interested in the effect of high order submission and cancellation rates *relative* to the data timestamp precision. To simulate this problem I truncate the timestamp precision at frequencies ranging from nanoseconds to seconds. This corresponds to a median number of quote changes during the time of trades ranging from 1 to 17. To analyze the problem of lagging transaction timestamps I add exponentially distributed noise to the original trade times.

The results provide a clear message: the new algorithm outperforms the traditional trade classification algorithms. First, at every considered timestamp precision the new algorithm does not perform worse than the others and it offers considerable improvement in classification accuracy at lower timestamp precisions. For example, for the data timestamped to the second the new algorithm correctly classifies the trade initiator for 95% of the trading volume, compared to 90% for the best competitor, the EMO algorithm. Similar improvements are achieved under randomly delayed trade times of average intensities of up to one centisecond. Moreover, the improved accuracy of the new algorithm translates into considerable improvements in the estimation of a typical indicator of informed trading, the order imbalance, and measures of market liquidity, i.e. the dollar effective spread, the dollar price impact, and the dollar realized spread. These results are robust against varying assumptions on the data structure that affect the way the new algorithm can use the data. An analysis of the determinants of misclassification shows that hidden orders are the main driver for falsely attributing the trade initiator to either the buyer or the seller. Other, economically relevant, factors such as return volatility or trading volume play

[6] For example, Angel et al. (2011) report that NASDAQ's market share in NASDAQ-listed stocks decreased from 53% in April 2005 to around 30% in April 2009.

less of a role.

The remainder of the paper is structured as follows. Section 2 introduces the established algorithms of Lee and Ready (1991), Ellis et al. (2000) and Chakrabarty et al. (2007), followed by Section 3, which introduces the algorithm proposed in this paper. Section 4 presents the data used to evaluate the algorithms. Section 5 presents the main results of the ability of the algorithms to recover the trade initiator from the data. The results from typical applications of the algorithms, estimating order imbalances and market liquidity, are presented in Section 6. Section 7 presents extensive robustness checks against varying assumption on the data structure. Section 8 offers some discussion and Section 9 concludes.

2 The LR, EMO and CLNV Algorithm

2.1 The Decision Rules

The LR algorithm (Lee and Ready, 1991) is the most popular choice to classify trade data into the orders of the liquidity demanding and supplying sides. It compares the transaction price to the mid-point of the ask and bid quote at the time the trade took place. If the transaction price is greater (smaller) than the mid-point the liquidity demanding side is the buyer (seller), i.e. the trade is buyer-(seller-)initiated. If the transaction price is equal to the mid-point, the trade initiator is assigned according to the tick-test. That is, if the transaction price is greater (smaller) than the last price that is not equal to the current transaction price, the trade was buyer-(seller-)initiated.

The common alternatives to the LR algorithm are the EMO and CLNV algorithms (Ellis et al., 2000; Chakrabarty et al., 2007). The EMO algorithm classifies a trade as buyer-(seller-)initiated if the transaction price is equal to the ask (bid) price. For all trades off the quotes the tick-test is used. The CLNV algorithm assigns the trade initiator to the buying (selling) side if the transaction price is equal to the ask (bid) or up to 30% of the spread below (above) the ask (bid). For all trades above (below) the ask (bid) or within a 40% range of the spread around the mid-point the tick-test is used. Table 3 in the appendix summarizes the classification algorithms in pseudo code.

2.2 Quote-Matching Rules

The LR, EMO and CLNV algorithms require assigning one bid and ask quote to each trade in order to classify it. In an ideal data environment where at the time of the trade we record only one quote change, we know that the quotes in effect at the time of the trade are the last ones recorded before the time of the trade. With several quote changes occurring at the same time as the trade, however, it is not clear which quotes to select for the classification procedure. For example, with one trade and three quote changes recorded at the same millisecond, the quotes corresponding to the trade could be the last quotes from before the millisecond or one of the first two recorded at the millisecond. The convention in such a case, in the absent of any misalignment between reported trade and quote times, is to take the last ask and bid from before the time of the trade. In the presence of a misalignment, the common procedure is to lag the time of quotes by the amount of the suspected delay in the reporting of the trades and then to match each trade with the last quote from before the time of the trade.

An alternative, recently suggested by Holden and Jacobsen (2014) to circumvent the problem of imprecise timestamps, advises transforming the timestamps to a higher precision. This is done by interpolating the recorded times according to the number of trades or quotes during that time. For example, for trades recorded at seconds, the interpolated time t is computed by

$$t = s + \frac{2i-1}{2I}, \quad i = 1, \dots, I$$

where s is the recorded time, and I is the number of trades at time s. The algorithms then use the last ask and bid price from before the time of the trade according to the interpolated time.

3 The Full-Information Classification Algorithm

Selecting a single pair of ask and bid quotes based on *ad hoc* rules is likely to induce errors in the classification results of the LR, EMO and CLNV algorithms under the described data deficiencies. The algorithm proposed here aims to reduce the number of erroneous classifications by allowing for more than one ask and bid quote to be considered and by using the full information provided by the data to assign a trade to a certain quote.

To understand how we can use the information contained in the data, consisting of transaction prices and volume, as well as the best ask and bid quotes and the volume available at the quotes, to determine the trade-quote correspondence we need to make some assumptions about the data structure.

Data Structure 1.

(i) Each transaction against a visible order leads to a corresponding reduction in volume available at the respective quote.

(ii) Trades and quotes are reported in the correct order.

Assumption (i) is certainly harmless if we consider a market order that trades against a single limit order. Consider, however, a market order that is too large to be filled by a single limit order. The assumption then states that the order book displays the successive steps in the completion of the market order. That is, if a market buy order for 100 shares trades against two limit orders for 50 shares each, the order book will first show a reduction of 50 shares at the bid and then another reduction of the same size, even though these changes happen basically instantaneously. Assumption (ii) is certainly harmless when we use data from a single exchange. For the data from the consolidated tape, however, due to the different latencies of the exchanges to the tape, trades and quotes can be out of order over short intervals of time. As both assumptions hold for the present data set, but they may not for others, I will relax them later and discuss the adjustments to the proposed algorithm.

Before describing the procedure of the algorithm let me introduce a bit of notation for the transaction and ask data. The notation for the bid data follows analogously to that of the ask.

Notation

- Transaction index: $i \in \{1, \ldots, I\}$
- Transaction price and volume: p_i and v_i
- Recorded transaction time: s_i
- Ask quote index: $j \in \mathcal{J}_a = \{1, \ldots, J_a\}$
- Ask price and volume: a_j and v_j^a

- Change in volume at the j-th ask to the next:

$$\Delta v_j^a = \begin{cases} v_j^a - v_{j+1}^a & \text{if } a_j = a_{j+1} \\ v_j^a & \text{if } a_j < a_{j+1} \\ -1 & \text{otherwise,} \end{cases}$$

  Explanation: If the ask price increases from j to $j+1$ ($a_j < a_{j+1}$), then all of the volume at that the j-th ask must have disappeared (either because of a trade or because the order was canceled), and hence $\Delta v_j^a = v_j^a$.[7] If the ask price decreases from j to $j+1$ ($a_j > a_{j+1}$), then a new sell order must have been submitted with a better limit price than that of the j-th quote. So a trade cannot have taken place at the j-th quote. This is indicated by -1.

- Recorded time of an ask: s_j^a. This indicates the time from which point on the j-th ask price and volume determine the best visible ask.
- The collection of ask quote indices with the same timestamp s: $\mathcal{N}_s^a = \{j \in \mathcal{J}_a \colon s_j^a = s\}$.
- Interpolated time of an ask: $t_j^a = s_j^a + n_j^a/(|N_s^a|+1)$ with $n_j^a \in \{1, \ldots, |N_s^a|\}$
- Auxiliary variable: l^a. This will be used to approximate the ask quote at the time of an execution against a hidden order
- Trade direction of the liquidity demanding party: o_i (1 for buy, -1 for sell)

The algorithm works as follows (see Figure 1 for a graphic representation):

Step 1 – Quote Selection and Matching: Starting with the first trade $i = 1$, we collect all ask and bid quotes against which the trade could have been executed only by considering the timing, i.e. for the ask

$$\mathcal{C}_a = \max\{j \in \mathcal{J}_a \colon s_j^a < s_i\} \cup (\mathcal{N}_{s_i}^a \setminus \max \mathcal{N}_{s_i}^a).$$

These are the last quotes from before the time of the trade and all but the last quote at the time of the trade. We initialize the variable $l^a = a_k$ with $k = \min \mathcal{C}_a$. Analogously, we obtain $\mathcal{C}_b$ and l^b for the bid.

Using transaction price and volume, search for the first match among the se-

[7] For the bid quote, the second case reads "if $b_j > b_{j+1}$".

lected ask and bid quotes:

$$\boldsymbol{\alpha} = \min\{j \in \mathcal{C}_a \colon p_i = a_j \text{ and } v_i = \Delta v_j^a\}.$$

Analogously we obtain the first match among the bid quotes denoted $\boldsymbol{\beta}$.

Step 2 – Unique Match: If we find a match among the ask quotes, but not for the bid quotes the trade has been executed on the ask and we set $o_i = 1$. Go back to Step 1 and proceed with trade $i+1$. If $s_{i+1} = s_i$ we use the same collection of ask and bid quotes and set $l^a = a_\alpha$, otherwise we update $\mathcal{C}_a$ and $\mathcal{C}_b$ and newly initialize l^a and l^b. (If we find that there is no match among the ask quotes, but at least one among the bids, all updates are made analogously.)

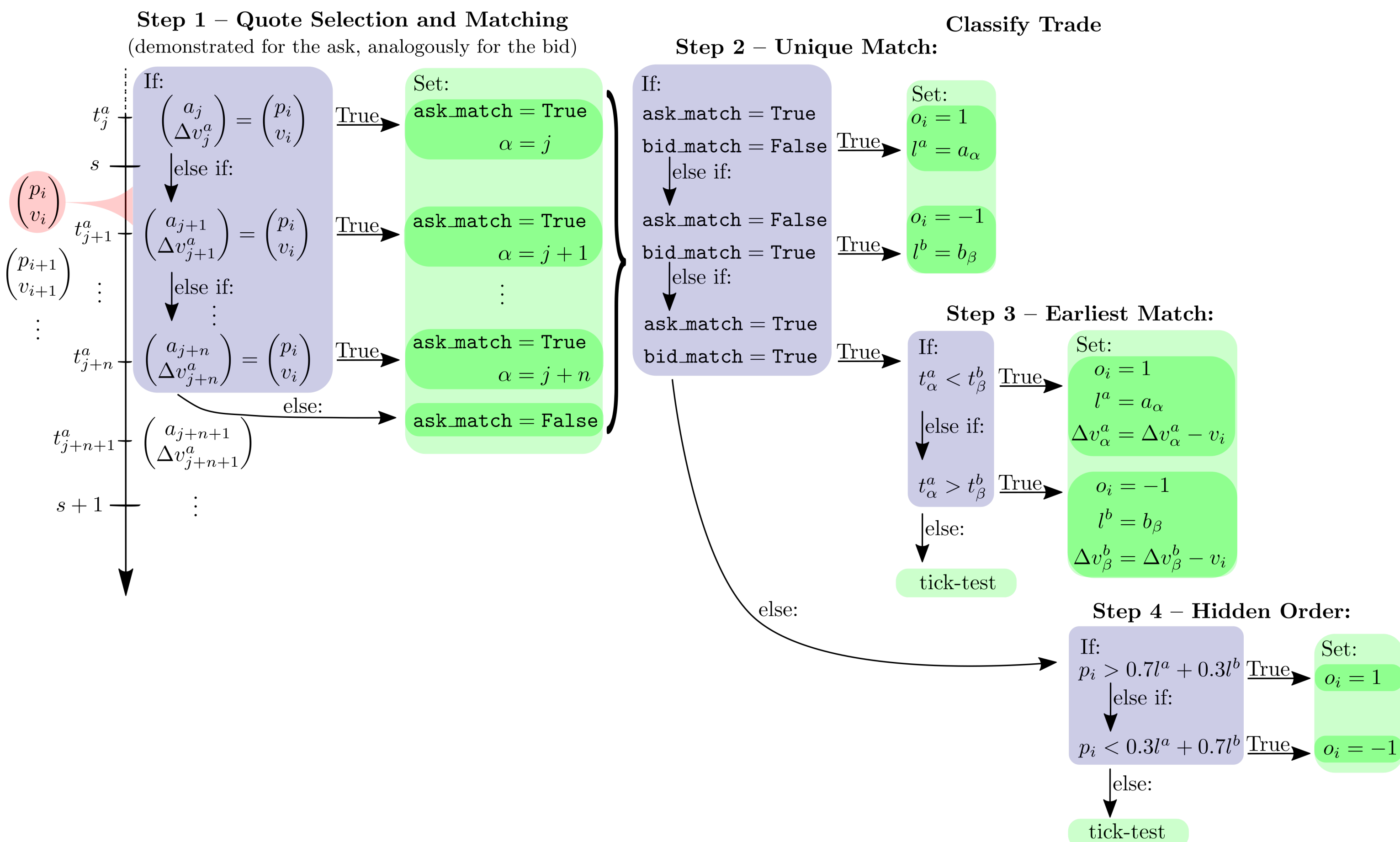
Step 3 – Earliest Match: If we find a match among both the ask and bid quotes, we classify the trade according to which quote seems to be affected first. That is, if $t^a_{\boldsymbol{\alpha}} < t^b_{\boldsymbol{\beta}}$, we set $o_i = 1$. Go back to Step 1 and proceed with trade $i+1$, making the same adjustments as described under Step 2. Additionally we omit the $\boldsymbol{\alpha}$*th*-ask from further comparisons by subtracting from $\Delta v^a_{\boldsymbol{\alpha}}$ the size of the transaction v_i. If $t^a_{\boldsymbol{\alpha}} > t^b_{\boldsymbol{\beta}}$, we set $o_i = -1$ and updates are made accordingly.

Step 4 – Hidden Order: If we cannot find a match among the ask and bid quotes, we are likely to face a trade against a hidden order. These are classified according to their position in the spread (similar to Chakrabarty et al., 2007), which is approximated by l^a and l^b. If $p_i > 0.7l^a + 0.3l^b$ and $l^a > l^b$ we set $o_i = 1$. If $p_i < 0.3l^a + 0.7l^b$ and $l^a > l^b$ we set $o_i = -1$. Go back to Step 1, proceed with trade $i+1$ and update $\mathcal{C}_a$ and $\mathcal{C}_b$ if $s_{i+1} > s_i$.

Step 5 – Tick-test: Any trade that could not be classified in Step 2 to Step 4 is classified by the tick-test.

Remark to Step 3 The idea of using the interpolated time to classify trades that match with both an ask and bid quote is as follows. Observing ask and bid quotes to equal each other within the same, say, second of a trade, may be due to the price impact of the trade. That is, quotes are updated in the direction of the trade initiator. This should be reflected in a relatively early interpolated time of the corresponding quote for the following reasons. First, in case of a buyer-initiated trade we may expect more activity on the ask side because of the information contained in the trade that leads to the price impact. Traders will either submit buy orders

Figure 1: The Full-Information Classification Algorithm



Notes: This figure shows the process of the Full-Information algorithm to classify a trade. The variables are defined in the **Notation** list. In **Step 1** we collect all ask and bid quotes against which the trade could have executed considering only the timing of the trade and the quotes. Starting with the first ask and bid quote, respectively, from these collections we search for an exact match of the quote and its volume change with the transaction price and volume. If a match could be found, we set an indicator variable to `True` and memorize the index of the respective quote. In **Step 2**, if only the ask/bid side matches the trade, we classify it as buyer-/seller-initiated and assign the respective quote to the auxiliary variable l^a/l^b, which is used to construct the spread in case of hidden order executions. In **Step 3**, if both sides match the trade, it is classified according to the interpolated time of the matched quotes. The corresponding quote is then omitted from further proceedings by subtracting the transaction volume from the volume change at the quote. In **Step 4** the trade is classified by the position of the price within the spread, which is approximated by the auxiliary variables. Trades not classified in any of these steps are classified by the tick-test.

to take advantage of stale limit orders or cancel their stale limit orders in response to the trade. Either way, $|N_s^a|$ will be relatively large. Second, in case of a buyer-initiated trade, the trade executed *first* on the ask and then bid quotes were updated subsequently upwards. That is, $\boldsymbol{\alpha}$ will be relatively small while $\boldsymbol{\beta}$ relatively large. In total, this means that $t^a_{\boldsymbol{\alpha}}$ will be smaller than $t^b_{\boldsymbol{\beta}}$.

To avoid further conflicts between ask and bid quotes with the price and volume characteristics, the quote to which the trade is matched is omitted from assignments of subsequent trades. This assumes that a quote can only be hit once, which means that we have eliminated the counter-party to each trade in the data. The counter-party is easily identified by the neighboring trade with the same price and volume, but opposite trade direction. Alternatively, if the counter-party is not omitted from the data for the classification process, we drop the corresponding quote after it has been assigned twice to a trade.

Remark to Step 4 The spread in Step 4 is constructed from the auxiliary variables l^a and l^b. They serve to approximate the ask and bid valid at the time of the execution of the hidden order. They are initialized to the first ask and bid valid during the time of the trade. If we are able to classify a trade involving a visible order, the corresponding auxiliary variable is updated. In that way, we obtain a better approximation of the spread at the time of the hidden order execution due to the correct order of the trades.

Remark to Step 5 I follow the design of the traditional algorithms and use the tick-test to classify the most ambiguous cases. This can be motivated by the finding of Perlin et al. (2014) who show that the misclassification rate of the tick-test is upper-bounded by 50%.

4 Data

The evaluation of the algorithms is based on equity trading data from NASDAQ's electronic limit order book constructed from NASDAQ's TotalView-ITCH data.[8] The trade data contain all transactions against visible and hidden limit orders with

[8] The reconstruction from the TotalView-ITCH data is done by the software LOBSTER, which in turn produces the order book data and messages files containing the information on the events causing the changes in the order book. A detailed description of how I obtain the trade and quote data from these files is given in the appendix.

information on the price and volume of the transaction. The order book data contain the development of the order book. That is, whenever a visible limit order that affects the best quotes is submitted, canceled (partially or completely) or executed, the order book contains an entry of the best bid or ask indicating the new price and volume available. Changes regarding hidden orders are not displayed in the order book.

The data covers the continuous trading phase from 9:30 am to 4 pm for all trading days during the 3 month period May to July 2011. I selected the 30 largest stocks (by market capitalization) in 2015 from the 11 NASDAQ industry sectors.[9] Following Chakrabarty et al. (2015), I drop stock-days with an end-of-day price of less than one dollar or with less than 10 trades, which leaves me with a total of 19842 stock-days. A list with all ticker names studied in this paper, as well as some summary statisitics are provided in Tables 2 and 4 in the appendix.

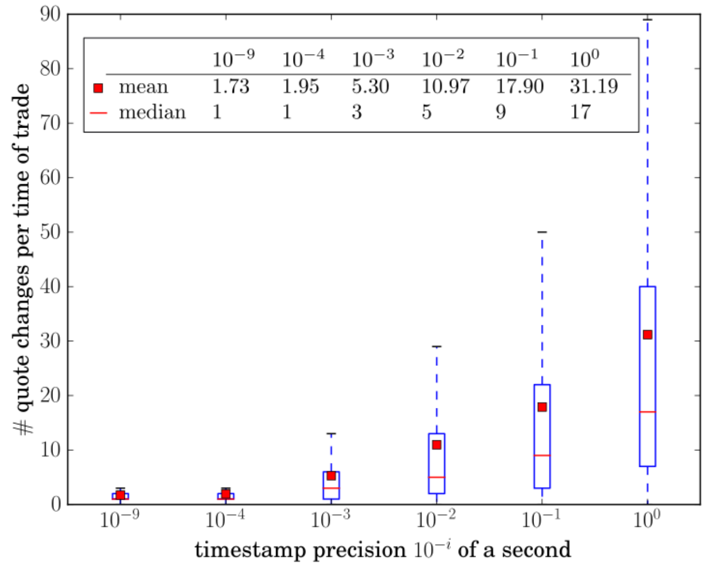
For the main analysis the quotation frequency, i.e. the number of quote changes at a given time of a trade at a given timestamp precision, is important. Figure 2, therefore, plots the distribution of the number of quote changes at the times of trades for different timestamp precisions: the original precision of nanoseconds (10^{-9} of a second), as well as 10^{-4} to 10^{0} of a second.

At a precision of 10^{-4} or less, most of the trade times experience only a small number of quote changes, which would allow us to match trades to their quotes based only on the timing of the two. With decreasing timestamp precision, however, this number increases quickly. At a precision of seconds, the median number of quote changes with the same timestamp as that of a trade is 17. With 17 quote changes at the time of a trade, we cannot deduce, just from the timing of the two, which quote belongs to the trade.

Figure 2 omits extreme values for illustrative purposes. Figure 3, therefore, gives a closer account of the distribution of the number of quote changes at the time of trades for the data timestamped to the second. We can see that a number of quote changes as high as 100 or more over an interval of one second occur more than 5% of the time. The figure also displays the fraction of cases where, during the second of a trade with a given number of quote changes, one of the bid quotes is at least as high as one of the ask quotes. We see, for example, that for all trading seconds with 17 quote changes 38% have at least one bid quote as high as one of the asks. In all these cases, the wrong choice of a quote can easily lead to a wrong classification of

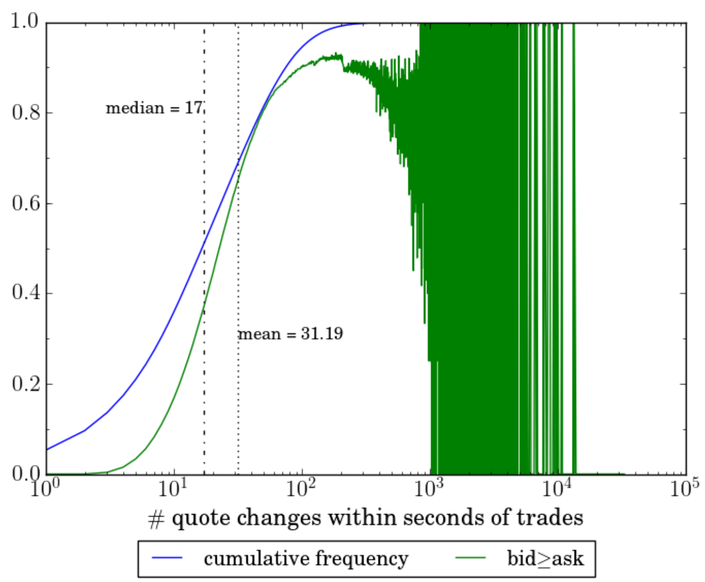
[9]These sectors are: Basic Industries, Finance, Capital Goods, Healthcare, Consumer Durables, Consumer Non-Durables, Public Utilities, Consumer Services, Technology, Energy, Transportation, Miscellaneous.

Figure 2: Distribution of the Number of Quote Changes at Different Frequencies



Notes: For each time where at least one trade takes place, I count the number of quote changes with the same timestamp. This figure shows the distribution of these counts in terms of boxplots for different timestamp precisions. For example, 50% of the milliseconds with at least one trade also display 3 or more quote changes.

Figure 3: Distribution of the Number of Quote Changes at Seconds and the Frequency of Crossing Quotes



Notes: At each second where at least one trade takes place, I count the number of quote changes with the same timestamp. The blue line shows the cumulative frequency of these counts. For example, 68% of the seconds with at least one trade experience 31 quote changes or less. The green line displays the fraction of cases where one of the bids is at least as high as one of the asks for a given number of quote changes at the second of the trade.

the trade.

5 Results

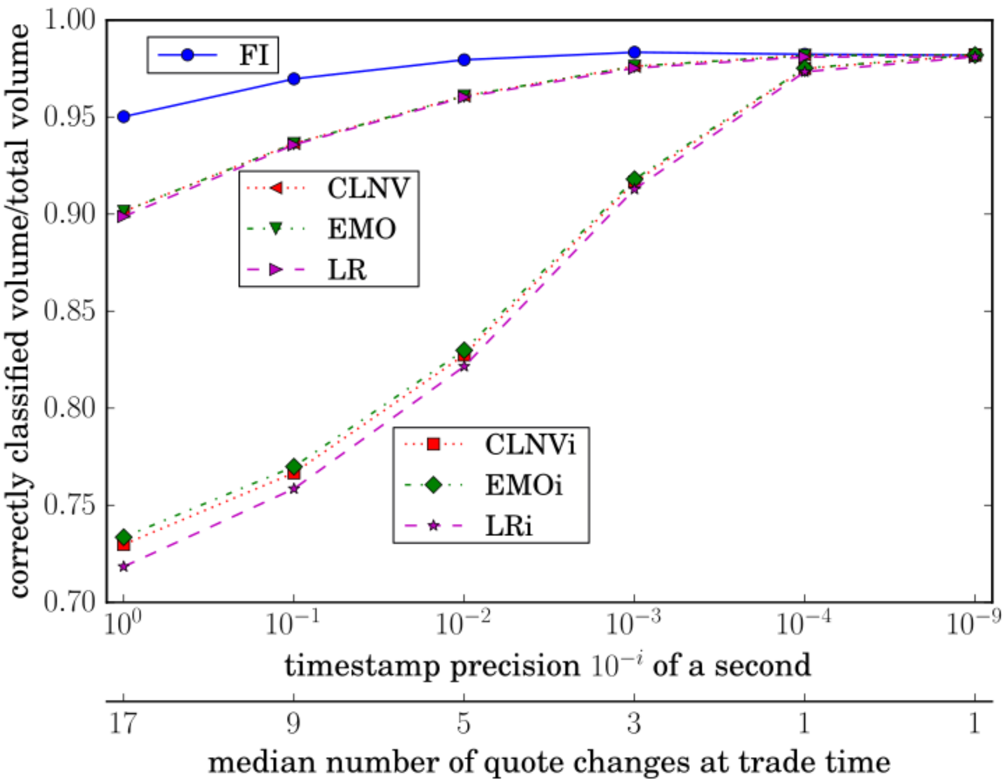
5.1 Classification Accuracy at Different Timestamp Precisions

I analyze the improvements we can achieve over the traditional algorithms by applying them and the proposed Full-Information algorithm (FI) to the data with varying timestamp precisions. The traditional algorithms are used in combination with the quote matching rule of using the last quotes from before the time of the trade (denoted by LR, EMO and CLNV), and using the interpolated time of trades and quotes (denoted by LRi, EMOi and CLNVi) proposed by Holden and Jacobsen (2014). The timestamp precisions are chosen to be of 10^{-i} of a second for $i = 0, 1, 2, 3, 4$, as well as the original data precision of nanoseconds (10^{-9} of a second). I evaluate the quality of the algorithms on the basis of correctly classified trading volume. Figure 4 presents the sum of correctly classified trading volume over total trading volume over the entire sample. Table 5 in the appendix provides the corresponding numbers, along with the means and standard deviations across the sample.

The results show that the FI algorithm dominates the others. At the original timestamp precision of nanoseconds all of the algorithms correctly classify around 98% of trading volume. Approaching the timestamp precision of seconds, however, the performance of the traditional algorithms falls off more quickly than that of the FI algorithm. At the precision of seconds the traditional algorithms correctly classify around 90% of trading volume (around 73% when using the interpolated time), in contrast to 95% correctly classified volume by the FI algorithm. That is, the FI algorithm reduces the number of misclassified shares by half.

Table 5 in the appendix shows that the FI algorithm also dominates in terms of the variation in classification accuracy. While the standard deviation of the stock-day classification accuracy barely moves for the FI algorithm (from 2.09%-points at nanosecond to 2.4 at second precision), the standard deviation of the traditional algorithms increases from the same level of around 2.1%-points to more than 3.4.

Figure 4: Classification Accuracy at Different Timestamp Precisions



Notes: This figure depicts the fraction of correctly classified trading volume by the FI algorithm and the traditional algorithms using the last quotes from before the time of the trade (EMO, CLNV and LR), and using the interpolated time of trades and quotes (EMOi, CLNVi and LRi). The algorithms are applied to the data with reduced timestamp precisions (10^{-i} of a second for $i = 0, \ldots, 4$), and using the original precision of nanoseconds (10^{-9} of a second). These correspond to median number of quote changes at the time of trades ranging from 17 (for $i = 0$) to 1 (for $i = 9$).

5.2 Comparison to Previous Studies

The study that is closest to the analysis in this paper is Chakrabarty et al. (2015) who study the accuracy of the LR algorithm for transactions from NASDAQ's ITCH data over the same time span. The comparison with their study is interesting, because they use the same data source for the transactions, but the quotes from the DTAQ. That is, a comparison of the classification accuracy of the LR algorithm can tell us something about the quality of the DTAQ quote data.

The classification accuracy of the LR algorithm is 10 to 15%-points lower than reported here, even though Chakrabarty et al. (2015) aggregate the classification results over time intervals such that opposite misclassifications cancel each other out.[10] Importantly, these results do not seem to be merely driven by possibly asynchronous timing of the transactions from the ITCH data and the quotes from the DTAQ. Chakrabarty et al. (2015) match the trades from the ITCH data to the trades from the DTAQ data and repeat the classification exercise solely for the trade and quote data from the DTAQ, but the classification accuracy remains overall low. That is, the DTAQ data apparently still pose substantial problems for accurate trade classification by the traditional algorithms.

Overall the traditional algorithms perform relatively well when compared to previous studies. Differences in classification accuracies result, in large part, from applications of the algorithms under different market structures and different ways of identifying the trade initiator. For example, Finucane (2000) equates the trade initiator with the trade direction of the market order, including market-crosses for which a definition of the trade initiator is not straight forward, and finds an accuracy of 84%. Lee and Radhakrishna (2000), on the other hand, who follow a similar but more restrictive definition by excluding market-crosses, find an accuracy of 93%. Yet, only 50% of their sample qualifies for their definition. The possibility of market orders trading against each other, however, should not be of any concern given today's fast execution times.

Another identification of the trade initiator is provided by Ellis et al. (2000) and Theissen (2001) who find accuracies of 81% and 75%, respectively. Both study dealer markets and identify the trade initiator by the trade direction opposite to the dealer, because the dealer is supposed to cater to the demand of the customer. In markets where dealers play a larger role, however, there is more room for individual deviations from a standard procedure of matching orders, which are not captured by the algorithms, and dealers may not always trade passively to manage their inventory.

In contrast to these studies, the liquidity demanding and supplying parties are unambiguously identified in the present data set, and the results show that the traditional classification algorithms are well suited to distinguish these parties in the environment of the simple and consistent mechanisms of an electronic limit order book, as long as the number of quote changes at the time of trades is not too high.

[10] See Panel A and B of Table 1 in Chakrabarty et al. (2015, p. 60). They aggregate the classification accuracy over different time intervals, because they compare the accuracy of the LR algorithm to that of the BVC algorithm, which is not meant to classify single trades. The numbers best comparable to those presented here are in the first row of the "Time bars" columns.

5.3 Explaining the Performance of Holden and Jacobsen's (2014) Interpolation Method

The results show that interpolating the trade and quote times to circumvent the problem of imprecise timestamps is not a fruitful alternative to the traditional quote matching approach. The idea behind the interpolation method is that trades and quotes are equally distributed over a given interval, e.g. a second. In that case, the best guess to when these trades and quote changes took place would be to distribute them equally over the interval.

Conditioned on the event of a trade, however, this reasoning is not valid. Given the event of a trade, the reason to observe a quote change is the trade itself. This is confirmed by studies which empirically model the events of trades and quote changes as mutually exciting point processes where the probabilities to observe a trade or quote change depend on each other (Bowsher, 2007). Moreover, a single trade may not only lead to a single quote change, but to several quote changes in response to the information contained in the trade. That is, the number of quote changes to the right of the quote change that was triggered by the trade is likely to be greater than the number of quote changes to its left. This implies that a trade is likely to be placed behind the quote change that was triggered by the trade if we interpolate following Holden and Jacobsen (2014). If we further conjecture that the price impact follows the direction of the trade, misclassification is often the result.

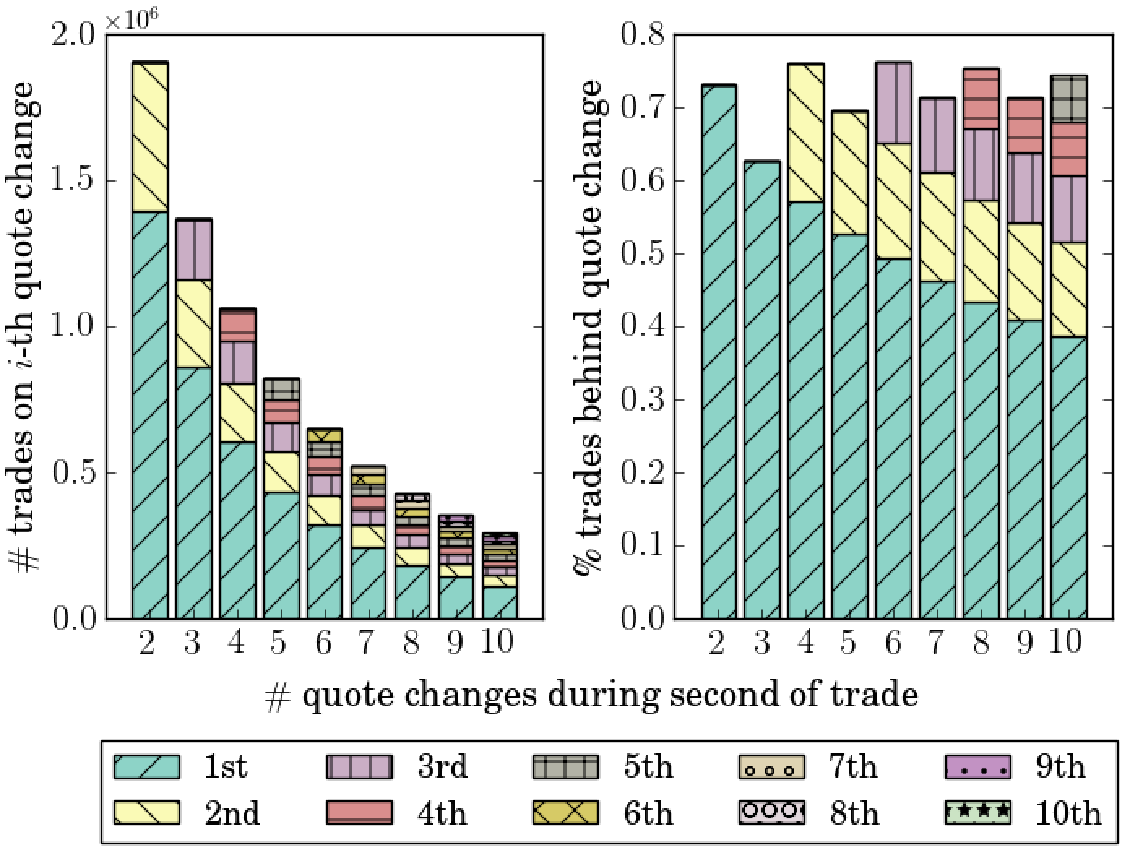
Figure 5 confirms these considerations. The left panel shows how often a trade triggered the first, second, third etc. quote change (y-axis) for a given number of quote changes during the second of the trade (x-axis).[11] For example, the first bar shows that almost 1.5 million trades triggered the first quote change, whereas only around 0.5 million were responsible for the second quote change in all cases where we observe 2 quote changes during the second of the trade. We see that even at seconds with a large number of quote changes, trades most often account for one of the first quote changes.

The right panel of Figure 5 shows the percentage of trades that are moved behind their corresponding quote changes by the interpolation method. It shows that quite generally in more than 70% of the seconds with a single trade, the trade is moved behind the quote change that was triggered by that trade.

Interestingly, Holden and Jacobsen (2014) report that by interpolating trade and quote times they increase the agreement of trade classification from the MTAQ data

[11] I used only those seconds with a single trade.

Figure 5: Quote Changes Caused by Trades during Single-Trade Seconds



Notes: The LEFT panel presents the number of times a trade is executed on the first, second, third, ..., 10-th quote (y-axis) within seconds of 2, 3, ..., 10 quote changes (x-axis). The RIGHT panel shows the percentage of trades that are placed behind the quote change that was triggered by the trade if trade and quote times are interpolated following Holden and Jacobsen (2014). Only seconds containing a single trade are used.

with the results obtained from the DTAQ data that are timestamped at higher precision than the MTAQ. In light of the results presented here this has important implications, namely that the problem of quotes being reported ahead of their corresponding trades prevails even today in the DTAQ data.[12] This adds further doubt on the ability to readily apply the traditional algorithms to the DTAQ data.

[12]Without a closer look into the DTAQ data, which is currently not available to this author, the assertion of a possible time delay of reported trades in the DTAQ must, of course, be viewed with caution. It is quite possible that the increase in agreement in the classification results between the DTAQ and MTAQ by using the interpolation method reported in Holden and Jacobsen (2014) is only due to chance. However, also the earlier comparison to the results presented in Chakrabarty et al. (2015) raised some concerns about the DTAQ data.

5.4 A Closer Look at the Classification Accuracy at the Individual Classification Steps

The Full-Information algorithm classifies trades at different steps, depending on the criteria that apply to the specific trade. The uncertainty of the classification increases with each step in the algorithm and we would, thus, expect the accuracy to differ with the different classification criteria.

Therefore, to study the performance of the individual classification steps, Table 1 presents the individual classification accuracies. Panel A shows the percentage of correctly classified volume at the individual classification step summed over the entire sample, while Panel B shows the percentage of trading volume that is classified at the respective classification step. The Table differentiates between trades executed against visible (visible = YES) and hidden (visible = NO) orders. The column "cl. step" refers to the classification steps (2 to 5) at which the trade initiator is assigned. "Cl. step 0" refers to cases where the trade direction of the liquidity demander could not be derived.

Trades against visible orders are almost exclusively classified during Step 2 or 3 of the classification process. That is, any trade that executed against a visible order must have at least one match among the quotes with the corresponding change in volume. Matches between quotes and trades that actually executed against hidden orders, on the other hand, are only accidental and occur rarely. With decreasing timestamp precision the number of hidden orders classified in Step 2 or 3 increases as the number of quotes that we consider during the classification of a trade increases. Overall, however, the number of hidden orders classified in Step 2 and 3 remains relatively small. Trades involving hidden orders are predominantly classified in Step 4 or 5 of the algorithm, as they are supposed to.

The accuracy of the assignments of visible orders in Step 2 of the algorithm is almost 100% at any timestamp precision. That is, an unambiguous match at one side of the order book leads almost always to the correct classification of the trade. With decreasing timestamp precision, however, the number of unambiguous assignments decreases and the algorithm refers to the interpolated times of the matched quotes more often. Though the interpolated time is a suitable indicator for the assignment with accuracies between 90 to 95% it does not provide the same certainty as a classification at the second step. In fact, the decrease in overall classification accuracy going from nanoseconds to seconds is largely driven by the substitution of assignments between Step 2 and 3.

Table 1: Accuracy of Individual Classification Criteria

		timestamp precision: 10^{-i} for $i =$					
visible	cl. step	0	1	2	3	4	9
Panel A: % correctly classified volume							
YES	0	–	–	–	–	–	–
	2	99.85	99.91	99.96	99.99	100.00	100.00
	3	90.31	94.18	95.98	95.11	68.96	–
	4	79.67	80.32	80.70	80.57	80.55	80.57
	5	55.97	55.90	56.63	56.31	31.74	29.80
NO	0	–	–	–	–	–	–
	2	67.25	74.49	84.52	94.50	99.79	99.98
	3	83.44	87.66	87.17	79.64	66.67	–
	4	92.59	94.79	95.75	95.58	94.39	93.86
	5	64.96	65.21	65.94	67.20	68.60	68.81
Panel B: % classified volume							
YES	0	0.00	0.00	0.00	0.00	0.00	0.00
	2	61.79	71.44	81.20	88.55	90.42	90.42
	3	28.55	18.93	9.19	1.86	0.00	0.00
	4	0.00	0.00	0.00	0.00	0.00	0.00
	5	0.07	0.05	0.03	0.01	0.00	0.00
NO	0	0.00	0.00	0.00	0.00	0.00	0.00
	2	1.87	1.84	1.70	1.33	0.80	0.72
	3	0.65	0.30	0.09	0.01	0.00	0.00
	4	4.01	4.23	4.34	4.29	3.87	3.79
	5	3.05	3.20	3.45	3.95	4.92	5.07

Notes: Panel A shows the classification accuracy of the different classification criteria applied by the FI algorithm. Panel B shows the percentage of trading volume that is classified by the respective criterion. The column "cl. step" refers to the step in the classification process at which the trade initiator is assigned (with 0 referring to cases which could not be classified).

The classification of trades involving hidden orders is inherently more difficult than for visible orders. At nanosecond precision, the number of misclassifications can almost entirely be attributed to trades involving hidden orders. Even though the position of the transaction price within the spread is informative as we can see from the classification accuracies of around 94% at Step 4, the economic reasoning that is behind the decision rule does not apply to all cases. With decreasing timestamp precision the classification accuracy of trades involving hidden orders, however, does not change much. The informativeness and the number of cases assigned to Step 4 is almost the same whether for data timestamped at nanoseconds or seconds. Most of the change in hidden orders classification accuracy is due to a shift from classifications by the tick-test (Step 5) to Step 2 of the algorithm.

The Appendix provides further details on the determinants of misclassification depending on economic variables external to the classification process. In an analysis that necessitates the estimation of the trade initiator, a variation of the classification accuracy in tandem with the variation of other variables entering the analysis can impact the statistical inference and, in the worst case, compromise the researcher's conclusions. The Appendix, therefore, presents results from a logistic regression following Finucane (2000), Ellis et al. (2000) and Chakrabarty et al. (2007) where the binary variable of a correct/incorrect classified trade is regressed on variables such as trade size, realized volatility, spread size, speed of trading, a dummy variable indicating hidden orders etc.

Similar to the results presented in this section, the analysis shows that hidden orders are the most important factor for the probability of misclassification. On average, a trade that executes against a hidden order as opposed to a visible order decreases the estimated probability of a correct classification by 11%-points. Other variables like the aforementioned, which may play a more significant role in economic and financial studies, do not strongly impact the classification accuracy.

5.5 Classification Accuracy under Delayed Timestamps

To see how the FI algorithm may help to alleviate the problem of delayed trade times consider the following example. A trade is timestamped at 9:45:50.9 and three quote changes are timestamped at 9:45:50.1, 9:45:50.3 and 9:45:50.5. Given the uncertainty revolving around the degree of the report delay we may want to consider all three of them for the classification procedure. In this particular case, the FI algorithm would allow us to do so by simply decreasing the timestamp precision to that of seconds. The results from the previous section show that we would lose little in terms of classification accuracy if, in fact, a higher timestamp precision would suffice, but we would ensure that the results are not driven by the noise in the timestamps.

To explore the effect of random delays in reported trade times on the classification performance of the traditional algorithms and the FI algorithm, I add exponentially distributed noise to the original trade timestamp at nanosecond precision. That way the time of the trade will lag behind the reported time of its corresponding quote change but to a varying degree from trade to trade. The exponential distribution is given by $F(x; \beta) = 1 - \exp\{-x/\beta\}$ for $x \geq 0$ and I choose $\beta = 10^{-j}$ for $j = 1, \ldots, 4$.[13] I then choose different timestamp precisions at which the algorithms are applied to the data.[14] The precision s ranges from 10^{-4} of a second to 2.5 seconds.

The appendix presents a brief derivation of how we can expect the classification accuracy to be affected by noisy timestamps. To give a quick idea, consider applying the algorithms at a precision of seconds. The reduction in classification accuracy compared to the situation without noise is only determined by the number of trades that are shifted outside the second at which they actually occurred. The average classification accuracy of these trades will tend towards 0.5, while the classification accuracy is unaffected for those trades that remain in the same second as they were in the absence of noise.[15] Choosing the optimal timestamp precision is, thus, a trade-off between a reduction in accuracy due to imprecise timestamps on the one hand, and a reduction in accuracy due to trades being reported outside their actual time interval at high timestamp precision on the other. Since the classification accuracy of the FI algorithm is greater-equal to the accuracy of the traditional algorithms at any timestamp precision, we would expect that the FI algorithm dominates the traditional algorithms under noise as well. Figure 6 presents the numerical results.

As expected, we find that the FI algorithm dominates the others. It is only at relatively high precision timestamps that the performance of the algorithms align, trending towards an accuracy that is not different from a random classification of the trade initiator. We observe, however, that the FI algorithm is better able to profit from the decrease in timestamp precision. The classification accuracy of the FI algorithm keeps increasing with growing interval length where the accuracy of the other algorithms stagnates or falls off.

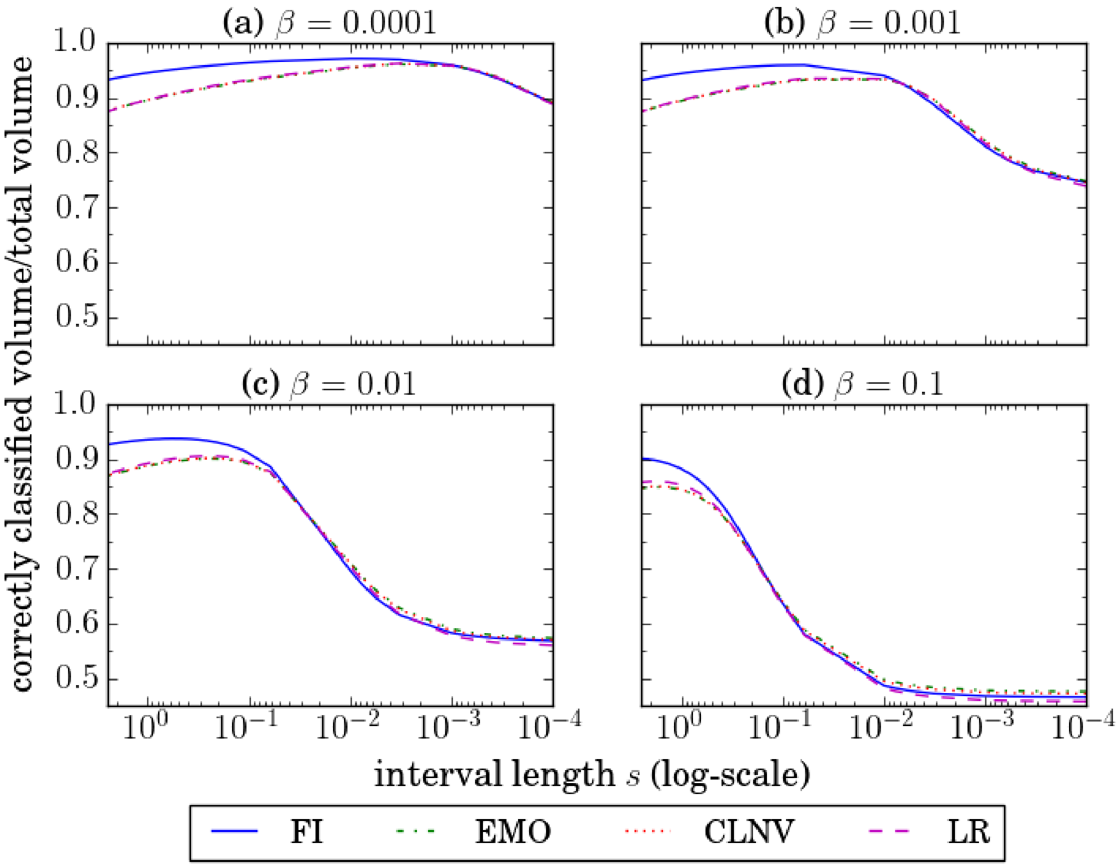
Importantly, the FI algorithm offers high and robust accuracies across the different noise intensities, especially for the probably more relevant range (a)-(c). For example, choosing a precision of a bit more than 0.5 of a second, the FI algorithm achieves accuracies of 94-95% for β between 10^{-4} and 10^{-2}. Even for relatively strong noise of $\beta = 0.1$, the FI algorithm correctly classifies more than 90% of trading volume if the timestamp precision is reduced below 1.9 of a second.

[13]The mean of the exponentially distributed variable is given by β and the q-th percentile by $-\ln(1-q)\beta$. For example, if $\beta = 1/10^3$, we expect a delay in the reported trade time of one millisecond and 99% of all trades to have a delay of less than 5 milliseconds.

[14]Note that I do not report the results using the interpolation method due to their relatively unsuccessful performance presented earlier.

[15]Since the FI algorithm makes use of the correct order of trades at least to some extent, the classification accuracy for those trades not shifted outside their actual time interval can still be affected if the order of the trades changes due to the noise. I discuss this issue in more detail below.

Figure 6: Classification Accuracy and Noisy Timestamps



Notes: This figure shows the fraction of correctly classified trading volume (y-axis) for the data with delayed trade times. The trade time equals the actual time plus ε, with $\varepsilon \sim \text{Exp}(1/\beta)$ and $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. To counteract the effect of the noise on the classification accuracy the algorithms FI, EMO, CLNV and LR are applied to the data with reduced timestamp precision (s) ranging from 10^{-4} of a second to 2.5 seconds presented on a $\log_{10}$-scale (x-axis).

For any level of accuracy of the traditional algorithms under a given noise intensity, we can find the same accuracy for the FI algorithm at lower timestamp precision, which then offers robustness against higher levels of noise. That is, by choosing the FI algorithm at decreased timestamp precision, one gains robustness against unknown degrees of noise without forfeiting classification accuracy against the alternative algorithms if the noise intensity is, in fact, smaller than suspected.

6 Application to Measuring Liquidity and Order Imbalances

6.1 The Order Imbalance

A frequent application where the initiator label enters the analysis is the estimation of the order imbalance

$$OI = \frac{V_B - V_S}{V},$$

where $V = V_B + V_S$ is the total trading volume, V_B is the volume of buyer-initiated trades and V_S the volume of seller-initiated trades. The order imbalance is often used, either directly or indirectly, as a measure of informed trading (see, e.g., Easley et al., 1996, 2012; Bernile et al., 2016; Brennan et al., 2018) or may be itself the variable of interest (e.g. Chordia et al., 2002; Dorn et al., 2008; Chordia et al., 2016).

To construct the order imbalance I split each stock-day into $\tau = 10, 100$ bins of equal volume size giving us $\tau \times 19842$ order imbalance estimates in total, with varying volume sizes across the stock-days.[16] Within each volume bin, the order imbalance is computed according to the above formula using the true trade initiator label and the labels provided by the classification algorithms.

I apply the algorithms to the data at different timestamp precisions ranging from seconds to milliseconds and under different degrees of delays in reported trade times. Trade times are either not delayed or the delay is governed by the exponential distribution with an intensity of $\beta = 10^{-3}, 10^{-2}$.

The statistics of interest are the root-mean-square error between the vector of true order imbalances and their estimates and the overall mean of the absolute order imbalance.[17] Figure 7 presents the results.[18] The red line in the mean-columns indicates the mean absolute order imbalance computed from the true trade initiator label. Numbers are displayed as percentages.

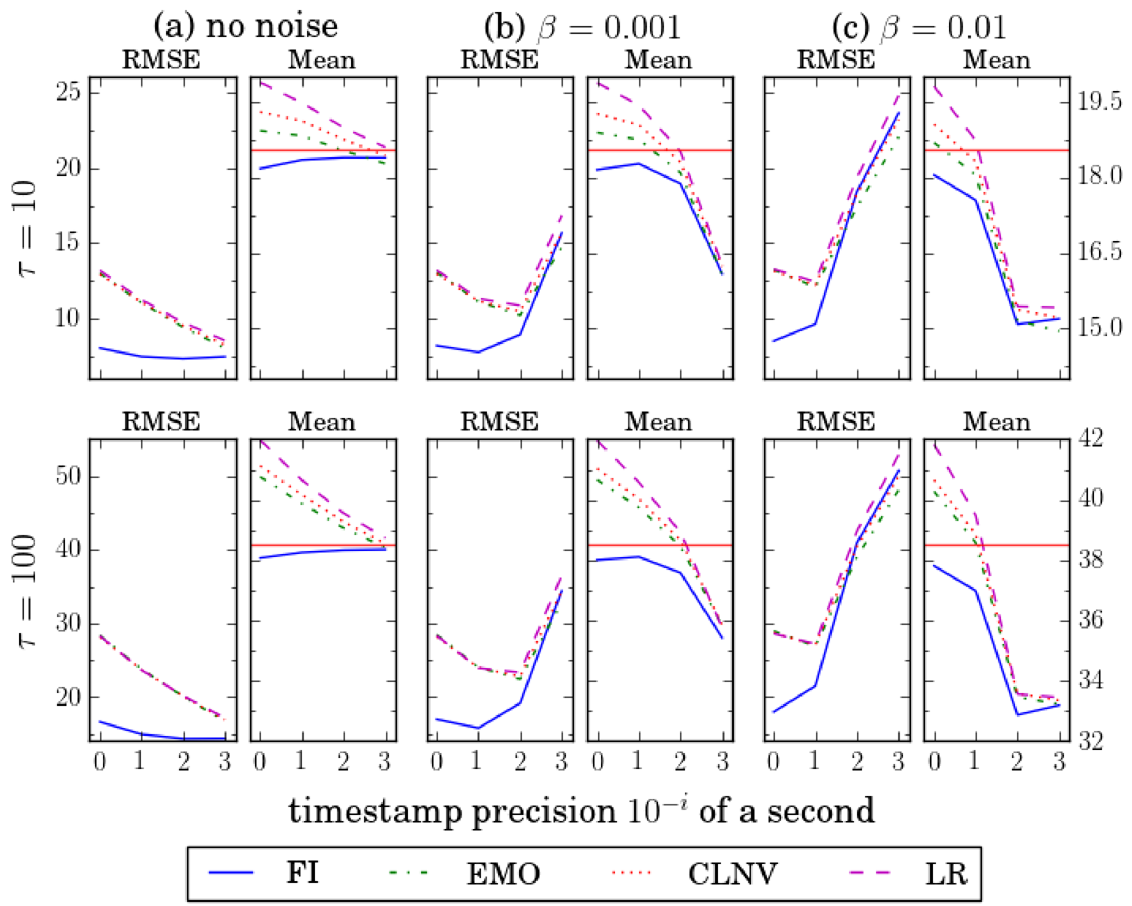
The root-mean-square errors of the estimates based on the classifications of the FI algorithm are generally the smallest. Again, the largest improvements occur at low timestamp precision. For the data split into 10 volume bins at each stock-day

[16] If necessary, the last trade in a bin is split between the two successive bins to ensure equal volume.

[17] RMSE $= \sqrt{\sum_{k,d,t}(OI_{k,d,t} - \widehat{OI}_{k,d,t})^2/|K \times D \times \tau|}$ and AbsoluteMean $= \sum_{k,d,t} |\widehat{OI}_{k,d,t}|/|K \times D \times \tau|$, where $|K \times D|$ is the number of stock-days

[18] The corresponding numbers are reported in Table 7 in the appendix.

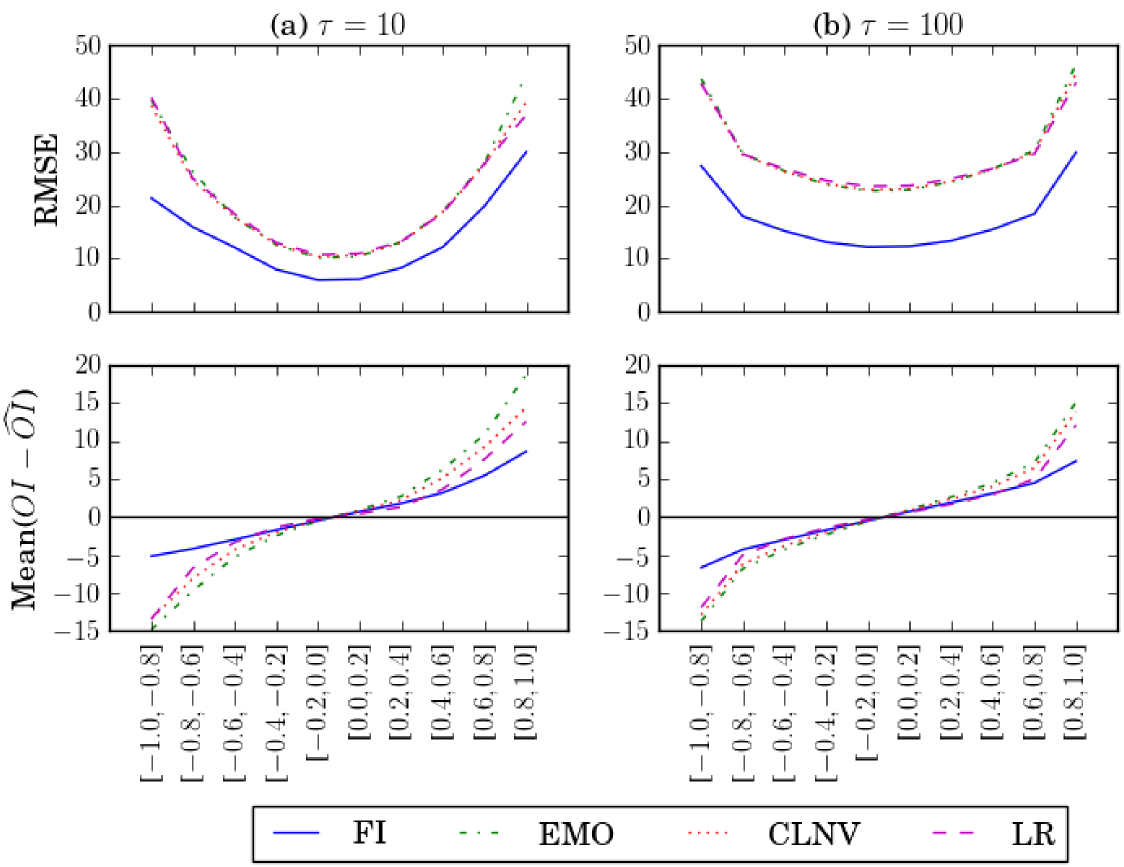
Figure 7: Estimating Order Imbalance



Notes: This figure shows the root-mean-square error between estimates of the order imbalance and the true order imbalance, as well as the sample averages of the absolute order imbalances displayed in percentages. The true values are computed from the true trade initiator labels and the estimates from the classification results of the different algorithms. For the computation of the order imbalance each stock-day is split into equally sized volume bins. The number of bins is chosen to be $\tau = 10, 100$. The algorithms are applied to the data with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds.

the average deviation of the estimated order imbalance based on the FI algorithm around the true order imbalance is 8.06%-points, while that of the best competitor, the CLNV algorithm, is 12.95%-points, a sizable difference considering the average absolute order imbalance of 18.56%. It is only at high timestamp precision (millisecond) that the estimation results of the traditional algorithms can compete with those of the FI algorithm at low timestamp precision. However, only at low timestamp precision are the results robust against moderate delays in reported trade times. Importantly, for the FI algorithm this robustness does not come at the cost of notably

Figure 8: Estimation Performance for Different Regions of the Order Imbalance



Notes: This figure shows the root-mean-square error and the bias between the estimated and true order imbalance for different regions (x-axis) of the level of the true order imbalance. The estimated order imbalances are computed from $\tau = 10, 100$ equal volume bins per stock-day and from the classification results of the algorithms applied to the data timestamped to the second without delay in trades times.

less accurate estimations of the order imbalance.

As the order imbalance is often used as a measure of informed trading indicated by a large absolute order imbalance, we may be particularly interested in the estimation performance depending on the level of the order imbalance. Figure 8 plots the estimation performance for the setup of a timestamp precision of seconds and no delay in trade times for different intervals of the true order imbalance.[19] In each interval I compute the root-mean-square error and the bias between the estimated and true order imbalance.[20] We see that the improvement in the estimation of the order imbalance provided by the FI algorithm is particularly strong in the interesting

[19]The corresponding numbers are presented in Table 8 in the appendix.

[20]The bias for a given interval $\Omega = [l, u]$ with $l < u$ is given by $\sum_{OI_j \in \Omega}(OI_j - \widehat{OI}_j)$.

region of the extremes of the order imbalance.The root-mean-square error for the true order imbalance being in a range of $[-1, -0.8]$ ($[0.8, 1]$) is 21%-points (30%-points) for the FI algorithm, compared to 39%-points (37%-points) for the best competitor, the CLNV (LR) algorithm. The order imbalance in the same region is under-estimated (absolutely speaking) on average by 5%-points (9%-points) for the FI algorithm, compared to 13%-points (13%-points) for the best competitor.

6.2 Measuring Liquidity

To analyze how the plus in classification accuracy translates into dollar values, I apply the different algorithms to the measurement of liquidity. The liquidity measures I consider are the dollar effective spread, the dollar price impact and the dollar realized spread for which the knowledge of the trade initiator plays a crucial role.

The effective spread is defined as

$$DES_i = 2o_i(p_i - m_i)$$

where p_i is the price per share of the i-th trade, m_i is the spread mid-point associated with the i-th trade and o_i is the direction of the trade initiator, that is $+1$ in case of buyer-initiated trades and -1 in case of seller-initiated trades. The effective spread measures the costs incurred by liquidity demanders relative to the ideal environment where trades execute at the mid-point.

Opposite to the costs of liquidity demanders are the profits of liquidity suppliers. These gains are usually measured by the realized spread which subtracts the price impact from the effective spread, since the price impact is detrimental to the liquidity provider's profits if prices move in the direction of the trade initiator.

The price impact is given by

$$DPI_i = 2o_i(m_{i+\Delta} - m_i)$$

where $m_{i+\Delta}$ is the mid-point at Δ units, here chosen to be 10 minutes, after the i-th trade and the realized spread is, thus,

$$\begin{aligned} DRS_i &= DES_i - DPI_i \\ &= 2o_i(p_i - m_{i+\Delta}). \end{aligned}$$

For each stock-day I compute the volume weighted averages $L = \sum_i v_i L_i / V$ for the respective liquidity measure $L_i \in \{DES_i, DPI_i, DRS_i\}$. I compare the measures computed from the true trade initiator label and the knowledge of which mid-point belongs to which trade, with the measures computed using the estimated trade initiator label and the associated mid-point provided by the algorithms. The setups for the application of the algorithms to the data are as in the previous section. The timestamps of the trades are either not affected by noise or they are delayed by exponential noise with intensity $\beta = 10^{-j}$ with $j = 2, 3$. The precision of the timestamp is reduced to 10^{-i} for $i = 0, \ldots, 3$.

Figure 9 shows the root-mean-square error and the estimated mean across all stock-days measured in cents.[21] The red line in the graphs for the mean indicates the mean measured from the true trade initiator label.

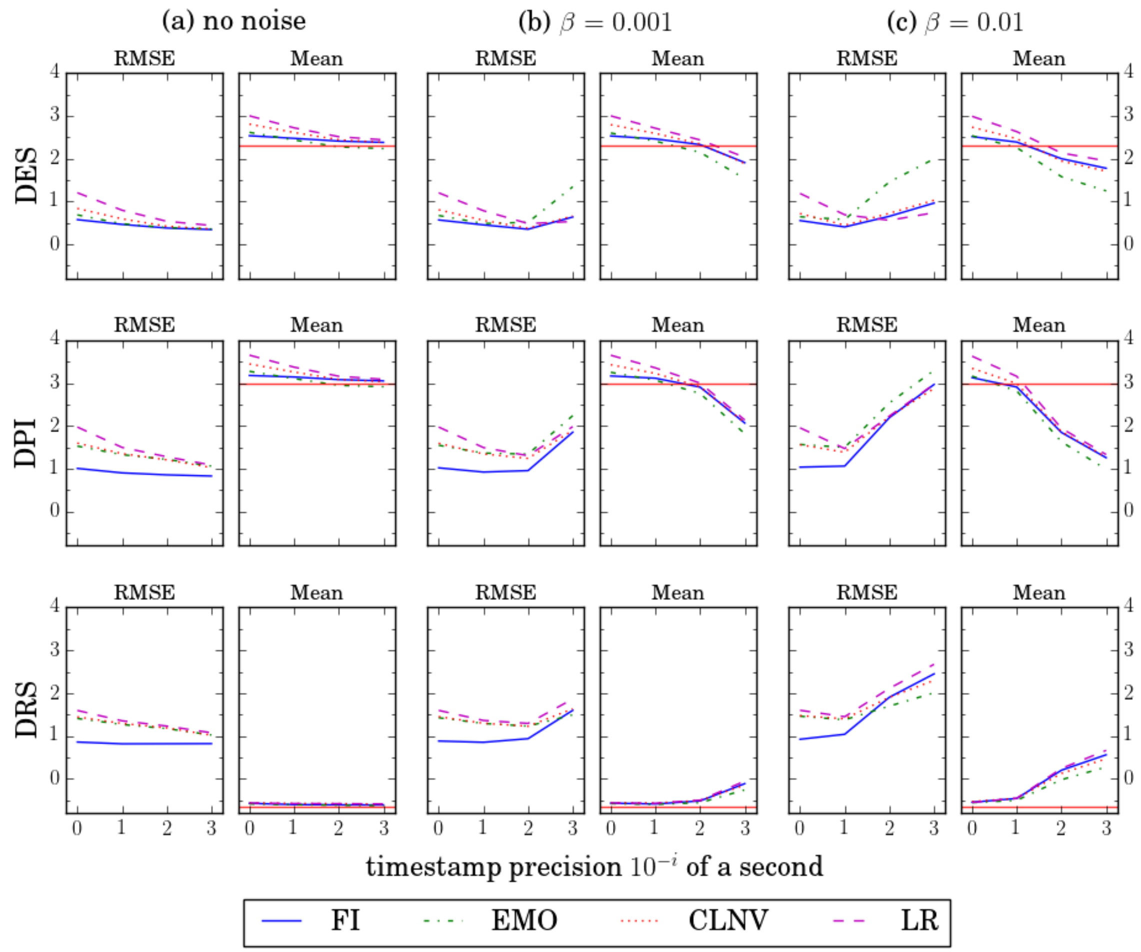
Again, the estimates based on the FI algorithm generally provide the smallest root-mean-square error. The improvements over the traditional algorithms are strongest for the dollar price impact and the dollar realized spread at the timestamp precisions of seconds and 10-th of a second. For example, for the data timestamped at seconds, the average deviation of the stock-day estimate of the realized spread around the true value is 0.9¢ for the FI algorithm, while it is 1.41¢ for the best competitor, the EMO algorithm. Even at high timestamp precision, the estimates of the traditional algorithms do not provide the same precision as the ones of the FI algorithm at lower timestamp precision.

Also the overall sample means are generally estimated closer to the true ones for the FI algorithm than for the other algorithms. Only the EMO algorithm provides very similar mean estimates. The EMO algorithm is, however, more strongly affected at high timestamp precisions than the other algorithms.

The advantage of applying the FI algorithm at lower timestamp precision is again visible in columns (b) and (c), where trade times are affected by noise. The performance of the algorithms deteriorates at high timestamp precision if trades are reported with even mild delay, which translates into poor estimates of the liquidity measures. In the absence of noise the estimates of the FI algorithm at low timestamp precision, however, are barely different from the ones at high timestamp precision, but offer strong robustness against the report delay.

[21] Table 9 provides the corresponding numbers. The root-mean-square error is given by RMSE = $\sqrt{\sum_{k,d}(L_{k,d} - \hat{L}_{k,d})^2/|K \times D|}$, where $|K \times D|$ is the number of stock-days.

Figure 9: Estimating Liquidity



Notes: This figure shows the sample averages and the root-mean-square error between the stock-day estimates and the true values of the dollar effective spread (DES), the dollar price impact (DPI) and the dollar realized spread (DRS) as defined in the text displayed in cents. The true values are computed from the true trade initiator label and the knowledge of the ask and bid in place at the time of any given trade. The estimates are constructed from the classification results of the different algorithms and the ask and bid quotes that they assume to be in effect at the time of the trade. The algorithms are applied to the data with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds.

7 Adjusting the FI Algorithm to Different Data Structures

So far, we assumed the same level of data granularity (summarized in Data Structure 1) that is provided by the reconstructed limit order book from the NASDAQ TotalView-ITCH data. The advantage of the FI algorithm over the traditional approaches feeds on the use of information offered from this granularity. In this section, I will relax the assumptions in Data Structure 1 and present appropriate adjustments to the algorithm. The results show that, despite a slight decline in the accuracy of the algorithm, it still provides sizable improvements over the traditional algorithms particularly with regard to estimation of the order imbalance and liquidity.[22]

7.1 Relaxing Assumption (i) of Data Structure 1

Assumption (i) of 1 states that each transaction against a visible standing order is reflected by a corresponding decrease in volume at the respective quote. This includes transactions that are part of an order too big to be filled by a single standing order. Even though the transactions between the parties involved are carried out almost instantaneously in the order book, we assumed that the data displays the successive steps in the execution according to its order precedence rules.[23] In this section, we assume instead that at the time of a trade the order book displays the state of the order book after the completion of the order that led to the trade.

Data Structure 2. *Aggregated Quote Changes*

(i) At the time of a trade, the order book displays the new state of the order book after the completion of all transactions that were carried out due to the same buy or sell order.

(ii) Trades and quotes are reported in the correct order.

7.1.1 The FI Algorithm under Data Structure 2

The change in the data structure means that we cannot use the strict equality between the transaction volume and the change in volume at the quote to eliminate potential matches. If an order for 100 shares trades against two limit orders for 50 shares each, posted at the same price, the trade data record two transactions for 50

[22] The results for the traditional algorithms in terms of their classification accuracy under the changes to the data structure are not reported. The simplicity of the traditional algorithms means that they are affect by the upcoming changes only in very special cases. The appendix presents the results form the application under noise and for the empirical application. From these one can see that the impact of the different data structures on the traditional algorithms are indeed negligible.

[23] Order precedence rules determine the order in which standing orders are executed when a marketable order enters the order book. Usually, the order offering the best price is executed first. If several visible orders offer the same price, the one that was submitted earliest is executed first (visible orders are usually precedented over hidden orders at the same price, even if the hidden order was submitted first), and so on.

shares each, while the order book data shows a decrease in volume at the respective quote by 100 shares.

Hence, we change the search for a match among the ask quotes in Step 2 of the algorithm to

$$\boldsymbol{\alpha} = \min\{j \in \mathcal{J}_a \colon p_i = a_j \text{ and } v_i \leq \Delta v_j^a\},$$

and analogously for the bid.

If we find a match at one or both sides of the order book, the algorithm proceeds as before. If, however, we cannot find a match on either side of the order book, we need to insert two additional steps before we can conclude that we apparently face a transaction involving a hidden order, which would be classified under Step 4.

Consider a market order for a number of shares greater than what is available at the best quote. The trade data will show the corresponding transaction at the next-best quote, but the order book data will not show any decrease in volume at that quote. In the extreme case, where the market order is so large that it will go through several levels of the order book, the order book data will not even show the quotes against which the order executed on its way to the last quote.

To accommodate these cases the adjusted algorithm injects two additional searches for a match at the ask or bid side before it proceeds with Step 4. The first search (demonstrated for the ask) under Step 4a is conducted as

$$\widetilde{\boldsymbol{\alpha}} = \min\{j \in \mathcal{J}_a \colon p_i = a_j \text{ and } a_{j-1} < a_j\}.$$

In case we find a match on one or both sides of the order book we proceed as prescribed by Step 2.[24]

The second additional search for a match among the quotes if we cannot find one under Step 4a, is conducted under Step 4b (again demonstrated for the ask) as

$$\hat{\boldsymbol{\alpha}} = \min\{j \in \mathcal{J}_a \colon p_i > a_j \text{ and } a_{j+1} < p_i\},$$

and analogously for the bid, proceeding exactly as under Step 4a if a match on one or both sides can be found. If again neither a match at the ask side nor the bid side can be found, we are likely facing a hidden order and the classification is derived under Step 4 as before.

[24]Note that if we classify the transaction according to the interpolated time, we do not adjust the volume at the matched quote, as there was no corresponding volume change to begin with.

7.1.2 Results for Varying Timestamp Precisions

The evaluation of the adjusted FI algorithm (FI_{DS2}) is conducted as before. The order book data, however, has been changed to reflect the new data structure. That is, all intermediate changes in the order book due to a trades that are executed against more than one counter-party are neglected. Table 10 in the appendix presents the results.

The conclusions from the application of the algorithm to the more granular data do not change. The classification accuracies at relatively high timestamp precisions are not different to the accuracies under more granular data. The classification accuracy suffers a bit from the loss of information in the order book data only for the data timestamped at seconds. Still, the FI_{DS2} algorithm improves the classification accuracy of the traditional algorithms by more than 3%-points.

7.1.3 Results for Noisy Timestamps

The results from the application of the FI_{DS2} algorithm to the data with delayed transaction timestamps are presented in Figure 11 in the appendix.[25] Again, the conclusions from the earlier exercise using the original data structure do not change. At high timestamp precision the classification accuracy of all algorithms is strongly affected even by relatively moderate noise intensities. Decreasing the timestamp precision, however, helps to counteract this adverse effect. Over the range of timestamp precisions that show stable results over different noise intensities the FI_{DS2} algorithm is more accurate than the traditional ones.

7.1.4 Estimating Liquidity and Order Imbalances

The results for the estimation of the order imbalance and liquidity measures under the new data structure are presented in Figures 12 and 14, along with the corresponding Tables 11 and 13, in the appendix. As before, the algorithms are applied to the data with timestamp precisions ranging from seconds to milliseconds and under varying degrees of noise added to the trade times ranging from no noise to exponential noise with intensities $\beta = 10^{-3}, 10^{-2}$. The results from the estimation of the order imbalance for different ranges of the true order imbalance are presented in Figure 13

[25] Due to the more granular data structure in the previous sections, we were basically treating each transaction as a single trade. Each transaction was therefore shocked by a separate noise realization. Here, since we count transactions that belong to the same marketable order as a single trade, I shock transactions belonging to the same order by the same noise realization.

and Table 12.

The conclusions regarding the improvements achieved by the FI_{DS2} algorithm do not change. By and large, the estimates based on the FI_{DS2} algorithm provide the smallest root-mean-square error. That is, the FI_{DS2} algorithm provides the most precise estimates. For example, the average deviation of the order imbalances based on the FI_{DS2} algorithm around the true ones is 9%-points compared to 13%-points for the competitors when applied at second precision timestamps for any considered degree of noise. As before, the RMSE and Bias show particular improvement in the extremes of the order imbalance. The RMSE of the dollar realized spread under no noise and a timestamp precision of seconds is 0.98¢ per share for the FI_{DS2} algorithm compared to 1.43¢ for the EMO.

7.2 Relaxing Assumption (ii) of Data Structure 1 and 2: Randomized Order of Trades

In certain datasets, trades may not follow the actual order in which they were executed (e.g. Easley et al., 2016). That may be due to two reasons. First, the legal framework may allow for some delay in reporting trades. Depending on whether the timestamp of the data reflects the time of the report or the time of the actual trade and depending on the extent to which trading institutions exploit their right of delayed reporting, trades may be out of order. Second, for data from a consolidated tape, which timestamps trades when the corresponding data are processed, trades are out of order due to different latencies for sending information from different market places to the same data processor. These latencies can be expected to be small, but large enough to affect trades that are executed over small intervals.[26]

Nevertheless, we can expect that the FI algorithm will be little, if at all, affected by trades being out of order. The correct order of trades plays a role for the FI algorithm only if it uses the tick-test, which it rarely does.[27]

[26] For example, the speed of light in a vacuum is roughly $300 * 10^6$ m/s. Sending data from Chicago to New York (a distance of around 1300km) at the speed of light would thus take 4ms. So even at this physically lower limit of transmission time the report delay of a Chicago trade is 4ms compared to a trade at the NYSE where the consolidated tape is located.

[27] To a lesser extent, the correct order of trades also plays a role if the algorithm uses the interpolated time of quotes to classify a trade. If a trade is classified using the interpolated time, the volume at the quote that is matched to the trade is reduced by the size of the transaction. This is done because two different trades cannot cause the same quote change and to avoid that this quote causes further conflicts between an assignment of trades to either the ask or the bid side. If there are several trades with the same price and volume and these trades are out of order, it is possible that the FI algorithm assigns these trades to the conflicting ask and bid quotes in

We already examined the consequences of trades being out of order in the sections where we delayed the trade times by exponential noise (though we did not mentioned it explicitly). When we add to each trade time an independently distributed exponential variable, the order of trades can change. For example, for two trades with the second trade following one millisecond after the first trade, the probability that the first trade will be shifted behind the second one if both trades are affected by exponential noise with $\beta = 1/10^3$ is 0.1839.[28] We saw from the previous exercises under noisy trade times that the randomization of the order of trades did not greatly influenced the accuracy of both versions of the FI algorithm and did not greatly affect the performance against the traditional algorithms.

7.3 Relaxing Assumption (ii) of Data Stucture 2: Randomized Order of Trades and Quotes

The reasons for the possibility that trades could be out of order apply to the recorded quotes, at least for a consolidated tape, just as well. If, indeed, quote changes are out of order the decision criteria of the FI algorithm have to be adjusted. The definition of the change in volume used to find matches between transactions and quotes is only meaningful if the order of the quotes is correct.

Data Structure 3. *Aggregated Quote Changes and Random Trade and Quote Order*

(i) At the time of a trade the order book displays the new state of the order book after the completion of all transactions that were carried out due to the same buy or sell order.

(ii) Trades and quotes can be out of order.

---

the exact opposite order in which they actually occurred. However, for statistics like the order imbalance such errors are irrelevant.

[28]More formally, for two trades at time t_1 and t_2 with $t_2 = t_1 + \Delta$ and $\Delta \geq 0$ the probability that the first trade is shifted behind the second one due to noise is given by

$$P(t_1 + \varepsilon_1 > t_2 + \varepsilon_2) = \int_{\Delta}^{\infty} f(\varepsilon_1) F(\varepsilon_1 - \Delta)\, \mathrm{d}\varepsilon_1,$$

with $\varepsilon_i \in \mathbb{R}_{\geq 0}$ and $\varepsilon_i \overset{\text{iid}}{\sim} F$ for some distribution function F with density f. For F being the exponential distribution $\text{Exp}(1/\beta)$ this is given by $\exp\{-\Delta/\beta\}/2$.

7.3.1 The FI Algorithm under Data Structure 3

Instead of the change in volume we can now rely only on the absolute volume displayed at the respective quote. For a transaction to be executed at a particular quote the volume of the transaction cannot exceed that of the volume available at the quote. Therefore, the search of a match between a transaction and a quote in Step 2 is changed to (demonstrated for the ask)

$$\boldsymbol{\alpha} = \min\{j \in \mathcal{J}_a : p_i = a_j \text{ and } v_i \leq v_j^a\},$$

and analogously for the bid. From here, the adjusted algorithm proceeds as the baseline version.[29] If the classification is derived under Step 3 the algorithm subtracts the transaction volume from the volume available at the matched quote. Step 4a and Step 4b from the previous adjustment to the algorithm do not apply here because they rely on the correct order of the quotes.

7.3.2 Results for Varying Timestamp Precisions

Table 14 in the appendix shows the results of the application of the FI algorithm adjusted for the new data structure (FI_{DS3}) applied to the data not being affected by noise. Although the order of trades and quotes will thus not be affected, the exercise demonstrates the loss in classification accuracy we have to incur for not allowing us to use the full amount of information. We see that the algorithm, albeit precise at high timestamp precisions, reacts with greater sensitivity to the reduction in timestamp precision than the previous versions, as there are more situations where both ask and bid quotes seem to provide a match to the transaction. However, the FI_{DS3} algorithm still achieves a 1.6 to 3.0%-points improvement over the traditional algorithms at a timestamp precision of seconds.

7.3.3 Results for Noisy Timestamps

To study the effect of random trade and quote order, I add exponential noise to the timestamps of both trade and quote data. Note that in doing so, not only will the order of trades and quotes change, but trades may now also be reported before their corresponding quote change.

Figure 15 in the appendix shows that, contrary to the above setups, there are regions of timestamp precisions and noise where the traditional algorithms outper-

[29] Note, however, that the auxiliary variables l^a and l^b are not updated after a classification.

form the $\mathrm{FI}_{\mathrm{DS3}}$ algorithm. At these regions of higher timestamp precision, however, classification accuracy is quite low for all the algorithms and not stable across the different degrees of noise. At the timestamp precisions that ensure that the algorithms are not too strongly affected by noise, the $\mathrm{FI}_{\mathrm{DS3}}$ algorithm again outperforms the others.

7.3.4 Estimating Liquidity and Order Imbalances

Importantly, with respect to estimating the order imbalance and measuring liquidity Figures 16, 17 and 18, as well as Tables 15, 16 and 17 in the appendix show that despite the decrease in classification accuracy due to the changes in the data structure, the $\mathrm{FI}_{\mathrm{DS3}}$ still achieves considerable improvements in terms of the root-mean-square error. Also, the bias for the estimation in the extremes of the true order imbalance is considerably reduced by an application of the $\mathrm{FD}_{\mathrm{DS3}}$ algorithm.

8 Discussion

8.1 Further Data Structures

The algorithm (and its various versions) has been designed with data structures in mind that are likely to be faced by many researchers. Still, there are likely to be various databases that do not directly meet the conditions necessary for any of the versions of the FI algorithm to be directly applicable. For example, some exchanges allow the use of so-called iceberg orders, a specific type of limit order which displays only a fraction of the total volume. When trading against such an iceberg order it is quite possible that the transaction volume is greater than what was displayed to be available in the order book.

In that case, we could further simplify the algorithm by using $\boldsymbol{\alpha} = \min\{j \in \mathcal{J}_a : p_i = a_j\}$ in the version for Data Structure 3, which would still provide an improvement over the traditional algorithms. At some point, however, the loss in classification accuracy due to discarding certain criteria in matching trades to their quotes, because these criteria do not strictly comply with the trading rules, may outweigh the loss in classification accuracy that is incurred by applying these criteria nonetheless. In our example, this may be the case, because iceberg orders are rarely used. It is at the discretion of the researcher to make that call.

The structure of the algorithm is easy enough for a researcher to cancel certain criteria, mix certain criteria of the different versions of the algorithm presented here or even add some new criteria to match trades to their quotes if the researcher sees them fit for the specific data at hand. This paper shows that there is an easy gain in classification accuracy by using even very little extra information contained in the data which is not utilized by the traditional algorithms.

8.2 The Speed of the FI Algorithm

An important aspect of the FI algorithm is that it processes more information than the traditional algorithms. This raises the question of computational feasibility. The extensive simulation exercises performed in this study should already have made evident that the application of the FI algorithm does not bring about any serious computational constrains.

On the entire data set consisting of 19842 stock-days with a total of 134,449,578 trades the classification function needs 20.91 minutes on an Intel Core i7 CPU with 3.6 GHz (*not* using the different cores for parallel computing which can easily be done) to sign all trades for the data timestamped to a second, which is on average 9.3 microseconds for a single trade. This includes the time for reading the data from an SQLite database—a task that has to be performed by any algorithm and which actually accounts for the largest fraction of the run-time. It took on average 0.34 seconds for a stock-day of order book data and 0.1 seconds for a stock-day of transaction data to be read into memory. In sum, the use of the additional information by the FI algorithm to arrive at an improved classification accuracy does not come at any noteworthy computational costs.

Of course, the run-time depends on the specific coding design and the program language. I implemented the FI algorithm in the by now very popular Python language using the Cython hybrid language for the computationally more intense parts. The code will be made available and can be used easily by anyone slightly familiar with Python.

9 Conclusion

This paper proposes a new trade classification algorithm that improves the classification of trades into the liquidity demanding and supplying side under the characteristics of today's markets and data records. In particular, the high frequency of quote submission and cancellation pose a problem for established classification algorithms. Under a median of 17 quote changes at the time of a trade, for example, the new algorithm manages to reduce misclassification rates by half. The mechanism of the new algorithm can also be used to ameliorate the problem of delayed reporting of trades that is usually a concern when analyzing historical data from the consolidated tape of US stock exchanges. The improvements in classification rates also translate into considerable improvements in the estimation of transaction costs and order imbalances.

The evidence presented in this paper also raises some concern about using the DTAQ data in combination with the traditional classification algorithms without worrying about data quality. Unfortunately, not having access to the DTAQ data I cannot address to which extent the proposed algorithm is able to improve the classification of trades from the consolidated tape. To the extent that the DTAQ suffers from the different data deficiencies analyzed in this paper, however, we saw that the improvement can be sizable.

References

Angel, J. J., Harris, L. E., Spatt, C. S., 2011. Equity trading in the 21st century. The Quarterly Journal of Finance 1 (01), 1–53.

Angel, J. J., Harris, L. E., Spatt, C. S., 2015. Equity trading in the 21st century: An update. The Quarterly Journal of Finance 5 (01), 1–39.

Bernile, G., Hu, J., Tang, Y., 2016. Can information be locked up? Informed trading ahead of macro-news announcements. Journal of Financial Economics 121 (3), 496–520.

Bessembinder, H., 2003. Issues in assessing trade execution costs. Journal of Financial Markets 6 (3), 233–257.

Blais, M., Protter, P., 2012. Signing trades and an evaluation of the Lee–Ready algorithm. Annals of Finance 8 (1), 1–13.

Bowsher, C. G., 2007. Modelling security market events in continuous time: Intensity based, multivariate point process models. Journal of Econometrics 141 (2), 876–912.

Brennan, M. J., Huh, S.-W., Subrahmanyam, A., 2018. High-frequency measures of informed trading and corporate announcements. The Review of Financial Studies 31 (6), 2326–2376.

Chakrabarty, B., Li, B., Nguyen, V., Van Ness, R. A., 2007. Trade classification algorithms for electronic communications network trades. Journal of Banking & Finance 31 (12), 3806–3821.

Chakrabarty, B., Moulton, P. C., Shkilko, A., 2012. Short sales, long sales, and the Lee–Ready trade classification algorithm revisited. Journal of Financial Markets 15 (4), 467–491.

Chakrabarty, B., Pascual, R., Shkilko, A., 2015. Evaluating trade classification algorithms: Bulk volume classification versus the tick rule and the Lee-Ready algorithm. Journal of Financial Markets 25, 52–79.

Chordia, T., Goyal, A., Jegadeesh, N., 2016. Buyers versus sellers: Who initiates trades, and when? Journal of Financial and Quantitative Analysis 51 (5), 1467–1490.

Chordia, T., Hu, J., Subrahmanyam, A., Tong, Q., 2017. Order flow volatility and equity costs of capital. Management Science Forthcoming.

Chordia, T., Roll, R., Subrahmanyam, A., 2002. Order imbalance, liquidity, and market returns. Journal of Financial economics 65 (1), 111–130.

Dorn, D., Huberman, G., Sengmueller, P., 2008. Correlated trading and returns. The Journal of Finance 63 (2), 885–920.

Easley, D., de Prado, M. L., O'Hara, M., 2016. Discerning information from trade data. Journal of Financial Economics 120 (2), 269–285.

Easley, D., de Prado, M. M. L., O'Hara, M., 2012. Flow toxicity and liquidity in a high-frequency world. Review of Financial Studies 25 (5), 1457–1493.

Easley, D., Kiefer, N. M., O'hara, M., Paperman, J. B., 1996. Liquidity, information, and infrequently traded stocks. The Journal of Finance 51 (4), 1405–1436.

Ellis, K., Michaely, R., O'Hara, M., 2000. The accuracy of trade classification rules: Evidence from Nasdaq. Journal of Financial and Quantitative Analysis 35 (04), 529–551.

Finucane, T. J., 2000. A direct test of methods for inferring trade direction from intra-day data. Journal of Financial and Quantitative Analysis 35 (04), 553–576.

Fong, K. Y., Holden, C. W., Trzcinka, C. A., 2017. What are the best liquidity proxies for global research? Review of Finance 21 (4), 1355–1401.

Hautsch, N., Huang, R., 2012. Limit Order Flow, Market Impact and Optimal Order Sizes: Evidence from NASDAQ TotalView-ITCH Data. In: Abergel, F., Bouchaud, J.-P., Foucault, T., Lehalle, C.-A., Rosenbaum, M. (Eds.), Market Microstructure: Confronting Many Viewpoints. Wiley, pp. 136–161.

Henker, T., Wang, J.-X., 2006. On the importance of timing specifications in market microstructure research. Journal of Financial Markets 9 (2), 162–179.

Holden, C. W., Jacobsen, S., 2014. Liquidity measurement problems in fast, competitive markets: expensive and cheap solutions. The Journal of Finance 69 (4), 1747–1785.

Hu, J., 2014. Does option trading convey stock price information? Journal of Financial Economics 111 (3), 625–645.

Hu, J., 2017. Option listing and information asymmetry. Review of Finance Forthcoming.

Huang, R., Polak, T., 2011. Lobster: Limit order book reconstruction system. Tech. rep., School of Business and Economics, Humboldt Universität zu Berlin.
URL `https://lobsterdata.com/info/docs/LobsterReport.pdf`

Huang, R. D., Stoll, H. R., 1996. Dealer versus auction markets: A paired comparison of execution costs on NASDAQ and the NYSE. Journal of Financial economics 41 (3), 313–357.

King, G., Zeng, L., 2001. Logistic regression in rare events data. Political analysis, 137–163.

Kremer, S., Nautz, D., 2013. Causes and consequences of short-term institutional herding. Journal of Banking & Finance 37 (5), 1676–1686.

Lee, C., Ready, M. J., 1991. Inferring trade direction from intraday data. The Journal of Finance 46 (2), 733–746.

Lee, C. M., Radhakrishna, B., 2000. Inferring investor behavior: Evidence from torq data. Journal of Financial Markets 3 (2), 83–111.

Liu, L. Y., Patton, A. J., Sheppard, K., 2015. Does anything beat 5-minute RV? A comparison of realized measures across multiple asset classes. Journal of Econometrics 187 (1), 293–311.

Muravyev, D., 2016. Order flow and expected option returns. The Journal of Finance 71 (2), 673–708.

Odders-White, E. R., 2000. On the occurrence and consequences of inaccurate trade classification. Journal of Financial Markets 3 (3), 259–286.

O'Hara, M., 2015. High frequency market microstructure. Journal of Financial Economics 116 (2), 257–270.

Panayides, M., Shohfi, T., Smith, J., September 2014. Comparing trade flow classification algortihms in the electronic era: The good, the bad, and the uninformative, mimeo.

Perlin, M., Brooks, C., Dufour, A., 2014. On the performance of the tick test. The Quarterly Review of Economics and Finance 54 (1), 42–50.

Peterson, M., Sirri, E., 2003. Evaluation of the biases in execution cost estimation using trade and quote data. Journal of Financial Markets 6 (3), 259–280.

Piwowar, M. S., Wei, L., 2006. The sensitivity of effective spread estimates to trade–quote matching algorithms. Electronic Markets 16 (2), 112–129.

Rosenthal, D. W., 2012. Modeling trade direction. Journal of Financial Econometrics 10 (2), 390–415.

Theissen, E., 2001. A test of the accuracy of the Lee/Ready trade classification algorithm. Journal of International Financial Markets, Institutions and Money 11 (2), 147–165.

Vergote, O., March 2005. How to match trades and quotes for nyse stocks? Center for Economic Studies Discussion Paper Series.

A Extracting Trade and Quote Data from LOBSTER's Message Files

The software LOBSTER reconstructs from the original NASDAQ TotalView-ITCH data feed the full limit order book, as well as a message file containing information on the events causing the changes in the order book.[30]

Figure 10: Data Construction

Message File					Order Book Data			
time	event	size	price	direction	ask price	ask size	bid price	bid size
t1	1	200	105	-1	105	200	-999999	0
t2	1	150	100	1	105	200	100	150
t3	4	80	105	-1	105	120	100	150
t4	5	50	103	1	105	120	100	150
t5	2	70	105	-1	105	50	100	150
t6	4	50	100	1	105	50	100	100
t7	3	100	100	1	105	50	-999999	0

Transaction Data

time	price	size	direction	initiator
t3	105	80	-1	1
t4	103	50	1	-1
t6	100	50	1	-1

Ask Data

time	price	size
t1	105	200
t3	105	120
t5	105	50

Bid Data

time	price	size
t2	100	150
t6	100	100
t7	-999999	0

Notes: Each row in the Message File describes the cause of the change in the order book from the previous row to the next. Event types 1, 2 and 3 refer to the submission, partial cancellation and total deletion of a limit order, events 4 and 5 to the execution of a visible and hidden limit order, respectively. The direction 1 (-1) refers to a buy (sell) limit order.

The gray shaded area in Figure 10 provides an example of the design of LOBSTER's message and order book files. The k-th row of the message file describes the cause of the change in the order book from the $(k-1)$-th row to the k-th row. The events 1, 2 and 3 refer to the submission, partial cancellation and total deletion of a limit order. The events 4 and 5 refer to the execution of a visible and hidden limit order, respectively. The direction indicates whether a buy (+1) or sell (-1) limit order is affected. If a hidden order is executed, the order book is not visibly

[30]For more information on the TotalView-ITCH data feed and the order book reconstruction by LOBSTER, see Hautsch and Huang (2012) and Huang and Polak (2011).

affected. In that case, to maintain a symmetric output, the LOBSTER order book data displays the order book's state after the execution of the hidden order.

As an example take the first row of the order book and message file. We start here with an empty order book which I indicated by negative quotes. At `t1` the message file indicates a submission of a limit sell order for a price of `105` per share for a total of `200` shares. In the same row, the order book displays its new state. The bid side is still empty and the ask side is now displaying the price and volume of the limit sell order.

Below the gray shaded area in Figure 10, it is illustrated how I extract the trade and quote data from the order book and message file. I construct the trade data by extracting all visible and hidden executions of limit orders (events 4 and 5) from the message file, with the respective information on the price and volume of the transactions. As the direction in the message file refers to the limit order, the initiator is given by the opposite party to the trade. Note that I omit the active counter-party to each trade from the trade data. In doing so, however, I do not omit any relevant information as the counter-party simply mirrors the passive trade with opposite trade direction. I remind the reader of that decision in the main text at any point where it is relevant, and discuss the alternative of including the counter-party to each trade.

The data for the ask side of the order book is constructed by extracting the state of the order book at any point the ask side is affected by the submission, cancellation, deletion or execution of a visible sell limit order (events 1, 2, 3, 4). Any event that is related to a hidden order is omitted. The construction of the bid side follows analogously.

B Approximation of the Reduction in Classification Accuracy due to Noisy Timestamps

To get a feeling of how we can expect the results to be affected if the reported time of trades is delayed by a random amount, we can calculate the reduction in classification accuracy under a few distributional assumption. The approximation may also help the practitioner to choose the appropriate timestamp precision at which to apply the classification algorithm in her data set, if she has a rough idea of the degree of noise.

Let the timestamp of a trade reflect the actual trade time plus noise, $\varepsilon \in \mathbb{R}_{\geq 0}$, which follows the distribution $F(\varepsilon)$. Let the probability of a trade at some point x over an interval of length s be determined by the density $g(x)$, $0 \leq x < s$. The fraction of trades that is placed outside the interval in which they actually occurred is then given by $\int_0^s g(x)(1 - F(s - x))\,\mathrm{d}x$.

For example, if trades are equally distributed over the interval s and the delay in reported time follows the exponential distribution, $\varepsilon \sim \mathrm{Exp}(1/\beta)$, the fraction of trades placed to the right of the interval during which they actually occurred is $\beta(1-\exp\{-s/\beta\})/s$. For an interval of the length of a second ($s = 1$) and an average delay of one 10-th of a second ($\beta = 0.1$), that would mean that 10% of trades are reported outside the second in which they occurred.

Denoting the classification accuracy of all trades that lie in the correct interval by $A(s)$, the overall classification accuracy is given by

$$A(s) \int_0^s g(x)F(s - x)\,\mathrm{d}x + 0.5 \int_0^s g(x)(1 - F(s - x))\,\mathrm{d}x$$

assuming that the average classification accuracy of trades outside their actual time interval is 0.5. That is, the reduction in the accuracy due to delayed report times is given by

$$\int_0^s g(x)(1 - F(s - x))\,\mathrm{d}x\ (A(s) - 0.5).$$

For example, given the above classification accuracy of 95% for data timestamped at seconds ($s = 1$, with a median of 17 quote changes at the time of trades), we would expect the reduction in accuracy to be around 4.5%-points due to noise of intensity $\beta = 0.1$. Note that the reduction in accuracy may exceed the 4.5%-points if the classification accuracy for trades not shifted outside their interval is affected by the permutation of trades or if the accuracy of trades shifted just behind the interval at which they actually occurred is less than 0.5.

Given that at any timestamp precision s we found that in the absence of noise $A(s; \mathrm{FI}) \gtrsim A(s; j)$ for $j = \mathrm{EMO}$, CLNV, LR, we would expect the FI algorithm to dominate under noise as well.

C Figures and Tables

Table 2: Ticker Names

AAPL	AA	ABB	ABT	ACE	ACN	ADBE	ADM	ADP
ADS	AEP	AGN	AGU	AIG	AKAM	ALK	ALL	AME
AMGN	AMT	AMX	AMZN	AN	AON	AOS	APC	APD
APH	ASH	ASR	AVGO	AVY	AXP	AYI	AZN	BAC
BAM	BAX	BA	BBL	BBT	BCE	BEAV	BEN	BHI
BHP	BIDU	BIIB	BK	BLK	BLL	BMS	BMY	BP
BRFS	BR	BTI	BT	BUD	BX	CAJ	CAT	CCK
CELG	CF	CHA	CHL	CHRW	CHT	CHU	CLX	CL
CMCSA	CMCSK	CME	CMI	CM	CNI	CNQ	COF	COP
COST	CPA	CPRT	CP	CRH	CRM	CSCO	CSGP	CSX
CTRP	CTSH	CUK	CVS	CVX	C	DAL	DCM	DD
DEO	DE	DHR	DISH	DIS	DOW	DTV	DUK	DVN
D	EBAY	ECL	EL	EMC	EMR	ENB	ENR	EOG
EPD	ESRX	ETE	ETN	EXC	EXPD	E	FCX	FDX
FIS	FLT	FMX	F	GD	GE	GG	GILD	GIS
GLW	GMCR	GM	GOOG	GPK	GPN	GPRO	GSK	GS
GT	GWR	HAL	HD	HMC	HON	HPQ	HSY	IBM
IBN	IGT	ILMN	IMO	INFY	INTC	IP	IR	ITW
JAH	JBHT	JBLU	JCI	JPM	KAR	KMB	KMX	KO
KR	KSU	K	LBTYA	LBTYK	LEG	LFL	LLY	LMT
LNKD	LOW	LO	LUV	LVS	LYB	MA	MCD	MCK
MELI	MET	MGA	MHK	MJN	MMC	MMM	MON	MOS
MO	MPC	MRK	MSCI	MSFT	MS	MT	NCR	NEE
NGG	NKE	NLSN	NOC	NSC	NTES	NTT	NUE	NVO
NVS	ODFL	ORCL	OXY	PAC	PBR	PCAR	PCLN	PCP
PEP	PFE	PG	PHG	PH	PKG	PKX	PM	PNC
POT	PPG	PRU	PSA	PTR	PX	QCOM	RAI	REGN
RIO	RKT	ROP	RTN	RYAAY	SAP	SAVE	SBUX	SCCO
SCHW	SIAL	SLB	SNE	SNP	SNY	SON	SO	SPB
SPG	SRE	STO	STT	STZ	SU	SWFT	SYT	SYY
TEF	TEL	TEVA	TGT	TJX	TMO	TM	TOT	TRP
TRV	TSLA	TSM	TSS	TS	TTM	TWC	TWX	TXN
T	UAL	UL	UNH	UNP	UN	UPS	USB	UTX
VALE	VFC	VLO	VMW	VRX	VZ	V	WFC	WHR
WIT	WMB	WMT	WM	WPZ	WU	XOM	XRX	YHOO
YUM	Z							

Notes: This table provides the ticker names of all stocks included in the sample. However, not all of these stocks are analyzed over the whole range of the sample as some stock-days may not have fulfilled the criteria mentioned in the data section (day-end price ≥ 1 \$ and number of trades $\geq$ 10), or due to an initial public offering during the sample period (e.g. Z).

Table 3: Traditional Trade Classification Algorithms

Variables: p_i – transaction price of the i^{th} trade; a_i, b_i – ask, bid price corresponding to the i^{th} trade; o_i – trade initiator; $i = 1, \ldots, I$

Tick-Test	**LR** (Lee and Ready, 1991)	**EMO** (Ellis et al., 2000)	**CLNV** (Chakrabarty et al., 2007)

for $i = 2 : I$ **do**
  $j = 0$
  while $i - j > 0$ **do**
    $j = j + 1$
    if $p_i > p_{i-j}$ **then**
      o_i = buyer
      break
    else if $p_i < p_{i-j}$ **then**
      o_i = seller
      break
	for $i = 1 : I$ **do**
	  $m_i = (a_i + b_i)/2$
	  if $p_i > m_i$ **then**
	    o_i = buyer
	  else if $p_i < m_i$ **then**
	    o_i = seller
	  else
	    apply tick-test
		for $i = 1 : I$ **do**
		  if $p_i = a_i$ **then**
		    o_i = buyer
		  else if $p_i = b_i$ **then**
		    o_i = seller
		  else
		    apply tick-test
			for $i = 1 : I$ **do**
			  $\underline{a} = 0.7a_i + 0.3b_i$
			  $\overline{b} = 0.3a_i + 0.7b_i$
			  if $\underline{a} < p_i \leq a_i$ **then**
			    o_i = buyer
			  else if $b_i \leq p_i < \overline{b}$ **then**
			    o_i = seller
			  else
			    apply tick-test

Table 4: Summary Statistics of NASDAQ's Transaction and Quote Data

	mean	std	min	25%	median	75%	max
T	6776.01	7647.55	17	1673	4397	9247.75	106407
V	1131.88	2347.50	1.37	165.53	467.49	1160.76	58115.01
V/T	129.02	87.20	44.65	95.50	108.31	129.33	3573.15
$\%\{V \geq 100\}$	0.77	0.12	0.21	0.71	0.79	0.87	0.98
$\%\{V = 100\}$	0.62	0.12	0.19	0.55	0.63	0.70	0.92
P	59.01	54.25	4.48	32.07	48.22	69.96	623.37
$\#Q$	104.10	102.26	1.16	28.62	70.39	150.38	908.74

Notes: This table provides summary statistics to the following variables computed for each stock-day: T – Number of trades, V – Trading volume in 1000 shares, V/T Volume per trade (stock-day average), $\%\{V \geq 100\}$ – Percentage of trades with volume greater or equal to 100 shares, $\%\{V = 100\}$ – Percentage of trades with trading volume equal to 100 shares, P – Price per share (stock-day average), $\#Q$ – Number of quote changes in 1000.

Compared to Chakrabarty et al. (2015), who study ITCH-data from the same time span for a size stratified sample of 300 stocks, my sample displays slightly higher trading activity measured by the daily, cross-sectional average of the total number of shares traded. Also, trades in my sample tend to be smaller and higher priced. However, the sample retains a great deal of variability and as the purpose is to analyze trade classification under the problem of imprecise timestamps, a focus on slightly more frequently traded stocks seems only proper. The average number of transactions on a stock-day is 6776, which is less than a trade per second. The average number of quote changes, however, is substantially larger with 104,100 quote updates on a stock-day, which is almost 4.5 quote updates per second. If we condition only on seconds where trades took place, the average number of quote updates per second increases to 31.19 (see Figure 2). It is this large number of quote changes that makes accurate trade classification with imprecise timestamps challenging.

Table 5: Classification Accuracy at Different Timestamp Precisions

	timestamp precision 10^{-i} of a second with $i =$					
	0	1	2	3	4	9
Panel A: total volume						
FI	95.02	96.97	97.95	98.34	98.24	98.18
EMO	90.13	93.64	96.08	97.61	98.19	98.20
CLNV	90.14	93.63	96.08	97.61	98.17	98.19
LR	89.88	93.57	96.02	97.52	98.10	98.10
EMOi	72.98	76.67	82.74	91.66	97.49	98.20
CLNVi	73.36	76.99	82.98	91.80	97.52	98.19
LRi	71.84	75.85	82.16	91.28	97.33	98.10
Panel B: average volume						
FI	94.52	96.47	97.38	97.71	97.53	97.44
	(2.40)	(1.91)	(1.91)	(2.04)	(2.05)	(2.09)
EMO	89.49	92.75	94.93	96.52	97.46	97.43
	(3.41)	(2.99)	(2.86)	(2.60)	(2.06)	(2.11)
CLNV	89.39	92.69	94.92	96.55	97.42	97.42
	(3.40)	(3.01)	(2.85)	(2.54)	(2.07)	(2.10)
LR	89.31	92.66	94.79	96.31	97.26	97.21
	(3.55)	(3.08)	(3.00)	(2.77)	(2.26)	(2.32)
EMOi	73.04	77.07	82.70	90.60	96.52	97.43
	(7.51)	(6.98)	(5.65)	(3.77)	(2.53)	(2.11)
CLNVi	74.38	78.21	83.57	91.10	96.61	97.42
	(8.05)	(7.35)	(5.68)	(3.40)	(2.38)	(2.10)
LRi	70.54	75.12	81.25	89.71	96.15	97.20
	(6.81)	(6.42)	(5.54)	(4.20)	(2.90)	(2.33)

Notes: This table shows the percentage of correctly classified trading volume by the FI algorithm and the traditional algorithms using the last quotes from before the time of the trade (EMO, CLNV and LR) and using the interpolated time of trades and quotes (EMOi, CLNVi and LRi) as suggested by Holden and Jacobsen (2014). The algorithms are applied to the data with reduced timestamp precisions (10^{-i} of a second for $i = 0, \ldots, 4$) and using the original precision of nanoseconds (10^{-9} of a second). These correspond to a median number of quote changes at the time of trades ranging from 17 (for $i = 0$) to 1 (for $i = 9$). Panel A shows the percentage of correctly classified volume summed over the entire sample. Panel B shows the average of correctly classified volume over the 19842 stock-days with the standard deviations in brackets.

Table 6: Quote Changes Caused by a Trade

	Number of Quotes									
Quote change due to trade	2	3	4	5	6	7	8	9	10	$\sum$
1st	1,393	859	606	434	321	243	185	144	114	4,299
2nd	513	300	199	139	103	77	60	47	38	1,476
3rd		211	142	99	72	54	42	33	27	680
4th			112	81	61	46	35	27	22	384
5th				69	50	40	30	24	19	232
6th					44	34	28	22	18	145
7th						31	24	20	16	92
8th							22	18	15	55
9th								17	14	31
10th									13	13
$\sum$	1,906	1,370	1,060	821	650	525	427	353	295	7,407

Notes: This table shows the number of times a trade is executed on the first, second, third, ..., 10-th quote within seconds of 2, 3, ..., 10 quote changes. Only seconds with a single trade were used for the computation. All numbers are presented in thousands. For example, entry (1st, 2) means that within seconds containing 2 quote updates and one trade (of which there are 1.9 million), 1.4 million of the first quote changes were due to a trade.

Table 7: RMSE and Mean Order Imbalance for Data Structure 1

		RMSE				Mean($\|OI\|$)			
		timestamp precision 10^{-i} of a second							
β	method	$i = 0$	1	2	3	0	1	2	3
Panel A: $\tau = 10$, true average absolute order imbalance = 18.56									
no noise	FI	8.06	7.49	7.34	7.48	18.18	18.36	18.40	18.40
	CLNV	12.95	11.08	9.53	8.27	19.31	19.13	18.76	18.44
	EMO	13.01	11.04	9.43	8.05	18.94	18.83	18.54	18.28
	LR	13.21	11.26	9.75	8.56	19.90	19.49	19.00	18.61
0.001	FI	8.22	7.78	8.95	15.68	18.16	18.28	17.88	16.09
	CLNV	12.98	11.17	10.48	15.69	19.27	19.05	18.31	16.23
	EMO	13.07	11.13	10.22	14.75	18.90	18.76	18.10	16.07
	LR	13.22	11.33	10.87	16.84	19.89	19.44	18.52	16.26
0.010	FI	8.52	9.65	18.43	23.62	18.06	17.56	15.09	15.20
	CLNV	13.16	12.23	18.27	23.18	19.06	18.32	15.38	15.22
	EMO	13.25	12.08	17.40	22.05	18.70	18.05	15.16	14.95
	LR	13.30	12.44	19.38	24.78	19.82	18.74	15.44	15.42
Panel B: $\tau = 100$, true average absolute order imbalance = 38.52									
no noise	FI	16.63	14.93	14.33	14.37	38.07	38.25	38.33	38.35
	CLNV	28.16	23.71	20.11	16.92	41.12	40.16	39.27	38.55
	EMO	28.39	23.79	20.07	16.73	40.76	39.87	39.06	38.42
	LR	28.31	23.66	20.18	17.15	41.98	40.65	39.55	38.73
0.001	FI	16.98	15.78	19.16	34.43	38.00	38.10	37.58	35.43
	CLNV	28.21	23.95	22.80	34.37	41.03	40.03	38.66	35.86
	EMO	28.46	24.03	22.45	32.84	40.66	39.74	38.46	35.82
	LR	28.34	23.93	23.33	36.51	41.96	40.58	38.92	35.75
0.010	FI	17.97	21.51	40.98	50.74	37.81	36.98	32.87	33.19
	CLNV	28.72	27.08	40.46	49.98	40.65	38.85	33.58	33.35
	EMO	28.98	26.88	39.03	48.12	40.28	38.58	33.46	33.23
	LR	28.61	27.22	42.56	52.89	41.84	39.48	33.56	33.45

Notes: This table corresponds to Figure 7. It shows the root-mean-square error between estimates of the order imbalance and the true order imbalance, as well as the sample averages of the absolute order imbalances displayed in percentages. The true values are computed from the true trade initiator labels and the estimates from the classification results of the different algorithms. For the computation of the order imbalance each stock-day is split into equally sized volume bins. The number of bins is chosen to be $\tau = 10, 100$. The algorithms are applied to the data with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds.

Table 8: RMSE and Bias for Different Intervals of the Order Imbalance under Data Structure 1

τ	method	$[-1,-0.8]$	$[-0.8,-0.6]$	$[-0.6,-0.4]$	$[-0.4,-0.2]$	$[-0.2,0]$	$[0,0.2]$	$[0.2,0.4]$	$[0.4,0.6]$	$[0.6,0.8]$	$[0.8,1]$
Panel A: RMSE											
10	FI	21.36	15.90	12.05	7.92	5.95	6.11	8.24	12.15	19.88	30.04
	EMO	39.76	26.29	17.72	12.58	10.11	10.41	13.14	18.89	28.07	44.54
	CLNV	38.73	24.63	17.78	12.60	10.32	10.62	13.02	18.77	27.67	39.44
	LR	40.26	25.04	18.34	12.96	10.68	10.96	13.20	18.57	27.87	36.95
100	FI	27.41	17.91	15.17	13.07	12.16	12.29	13.33	15.47	18.41	29.94
	EMO	43.77	29.70	26.44	23.93	22.72	22.90	24.40	26.83	30.33	46.47
	CLNV	42.93	29.50	26.37	24.06	22.90	23.09	24.51	26.76	30.00	44.77
	LR	42.84	29.63	26.81	24.65	23.57	23.68	24.98	26.91	29.67	43.06
Panel B: Mean$(OI-\widehat{OI})$											
10	FI	−5.15	−4.17	−2.96	−1.70	−0.47	0.71	1.80	3.20	5.47	8.63
	EMO	−14.82	−9.68	−5.27	−2.44	−0.51	0.92	2.74	6.26	10.88	18.72
	CLNV	−13.38	−8.06	−4.28	−2.00	−0.41	0.78	2.24	5.16	9.22	14.40
	LR	−13.41	−6.71	−3.32	−1.34	−0.20	0.45	1.32	3.68	7.63	12.56
100	FI	−6.65	−4.27	−2.98	−1.71	−0.49	0.72	1.91	3.12	4.49	7.37
	EMO	−13.68	−6.78	−4.29	−2.30	−0.57	0.97	2.65	4.51	7.17	15.12
	CLNV	−12.86	−6.18	−3.80	−2.00	−0.46	0.87	2.36	4.01	6.44	13.81
	LR	−11.87	−4.99	−2.88	−1.42	−0.31	0.61	1.68	2.98	4.99	12.10

Notes: This table corresponds to Figure 8. It shows the root-mean-square error and the bias between the estimated and true order imbalance for different regions of the true order imbalance. The estimated order imbalances are computed from $\tau = 10, 100$ equal volume bins per stock-day and from the classification results of the algorithms applied to the data timestamped to the second without delay in trades times.

Table 9: Estimating Liquidity under Data Structure 1

β	method	RMSE timestamp precision 10^{-i} of a second $i = 0$	1	2	3	Mean 0	1	2	3
Panel A: Dollar Effective Spread, true DES = 2.30¢									
no noise	FI	0.57	0.46	0.38	0.34	2.53	2.47	2.41	2.38
	CLNV	0.84	0.59	0.42	0.36	2.80	2.61	2.44	2.38
	EMO	0.69	0.48	0.39	0.36	2.61	2.43	2.27	2.23
	LR	1.20	0.79	0.54	0.44	3.00	2.72	2.51	2.44
0.001	FI	0.57	0.45	0.35	0.64	2.53	2.46	2.33	1.91
	CLNV	0.80	0.56	0.37	0.66	2.79	2.59	2.34	1.88
	EMO	0.67	0.48	0.51	1.34	2.60	2.41	2.15	1.53
	LR	1.20	0.78	0.48	0.53	3.00	2.71	2.43	2.03
0.010	FI	0.55	0.40	0.65	0.96	2.51	2.39	2.00	1.77
	CLNV	0.72	0.44	0.72	1.03	2.73	2.46	1.94	1.71
	EMO	0.64	0.58	1.46	1.99	2.53	2.26	1.58	1.25
	LR	1.19	0.69	0.56	0.74	2.99	2.63	2.13	1.95
Panel B: Dollar Price Impact, true DPI = 3.00¢									
no noise	FI	1.01	0.90	0.86	0.83	3.18	3.14	3.08	3.05
	CLNV	1.60	1.35	1.21	1.03	3.45	3.26	3.08	3.03
	EMO	1.53	1.33	1.22	1.06	3.27	3.10	2.95	2.91
	LR	1.97	1.50	1.28	1.09	3.65	3.37	3.16	3.09
0.001	FI	1.02	0.92	0.96	1.85	3.17	3.11	2.90	2.07
	CLNV	1.59	1.34	1.24	1.90	3.43	3.22	2.91	2.07
	EMO	1.55	1.36	1.34	2.24	3.25	3.06	2.76	1.82
	LR	1.97	1.49	1.30	1.98	3.65	3.35	3.00	2.14
0.010	FI	1.04	1.06	2.19	2.97	3.13	2.91	1.85	1.26
	CLNV	1.56	1.38	2.22	2.87	3.34	2.99	1.86	1.27
	EMO	1.57	1.50	2.54	3.30	3.16	2.81	1.63	0.99
	LR	1.95	1.47	2.23	2.97	3.63	3.16	1.94	1.33
Panel C: Dollar Realized Spread, true DRS = -0.64¢									
no noise	FI	0.86	0.82	0.82	0.82	-0.57	-0.60	-0.61	-0.61
	CLNV	1.45	1.29	1.19	1.02	-0.56	-0.58	-0.58	-0.59
	EMO	1.41	1.27	1.17	1.02	-0.59	-0.60	-0.62	-0.63
	LR	1.60	1.35	1.23	1.08	-0.56	-0.58	-0.58	-0.59
0.001	FI	0.88	0.85	0.94	1.60	-0.57	-0.59	-0.52	-0.11
	CLNV	1.45	1.30	1.23	1.64	-0.55	-0.56	-0.51	-0.14
	EMO	1.43	1.29	1.23	1.49	-0.58	-0.59	-0.56	-0.26
	LR	1.60	1.36	1.29	1.86	-0.56	-0.57	-0.50	-0.05
0.010	FI	0.92	1.04	1.91	2.45	-0.55	-0.46	0.20	0.56
	CLNV	1.48	1.40	1.90	2.30	-0.53	-0.46	0.13	0.47
	EMO	1.46	1.38	1.70	2.01	-0.56	-0.50	-0.02	0.27
	LR	1.60	1.45	2.11	2.67	-0.55	-0.45	0.24	0.66

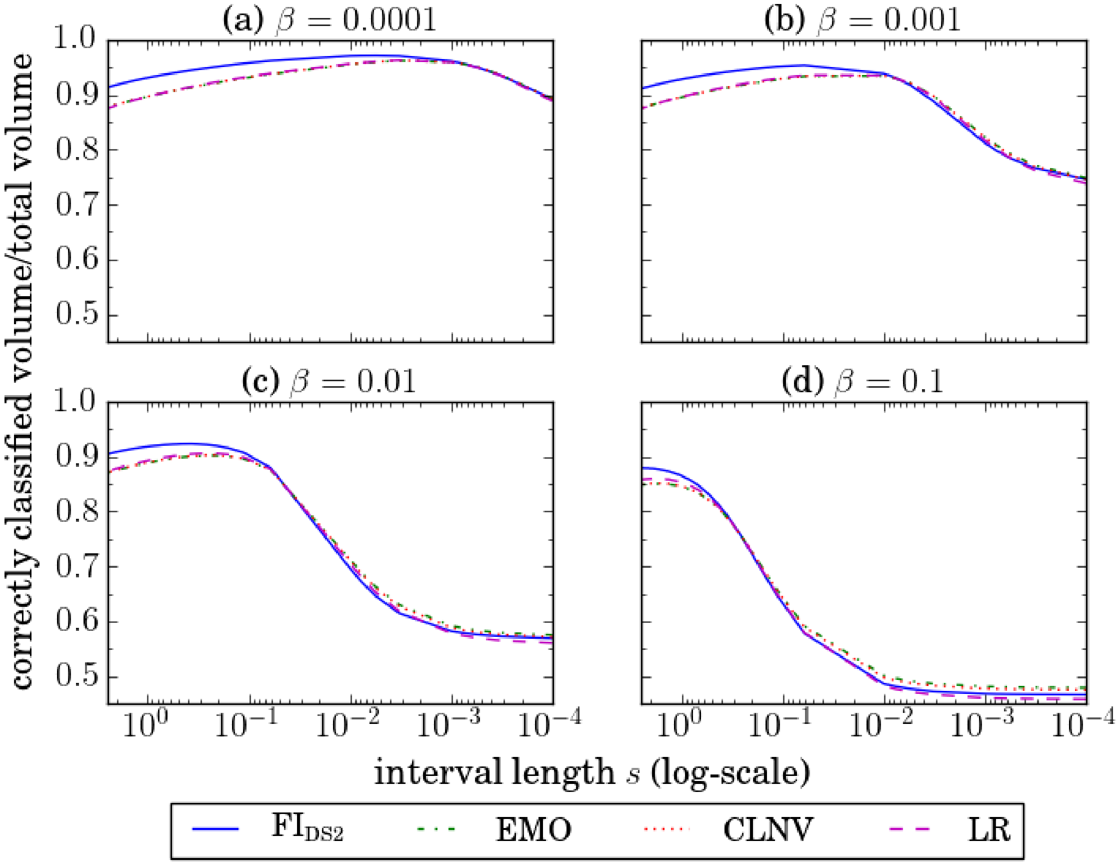
Notes: This tabel corresponds to Figure 9. It shows the sample averages and the root-mean-square error between the stock-day estimates and the true values of the dollar effective spread (DES), the dollar price impact (DPI) and the dollar realized spread (DRS) in cents. The estimates are constructed from the classification results of the different algorithms applied to the data with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds. Note that the difference between the average DES and DPI does not exactly equal the average DRS due to the numerical impact of the order of summation.

Table 10: Classification Accuracy of FI_{DS2}

		timestamp precision: 10^{-i} of a second for $i =$					
		0	1	2	3	4	9
Panel A: overall correctly classified volume (in %)							
total		93.36	96.16	97.71	98.30	98.25	98.21
mean		93.42	95.93	97.18	97.65	97.54	97.45
std		2.64	2.02	1.96	2.06	2.05	2.09
Panel B: % correctly classified volume in each classification category							
visible	cl. step						
YES	0	–	–	–	–	–	–
	2	99.50	99.76	99.94	99.99	100.00	100.00
	3	90.80	94.52	96.78	97.34	67.51	–
	4a	94.79	98.34	99.72	99.98	100.00	100.00
	4b	98.73	99.57	99.90	99.99	100.00	100.00
	4	49.78	55.15	59.45	66.66	75.93	76.00
	5	53.39	52.45	51.94	50.54	38.99	36.09
NO	0	–	–	–	–	–	–
	2	71.68	79.77	88.60	95.57	99.77	99.96
	3	87.78	92.83	95.55	96.81	97.13	–
	4a	81.82	88.50	91.88	95.52	99.88	100.00
	4b	48.50	63.19	84.89	95.75	99.85	100.00
	4	90.11	92.52	93.41	93.16	92.15	91.53
	5	64.87	64.73	65.87	67.56	68.99	69.14
Panel C: % classified volume in each classification category							
visible	cl. step						
YES	0	0.00	0.00	0.00	0.00	0.00	0.00
	2	46.52	59.06	72.95	85.69	89.49	89.49
	3	43.48	30.85	16.84	3.94	0.00	0.00
	4a	0.22	0.27	0.32	0.39	0.46	0.46
	4b	0.12	0.19	0.29	0.40	0.46	0.46
	4	0.01	0.01	0.01	0.00	0.00	0.00
	5	0.07	0.04	0.03	0.01	0.00	0.00
NO	0	0.00	0.00	0.00	0.00	0.00	0.00
	2	2.89	3.04	3.06	2.67	1.80	1.67
	3	1.47	0.82	0.33	0.06	0.00	0.00
	4a	0.53	0.29	0.07	0.02	0.00	0.00
	4b	0.46	0.25	0.15	0.13	0.12	0.12
	4	1.82	2.35	2.69	2.79	2.74	2.74
	5	2.40	2.82	3.28	3.91	4.92	5.05

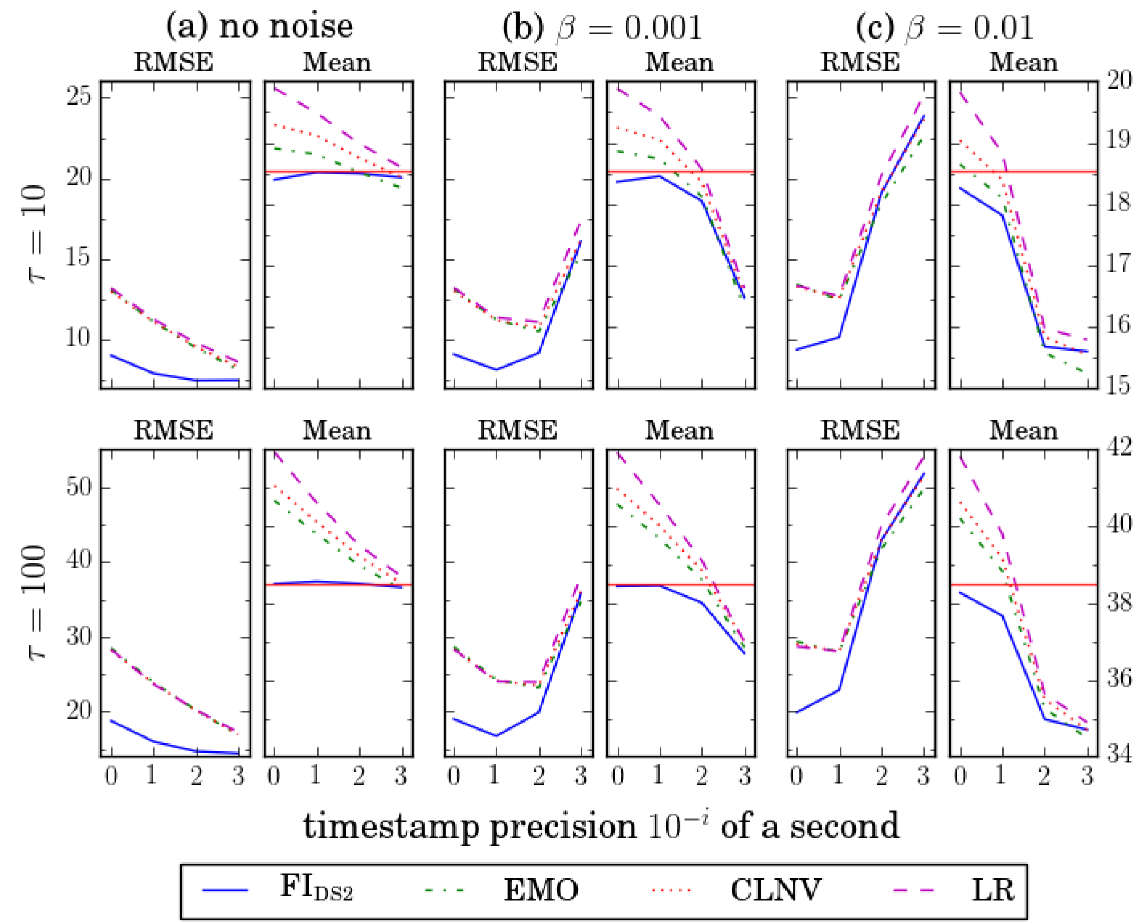
Notes: This table shows the percentage of correctly classified trading volume for the FI algorithm adjusted to the Data Structure 2. "cl. step" refers to the accuracy at the corresponding step of the classification procedure.

Figure 11: Classification Accuracy under Delayed Trade Times of Data Structure 2



Notes: This figure shows the fraction of correctly classified trading volume (y-axis) for the data with delayed trade times under Data Structure 2. The trade time equals the actual trade time plus ε, with $\varepsilon \sim \text{Exp}(1/\beta)$ and $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The classification algorithms FI_{DS2}, EMO, CLNV and LR are apply to the data with reduced timestamp precision (s) ranging from 10^{-4} of a second to 2.5 seconds presented on $\log_{10}$-scale (x-axis).

Figure 12: Estimating Order Imbalances under Data Structure 2



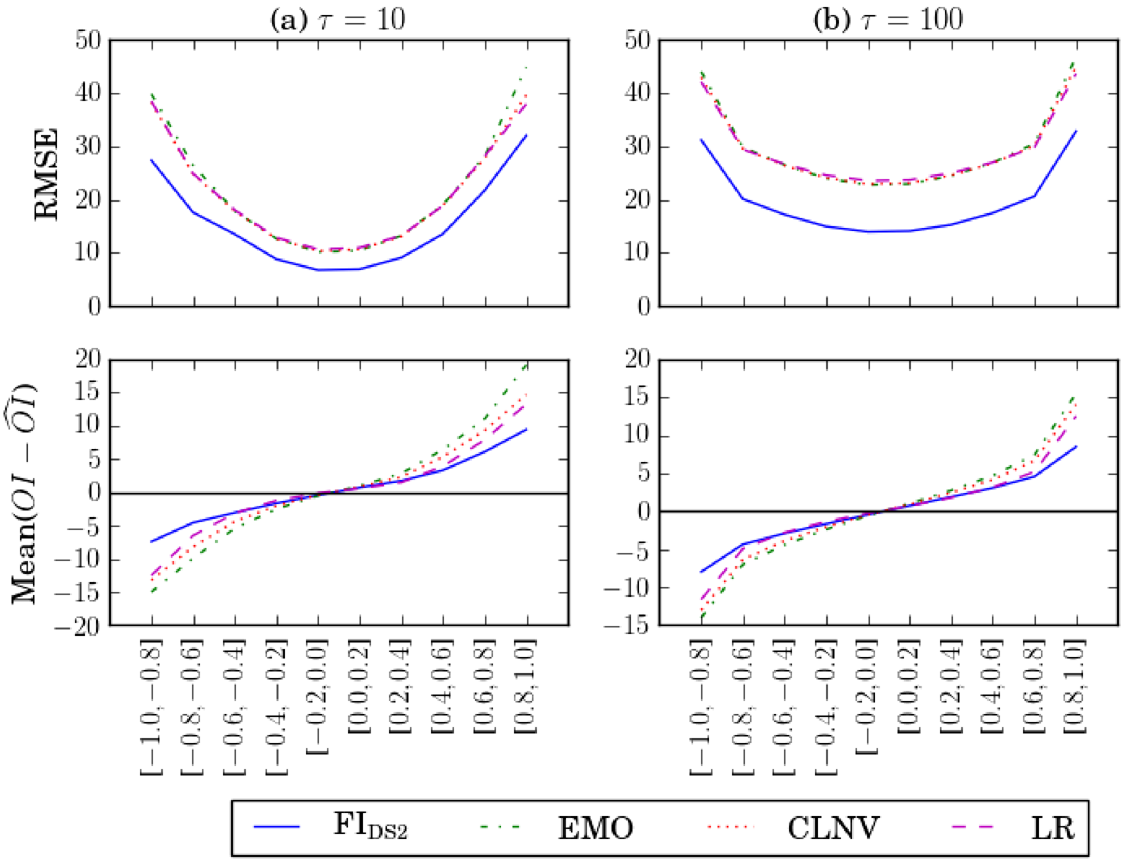
Notes: This figure shows the root-mean-square error between estimates of the order imbalance and the true order imbalance, as well as the sample averages of the absolute order imbalances displayed in percentages. The true values are computed from the true trade initiator labels and the estimates from the classification results of the different algorithms under Data Structure 2. For the computation of the order imbalance each stock-day is split into equally sized volume bins. The number of bins is chosen to be $\tau = 10, 100$. The algorithms are applied to the data with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds.

Table 11: RMSE and Mean Order Imbalance for Data Structure 2

		RMSE				Mean($\|OI\|$)			
		timestamp precision 10^{-i} of a second							
β	method	$i=0$	1	2	3	0	1	2	3
Panel A: $\tau = 10$, true average absolute order imbalance = 18.56									
no noise	FI	9.04	7.92	7.49	7.51	18.40	18.52	18.50	18.44
	CLNV	12.99	11.11	9.58	8.32	19.29	19.12	18.76	18.44
	EMO	13.09	11.12	9.51	8.14	18.91	18.81	18.52	18.26
	LR	13.21	11.29	9.82	8.63	19.89	19.48	18.99	18.60
0.001	FI	9.11	8.14	9.18	16.11	18.36	18.45	18.05	16.48
	CLNV	13.03	11.23	10.74	16.23	19.25	19.05	18.36	16.54
	EMO	13.15	11.24	10.52	15.38	18.86	18.74	18.12	16.33
	LR	13.23	11.39	11.11	17.41	19.89	19.44	18.57	16.63
0.010	FI	9.39	10.16	19.08	23.85	18.26	17.82	15.68	15.60
	CLNV	13.30	12.56	19.16	23.64	19.04	18.40	15.84	15.55
	EMO	13.44	12.48	18.42	22.63	18.65	18.10	15.58	15.24
	LR	13.36	12.71	20.17	25.13	19.82	18.83	15.98	15.79
Panel B: $\tau = 100$, true average absolute order imbalance = 38.52									
no noise	FI	18.79	16.01	14.72	14.44	38.51	38.57	38.51	38.41
	CLNV	28.23	23.77	20.17	16.98	41.07	40.12	39.23	38.52
	EMO	28.52	23.93	20.22	16.90	40.68	39.80	38.99	38.36
	LR	28.30	23.70	20.26	17.25	41.94	40.62	39.51	38.69
0.001	FI	19.03	16.75	19.90	35.83	38.44	38.46	38.01	36.69
	CLNV	28.36	24.14	23.49	36.08	40.98	40.02	38.85	36.99
	EMO	28.67	24.30	23.23	34.70	40.58	39.69	38.61	36.85
	LR	28.35	24.05	23.97	38.13	41.93	40.56	39.11	36.98
0.010	FI	19.91	22.89	42.89	51.80	38.28	37.67	34.98	34.70
	CLNV	29.06	28.04	42.98	51.47	40.64	39.18	35.50	34.70
	EMO	29.41	27.97	41.74	49.77	40.21	38.84	35.26	34.48
	LR	28.72	28.07	44.81	54.07	41.83	39.81	35.64	34.89

Notes: This table corresponds to Figure 12. It shows the root-mean-square error between estimates of the order imbalance and the true order imbalance, as well as the sample averages of the absolute order imbalances displayed in percentages. The true values are computed from the true trade initiator labels and the estimates from the classification results of the different algorithms under Data Structure 2. For the computation of the order imbalance each stock-day is split into equally sized volume bins. The number of bins is chosen to be $\tau = 10, 100$. The algorithms are applied to the data with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds.

Figure 13: Estimation Performance for Different Regions of the Order Imbalance under Data Structure 2



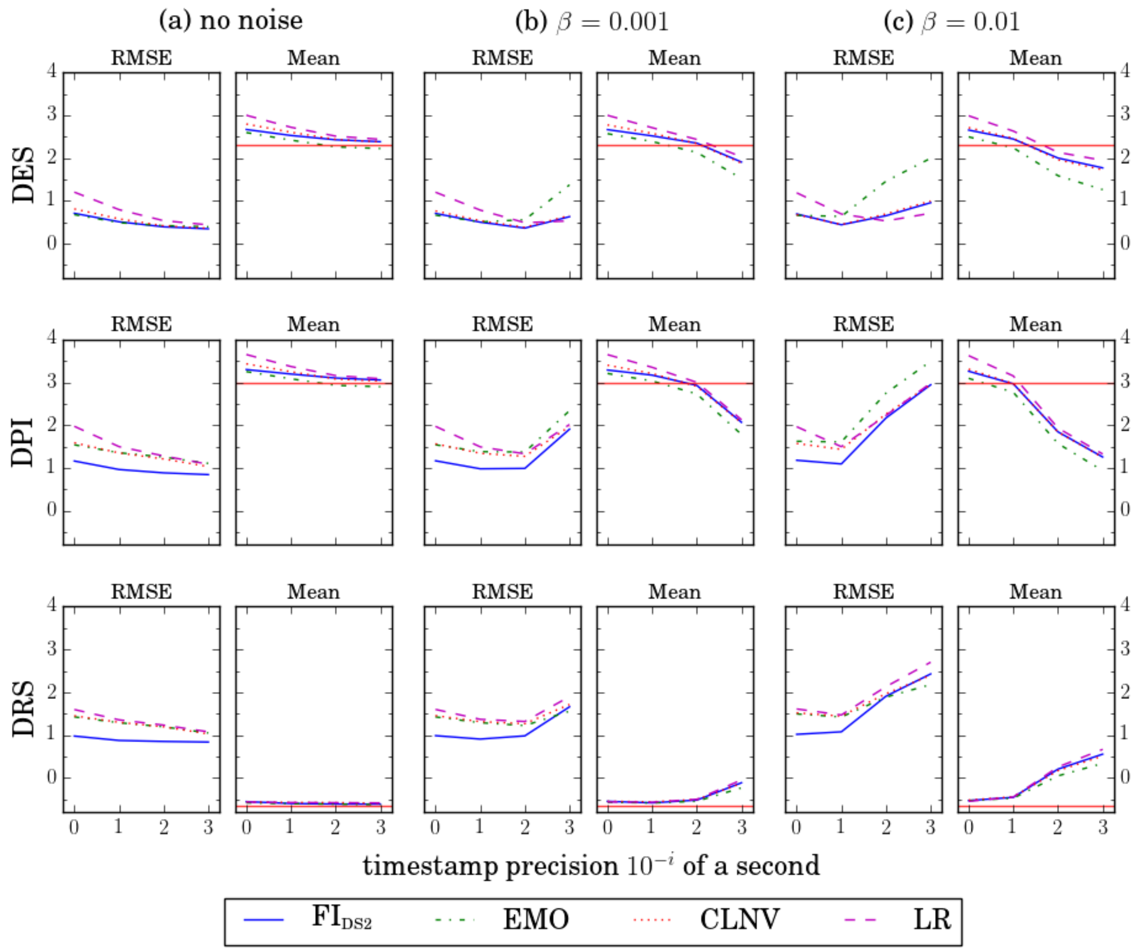
Notes: This figure shows the root-mean-square error and the bias between the estimated and true order imbalance for different regions (x-axis) of the level of the true order imbalance under Data Structure 2. The estimated order imbalances are computed from $\tau = 10, 100$ equal volume bins per stock-day and from the classification results of the algorithms applied to the data timestamped to the second without delay in trades times.

Table 12: RMSE and Bias for Different Intervals of the Order Imbalance under Data Structure 2

τ	method	$[-1,-0.8]$	$[-0.8,-0.6]$	$[-0.6,-0.4]$	$[-0.4,-0.2]$	$[-0.2,0]$	$[0,0.2]$	$[0.2,0.4]$	$[0.4,0.6]$	$[0.6,0.8]$	$[0.8,1]$
Panel A: RMSE											
10	FI	27.41	17.62	13.50	8.80	6.75	6.92	9.09	13.56	21.73	32.06
	EMO	39.89	26.51	17.85	12.68	10.16	10.45	13.22	19.04	28.14	44.94
	CLNV	38.31	24.83	17.84	12.65	10.35	10.66	13.09	18.91	27.75	39.69
	LR	38.53	24.92	18.07	12.88	10.69	10.98	13.24	18.85	28.18	37.99
100	FI	31.21	20.13	17.19	14.96	13.94	14.09	15.28	17.49	20.64	32.87
	EMO	44.08	29.88	26.53	23.99	22.78	22.95	24.47	26.96	30.52	46.81
	CLNV	42.99	29.61	26.42	24.09	22.95	23.13	24.57	26.86	30.15	44.99
	LR	42.15	29.44	26.69	24.60	23.56	23.70	25.03	27.01	29.85	43.69
Panel B: Mean$(OI-\widehat{OI})$											
10	FI	−7.43	−4.58	−3.07	−1.66	−0.44	0.71	1.72	3.31	6.09	9.42
	EMO	−15.01	−9.93	−5.44	−2.53	−0.52	0.96	2.82	6.43	11.02	19.16
	CLNV	−13.24	−8.22	−4.40	−2.05	−0.41	0.82	2.31	5.33	9.29	14.68
	LR	−12.52	−6.53	−3.14	−1.24	−0.12	0.56	1.47	3.92	7.88	13.23
100	FI	−7.97	−4.35	−2.91	−1.67	−0.45	0.72	1.89	3.07	4.57	8.49
	EMO	−14.01	−7.01	−4.43	−2.40	−0.59	1.01	2.75	4.68	7.43	15.47
	CLNV	−13.03	−6.34	−3.89	−2.06	−0.48	0.91	2.44	4.15	6.65	14.05
	LR	−11.61	−4.92	−2.81	−1.36	−0.24	0.70	1.78	3.14	5.21	12.51

Notes: This table corresponds to Figure 13. It shows the root-mean-square error and the bias between the estimated and true order imbalance for different regions of the true order imbalance under Data Structure 2. The estimated order imbalances are computed from $\tau = 10, 100$ equal volume bins per stock-day and from the classification results of the algorithms applied to the data timestamped to the second without delay in trades times.

Figure 14: Estimating Liquidity under Data Structure 2



Notes: This figure shows the sample averages and the root-mean-square error between the stock-day estimates and the true values of the dollar effective spread (DES), the dollar price impact (DPI) and the dollar realized spread (DRS) for the FI$_{\mathrm{DS2}}$ algorithm and the traditional algorithms under Data Structure 2. The algorithms are applied to the data with and without delayed trade times, where the delay is given by $\varepsilon \sim \mathrm{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds.

Table 13: Estimating Liquidity under Data Structure 2

		RMSE				Mean			
		timestamp precision 10^{-i} of a second							
β	method	$i = 0$	1	2	3	0	1	2	3
Panel A: Dollar Effective Spread, true DES = 2.30¢									
no noise	FI	0.71	0.51	0.39	0.35	2.67	2.53	2.43	2.38
	CLNV	0.81	0.58	0.42	0.36	2.79	2.60	2.43	2.38
	EMO	0.68	0.49	0.42	0.39	2.59	2.42	2.26	2.22
	LR	1.20	0.79	0.54	0.44	3.00	2.72	2.51	2.44
0.001	FI	0.71	0.50	0.36	0.63	2.66	2.52	2.35	1.91
	CLNV	0.76	0.54	0.38	0.66	2.77	2.58	2.34	1.88
	EMO	0.67	0.51	0.56	1.37	2.57	2.38	2.13	1.52
	LR	1.20	0.78	0.49	0.53	3.00	2.71	2.43	2.03
0.010	FI	0.69	0.44	0.65	0.96	2.65	2.44	2.00	1.77
	CLNV	0.68	0.45	0.69	1.00	2.71	2.46	1.96	1.72
	EMO	0.66	0.64	1.46	2.00	2.49	2.24	1.59	1.26
	LR	1.19	0.69	0.53	0.71	2.99	2.63	2.13	1.95
Panel B: Dollar Price Impact, true DPI = 3.00¢									
no noise	FI	1.16	0.96	0.89	0.84	3.30	3.19	3.10	3.06
	CLNV	1.59	1.35	1.21	1.03	3.43	3.25	3.08	3.03
	EMO	1.54	1.36	1.26	1.11	3.25	3.08	2.93	2.90
	LR	1.97	1.50	1.28	1.09	3.65	3.37	3.16	3.09
0.001	FI	1.17	0.98	0.99	1.91	3.29	3.17	2.93	2.07
	CLNV	1.56	1.34	1.27	1.97	3.40	3.21	2.91	2.05
	EMO	1.54	1.38	1.37	2.34	3.21	3.03	2.73	1.78
	LR	1.97	1.49	1.33	2.02	3.65	3.35	3.00	2.13
0.010	FI	1.18	1.09	2.17	2.94	3.26	2.97	1.84	1.26
	CLNV	1.57	1.43	2.26	2.95	3.31	2.97	1.84	1.25
	EMO	1.62	1.61	2.75	3.49	3.09	2.77	1.57	0.94
	LR	1.96	1.50	2.23	2.98	3.62	3.15	1.93	1.33
Panel C: Dollar Realized Spread, true DRS = -0.64¢									
no noise	FI	0.98	0.88	0.85	0.84	-0.55	-0.60	-0.61	-0.61
	CLNV	1.45	1.30	1.19	1.03	-0.56	-0.57	-0.58	-0.59
	EMO	1.43	1.29	1.19	1.05	-0.58	-0.60	-0.61	-0.62
	LR	1.60	1.36	1.23	1.08	-0.56	-0.57	-0.58	-0.58
0.001	FI	0.99	0.91	0.98	1.66	-0.55	-0.58	-0.52	-0.11
	CLNV	1.46	1.32	1.27	1.72	-0.55	-0.56	-0.50	-0.13
	EMO	1.43	1.29	1.22	1.56	-0.57	-0.58	-0.55	-0.23
	LR	1.60	1.37	1.32	1.90	-0.56	-0.57	-0.50	-0.05
0.010	FI	1.02	1.08	1.91	2.43	-0.53	-0.45	0.21	0.56
	CLNV	1.52	1.44	1.97	2.40	-0.52	-0.44	0.17	0.51
	EMO	1.49	1.42	1.89	2.18	-0.53	-0.47	0.05	0.34
	LR	1.61	1.47	2.13	2.70	-0.55	-0.45	0.26	0.67

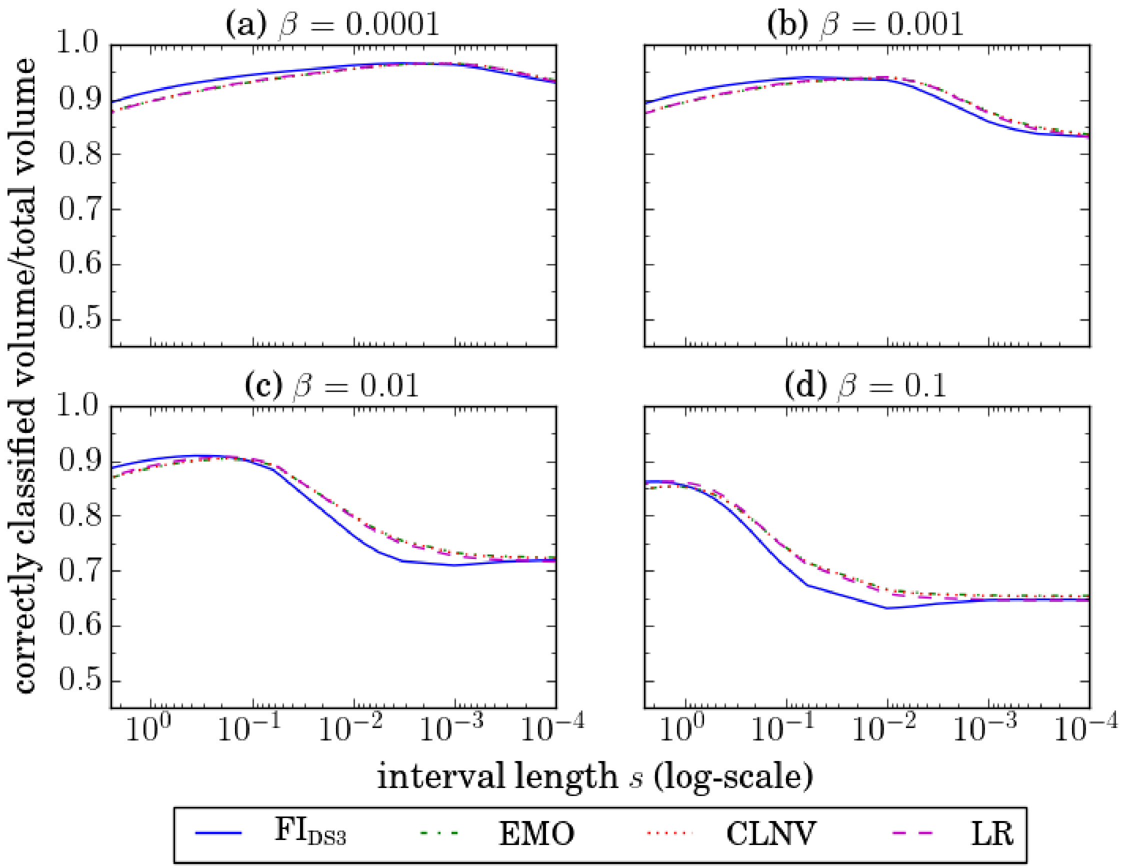
Notes: This tabel corresponds to Figure 14. It shows the sample averages and the root-mean-square error between the stock-day estimates and the true values of the dollar effective spread (DES), the dollar price impact (DPI) and the dollar realized spread (DRS) in cents. The estimates are constructed from the classification results of the different algorithms applied to the data under Data Structure 2 with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds. Note that the difference between the average DES and DPI does not exactly equal the average DRS due to the numerical impact of the order of summation.

Table 14: Classification Accuracy of FI_{DS3}

		timestamp precision: 10^{-i} of a second for $i =$					
		0	1	2	3	4	9
Panel A: overall correctly classified volume (in %)							
total		91.73	94.84	96.82	97.92	98.22	98.21
mean		92.33	94.86	96.43	97.40	97.52	97.45
std		2.97	2.24	2.00	2.05	2.05	2.09
Panel B: % correctly classified volume in each classification category							
visible	cl. step						
YES	0	–	–	–	–	–	–
	2	99.70	99.81	99.93	99.99	100.00	100.00
	3	89.36	94.14	97.24	98.88	98.15	–
	4	98.30	99.15	99.60	99.82	99.90	99.90
	5	51.77	50.85	48.37	42.03	41.39	38.94
NO	0	–	–	–	–	–	–
	2	67.64	66.03	71.96	83.44	97.17	99.81
	3	89.87	93.29	93.49	84.18	89.45	–
	4	83.29	89.46	92.33	93.00	92.50	91.95
	5	64.53	64.39	65.42	67.27	68.95	69.14
Panel C: % classified volume in each classification category							
visible	cl. step						
YES	0	0.00	0.00	0.00	0.00	0.00	0.00
	2	35.66	44.61	59.19	78.89	89.47	89.49
	3	54.56	45.54	30.84	10.95	0.05	0.00
	4	0.14	0.22	0.35	0.57	0.90	0.93
	5	0.06	0.06	0.04	0.01	0.00	0.00
NO	0	0.00	0.00	0.00	0.00	0.00	0.00
	2	2.78	3.13	3.39	3.04	1.87	1.68
	3	2.43	1.58	0.69	0.12	0.00	0.00
	4	1.95	2.11	2.34	2.65	2.84	2.85
	5	2.41	2.76	3.16	3.77	4.87	5.05

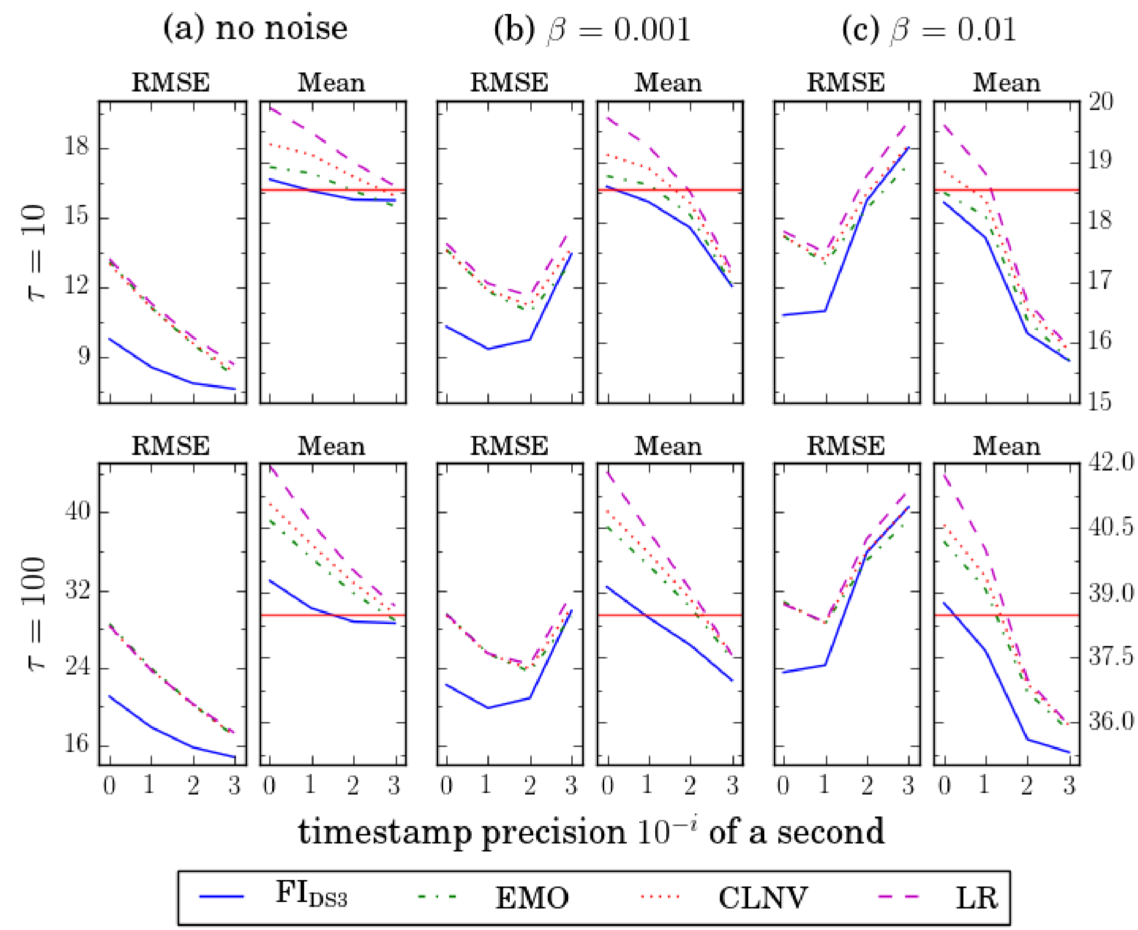
Notes: This table shows the percentage of correctly classified trading volume for the FI algorithm adjusted to the Data Structure 3. “cl. step” refers to the accuracy at the corresponding step of the classification procedure.

Figure 15: Classification Accuracy under Random Trade and Quote Times of Data Structure 3



Notes: This figure shows the fraction of correctly classified trading volume (y-axis) for the data with noisy quote and trade times (Data Structure 3). The recorded time of trades and quotes equals the actual time plus ε, with $\varepsilon \sim \text{Exp}(1/\beta)$ and $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The classification algorithms FI_{DS3}, EMO, CLNV and LR are apply to the data with reduced timestamp precision (s) ranging from 10^{-4} of a second to 2.5 seconds presented on $\log_{10}$-scale (x-axis).

Figure 16: Estimating Order Imbalances under Data Structure 3



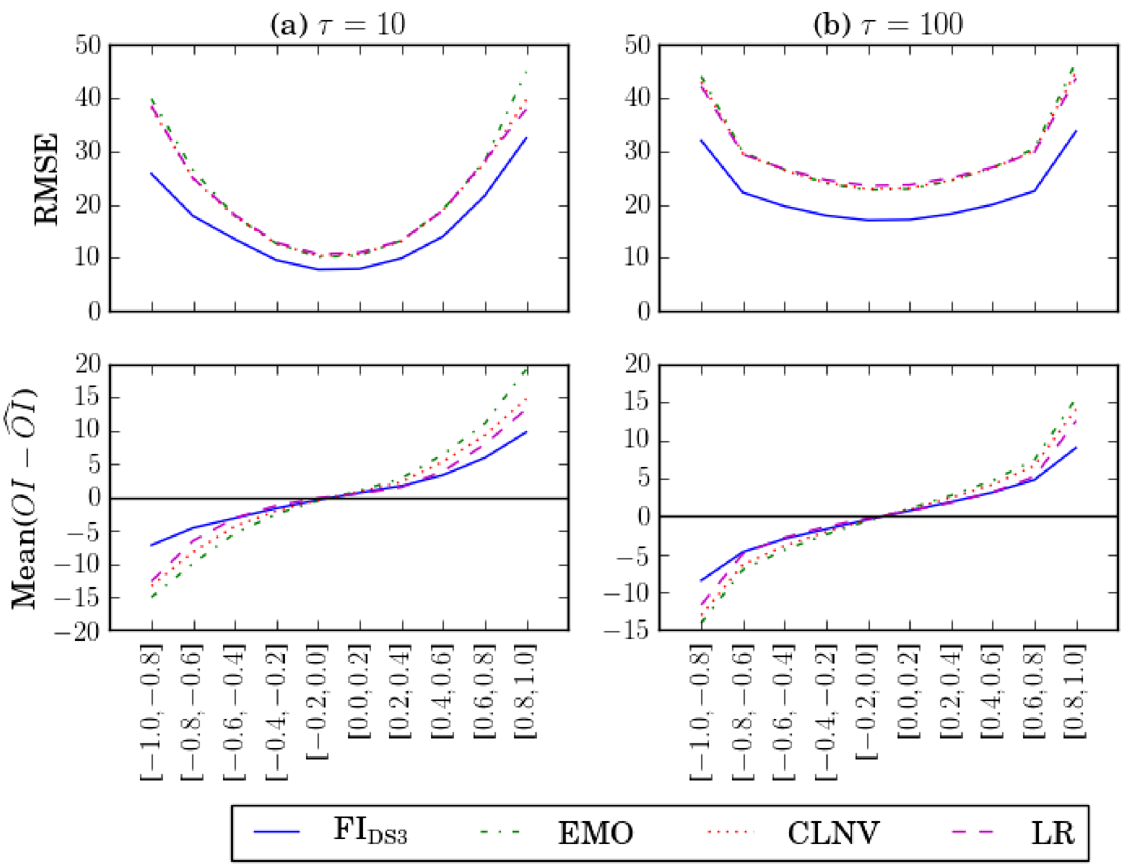
Notes: This figure shows the sample averages and the root-mean-square error between estimates of the order imbalance and the true order imbalance displayed in percent for the data with random trade and quote order. For the computation of the order imbalance each stock-day is split into equally sized volume bins. The number of bins is chosen to be $\tau = 10, 100$. The algorithms are applied to the data with and without noise. The noise is applied to both trade and quote times, where the noise is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$. The timestamp precision ranges from seconds to milliseconds.

Table 15: RMSE and Mean Order Imbalance for Data Structure 3

		RMSE				Mean($\|OI\|$)			
		timestamp precision 10^{-i} of a second							
β	method	$i=0$	1	2	3	0	1	2	3
Panel A: $\tau = 10$, true average absolute order imbalance $= 18.56$									
no noise	FI	9.76	8.56	7.85	7.61	18.71	18.52	18.37	18.36
	CLNV	13.00	11.11	9.59	8.33	19.29	19.11	18.75	18.43
	EMO	13.10	11.14	9.54	8.17	18.91	18.81	18.52	18.26
	LR	13.22	11.30	9.84	8.66	19.89	19.48	18.99	18.60
0.001	FI	10.30	9.33	9.73	13.43	18.59	18.33	17.90	16.93
	CLNV	13.58	11.86	11.21	13.69	19.12	18.89	18.31	17.11
	EMO	13.62	11.81	10.92	13.03	18.76	18.62	18.12	16.96
	LR	13.87	12.15	11.62	14.55	19.73	19.25	18.50	17.19
0.010	FI	10.79	10.97	15.73	18.00	18.33	17.74	16.15	15.70
	CLNV	14.19	13.16	16.05	18.12	18.84	18.37	16.55	15.87
	EMO	14.19	13.02	15.39	17.29	18.49	18.09	16.38	15.71
	LR	14.40	13.47	16.80	19.14	19.61	18.81	16.67	15.93
Panel B: $\tau = 100$, true average absolute order imbalance $= 38.52$									
no noise	FI	21.07	17.91	15.84	14.82	39.28	38.64	38.33	38.30
	CLNV	28.26	23.80	20.22	17.04	41.07	40.12	39.23	38.52
	EMO	28.54	23.97	20.27	16.97	40.68	39.80	39.00	38.36
	LR	28.34	23.74	20.32	17.33	41.94	40.62	39.52	38.70
0.001	FI	22.26	19.86	20.89	29.86	39.13	38.42	37.79	36.97
	CLNV	29.39	25.41	23.90	30.20	40.90	39.92	38.86	37.56
	EMO	29.56	25.43	23.61	29.24	40.53	39.63	38.67	37.46
	LR	29.52	25.48	24.35	31.59	41.81	40.42	39.07	37.56
0.010	FI	23.54	24.27	35.88	40.52	38.76	37.65	35.60	35.30
	CLNV	30.64	28.61	36.00	40.52	40.58	39.37	36.88	35.93
	EMO	30.80	28.53	35.04	39.24	40.19	39.08	36.70	35.80
	LR	30.54	28.70	37.26	42.28	41.74	40.00	36.99	35.93

Notes: This table corresponds to Figure 16. It shows the root-mean-square error between estimates of the order imbalance and the true order imbalance, as well as the sample averages of the absolute order imbalances displayed in percentages. The true values are computed from the true trade initiator labels and the estimates from the classification results of the different algorithms under Data Structure 3. For the computation of the order imbalance each stock-day is split into equally sized volume bins. The number of bins is chosen to be $\tau = 10, 100$. The algorithms are applied to the data with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds.

Figure 17: Estimation Performance for Different Regions of the Order Imbalance under Data Structure 3



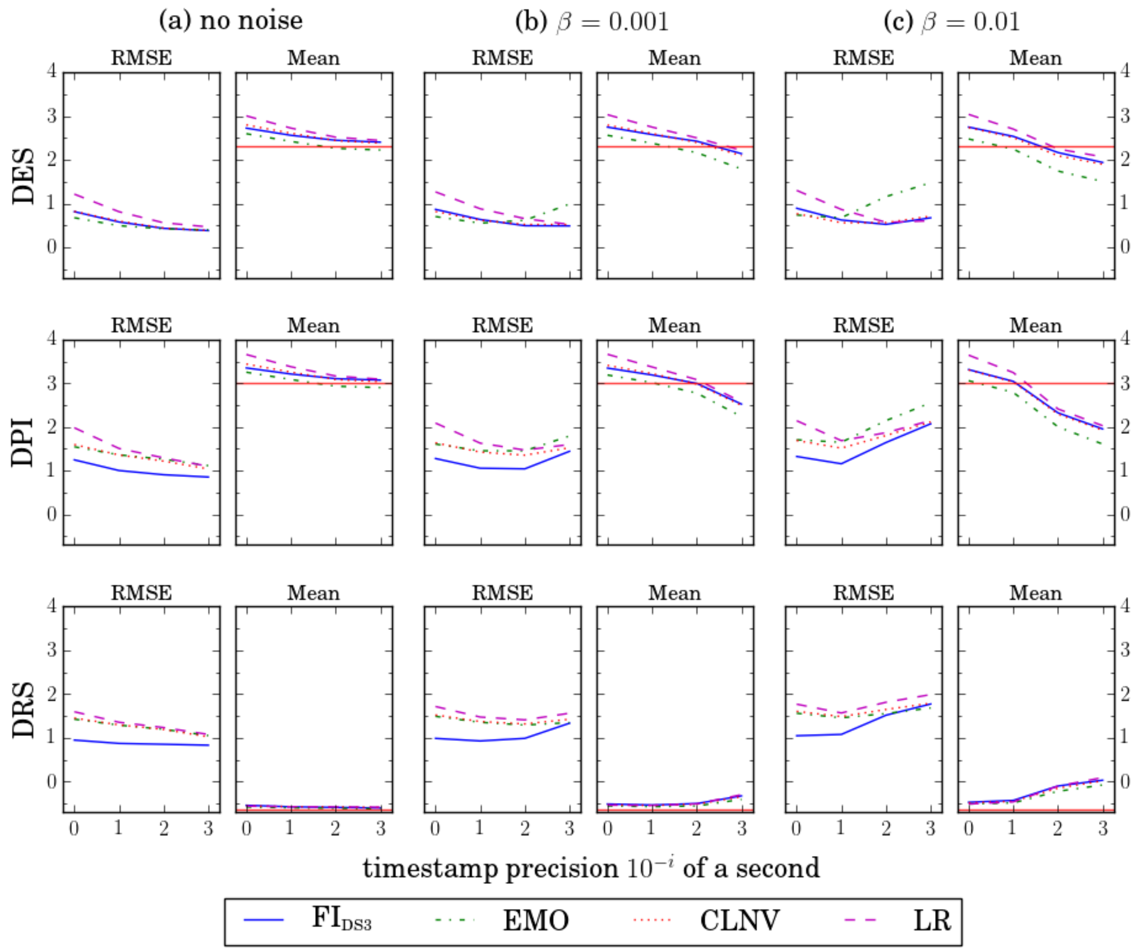
Notes: This figure shows the root-mean-square error and the bias between the estimated and true order imbalance for different regions (x-axis) of the level of the true order imbalance under Data Structure 3. The estimated order imbalances are computed from $\tau = 10, 100$ equal volume bins per stock-day and from the classification results of the algorithms applied to the data timestamped to the second without delay in trades times.

Table 16: RMSE and Bias for Different Intervals of the Order Imbalance under Data Structure 3

τ	method	$[-1,-0.8]$	$[-0.8,-0.6]$	$[-0.6,-0.4]$	$[-0.4,-0.2]$	$[-0.2,0]$	$[0,0.2]$	$[0.2,0.4]$	$[0.4,0.6]$	$[0.6,0.8]$	$[0.8,1]$
Panel A: RMSE											
10	FI	25.82	17.88	13.51	9.57	7.80	7.92	9.91	13.99	21.66	32.54
	EMO	39.91	26.55	17.84	12.69	10.17	10.46	13.23	19.04	28.20	45.03
	CLNV	38.28	24.92	17.79	12.64	10.36	10.66	13.09	18.95	27.83	39.74
	LR	38.62	24.99	18.08	12.89	10.70	10.99	13.24	18.87	28.26	37.95
100	FI	32.04	22.28	19.70	17.95	17.06	17.17	18.26	20.02	22.56	33.82
	EMO	44.13	29.91	26.55	24.01	22.79	22.97	24.49	26.98	30.53	46.87
	CLNV	43.04	29.65	26.44	24.11	22.96	23.14	24.58	26.89	30.17	45.06
	LR	42.28	29.47	26.70	24.62	23.57	23.71	25.04	27.04	29.87	43.78
Panel B: Mean$(OI - \widehat{OI})$											
10	FI	−7.20	−4.62	−3.14	−1.66	−0.40	0.68	1.68	3.33	5.93	9.78
	EMO	−15.05	−9.96	−5.44	−2.53	−0.52	0.96	2.83	6.43	11.04	19.26
	CLNV	−13.36	−8.24	−4.38	−2.05	−0.41	0.82	2.30	5.35	9.31	14.78
	LR	−12.65	−6.57	−3.14	−1.24	−0.12	0.56	1.47	3.91	7.94	13.24
100	FI	−8.45	−4.70	−2.97	−1.69	−0.43	0.72	1.89	3.14	4.80	9.02
	EMO	−14.05	−7.02	−4.44	−2.40	−0.59	1.01	2.76	4.69	7.44	15.50
	CLNV	−13.06	−6.36	−3.89	−2.07	−0.48	0.91	2.44	4.16	6.66	14.09
	LR	−11.67	−4.94	−2.81	−1.36	−0.25	0.70	1.78	3.15	5.22	12.55

Notes: This table corresponds to Figure 17. It shows the root-mean-square error and the bias between the estimated and true order imbalance for different regions of the true order imbalance under Data Structure 3. The estimated order imbalances are computed from $\tau = 10, 100$ equal volume bins per stock-day and from the classification results of the algorithms applied to the data timestamped to the second without delay in trades times.

Figure 18: Estimating Liquidity under Data Structure 3



Notes: This figure shows the sample averages and the root-mean-square error between the stock-day estimates and the true values of the dollar effective spread (DES), the dollar price impact (DPI) and the dollar realized spread (DRS) for the FI$_{DS2}$ algorithm and the traditional algorithms under the data structure with random trade and quote order (Data Structure 3). The algorithms are applied to the data with and without noise. The noise is applied to both trade and quote times, where the noise is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$. The timestamp precision ranges from seconds to milliseconds.

Table 17: Estimating Liquidity under Data Structure 3

		RMSE				Mean			
		timestamp precision 10^{-i} of a second							
β	method	$i = 0$	1	2	3	0	1	2	3
Panel A: Dollar Effective Spread, true DES = 2.30¢									
no noise	FI	0.81	0.58	0.43	0.38	2.72	2.56	2.45	2.40
	CLNV	0.82	0.60	0.44	0.38	2.80	2.61	2.44	2.38
	EMO	0.68	0.49	0.42	0.40	2.60	2.42	2.26	2.22
	LR	1.22	0.81	0.56	0.47	3.01	2.73	2.52	2.44
0.001	FI	0.87	0.64	0.49	0.49	2.75	2.58	2.43	2.14
	CLNV	0.82	0.62	0.52	0.52	2.79	2.60	2.40	2.10
	EMO	0.70	0.55	0.62	0.99	2.56	2.38	2.16	1.79
	LR	1.27	0.88	0.66	0.52	3.03	2.75	2.50	2.22
0.010	FI	0.89	0.62	0.52	0.67	2.75	2.54	2.17	1.94
	CLNV	0.77	0.56	0.56	0.71	2.74	2.51	2.09	1.90
	EMO	0.74	0.68	1.16	1.48	2.48	2.25	1.74	1.51
	LR	1.30	0.87	0.57	0.61	3.04	2.70	2.25	2.07
Panel B: Dollar Price Impact, true DPI = 3.00¢									
no noise	FI	1.25	1.00	0.91	0.86	3.35	3.21	3.11	3.07
	CLNV	1.60	1.36	1.22	1.04	3.44	3.26	3.09	3.04
	EMO	1.55	1.36	1.26	1.12	3.25	3.09	2.93	2.90
	LR	1.98	1.51	1.29	1.10	3.66	3.38	3.17	3.09
0.001	FI	1.28	1.06	1.04	1.44	3.35	3.19	3.00	2.52
	CLNV	1.64	1.42	1.36	1.52	3.41	3.22	2.98	2.50
	EMO	1.61	1.45	1.46	1.79	3.19	3.01	2.77	2.24
	LR	2.09	1.63	1.47	1.60	3.66	3.37	3.08	2.57
0.010	FI	1.33	1.16	1.65	2.07	3.31	3.04	2.32	1.96
	CLNV	1.69	1.52	1.80	2.11	3.30	3.03	2.29	1.93
	EMO	1.71	1.65	2.15	2.55	3.06	2.79	2.00	1.61
	LR	2.14	1.69	1.87	2.15	3.64	3.24	2.41	2.03
Panel C: Dollar Realized Spread, true DRS = -0.64¢									
no noise	FI	0.95	0.87	0.85	0.83	-0.54	-0.58	-0.59	-0.61
	CLNV	1.45	1.30	1.19	1.02	-0.56	-0.57	-0.58	-0.59
	EMO	1.43	1.29	1.20	1.05	-0.58	-0.60	-0.61	-0.62
	LR	1.60	1.36	1.24	1.08	-0.56	-0.57	-0.58	-0.58
0.001	FI	0.99	0.93	0.99	1.34	-0.51	-0.54	-0.50	-0.33
	CLNV	1.52	1.37	1.32	1.43	-0.53	-0.55	-0.52	-0.35
	EMO	1.49	1.36	1.29	1.35	-0.56	-0.57	-0.56	-0.41
	LR	1.72	1.48	1.41	1.56	-0.54	-0.54	-0.51	-0.30
0.010	FI	1.05	1.08	1.52	1.77	-0.47	-0.43	-0.10	0.03
	CLNV	1.61	1.49	1.65	1.79	-0.49	-0.45	-0.15	0.02
	EMO	1.56	1.46	1.56	1.68	-0.51	-0.49	-0.22	-0.08
	LR	1.77	1.57	1.81	1.99	-0.51	-0.46	-0.10	0.09

Notes: This table corresponds to Figure 18. It shows the sample averages and the root-mean-square error between the stock-day estimates and the true values of the dollar effective spread (DES), the dollar price impact (DPI) and the dollar realized spread (DRS) in cents. The estimates are constructed from the classification results of the different algorithms applied to the data under Data Structure 3 with and without delayed trade times, where the delay is given by $\varepsilon \sim \text{Exp}(1/\beta)$ with $\beta = 10^{-3}, 10^{-2}$, and with varying timestamp precision ranging from seconds to milliseconds. Note that the difference between the average DES and DPI does not exactly equal the average DRS due to the numerical impact of the order of summation.

D Determinants of Misclassification

In an analysis that necessitates the estimation of the trade initiator, a variation of the classification accuracy in tandem with the variation of other variables entering the analysis can impact the statistical inference and, in the worst case, compromise the researcher's conclusions. For this reason, I analyse the determinants of misclassification by the FI algorithm in more detail using a logistic regression following Finucane (2000), Ellis et al. (2000) and Chakrabarty et al. (2007).

The previous studies focused on the LR algorithm (only Chakrabarty et al. (2007) also analyzed the determinants of misclassification for the EMO and the CLNV algorithm) and found that the most important determinant for correct classification is the execution of a trade against the quotes. The influence of other explanatory variables like trade size, spread, stock volume, firm size and the speed of trading does not always agree across the studies and is generally small in terms of their marginal effects. Still, slow trading and larger spreads seem to help the LR algorithm to infer the trade initiator.

D.1 Variable Selection

The logistic model is given by

$$P(y_i = 1 \mid x_i) = \Lambda(x_i'\beta)$$

where $y_i = 1$ is the event of a correct classification (and $y_i = 0$ the event of a misclassification) and $\Lambda(w)$ is the logistic distribution function, $\Lambda(w) = 1/(1 + \exp\{-w\}$. The explanatory variables, x_i, for the regression exercise are chosen as follows.

Section 5.4 already revealed that hidden orders are particularly difficult to classify. I will, thus, include a dummy variable, labeled Hidden, that takes the value 1 if the transaction involved a hidden order and 0 otherwise. Mid-point trades received a lot of attention in previous studies due to the reliance of the LR algorithm on the tick-test for such trades. I, therefore, include a variable that captures the distance of the transaction price to the mid-point at the time of the trade constructed as $\mathsf{Mid} = 1-2|p_t - m_t|/(a_t - b_t)$, where p_t is the execution price, m_t is the corresponding mid-point and a_t, b_t are the corresponding ask and bid quote, respectively. That is, Mid takes the value 1 if the trade executed at the mid-point and decreases towards 0 with the price approaching one of the quotes. What might be generally more important for accurate classification, however, is the proximity of buyer-initiated trades to the ask and that of seller-initiated trades to the bid. Hence, I include the variable Q-Dist $= |D_t - p_t|/(a_t - b_t)$ where $D_t = a_t$ if the trade is buyer-initiated and $D_t = b_t$ if the trade is seller-initiated.

The consideration of the mentioned variables is motivated by the classification criteria of the FI algorithm, and the obvious difficulty of classifying trades that deviate from the reasoning behind these criteria. To consider variables that could interact with the classification accuracy, though in a less obvious way, and play a role in more general economic and financial analyses I choose the following. I include the squared return of each transaction defined as $R^2 = (\log(p_i) - \log(p_{i-1}))^2$, where i is the i-th transaction, the size of each transaction in 100 shares (Size), the absolute spread size at the time of a trade measured in dollars (Spread), the total trading volume of the stock-day in 10^5 shares (Vol), the 5-minute realized variation (see Liu et al., 2015) over each stock-day (RV), the distance of each transaction to the previous trade in seconds (Δt-Trade), the distance of each transaction to the last quote change in seconds (Δt-Q), the number trades during the second of each trade (#Trades), the number of quote changes during the second of each trade (#Q) and a dummy variable indicating whether a transaction was part of a trade involving more than one counter party (MultiTrade). The latter is identified by observing more than one execution on the same side of the order book during the same nanosecond.

The following section provides summary statistics of the explanatory variables, a bi-variate correlation analysis, as well as a description of the filtering of the data before the actual estimation procedure. For numerical stability in the optimization procedure and to allow for a better comparison of the marginal effects across the variables, R^2 to #Q (i.e. all variables not ranging in $[0, 1]$) are standardized to have zero mean and unit variance.[31]

E Data Construction for Logistic Regression

The data used in the regression analysis is filtered as follows. Observations where either an ask or bid price is not quoted at the time of the trade are dropped because the spread is not defined in these cases. I also drop the first trade, because neither R^2 nor Δt-Trade are defined for the first trade. If a trade is not preceded by at least

[31] Due to the large number of zeros in R^2 (caused by many multi-party or successively placed, small trades), zero-observations have been left out in the standardization procedure for R^2. That is, they did not enter the computation for the mean and variance. Otherwise one would divide by near-zero and inflate the non-zero observations.

Table 18: Summary Statistics of Explanatory Variables

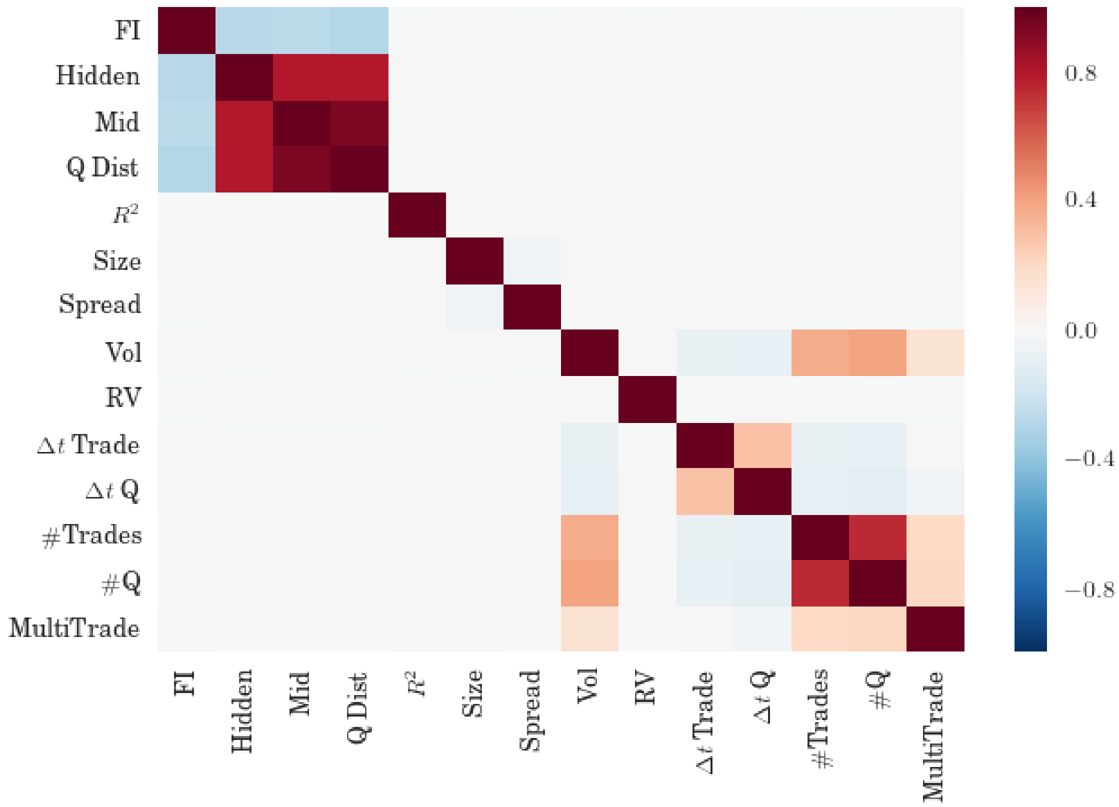
	mean	std	min	25%	50%	75%	max
Hidden	0.09	0.29	0	0	0	0	1
Mid	0.05	0.22	0	0	0	0	1
Q-Dist	0.03	0.12	0	0	0	0	1
$R^2 * 100$	0	0.11	0	0	0	0	1213.55
Size	1.67	4.50	0.01	1	1	1	4310.75
Spread	0.02	0.04	0.01	0.01	0.01	0.02	9.08
Vol	31.30	46.96	0.01	7.71	15.25	34.17	581.15
RV∗100	0.03	0.73	0	0.01	0.02	0.03	75.31
Δt Trade	6.62	23.01	0	0.01	0.48	5.21	12764.04
Δt Q	0.50	2.26	0	0	0.01	0.20	1098.91
# Trades	13.38	20.92	1	3	7	16	1020
# Q	76.70	104.44	0	18	46	98	32983
MultiTrade	0.65	0.48	0	0	1	1	1

Notes: This table shows the summary statistics of the explanatory variables. Hidden refers to a dummy variable taking the value 1 if the trade executed against a hidden order and 0 otherwise. Mid measures the distance of the execution price to the mid-quote, defined as $1-2|p_t-m_t|/(a_t-b_t)$, where p_t is the execution price, m_t is the corresponding mid-point and a_t, b_t are the corresponding ask and bid quote, respectively. Q-Dist measures the distance of the execution price to the quotes, defined as $|D_t - p_t|/(a_t - b_t)$ where $D_t = a_t$ if the trade is buyer-initiated and $D_t = b_t$ if the trade is seller-initiated. R^2 is the squared log-return of a transaction. Size is the number of shares exchanged in the transaction divided by 100. Spread is the absolute dollar spread at the time of the trade. Vol is the total trading volume of the stock-day divided by 10^5. RV is the 5-minute realized variation over the stock-day. Δt-Trade is the number of seconds since the previous trade. Δt-Q is the number of seconds since the last quote change. #Trades is the number of transactions during the same second of the trade. #Q is the number of quote changes during the second of the trade. MultiTrade is a dummy variable taking the value 1 if the transaction is part of a trade involving more than one counter-party and 0 otherwise.

one quote change, it is also dropped from the sample. Due to several quote changes happening at the same nanosecond it is possible that trades appear to be executed at negative spreads. Though the number of these instances is small, these observations are dropped. Due to several transactions taking place at the same nanosecond, it is also possible that trades appear to be executed outside the spread such that Q-Dist and Mid become negative. In these cases their values are truncated to 0. Table 18 presents summary statistics after this filtering process, which gives still over 134 million transactions to analyze.

To get a first idea of the influence of the explanatory variables on the event of a correct classification, as well as on possible cross-correlations among the regressors,

Figure 19: Correlation Matrix



Notes: This figure shows the correlations between the explanatory variables of the regression model, which are summarized in Table 18, and the dependent binary variable of a correct/false classification of a trade by the FI algorithm, as well as the correlations between the explanatory variables themselves depicted as a heat map.

Figure 19 depicts the correlation matrix of the explanatory variables and the dependent variable (the event of a correct classification by the FI algorithm applied at a timestamp precision of seconds). We see that, indeed, the number of misclassified transactions is higher for hidden orders, trades close to the mid-point and trades that execute away from the quote against which we would expect them to execute in the absence of hidden orders. As only trades that execute against a hidden order can be executed inside the spread, we see a strong, positive correlation between Hidden and the distance to the mid-point (Mid) or the distance to the quotes (Q-Dist).

In the regression analysis one could be worried that the coefficient of the event of a hidden order might overestimate the true effect of such an event as the placement of hidden orders and the execution against those may be viewed as endogenous decisions. In times of larger spreads, there is more room for placing hidden orders inside the spread, and traders searching for cheap execution prices may place successively

Table 19: Summary Statistics of Explanatory Variables for the Restricted Sample

	mean	std	min	25%	50%	75%	max
Hidden	0.09	0.28	0	0	0	0	1
Mid	0.05	0.21	0	0	0	0	1
Q-Dist	0.03	0.12	0	0	0	0	1
$R^2 * 10^6$	0.01	0.04	0	0	0	0	0.42
Size	1.38	1.46	0.01	1	1	1	13.99
Spread	0.02	0.02	0.01	0.01	0.01	0.01	0.15
Vol	27.52	34.51	0.01	7.83	15.12	31.78	233.70
RV*100	0.02	0.02	0	0.01	0.02	0.03	0.12
Δt Trade	4.98	10.46	0	0	0.45	4.83	81.36
Δt Q	0.34	0.89	0	0	0.01	0.18	7.75
#Trades	11.69	13.32	1	3	7	15	91
#Q	69.02	71.11	0	19	46	95	468
MultiTrade	0.65	0.48	0	0	1	1	1
N Obs.: 124254433							

Notes: This table shows the summary statistics of the sample presented in Table 18 restricted to observations where the variables R^2 to #Q do not exceed their 99th-percentile. This sample provides the baseline for the logistic regression of the probability of a correct classification of a transaction by the FI algorithm.

small orders to find those hidden orders. The correlation matrix suggests that such concerns can be neglected. In fact, the bivariate correlation analysis suggests that other than the variables Hidden, Mid and Q-Dist there is no strong, linear impact of the explanatory variables on the event of a correct classification.

To insure that we do not encounter problems with a few extreme outliers in the estimation procedure, I did not consider observations where one of the variables from R^2 to #Q exceeded their 99th-percentile. The summary statistics of this restricted sample used in the regression analysis are presented in Table 19.

E.1 Estimation Results

Due to the large number of observations and computational constraints, I did not use the full sample in the logistic regression. Instead, I selected all observations where the FI algorithm misclassified the transaction (almost 7.6 million observations) and randomly selected (without replacement) a sample of equal size from the observations where the FI algorithm correctly classified the trade. Maximum likelihood estimation of the logistic model still yields consistent estimates, except for the constant, which

Table 20: Logistic Regression Result

No. Observations:				15190680
McFadden R-squ.:				0.2205
Log-Likelihood:				-8.2072e+06
LL-Null:				-1.0529e+07
			marginal effects	
	β	std err	at the mean	overall
const	3.1719	0.001		
Hidden	-2.3120	0.003	-0.2055	-0.1130
Mid	0.8656	0.008	0.0299	0.0423
Q-Dist	-3.6268	0.015	-0.1254	-0.1773
R^2	0.0479	0.001	0.0017	0.0023
Size	0.1714	0.001	0.0059	0.0084
Spread	0.3403	0.001	0.0118	0.0166
Vol	0.3388	0.001	0.0117	0.0166
RV	-0.0912	0.001	-0.0032	-0.0045
Δt Trade	0.3880	0.001	0.0134	0.0190
Δt Q	0.2853	0.001	0.0099	0.0140
# Trades	0.2157	0.001	0.0075	0.0105
# Q	-0.7683	0.001	-0.0266	-0.0376
MultiTrade	0.5896	0.001	0.0223	0.0307

Notes: This table shows the regression results of a maximum likelihood estimation of the model

$$P(y_i = 1|x_i) = \Lambda(x_i'\beta)$$

with $y_i = 1$ being the event of a correct classification by the FI algorithm applied to the data timestamped to the second, and x_i containing the explanatory variables described in Table 18. $\Lambda(\cdot)$ is the logistic distribution function. The variables R^2 to #Q have been standardized. The estimates are based on a sub-sample containing all observations where $y_i = 0$ and a random selection of equal size of observations where $y_i = 1$. This yields consistent estimates except for the coefficient of the constant term, β_0. To obtain a consistent $\hat{\beta}_0$ one simply subtracts $\log((1-p)\bar{y}/p(1-\bar{y}))$, where p is the frequency of $y_i = 1$ in the full sample and $\bar{y}$ the corresponding frequency in the sub-sample. $\hat{\beta}_0$ in the Table is the bias corrected estimate. The marginal effects are evaluated at the sample mean, as well as evaluated at each data point of the standardized data and then averaged, i.e.:

$$\text{at the mean:} \quad \frac{\partial P(y=1|\bar{x})}{\partial x_k} = \Lambda(\bar{x}_i'\hat{\beta})(1-\Lambda(\bar{x}_i'\hat{\beta}))\hat{\beta}_k$$

$$\text{overall:} \quad \frac{\partial P(y=1|x)}{\partial x_k} = \sum_i \Lambda(x_i'\hat{\beta})(1-\Lambda(x_i'\hat{\beta}))\hat{\beta}_k/N.$$

For the dummy variables the effects are computed analogously using

$$P(y_i = 1|x_{ik} = 1, x_i) - P(y_i = 1|x_{ik} = 0, x_i).$$

can be easily corrected to obtain consistency (see e.g. Eq. (7) and Appendix B in King and Zeng, 2001).[32] Table 20 presents the regression results. The FI algorithm is applied to the data timestamped to the second.

We see that the single most important determinant for misclassification is the execution against a hidden order. On average, a trade that executes against a hidden order as opposed to a visible order decreases the estimated probability of a correct classification by 11%-points. Once controlled for the impact of a hidden order, the effect of the distance to the quotes or the proximity to the midpoint seems less important. For example, moving 10% of the spread size away from the quote against which we would expect the trade to execute decreases the estimated probability of a correct classification by approximately 1.8%-points on average.

The variables that may play a more decisive role in more general economic and financial studies involving the estimation of the trade initiator, like total trading volume, the realized variation or the speed of trading, do not strongly impact the classification accuracy. For example, a one standard deviation increase in the realized variation decreases the estimated probability of correct classification by only about 0.45%-points. Among these variables, frequent quote changes during the second of the trade (#Q) exhibit the strongest effect on the classification accuracy: a one standard deviation increase in the number of quote changes during the second of the trade decreases its probability of being correctly classified by around 3.8%-points on average.[33] Moreover, the correlation matrix presented in Figure 19 in the appendix suggests that the effect of hidden orders would not indirectly feed into an analysis involving the other economically relevant variables as the correlations between hidden orders and the other variables are near zero.

[32]The only requirement for consistency is that the conditional densities of the subsampled data $(x|y)$ matches the conditional density of the full sample $(X|Y)$, i.e. $P(x \mid y = 1) = P(X \mid Y = 1)$ and $P(x \mid y = 0) = P(X \mid Y = 0)$. The latter is trivially satisfied in my case as I select all observations where $y_i = 0$ and the former should be satisfied by the random subsampling scheme.

[33]Note that despite the predictive content of the explanatory variables for the probability of a correct classification, this does not imply a predictive power of these variables for the trade initiator label. The analysis only identifies environments under which it is more difficult to arrive at the true initiator label, but it did not identify the direction of the misclassification.