



CME iceberg order detection and prediction

DMITRY ZOTIKOV *[†] and ANTON ANTONOV [‡]

[†]Devexperts LLC, Barochnaya 10-1, St. Petersburg 197110, Russia

[‡]dxFeed Solutions DE GmbH, Mies-van-der-Rohe-Straße 6, München 80807, Germany

(Received 22 November 2019; accepted 7 August 2020; published online 26 October 2020)

We propose a method for detection and prediction of native and synthetic iceberg orders on the Chicago Mercantile Exchange. Native (managed by the exchange) iceberg orders are detected using discrepancies between the resting volume of an order and the actual trade size as indicated by trade summary messages, as well as by the tracking order modifications that follow trade events. Synthetic (managed by market participants) iceberg orders are detected by observing limit orders arriving within a short time frame after a trade. The obtained iceberg orders are then used to train a model based on the Kaplan–Meier estimator, accounting for orders that were cancelled after a partial execution. The model is utilised to predict the total size of newly detected iceberg orders. Out of sample validation is performed on the full order depth data, performance metrics and quantitative estimates of hidden volume are presented.

Keywords: Market microstructure; Limit order market; Stochastic models; Statistical methods; Statistics

JEL Classification: C14, C51, C52, C24

1. Introduction

On financial exchanges, an *iceberg order* is a limit order where only a fraction of the total order size (*display quantity*) is shown in the limit order book (LOB) at any one time (*peak*), with the remainder of volume hidden (Christensen and Woodmansey 2013). When the peak is executed, the next part of the iceberg's hidden volume (*tranche* or *refill*) gets displayed in the LOB. This process is repeated until the initial order is fully traded or cancelled.

The hidden volume, although not being directly observed, is de facto present in the LOB and hence can be traded against or affect trading decisions. This makes the detection of hidden liquidity a desirable goal for interested parties, e.g. traders and market makers.

In this paper, we propose a method for detecting and predicting hidden liquidity on the Chicago Mercantile Exchange (CME). The model is fit and assessed out of sample using historical data. We treat it both as a classification and a regression model and discuss relevant performance metrics.

1.1. Data

We had access to four days of full-order depth (FOD) LOB data of a September E-Mini S&P 500 futures contract,

existing at that time under the ticker symbol ESU19, for the period from 2019-06-17, 11:00:00 CDT to 2019-06-21, 16:00:00 CDT. The chosen interval is especially interesting from a trading activity standpoint: as the front month contract ESM19 approaches its expiration, a majority of the open interest gets transferred onto the next one, creating an increased demand for hidden liquidity vehicles. In addition, a special FOMC announcement took place within the observation interval,[†] which could have resulted in a significant impact on the trading activity. To compensate for that, we include that particular trading session into the learning period of our model, thus capturing both pre- and post-announcement market regimes. This is discussed further in section 7.1.

Each order was described by a sequence of fields presented in table 1.

In addition, trade summary messages (CME 2019d) were present in the data. Each trade event against a resting order corresponded to a trade record in the log with the aforementioned fields, 'Action' set to 'Trade' and an extra field for the passive order ID.

For more information about CME Market-by-Order book management, see CME (2019c). A complete product specification is available on CME Globex website, CME (2019a). In particular, the matching algorithm is FIFO: all orders at the same price level are filled according to time priority.

*Corresponding author. Email: dmitry.zotikov@devexperts.com; dmitry.zotikov@ungrund.org

[†] <https://www.federalreserve.gov/monetarypolicy/fomccalendars.htm>

Table 1. Order log fields for the ESU19 data.

Field	Time	Order ID	Side	Action	Price	Volume
Possible values	Timestamp, millisecond resolution	12-digit identifier	‘B’ (buy), ‘S’ (sell)	‘Limit’ (new), ‘Modify’ (update), ‘Delete’	Non-negative real	Non-negative integer

Our algorithm works in the ‘historical’ mode by reading a pre-recorded LOB log. Since it is not forward-looking, it can be easily modified to work with real-time streaming data.

1.2. CME iceberg orders

As briefly mentioned in the introduction, an iceberg order is characterised by the fact that only a part of its total volume V_{total} is visible in the LOB. Upon placing the order, a trader may choose to display a quantity V_{peak} —that is, the volume of the visible part of the order. This effectively splits the order \mathcal{I} into M tranches \mathcal{T}_r , $r \in 1 : M$. Let $\text{vol}(\cdot)$ denote the volume of either an iceberg order or an iceberg tranche; then

$$\mathcal{I} = (\mathcal{T}_1, \dots, \mathcal{T}_M),$$

so that

$$V_{\text{total}} = \text{vol}(\mathcal{I}) = \sum_{r=1}^M \text{vol}(\mathcal{T}_r) = \sum_{r=1}^M V_{\text{peak}}^{(r)}.$$

Note that in general V_{total} may not be divisible by V_{peak} , so it is natural to expect $V_{\text{peak}}^{(r)}$ to be constant for all tranches but the last: $V_{\text{peak}}^{(M)} \leq V_{\text{peak}}^{(r)}$, $r \in 1, \dots, M-1$.

We would expect tranches \mathcal{T}_r to sequentially appear in book as visible volume once the previous tranche has been traded—until the order is fully traded or cancelled. However, in reality not all tranches may enter the book, as sometimes the hidden volume is traded directly without becoming visible first, so $\text{vol}(\mathcal{T}_r) \neq V_{\text{peak}}^{(r)}$. Additionally, V_{peak} may change as the order is being executed, so $V_{\text{peak}}^{(r)} \neq V_{\text{peak}}$, $r \in 1 : M$.

Detection-wise, let \mathcal{A} denote a LOB message as in table 1, which we will refer to as an ‘action’. Then it is possible to identify a finite sequence $(\mathcal{A}_{r,1}, \mathcal{A}_{r,2}, \dots)$ that constitutes the tranche \mathcal{T}_r , $r \in 1 : M$ of the iceberg order \mathcal{I} . The ultimate detection goal is to identify all such actions in the stream and form a data structure that would describe the iceberg order \mathcal{I} , from which it would be possible to compute V_{total} and infer V_{peak} .

Depending on how new tranches appear in the LOB, one may distinguish between two types of iceberg orders on CME: *native* and *synthetic*. The mechanics of the two are quite different, which is reflected in how the detection is performed.

Native icebergs are managed by the exchange itself (CME 2019b). All new tranches are submitted as modifications of the initial order; this means that the original order ID is preserved throughout the whole lifetime of the iceberg. Moreover, any trades involving the iceberg

order will indicate the total volume of the trade, including the hidden part of the iceberg. Using these two properties, it is possible to unambiguously detect a sequence of new–trade–update–delete actions that forms an iceberg.

Synthetic icebergs are submitted by independent software vendors (ISVs), whose infrastructure is physically separated from the exchange. ISVs split the initial iceberg order, submit new tranches as new limit orders and track their execution. These tranches are indistinguishable from usual limit orders submitted by other participants. Thus synthetic icebergs, being identical to non-iceberg orders in how they are processed by the exchange, can only be detected heuristically relying on a set of assumptions which are introduced below.

We equally focus on detection and prediction of both synthetic and native icebergs.

2. Existing literature overview

For a good literature overview, see Christensen and Woodmansey (2013, p. 7). In particular, some authors had access to order logs in which iceberg orders were explicitly identified, so that an unambiguous reconstruction of both displayed and hidden LOBs was possible, e.g. Frey and Sandås (2017) for the XETRA exchange. There are a few articles that provide hidden liquidity estimates—see Hautsch and Huang (2010, p. 5) and Christensen and Woodmansey (2013, p. 7) for such lists. These estimates differ across authors, exchanges and instrument types, ranging from 2% (Fleming *et al.* 2018) to 52% (Moro *et al.* 2009). That being said, most of the papers were published almost a decade ago, so one may argue that contemporary markets have different hidden volume properties.

To our knowledge, there are virtually no articles that have the premise of simultaneous iceberg detection and prediction in the same setting as ours, with the exception of Christensen and Woodmansey (2013), who propose a solution to the very problem with which the authors of the present paper are concerned. Their approach consists of the following three phases:

- (i) A series of tranches are identified in the data as belonging to larger iceberg orders (the ‘detection’ step).

- (ii) Using the detected icebergs, a statistical model is fit that captures the correspondence between the peak size and the total iceberg size (the ‘learning’ step).
- (iii) The detection step is repeated, but once a new iceberg order is detected, a prediction of the total size of the iceberg is made using the model obtained at the previous step (the ‘prediction’ step).

We have identified a few aspects by which the proposed approach could be further improved:

- The authors did not have access to FOD MBO data at the time of writing. In particular, the order ID for each action or trade was not available, yet that drastically changes the logic of the detection step.
- No distinction between synthetic and native icebergs is made. Namely, it is assumed that trades can sometimes be larger than the size of the resting order being traded, which is specific for native icebergs; however, iceberg tranches arrive as new limit orders, and that is an attribute of synthetic icebergs.
- During the learning phase, a bivariate Gaussian kernel density estimate of peak and total size is built, which is then optimised for the global maximum given a peak size. For the purpose of prediction, where only one value of the total size corresponding to the maximum probability given a peak size is necessary, this complication is questionable as a simpler model is sufficient.[†] Kernel density estimate may be desired if the algorithm operates on instruments with a relatively low daily trading volume, and this is not the case with our data. In addition, by omitting this step we do not have to resort to numerical methods when optimising for conditional maxima.
- All incomplete icebergs, i.e. those that were cancelled before being fully executed—are not included into the learning phase. However, our calculations show that more than half of all synthetic icebergs are cancelled, thus it is highly desirable to include the information about incomplete executions into the model.

In our work, we adapt the aforementioned detection–learning–prediction scheme, albeit with the following notable differences and enhancements:

- FOD MBO data containing order IDs are used. In particular, this allows for unambiguous detection of native icebergs. Synthetic icebergs are also detected differently, as we additionally account for the uncertainty related to selecting the right limit orders as iceberg tranches.
- A different prediction model is utilised, which is computationally simpler and allows for inclusion of incomplete icebergs.

[†] Moreover, continuous kernels will assign non-zero densities to fractional volume ranges, which are not supported by the majority of established trading venues. It should be noted that the authors consider discrete kernel estimation, but opt to use the Gaussian kernel ‘on the basis of simplicity’.

3. Detection

3.1. Native icebergs

We would like to motivate the introduction of the detection algorithm with the following example. Consider the data presented in table 2.

- (i) Order #6...30354 enters the book and immediately gets traded at 2931.75 for the total of 12 units of volume. The remainder—6 units—is placed at the same level. At this point we do not know whether the order has any hidden liquidity or not. Moreover, assuming that it does, the peak size cannot be precisely determined; but since $12 + 6 = 18$, it is one of the divisors of 18 greater than or equal to 6, i.e. 6, 9 or 18.
- (ii) The next trade has volume 8 which is larger than the resting volume of 6. This is sufficient to mark order #6...30354 as an active iceberg.
- (iii) The next tranche volume is 7. Note that $8 - 6 = 2$ units of volume were traded against a tranche that had not entered the book. This means that V_{peak} can be determined precisely as $7 + (8 - 6) = 9$. The trade could have been large enough to consume several hidden tranches.
- (iv) The next several trades are smaller in volume than the resting order. The trade initiated by order #6...30365 is equal to the resting volume of 1. Consequently, the next modify action is seen to refresh the visible volume by the peak size of 9 (which agrees with the previous calculations).
- (v) Finally, the last update action has volume 5, which, accounting for the hidden trade of $9 - 7 = 2$ results in peak size of 7. The trade for 5 units completes the sequence as no more refresh messages are seen and the order is deleted from the book.

Therefore, we infer that the iceberg $\mathcal{I} = (\mathcal{T}_1, \dots, \mathcal{T}_4)$ had four tranches of sizes $\text{vol}(\mathcal{T}_1) = 20$, $\text{vol}(\mathcal{T}_2) = 7$, $\text{vol}(\mathcal{T}_3) = 11$, $\text{vol}(\mathcal{T}_4) = 5$ giving $V_{\text{total}} = \sum_{r=1}^4 \text{vol}(\mathcal{T}_r) = 43$. $V_{\text{peak}}^{(1)} \in \{6, 9, 18\}$, $V_{\text{peak}}^{(2)} = V_{\text{peak}}^{(3)} = 9$, $V_{\text{peak}}^{(4)} = 7$ and $V_{\text{peak}} = 9$. Note that the IDs of all constituent LOB messages (action) are equal to the iceberg order ID, so $\text{id}(\mathcal{I}) = \text{id}(\mathcal{T}_r) = \text{id}(\mathcal{A}_{\cdot})$.

From the implementation standpoint, an iceberg $\mathcal{I} = (\mathcal{T}_1, \dots, \mathcal{T}_M)$ is simply an ordered collection of tranches, whereas each tranche $\mathcal{T}_r = (\mathcal{A}_{r,1}, \mathcal{A}_{r,2}, \dots)$ is an ordered collection of LOB messages. It is convenient to implement the algorithm in the OOP-fashion and equip the objects with additional fields to store V_{peak} and related quantities. In what follows we use the functional notation $\text{FUNCTION}(\mathcal{O})$ to call methods and retrieve field values of an object \mathcal{O} .

We can generalise the algorithm as follows. Clearly, the incoming LOB messages modify the state of an order. The most natural way to formalise this is to describe the transition between states using a finite state automaton (FSA), see figure 1. An iceberg enters the book as a new limit order, possibly following a sequence of trades. It is then traded, and usually—but not always—each trade corresponds to one trade summary message, in which case it is followed by an update action, specifying the currently resting order volume. If more than one trade message is seen before the next update action,

Table 2. Sample native iceberg order log data.

Time	Order ID	Side	Action	Price	Volume	Affected
14:05:33.416	6...30354	S	Trade	2931.75	2	6...30338
14:05:33.416	6...30354	S	Trade	2931.75	10	6...30339
14:05:33.416	6...30354	S	Limit	2931.75	6	—
14:05:33.416	6...30360	B	Trade	2931.75	8	6...30354
14:05:33.416	6...30354	S	Modify	2931.75	7	—
14:05:33.416	6...30361	B	Trade	2931.75	3	6...30354
14:05:33.416	6...30354	S	Modify	2931.75	4	—
14:05:33.416	6...30362	B	Trade	2931.75	2	6...30354
14:05:33.416	6...30354	S	Modify	2931.75	2	—
14:05:33.416	6...30363	B	Trade	2931.75	1	6...30354
14:05:33.416	6...30354	S	Modify	2931.75	1	—
14:05:33.416	6...30365	B	Trade	2931.75	1	6...30354
14:05:33.416	6...30354	S	Modify	2931.75	9	—
14:05:33.416	6...30366	B	Trade	2931.75	1	6...30354
14:05:33.416	6...30354	S	Modify	2931.75	8	—
14:05:33.416	6...25841	B	Trade	2931.75	1	6...30354
14:05:33.416	6...30354	S	Modify	2931.75	7	—
14:05:33.417	6...30382	B	Trade	2931.75	9	6...30354
14:05:33.417	6...30354	S	Modify	2931.75	5	—
14:05:33.417	6...30390	B	Trade	2931.75	5	6...30354
14:05:33.417	6...30354	S	Delete	2931.75	5	—

Note: The orders related to the same tranche are grouped.

then this should be accounted for. Moreover, all price adjustments which move the order to the top of the book are not disseminated by the exchange, meaning that even after the placement the order can again act as an aggressive order and initiate a trade. If at this point the order is deleted from the book or traded so that the trade volume is never greater than the resting volume, it is marked as ‘ordinary’ and removed from consideration. On the other hand, once a trade larger than the resting volume is detected, or the order is fully traded but then modified to have non-zero volume again, then the order is marked as an iceberg. The trade–modify cycle then continues until the order is completely executed or cancelled, resulting in its deletion from the book.

In addition to simply tracking the transitions through the state space, the following operations are performed:

- A tranche \mathcal{T}_r is formed and added to the iceberg for each transition $[\{\text{start, initial trade}\} \rightarrow \text{first tranche}]$, $[\text{first tranche traded} \rightarrow \text{next tranche}]$, or $[\text{next tranche traded} \rightarrow \text{next tranche}]$. All the following actions are then assigned to the tranche.
- The *total volume* V_{total} is conveniently computed as the sum of all tranche volumes; in turn $\text{vol}(\mathcal{T}_r)$ is determined as the total trade volume V_T of the constituent tranche actions (which may exceed the sum of limit and/or update volumes), plus any volume V_D that is explicitly deleted;
- The *currently resting volume* V_R is set to the initial V_L , then adjusted by modify message volumes V_M , and finally set to 0 upon order deletion;
- The *peak size* V_{peak} is determined iteratively:
 - If the iceberg order enters the book directly as a limit order, then the limit order volume is indeed the peak size.

- If a series of trades precedes the limit order placement, then $V_L = V_{\text{peak}} - V_T \bmod V_{\text{peak}}$. Therefore, $V_T - kV_{\text{peak}} = V_{\text{peak}} - V_L$ for some $k \in \mathbb{N}_0$, from which we obtain

$$V_{\text{peak}} = \frac{V_T + V_L}{k + 1} = \frac{V_T + V_L}{d},$$

$$d = 1, \dots, V_T + V_L, V_{\text{peak}}$$

$$\in \{n \in \mathbb{N} : n \geq V_L\}.$$

If more than one admissible V_{peak} values are found, then the following heuristics apply.

- * If the first tranche is traded for exactly the resting volume, the following update message unambiguously identifies the peak size.
- * If the trade volume is greater than the resting volume, then

$$V_{\text{peak}} = V_M + (V_T - V_R) \bmod V_{\text{peak}}^*,$$

where V_{peak}^* is one of previously computed values. Only the values that satisfy this equation are kept.

- If a resting tranche is modified, then the peak volume is reduced (increased) to the same extent as the resting volume:

$$V_{\text{peak}} = V_{\text{peak}} - (V_R - V_M).$$

Suppose the function $\text{FSSTEPNATIVE}(\mathcal{I}, \mathcal{A})$ implements the above logic given an iceberg \mathcal{I} and an incoming LOB message \mathcal{A} ; the IDs of the affected and of the incoming orders can be retrieved using $\text{AFFECTEDID}(\mathcal{A})$ and $\text{ORDERID}(\mathcal{A})$;

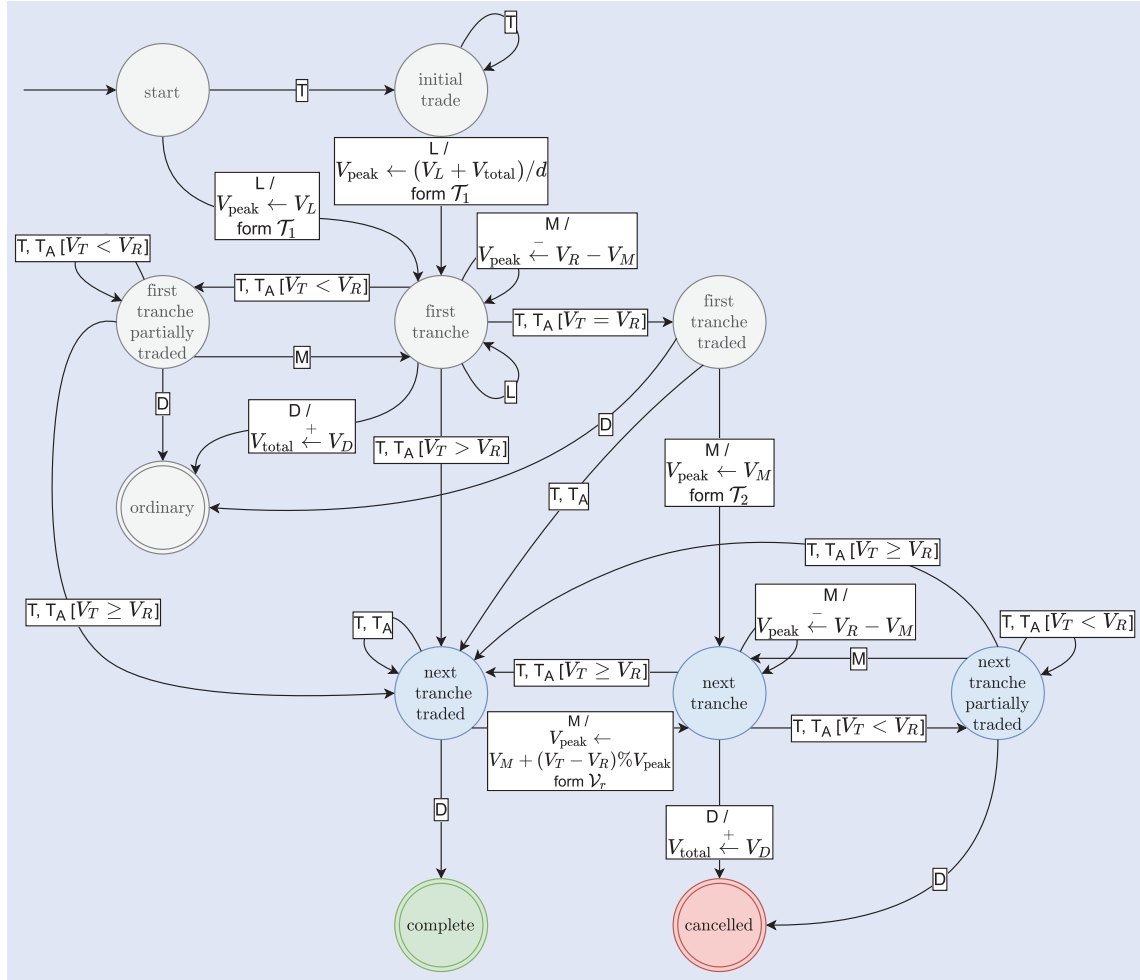


Figure 1. FSA for native icebergs. The nodes correspond to states of the finite state machine (order / iceberg states), the edges—to order actions: new (L), update (M), trade (T), ‘affected’ trade (T_A), delete (D). Trades T are initiated by the iceberg order, while ‘affected’ trades T_A —by other incoming orders. An optional condition is specified in square brackets; a side effect is specified after the slash (/). Different volumes V refer to the iceberg’s peak volume V_{peak} , total volume V_{total} , current resting volume V_R and the order’s trade (V_T), delete (V_D), modify (V_M) volumes. Node colours represent the status of the action sequence being tracked: grey for non-iceberg (‘ordinary’), blue for active (all statuses starting with ‘next tranche ...’), green for complete and red for cancelled.

the icebergs are stored in a pool from which they are retrieved using $\text{GETICEBERG}(\text{id}_{\mathcal{I}})$ and created using $\text{CREATEICEBERGNATIVE}(\text{id}_{\mathcal{I}})$. The algorithm 1 summarises the detection step.

Algorithm 1 Detection step for native icebergs.

```

procedure DETECTIONSTEPNATIVE( $\mathcal{A}$ )
  if AFFECTEDID( $\mathcal{A}$ )  $\neq \emptyset$  then
     $\mathcal{I} \leftarrow \text{GETICEBERG}(\text{AFFECTEDID}(\mathcal{A}))$ 
    if  $\mathcal{I} \neq \emptyset$  then
      FSASTEPNATIVE( $\mathcal{I}, \mathcal{A}$ )  $\triangleright$  What happens with
      the resting order?
    end if
  end if
   $\mathcal{I} \leftarrow \text{GETICEBERG}(\text{ORDERID}(\mathcal{A}))$ 
  if  $\mathcal{I} = \emptyset$  then
     $\mathcal{I} \leftarrow \text{CREATEICEBERG}(\text{ORDERID}(\mathcal{A}))$ 
  end if
  FSASTEPNATIVE( $\mathcal{I}, \mathcal{A}$ )  $\triangleright$  What happens with the
  incoming order?
end procedure

```

3.2. Synthetic icebergs

Unlike native icebergs, synthetic icebergs are managed by ISVs. After the tranche \mathcal{T}_r is fully executed, the ISV is expected to submit the next tranche \mathcal{T}_{r+1} as a *new limit order*. The ID of that order $\text{id}(\mathcal{A}_{r,\cdot}) \neq \text{id}(\mathcal{A}_{r+1,\cdot})$ would be different from that of the previously traded tranche \mathcal{T}_r , hence the native iceberg mechanics cannot be utilised.

One idea (see Christensen and Woodmansey 2013) is to exploit the fact that the ISV infrastructure is located outside the exchange, and thus a short non-constant delay (dubbed dt) is expected between the deletion of \mathcal{T}_r and the arrival of the new limit order corresponding to \mathcal{T}_{r+1} . Additionally, we assume that the trader chooses the price level and the LOB side apart from the V_{peak} value, and that they stay constant throughout the lifetime of the iceberg unless modified directly by a corresponding ‘modify’ message. Therefore we formally define a synthetic iceberg as a sequence of limit orders, each corresponding to a tranche ($\mathcal{T}_1, \dots, \mathcal{T}_M$), such that their price, side & volume are the same (unless changed by ‘modify’ messages during the execution) and such that \mathcal{T}_{r+1} arrives within dt seconds after \mathcal{T}_r was deleted from the book. If a tranche is

executed, but no refill orders follow within dt , the iceberg is considered *complete*; if a tranche is placed and later cancelled, the whole iceberg is considered cancelled (*incomplete*).

However, there are three issues with the definition:

- (i) When several limit orders of the same side, price and size (being a part of several icebergs) get executed and deleted from the book simultaneously, the next tranche can be ‘linked’ to any of those. Under the current model it is impossible to unambiguously tell to which iceberg the new tranche belongs, so we add it to all suitable icebergs, thus forming an iceberg *tree* \mathcal{I} . Each path from the root to any of the leaves of the tree—a linear sequence of tranches, which we call a tranche *chain*—represents a possible iceberg $\mathcal{I}^{(\ell)}$. See an example below.
- (ii) Under the current model we do not consider synthetic icebergs with varying peak sizes or price levels, even though they may technically exist. Otherwise icebergs would be indistinguishable from completely different phenomena, such as high-frequency trader activity or optimal execution algorithms. This in particular implies that if the iceberg is not a multiple of the display quantity, the last tranche will be smaller than all the previous tranches in volume, hence its detection using the current approach does not seem to be possible.
- (iii) If the activity in the LOB is high (as in the case with E-Mini S&P 500 contracts), more than one order of the target volume may arrive on the same price level within dt . Our very strong assumption is that the next tranche arrives faster than any other new limit order, so for each tranche there is only one child. A more sophisticated model would account for all possible children tranches, which would introduce a range of algorithmic challenges. This preference was made on the basis of simplicity and likeness to the native iceberg mechanics, where dt is effectively zero. The definition thus presents an opportunity for improvement, which we reserve for future research.

Suppose $dt = 0.3$ seconds and consider the following illustrative example (table 3).

- (i) Limit order #1 is placed as sell 2 @ 1000, shortly gets traded by #101 and removed from the book. A timer is set at this combination of side, price and volume.
- (ii) The following limit order #2 arrives after $0.1 < dt$ seconds having the same side, price and size as #1. We conclude that the two orders are a part of an iceberg \mathcal{I}_1 with one chain $\mathcal{I}_1^{(1)} = (\mathcal{T}_1, \mathcal{T}_2)$.
- (iii) Orders #4 and #5 are placed at ask 2 @ 1000 but not yet traded.
- (iv) #2 is traded by #102, #3 arrives within the dt , thus becoming the next tranche in $\mathcal{I}_1^{(1)} = (\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3)$.
- (v) After #4, #5 get traded simultaneously, order #6 arrives within dt , thus becoming the next tranche for both of the orders: $\mathcal{I}_2^{(1)} = (\mathcal{T}_4, \mathcal{T}_6)$, $\mathcal{I}_2^{(1)} = (\mathcal{T}_5, \mathcal{T}_6)$.
- (vi) #7 is placed at ask 2 @ 1000.

Table 3. Artificial data to demonstrate synthetic iceberg detection.

Time	Order ID	Side	Action	Price	Volume	Affected
18:22:12.00	1	S	Limit	1000	2	—
18:22:12.01	101	B	Trade	1000	2	1
18:22:12.01	1	S	Delete	1000	2	—
18:22:12.02	2	S	Limit	1000	2	—
18:22:12.04	4	S	Limit	1000	2	—
18:22:12.04	5	S	Limit	1000	2	—
18:22:13.00	102	B	Trade	1000	2	2
18:22:13.00	2	S	Delete	1000	2	—
18:22:13.01	3	S	Limit	1000	2	—
18:22:13.00	103	B	Trade	1000	2	4
18:22:13.00	104	B	Trade	1000	2	5
18:22:13.00	4	S	Delete	1000	2	—
18:22:13.00	5	S	Delete	1000	2	—
18:22:13.01	6	S	Limit	1000	2	—
18:22:14.00	7	S	Limit	1000	2	—
18:22:15.00	105	B	Trade	1000	2	3
18:22:15.00	106	B	Trade	1000	2	6
18:22:15.00	107	B	Trade	1000	2	7
18:22:15.00	3	S	Delete	1000	2	—
18:22:15.00	6	S	Delete	1000	2	—
18:22:15.00	7	S	Delete	1000	2	—
18:22:15.01	8	S	Limit	1000	2	—
18:22:16.00	108	B	Trade	1000	2	8
18:22:16.00	8	S	Delete	1000	2	—
18:22:16.01	9	S	Limit	1000	2	—
18:22:16.50	109	B	Trade	1000	2	9
18:22:16.50	9	S	Delete	1000	2	—

- (vii) The resting #3, #6, #7 are all traded by #103, #106, #107. #8 arrives within dt , thus the third iceberg $\mathcal{I}_3^{(1)} = (\mathcal{T}_7, \mathcal{T}_8)$ is formed. However, $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$ will all have to be merged together, which results in just one iceberg \mathcal{I}_1 with four chains $\mathcal{I}_1^{(1)} = (\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_8)$, $\mathcal{I}_1^{(2)} = (\mathcal{T}_4, \mathcal{T}_6, \mathcal{T}_8)$, $\mathcal{I}_1^{(3)} = (\mathcal{T}_5, \mathcal{T}_6, \mathcal{T}_8)$, $\mathcal{I}_1^{(4)} = (\mathcal{T}_7, \mathcal{T}_8)$.
- (viii) The next tranche #9 gets appended to all of the chains, then gets trade-deleted. If no other orders arrive at ask 2 @ 1000, then \mathcal{I}_1 will be considered complete.

See figure 2 for the resulting graph.

Implementation-wise, unlike native icebergs, it is convenient to implement an iceberg as a doubly linked list, since a tranche can belong to multiple tranche chains. Other than that, orders are now separate entities from icebergs, hence the need for two FSAs: one for tracking order/tranche states, the other one for tracking iceberg states, see figures 3 and 4.

For now, suppose an incoming limit order is not a part of an iceberg. After entering the book, the order might get traded, in which case we say the order starts ‘listening’ and set a timer corresponding to the order side, size and price level; alternatively, the order is cancelled, in which case we remove it from consideration. If at some point the order behaves like a native iceberg, we transition to the ‘native iceberg’ state and track the order state until it is deleted and then remove it from consideration as well. If, while rested, the order is modified, we save those changes, in particular, we adjust V_{peak} . Additionally, the order might initiate a trade while being in the LOB already (T instead of T_A)—this happens if the order was modified to

be relocated to another price level; CME does not disseminate the ‘modify’ message in that case.

If an incoming order is such that it matches one of the pre-set timers, then it either has to be linked to the corresponding iceberg, or the iceberg itself has to be merged with another iceberg, or the timer has to be deleted, see algorithm 2.

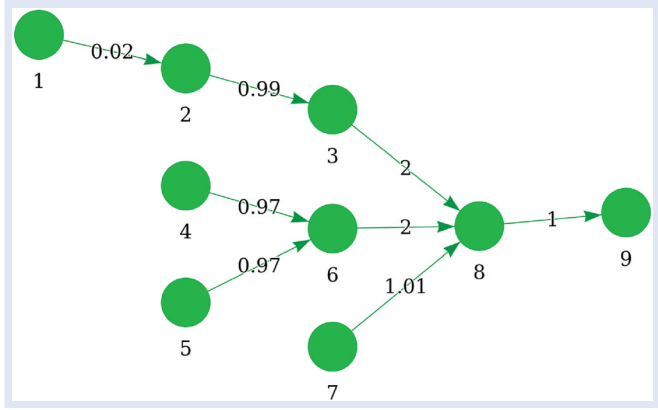


Figure 2. An iceberg tranche tree corresponding to table 3. Node labels are order IDs. Edge labels are time in seconds between subsequent tranches—note that these are different from dt as a tranche can remain indefinitely long in the book after its placement. The iceberg consists of either 3, 4 (two chains) or 5 tranches.

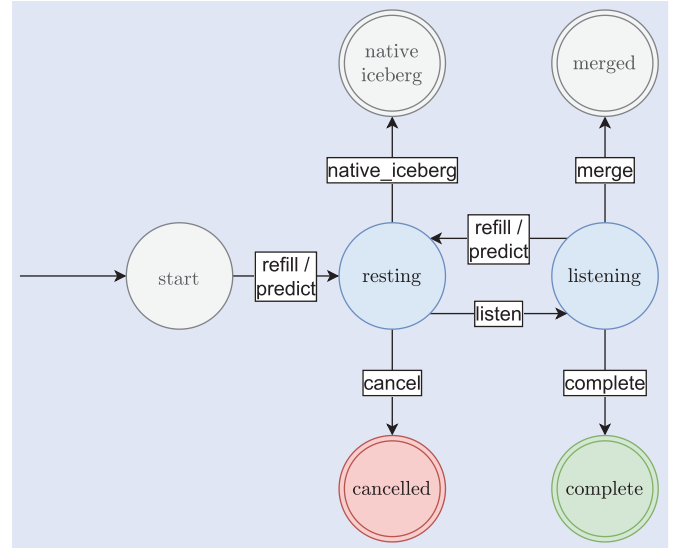


Figure 4. FSA for synthetic icebergs.

In the former two cases, the order is considered an iceberg tranche and we continue to track its state, changing the state of the underlying iceberg when necessary (using $FSA_{SI}(\cdot)$ side-effects).

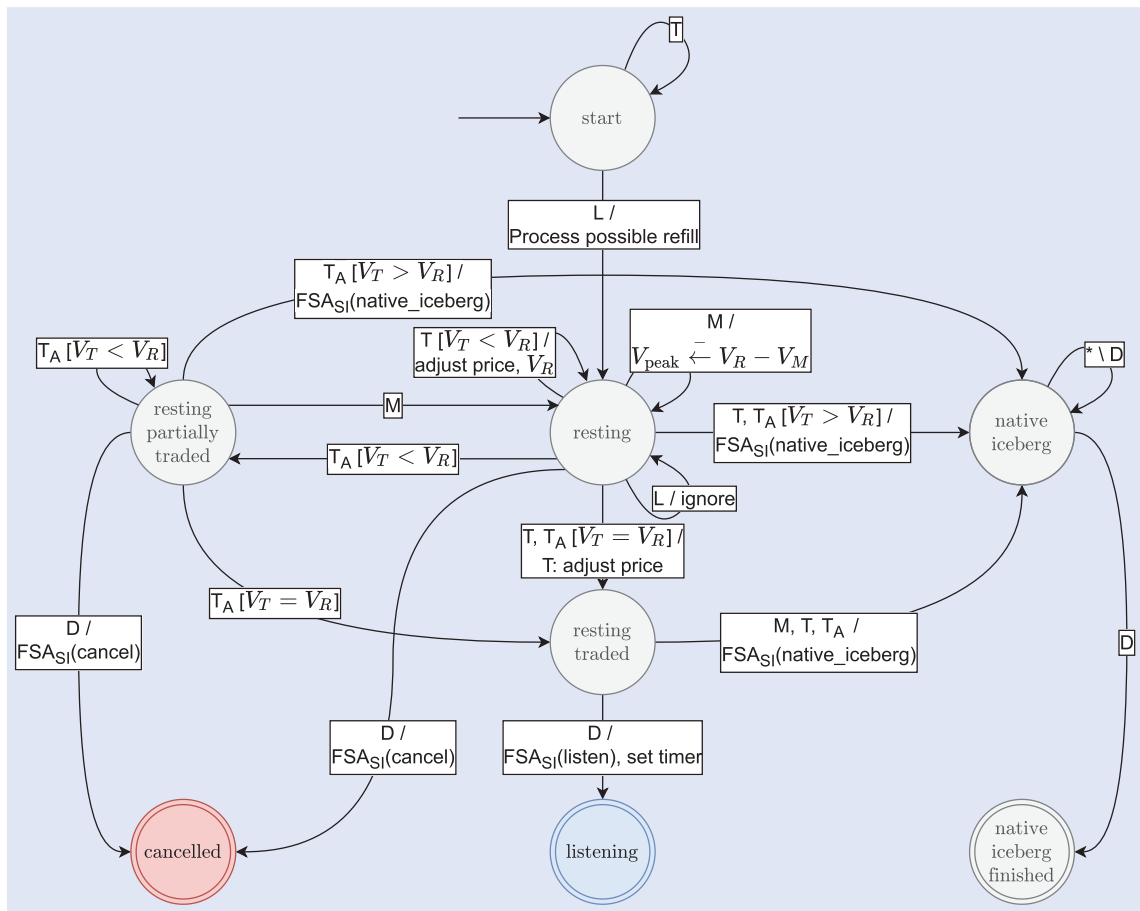


Figure 3. FSA for synthetic iceberg tranches. The nodes correspond to states of an order, the edges—to order actions, see figure 1 for details. $* \setminus D$ stands for ‘all but delete’. $FSA_{SI}(\cdot)$ means ‘change the state of the corresponding iceberg using the synthetic iceberg FSA’ (see figure 4).

We are also interested in computing the following related quantities per each chain $\mathcal{I}^{(\ell)}$:

- The *peak size* V_{peak} is set to be equal to the volume of the initial tranche (initial limit order and, possibly, trade volume);
- The *total volume* V_{total} is simply the sum of the tranche volumes.

A single value per iceberg \mathcal{I} is often required for inference, so V_{total} of different $\mathcal{I}^{(\ell)}$ can be combined into one value later on, if necessary.

Algorithm 2 Process possible refill.

```

procedure PROCESSPOSSIBLEREFILL( $\mathcal{T}$ )
   $\mathcal{I} \leftarrow \emptyset$ 
  for all  $\tilde{\mathcal{T}} \in \text{GETLISTENINGTRANCHES}(\mathcal{T})$  do  $\triangleright$  Going
    from least to most recent
     $\tilde{\mathcal{I}} \leftarrow \text{GETHOLDINGICEBERG}(\tilde{\mathcal{T}})$ 
    if  $\text{TIME}(\text{GETLASTACTION}(\mathcal{T})) -$ 
       $\text{TIME}(\text{GETLASTACTION}(\tilde{\mathcal{T}})) < dt$  then
      if  $\tilde{\mathcal{I}} = \emptyset$  then  $\triangleright$  A listening tranche without a
        holding iceberg
         $\tilde{\mathcal{I}} \leftarrow \text{CREATESYNTHETICICEBERG}(\tilde{\mathcal{T}})$ 
        end if
        if  $\mathcal{I} = \emptyset$  then
           $\mathcal{I} \leftarrow \tilde{\mathcal{I}}$ 
           $\triangleright$  Merge all further icebergs to this one
           $\text{LINKTRANCHE}(\mathcal{I}, \mathcal{T})$ 
        else
           $\text{MERGEICEBERG}(\mathcal{I}, \tilde{\mathcal{I}})$ 
        end if
      else
        if  $\tilde{\mathcal{I}} = \emptyset$  then
           $\text{DELETETRANCHE}(\tilde{\mathcal{T}})$   $\triangleright$  No iceberg
            associated with the tranche; delete it
          else
             $\text{FSASTEPSYNTHETICICEBERG}(\tilde{\mathcal{I}}, \text{complete})$ 
          end if
        end if
      end for
    if  $\mathcal{I} \neq \emptyset$  then
       $\text{FSASTEPSYNTHETICICEBERG}(\mathcal{I}, \text{refill})$ 
    end if
     $\text{DELETETIMERS}(\mathcal{T})$ 
end procedure

```

Finally, suppose the function $\text{FSASTEPSYNTHETICORDER}(\mathcal{T}, \mathcal{A}, dt)$ implements the above logic given a tranche \mathcal{T} , an incoming LOB message \mathcal{A} and a dt value; IDs of the affected and the incoming orders can be retrieved using $\text{AFFECTEDID}(\mathcal{A})$ and $\text{ORDERID}(\mathcal{A})$; the tranches are stored in a pool from which they are retrieved using $\text{GETTRANCHE}(\text{id}_{\mathcal{T}})$ and created using $\text{CREATETRANCHE}(\text{id}_{\mathcal{T}})$. The algorithm 3 summarises the detection step.

Algorithm 3 Detection step for synthetic icebergs.

```

procedure DETECTIONSTEP SYNTHETIC( $\mathcal{A}, dt$ )
  if  $\text{AFFECTEDID}(\mathcal{A}) \neq \emptyset$  then
     $\mathcal{I} \leftarrow \text{GETTRANCHE}(\text{AFFECTEDID}(\mathcal{A}))$ 
    if  $\mathcal{I} \neq \emptyset$  then
       $\text{FSASTEPSYNTHETICORDER}(\mathcal{T}, \mathcal{A}, dt)$   $\triangleright$  What
        happens with the resting order?
    end if
  end if
  if  $\mathcal{I} = \emptyset$  then
     $\mathcal{I} \leftarrow \text{GETTRANCHE}(\text{ORDERID}(\mathcal{A}))$ 
    if  $\mathcal{I} = \emptyset$  then
       $\mathcal{I} \leftarrow \text{CREATETRANCHE}(\text{ORDERID}(\mathcal{A}))$ 
    end if
     $\text{FSASTEPSYNTHETICORDER}(\mathcal{T}, \mathcal{A}, dt)$ 
     $\triangleright$  What happens with the incoming order?
  end if
end procedure

```

4. Learning

4.1. Kaplan–Meier estimation

Having detected sufficiently many iceberg orders, we would like to build a model that yields a prediction of the total iceberg size. Although one may propose various modelling techniques, we elaborate on the model proposed in Christensen and Woodmansey (2013). Namely, for each of P unique detected peak sizes, a distribution of possible total sizes is estimated; then, given the peak size $V_{\text{peak}} = p$ of a previously unseen iceberg, ‘the best’ total volume (in terms of conditional mean, median or mode) is returned as a prediction.

More precisely, from now on let V_p denote a random variable representing the total volume of an iceberg with peak size p . Then for each value of p we are interested in estimating the distribution of V_p . While a trivial empirical distribution might suffice, our experiments show that a significant amount of synthetic icebergs are cancelled before being completely executed (see figure 7). Hence for some icebergs only a lower bound on their total volume is known: for the i -th iceberg, $v_i \geq c_i$, $c_i \in \mathbb{N}$. In a survival analysis framework, these are *censored* observations. Usually survival analysis deals with the so-called ‘time to event data’: the primary interest is the time until the onset of an event for each member of the analysed group. If only upper (or lower, or both) bounds on time, but not exact event times, are known, the observations are considered censored. Instead of discarding these, it is possible to construct estimators that incorporate the uncertainty associated with censoring. In our case, accumulated iceberg volumes play the role of time to event durations, so the task is to estimate the distribution of V_p for each p using random right-censored data.

The proportion of cancelled native icebergs is much smaller, and could probably be disregarded for the purpose of distribution estimation. Nevertheless, we would like to utilise the same approach to simplify the analysis and to make the direct comparison between native and synthetic iceberg estimates possible.

The standard approach for a non-parametric distribution estimation of censored data is to use the *Kaplan–Meier*

estimate (Kaplan and Meier 1958). Let $F_p(v)$ be the cumulative distribution function of V_p , then $S_p(v) = 1 - F_p(v)$ is its survival function. Also define (for the given p)

- u_1, \dots, u_K unique volumes of all detected icebergs sorted in ascending order by size,
- d_j the number of complete icebergs of volume u_j , where $j = 1, \dots, K$,
- n_j the total number of both complete and incomplete icebergs of volumes u_j, \dots, u_K .

Then from the general theory of survival analysis it is known (Kalbfleisch and Prentice 2002) that the maximum likelihood estimate of S_p is

$$\hat{S}_p(v) = \prod_{j: u_j \geq v} \left(1 - \frac{d_j}{n_j}\right). \quad (1)$$

4.2. Weighted Kaplan–Meier estimation for synthetic icebergs

For synthetic icebergs, the estimation (1) cannot be performed directly, because, given a tranche tree, there is no unambiguous way to calculate the total iceberg size. Instead, we propose a weighting scheme that assigns weights to each chain within a tranche tree. Given the i -th tranche tree with h_i chains in total, the weights are

$$w_{i,\ell} = 1/h_i, \quad \ell \in 1 : h_i.$$

So for figure 2 each chain would have $w_{i,\ell} = 1/4$, $\ell \in 1 : 4$, or, equivalently, the contribution from chains of lengths 3 and 5 would be $1/4$, of length 4 – $1/2$. A weight can be also interpreted as a probability of the total iceberg volume being equal to the accumulated tranche chain volume. Then for the purpose of calculating $\hat{S}_p(v)$, instead of d_j, n_j , their weighted counterparts \tilde{d}_j, \tilde{n}_j are computed. Formally, given $\{u_j\}_{j=1,\dots,K}$ as defined above, let $\{v_{i,\ell}\}_{\ell=1,\dots,h_i}$ denote a set of unique volumes of the ℓ th chain of the i th tranche tree. Then

$$\tilde{d}_j = \sum_{i \in C} \sum_{\ell \in H_i} w_{i,\ell}, \quad H_i = \{\ell : v_{i,\ell} = u_j\},$$

where C is a set of indices of all complete icebergs and H_i is a set of tranche chain indices of the i th iceberg, having total volumes equal to u_j . \tilde{n}_j are computed similarly. Of course, when each tranche tree consists of only one chain, all weights are equal to 1 and we have $\tilde{d}_j = d_j$ and $\tilde{n}_j = n_j$. Since V_p only takes discrete values for all p , we finally obtain the weighted estimate

$$\hat{S}_p(u_j) = \prod_{k=1}^j \left(1 - \frac{\tilde{d}_k}{\tilde{n}_k}\right).$$

From \hat{S}_p an estimate of the probability mass function $f_p(u_j) = P(V_p = u_j)$ can be obtained in a trivial way. One notable problem with this estimate is that if $d_K = 0$, then $S_p(u_K) \neq 0$ and the probabilities do not sum up to 1. This is fixed trivially by normalising the probabilities.

4.3. Dirichlet process-based estimation

The utilised Kaplan–Meier estimate is arguably too simplistic. A number of more recent approaches and extensions exist, including a fully Bayesian model introduced by Mangili *et al.* (2015), where a *near-ignorant Dirichlet process* (IDP) is used as a prior to estimate survival curves for right-censored data.

For a detailed description of the estimation procedure, refer to Mangili *et al.* (2015). An implementation as an R package is available and can be used as a direct substitute to the Kaplan–Meier estimate, making it easy to run a direct analysis between these two. A detailed study showed that IDP performed slightly better in several test runs, though the performance gain was inconsistent and not substantial—not exceeding 0.01 for any of the reported classification metrics. For that reason, we do not discuss the evaluation details and only report results obtained with the Kaplan–Meier estimation procedure.

5. Prediction

The prediction step starts from detecting first several tranches of an iceberg. If the peak size p is precisely detected, a prediction of the total volume might be done.

We make predictions in terms of the conditional mean, median and mode. For a fixed iceberg, let v_r denote the currently accumulated volume up to, but not including, tranche number r and let $\mathcal{V}_p = \{u_j : u_j \geq v_r\}_{j=1,\dots,K_p}$ be the constrained optimisation space. Then define

- *mean* prediction based on

$$\begin{aligned} E(V_p | V_p > v_r) &= \frac{1}{P(V_p > v_r)} \sum_{u \in \mathcal{V}_p} u P(V_p = u) \\ &= \left(\sum_{u \in \mathcal{V}_p} f_p(u) \right)^{-1} \sum_{u \in \mathcal{V}_p} u f_p(u) \end{aligned}$$

and defined as

$$\hat{v}^{\text{mean}} = \left(\sum_{u \in \mathcal{V}_p^{(r)}} \hat{f}_p(u) \right)^{-1} \sum_{u \in \mathcal{V}_p} u \hat{f}_p(u),$$

rounded to the nearest integer;

- *median* prediction as

$$\begin{aligned} \hat{v}^{\text{median}} &= \max \left\{ u_j : \sum_{j=1}^J \hat{f}_p(u_j) \leq 0.5, \right. \\ &\quad \left. u_j \in \mathcal{V}_p \ \forall j = 1, \dots, |\mathcal{V}_p| \right\}; \end{aligned}$$

- *mode* prediction as

$$\hat{v}^{\text{mode}} = \operatorname{argmax}_{u \in \mathcal{V}_p} \hat{f}_p(u).$$

For synthetic icebergs, these quantities are computed per chain.

6. Evaluation

Given an estimate of $\hat{f}_p(v)$ and previously unseen data, the model can be evaluated both as a binary classifier and as a regression.

- In a *classification* scenario, for each observed iceberg we try to distinguish between events ‘there is still hidden liquidity’ (positive class) and ‘there is no hidden liquidity left’ (negative class). In the context of synthetic icebergs, the latter means that the iceberg is complete and no more tranches will follow. For native icebergs, it means the last seen tranche can only be traded for the volume not exceeding its currently visible volume, and that no more tranches will follow. Since the full information on a particular iceberg execution is available after we run the prediction algorithm (each iceberg is eventually complete or cancelled), the true total volume is known[†] and hence the classification results can be summarised in a confusion matrix, from which we compute the standard classification metrics: accuracy, precision, recall and F1 score. It is important to underline that we do not try to classify any existing order as being an iceberg or not. We only do so for an order that is marked as a potential candidate at some specific time point, and the prediction is not made prior to that. This explains why the described positive and negative classes contain significantly fewer events than the number of the LOB log messages (thousands, but not millions).
- The *regression* performance metric (MAE) show the degree to which the prediction is different from the true total volume.

In both scenarios, we omit icebergs when $\mathcal{V}_p = \emptyset$ – i.e. if the peak volume p hasn’t been previously seen or cannot be precisely determined, or if the accumulated volume is greater than all total volume estimates at that peak, $v_r > u_j \forall j \in 1 : K_p$.

Naturally, a prediction can be done each time the optimisation space gets smaller—after a trade or a new tranche arrival. To match the case of synthetic icebergs as closely as possible, we decided to evaluate the prediction results of native icebergs only after each new tranche. The metrics defined below are calculated across the whole set of icebergs, so we reintroduce the appropriate indexation.

Let $\hat{v}_{i,r}$ denote the predicted total size at tranche r (where ‘ \cdot ’ can be any of the ‘mean’, ‘median’ or ‘mode’), $v_{i,r}$ —the actual accumulated volume up to and including tranche r and R_i —the set of the i -th iceberg tranches.

Classification. The true class is given by the boolean value $r \neq |R_i|$ —that is, the tranche is not the last one; the class

induced by our prediction is $\hat{v}_{i,r} > v_{i,r}$ —there is some hidden liquidity. Comparing the two, we compose corresponding confusion matrices.

Regression. Let $e_{i,r} = v_i - \hat{v}_{i,r}$, $r \in R_i$ denote the residuals. We use mean absolute error (MAE) as the regression metric computed as

$$\text{MAE} = \frac{1}{|R|} \sum_{i \in C} \sum_{r \in R_i} |e_{i,r}|,$$

$$R = \bigcup_{i \in C} R_i, \quad C - \text{set of complete icebergs.} \quad (2)$$

We decide not to compute other metrics, in particular RMSE, as MAE, being expressed in the original units of data, can be clearly interpreted. In addition, we do not do model comparison and selection for which RMSE would have been more appropriate.

7. Results

We performed the following set of computational experiments. $f_p(v)$ was estimated and the model performance assessed on ESU19 (E-Mini S&P 500 futures contract) FOD LOB log data.

Native icebergs:

- Training period: 3 days from 2019-06-17 16:45:00 CDT, Monday, to 2019-06-20 16:00:00 CDT, Thursday (approx. 19 million records).
- Test period: 1 day from 2019-06-20 16:45:00 CDT, Thursday to 2019-06-21 16:00:00 CDT, Friday (approx. 6 million records).

Synthetic icebergs:

- Training period: 1 day from 2019-06-17 16:45:00 CDT, Monday, to 2019-06-18 16:00:00 CDT, Tuesday.
- Test period: 1 day from 2019-06-18 16:45:00 CDT, Tuesday to 2019-06-19 16:00:00 CDT, Wednesday.

A separate model was fit for each combination of the following parameters:

- Minimum number of tranches per iceberg (min_t): 1, 2, 3 (native); 3, 4, 5 (synthetic).
 - About 28% of all native icebergs had just one tranche and 1 unit of hidden volume, $V_{\text{total}} - V_{\text{peak}} = 1$. Presumably, by doing so market participants do not intend to hide liquidity. Including these icebergs into the model skews the estimation, so we also considered cases with 2, 3 minimum number of tranches per iceberg.
 - For synthetic icebergs, 2 is the minimum technical possible value of tranches. However, because of the inherent noise in the LOB message stream, values 3, 4 and 5 were considered.
- Minimum number of icebergs per estimate (min_i): 1, 2, 5, 10. That is, for a given combination of (p, v) , if the number of icebergs used to estimate

[†] For synthetic icebergs—only in terms of our definition.

$f_p(v)$ was less than a specified value, the estimate was removed entirely. This parameter has the interpretation of ‘certainty’: suppose for a given V_{peak} only one iceberg has been seen in the training sample, having some value of V_{total} . Then, if another iceberg arrives with the same V_{peak} , a guess of its V_{total} would be clearly uncertain. Hence it is fair to compare the performance of different values of the parameter.

Synthetic iceberg detection also depends on the value of dt . For their detection algorithm, which exploits the same idea of iceberg tranches arriving within a short time frame, Christensen and Woodmansey (2013) perform a statistical analysis of market data and propose the value $dt = 0.3$ seconds. After analysing a set of detected synthetic icebergs, we estimated 0.95- and 0.99- quantiles to be equal approximately 0.15 and 0.27, hence the maximum value of $dt = 0.3$ appears sufficient.

The choice of parameters and training intervals may be optimised further, but this falls outside of the scope of this article. Our evidence suggests that it is reasonable to include at least one trading session into the learning phase, thus capturing different order flow regimes throughout the day (see e.g. Bouchaud *et al.* 2018, chapter 4).

The implementation of the model is written in R programming language (R Core Team 2019).

7.1. LOB log statistics

The following figures were produced using the data for the aforementioned period. For synthetic icebergs, the longest chain volume aggregation is used.

Figure 5 summarises the distribution of actions; figure 6 shows the distribution of trade volumes across all four days. The shape of the trade volume distribution (notice the logarithmic scale of the y-axis) could possibly be interpreted as an evidence for iceberg placement, as larger orders are likely to be split into smaller ones.

7.2. Detection results

Figure 7 shows the proportion of completed and cancelled icebergs of both types by minimum number of tranches.

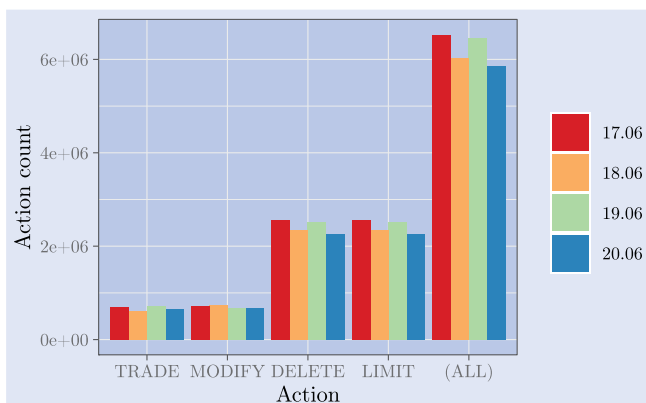


Figure 5. LOB log action distribution.

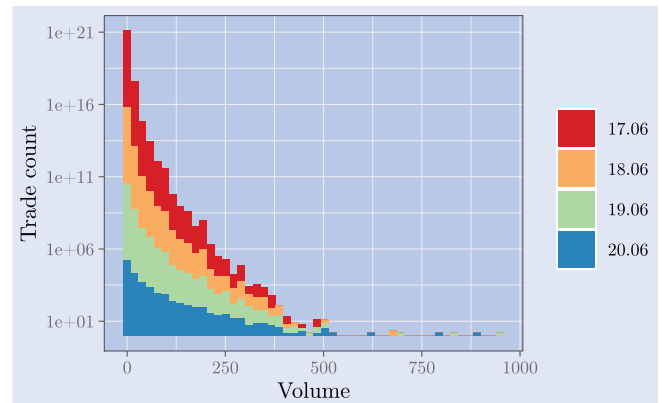


Figure 6. LOB log trade volume distribution (stacked).

The proportion of iceberg orders to all orders on one trading day is shown in figure 8 in terms of both volume and number of orders.

We divide the total volume of all iceberg orders by the total traded volume of all orders (like e.g. Frey and Sandås 2017 do) and not the total daily limit order volume. This ratio makes more sense because only executed icebergs can be detected, which surely constitute only a fraction of all resting hidden volume. We estimate that about 3.8% of all traded volume is contributed by native icebergs, while the volume contributed by synthetic icebergs ranges from 3.3% to 14.3%, depending on the minimum number of tranches. This is in agreement with some of the results reported in the literature as alluded to earlier in section 2.

Moreover, as Fleming *et al.* (2018) note, usually there is no hidden depth, but when it is present, it is substantial. This is especially true for native icebergs, that constitute 0.06% of all orders by number, but about 3.8% by volume; see figure 8.

In addition, the following size-related distributions are estimated:

- Trade volume (figure 9). At least with native icebergs, we confirm the finding of Christensen and Woodmansey (2013) that order sizes to be multiples of 5, like 15, 25, 50 or 100 as can be seen in the right panel—this might be indicative of a human bias.
- Peak volume (figure 10).
- Number of tranches per iceberg (figure 11).

Figure 12 visualises summary statistics related to the distributions of the number of tranches, the peak size and the total volume per order. Note that the total volume of both native and synthetic icebergs is significantly different from the size of all limit orders. Also, the median total volume is, in fact, identical for native and synthetic icebergs (being equal to 6), but the means are different due to some native icebergs having an extremely large size.

Lastly, figure 13 shows the distribution of arrival time differences between subsequent tranche placements. Note that this time delay should not be confused with dt , which is the interval between the trade message and the new tranche placement. Zero values are discarded for the purpose of drawing the

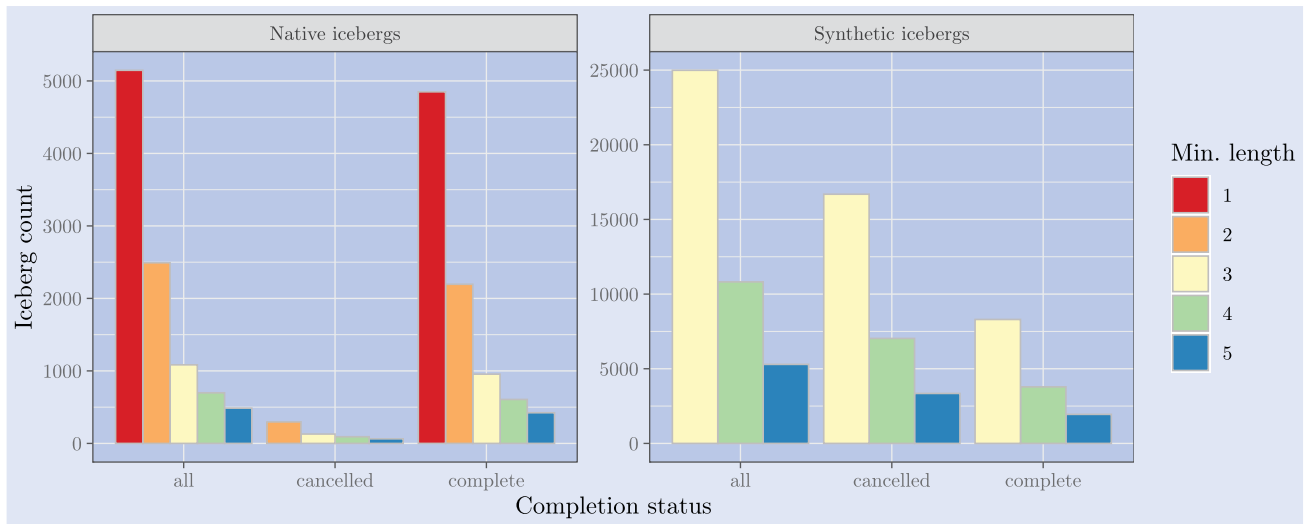


Figure 7. Iceberg completion status distribution by minimum number of tranches on the training sample.

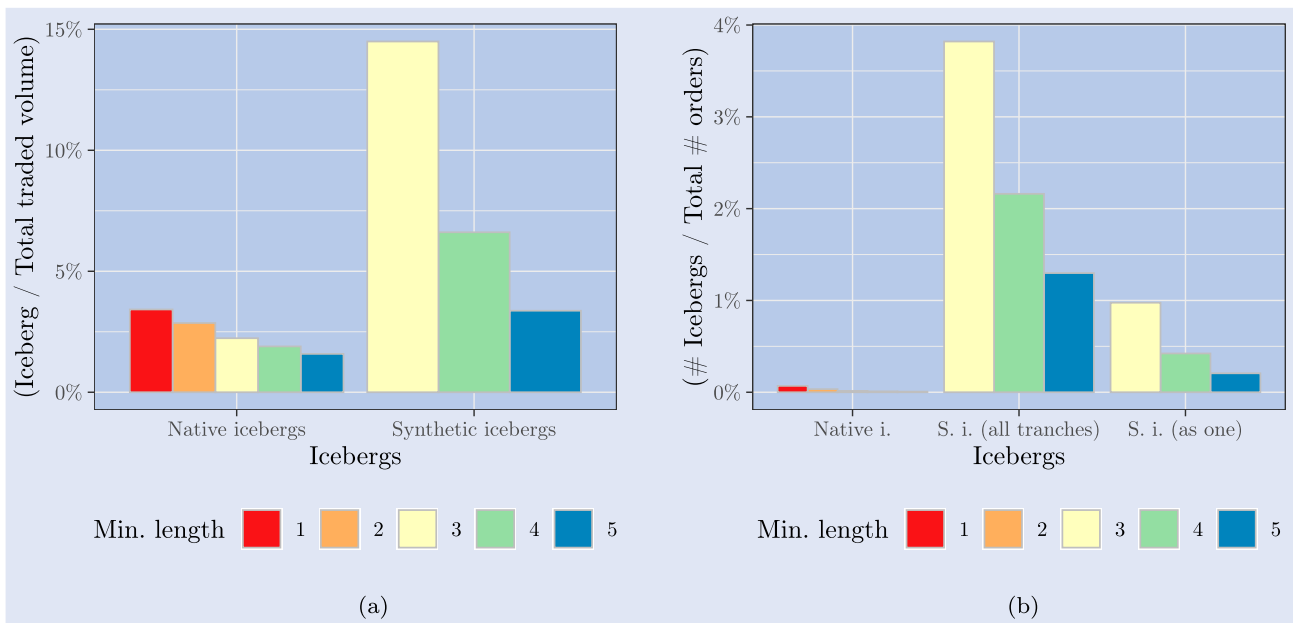


Figure 8. Proportion of iceberg orders to all orders on the training sample. (a) By volume and (b) By number of orders. Synthetic iceberg order tranches are counted either as one order, ‘s. i. (as one)’, or separate orders, ‘s. i. (all tranches)’.

plot, but they amount to 4.71%[†] and 38.94% of all values for synthetic and native icebergs, correspondingly. If the initial tranche is not considered, then it can be seen that the majority of tranches arrive less than one second after the previous tranche (before being traded). This suggests that the proposed detection algorithm is more suitable as an input to other trading algorithms, rather than a signal to a day trader, who would not be able to react sufficiently fast.

7.3. Prediction results

7.3.1. Native icebergs. We observed that min_t parameter influences results the most, i.e. for more than 10 percentage points for most of the classification metrics. For

$\text{min_t} = 1$ the accuracy plunges to about 55%, so we conclude that it is sensible not to include icebergs consisting of only one tranche into the model. On the contrary, min_i parameter influences the results for no more than 5 percentage points. Thus in the table 4 we list the metric values for $\text{min_t} \in \{2, 3\}$ separately, and summarise them as min-max ranges for $\text{min_i} \in \{1, 3, 5, 10\}$. Gains in the performance are associated with losses in the number of evaluations. Apparently, the loss is disproportional to the gain in some cases (40 p.p. v 5 p.p.), so in general, smaller values of min_t are preferable.

Table 5 provides a confusion matrix that summarises the performance of the model with parameters $\text{min_t} = 3$, $\text{min_i} = 1$. Note a mild (75–25) class imbalance. The ‘mode’ distribution averaging seems preferable, although perhaps not by a large margin.

Overall, the algorithm demonstrates a good performance as indicated by the $\approx .85$ F1 score and $\approx .75$ accuracy, although

[†] The fact that we observe zero delays for synthetic icebergs may be attributed to an insufficient accuracy of time records (millisecond resolution).

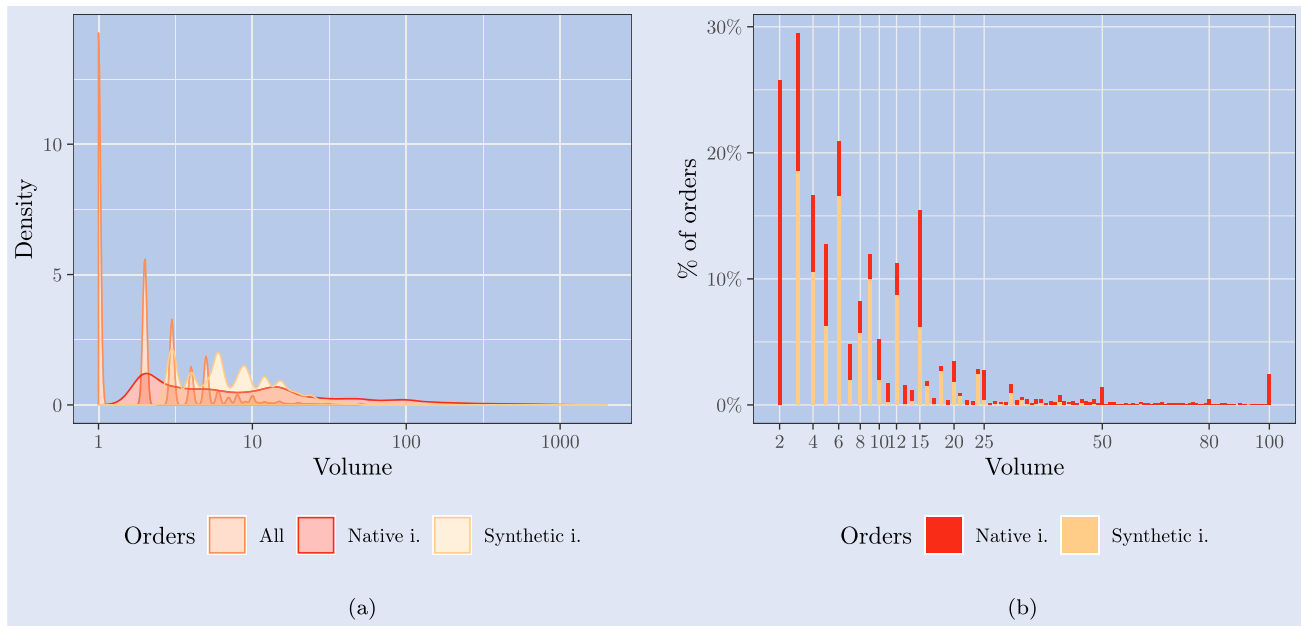


Figure 9. Order size distribution. (a) Density and (b) Individual probabilities (cutoff at 100 units).

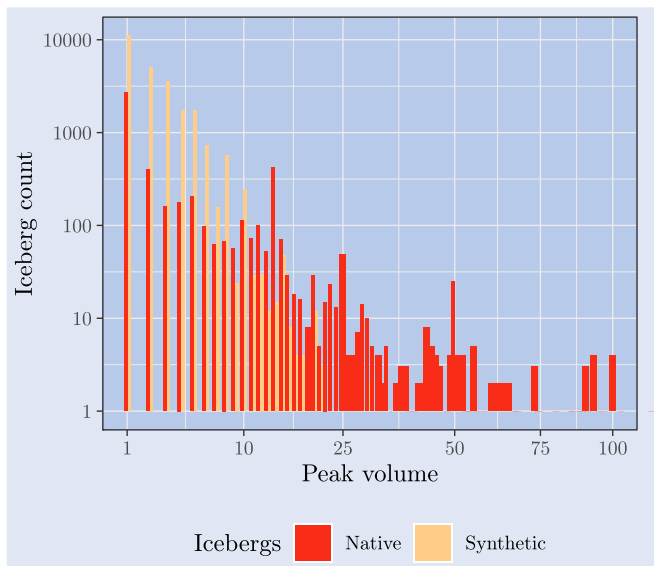


Figure 10. Peak volume distribution.

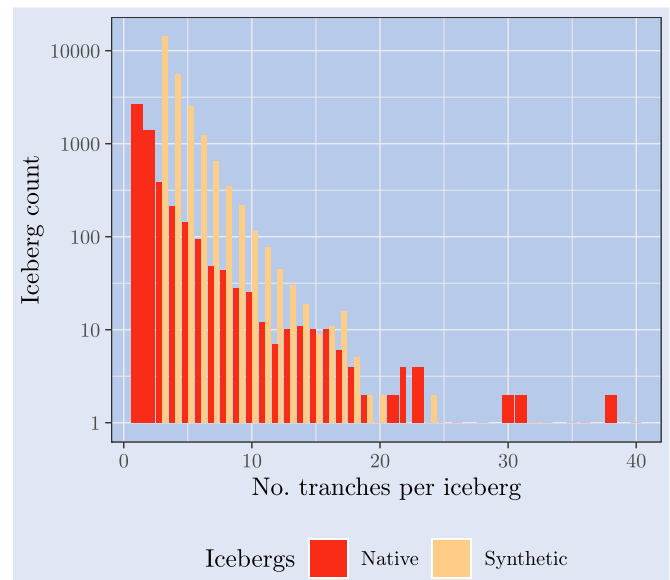


Figure 11. Number of tranches distribution.

it is mainly contributed by the true positives (the prediction that the iceberg is not complete).

7.3.2. Synthetic icebergs. As mentioned previously, a detected synthetic iceberg, being a tranche tree, requires an aggregation across all possible tranche chains. For the purpose of evaluation we take the longest chain of each iceberg. Other ways of averaging might provide better results—e.g. if a ‘voting’ among chains is performed and the value with the maximum number of supporting chains is chosen.

Similarly to native icebergs, we noted that min_t influences the results more than min_i , however, the difference is small. In the latter case it is negligible, so only one number is reported per metric in the table 6. A confusion matrix of

the model with parameters $\text{min_t} = 4$, $\text{min_i} = 5$ is provided in the table 7. Comparing the two, we note that the superior performance of the ‘mean’ version is mainly contributed by the large number of true positives and there are almost no true negatives. In contrast, for the ‘mode’ version the number of true negatives is significantly larger at the cost of having more false positives and negatives. Also note that there is virtually no class imbalance.

Overall, the performance is fair, and noticeably worse than in the case of native icebergs. This is not surprising, given the aforementioned conceptual difficulties associated with synthetic icebergs. Nevertheless, the model demonstrates useful predictive ability even in such circumstances.

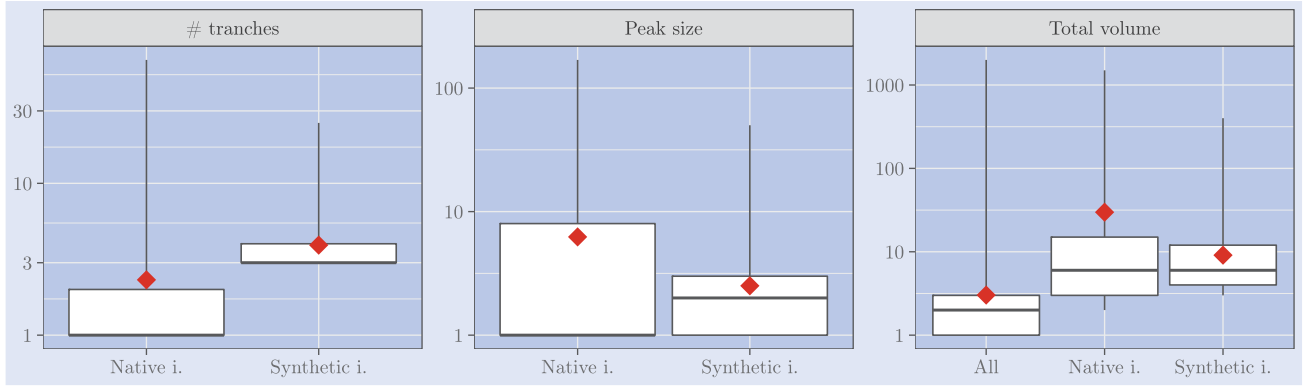


Figure 12. Summary of the distributions of the number of tranches per iceberg, the peak size and the total volume per order. The lower and upper hinges correspond to the first and third quartiles. The whiskers extend from the lower / upper hinge to the minimum / maximum value, respectively. The middle bar is the median, while the red diamond dot is the mean.

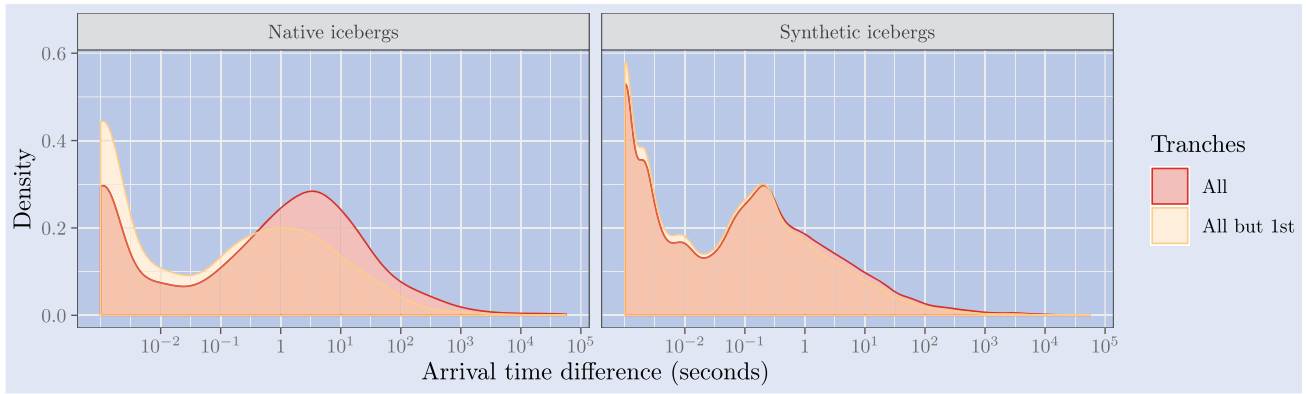


Figure 13. Tranche arrival time difference distributions (for values strictly greater than zero). It is instructive to compare two cases: when the initial tranche is included into and excluded from consideration—it might take longer time to execute the first tranche of an iceberg after its initial placement, but the following tranches get traded more rapidly.

Table 4. Evaluation metrics for native icebergs. The ‘Skipped’ row shows the proportion of all icebergs used for the evaluation *after* filtering out icebergs shorter than. The ‘Evaluated’ row shows the number of iceberg tranches used for the evaluation.

min_t		Mean	Median	Mode
3	F1 score	0.84–0.86	0.77–0.80	0.82–0.86
	Accuracy	0.73–0.77	0.65–0.70	0.72–0.77
	Precision	0.75–0.78	0.78–0.79	0.79–0.82
	Recall	0.96–0.99	0.76–0.82	0.85–0.89
	Skipped	25–60%	18–60%	18–60%
	Evaluated	510–973	510–1062	510–1062
2	F1 score	0.75–0.79	0.66–0.71	0.71–0.78
	Accuracy	0.63–0.66	0.55–0.62	0.68–0.71
	Precision	0.62–0.66	0.65–0.70	0.75–0.77
	Recall	0.95–0.98	0.68–0.71	0.67–0.80
	Skipped	18–50%	13–50%	13–50%
	Evaluated	948–1537	948–1632	948–1632

7.3.3. Residual analysis. It might be instructive to check the magnitude of the prediction error, as sometimes the classification fails because $V_{\text{total}} \neq \hat{V}_{\text{total}}$, but the difference of the two is not, in fact, too big. Simply computing MAE for the ‘best’ models of the section 7.3 (‘mode’ averaging) gives 117 and 2.1 for native and synthetic icebergs, correspondingly. However, examining the residual densities (see figure 14), we see that the residuals are centred at 0, and the large MAE value is contributed by rare outliers of outstandingly

large volume. The 0.9-quantile of the native iceberg residuals (‘mode’) is 494; computing MAE on the subset of data with $e_{\text{mode}}^{\text{mode}} \leq 494$ reduced it by almost a half to 68. The same data for synthetic icebergs shows that the ‘mean’, ‘median’ predictions are biased and constantly overestimate V_{total} , and so the ‘mode’ averaging is preferable.

We conclude that the regression approach is valuable for synthetic, but has limited applicability for native icebergs, unless we filter out rare events or increase the learning interval to capture more of them.

8. Discussion and future work

We have proposed an algorithm for detection and prediction of both native and synthetic iceberg orders on the CME. It is equally well suited for both streaming and pre-recorded data. The quantitative estimates of the observed hidden volume are new, and they are in agreement with what one may find in the existing literature.

8.1. Detection

Detection of native icebergs is straightforward as the information disseminated by the exchange is sufficient to reliably determine the sequence of tranches that constitute an iceberg

Table 5. Confusion matrix for native icebergs.

	Mean		Median		Mode	
	Actual incomplete	Actual complete	Actual incomplete	Actual complete	Actual incomplete	Actual complete
Predicted incomplete	735 (76%)	226 (23%)	631 (59%)	175 (16%)	736 (69%)	159 (15%)
Predicted complete	9 (1%)	3 (0%)	195 (18%)	61 (6%)	90 (8%)	77 (7%)

Table 6. Evaluation metrics for synthetic icebergs.

min_t		Mean	Median	Mode
4	F1 score	0.70	0.68	0.62
	Accuracy	0.54	0.53	0.54
	Precision	0.54	0.54	0.56
	Recall	0.10	0.91	0.68
	Skipped	0%	0%	0%
	Evaluated	7660	7678	7678
5	F1 score	0.72	0.69	0.58
	Accuracy	0.57	0.54	0.51
	Precision	0.57	0.56	0.57
	Recall	0.99	0.90	0.59
	Skipped	1%	0%	0%
	Evaluated	4149	4165	4165

order. However, precise details of the unambiguous detection are not widely known. We aim to address this by formalising the state transition mechanics.

On the other hand, detecting synthetic icebergs is conceptually more complicated and can only be attempted by relying on various heuristics. We introduce a formal notion of a synthetic iceberg as a sequence of events that closely resembles how an ISV would operate. An attempt to make a clear distinction between iceberg types is, in our opinion, a key factor in modern day studies of hidden liquidity. The presented definition and the detection algorithm, though

involving a number of inherent limitations, allow us to extend the quantitative assessment of LOB dynamics. That being said, for an end user interested in drawing conclusions from the current state of the book, it may not matter whether every detected limit order sequence is indeed a formal synthetic iceberg. Generally speaking, a repeatable *order flow pattern* gets detected, for which a satisfactory inference can be made. This information can in turn be used as an input to trading algorithms.

8.2. Learning and prediction

For a learning phase, we extend the approach of Christensen and Woodmansey (2013), by accounting for the fact that many icebergs get cancelled before being fully executed. By employing methods from survival analysis, we obtain very similar results for the Kaplan-Meier and the Dirichlet Process-based estimation. We try to avoid overfitting by carefully picking possible parameter values, and report ranges for classification metrics for a set of conducted numerical experiments. Our evidence suggests that the performance of the model is fairly robust.

The overall performance of the model classification varies from fair to decent, with F1 score around 0.6 and 0.85 for synthetic and native icebergs, correspondingly.

There is much space for improvement of the learning and prediction procedures. For native icebergs a variety of models

Table 7. Confusion matrices for synthetic icebergs.

	Mean		Median		Mode	
	Actual incomplete	Actual complete	Actual incomplete	Actual complete	Actual incomplete	Actual complete
Predicted incomplete	4145 (54%)	3481 (45%)	3798 (49%)	3239 (42%)	2842 (37%)	2203 (29%)
Predicted complete	16 (0%)	18 (0%)	372 (5%)	269 (4%)	1328 (17%)	1305 (17%)

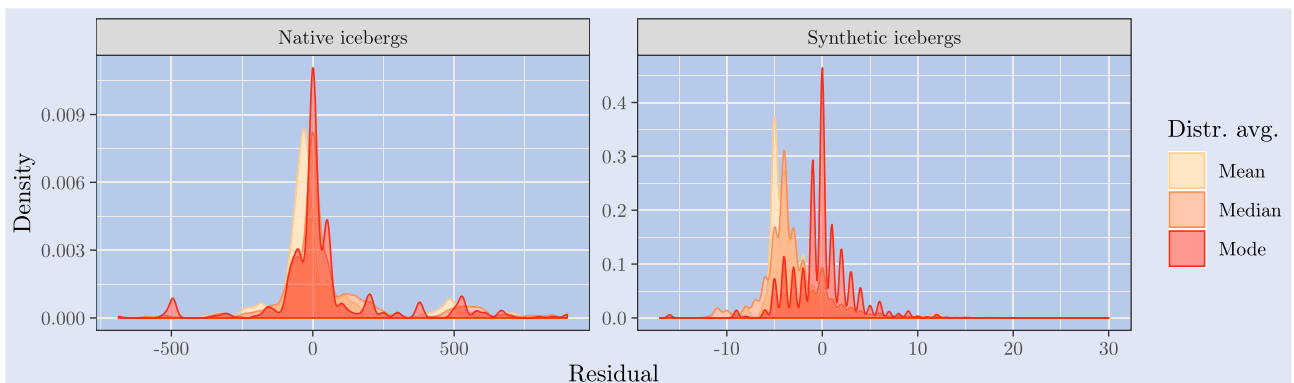


Figure 14. Residual densities for the 'best' native and synthetic iceberg models from section 7.3.

are available as we are not restricted to the methods of survival analysis. For synthetic icebergs a better choice would have been to utilise a semi-parametric relative risk (Cox) model and include covariates into the analysis, which would make the prediction more accurate, see e.g. Kalbfleisch and Prentice (2002). Relaxing our assumptions about synthetic icebergs, though conceptually challenging, may also prove insightful.

Finally, our algorithm may be adapted to use real-time data as an input stream. In this case, the model may be additionally improved to use constant updating of the mass function $f_p(u_j)$, as well as its preservation between consecutive trading sessions. We anticipate a further performance improvement and hope to explore these and other opportunities in our upcoming research on the topic.

Acknowledgments

The authors are grateful to the anonymous referees, whose remarks led to numerous improvements in the model performance. The would like to thank Anton Korenkov of dxFeed Solutions DE GmbH for helping us to obtain and process the CME order log data; Sergey Titov, Devexperts for his valuable comments on the native iceberg detection algorithm; and Olga Egorova, University of Southampton for reviewing a draft version of the paper.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Dmitry Zotikov  <http://orcid.org/0000-0002-2365-0634>

Anton Antonov  <http://orcid.org/0000-0002-9848-8854>

References

- Bouchaud, J.P., Bonart, J., Donier, J. and Gould, M., *Trades, Quotes and Prices: Financial Markets Under the Microscope*, 2018 (Cambridge University Press: Cambridge).
- Christensen, H. and Woodmansey, R., Prediction of hidden liquidity in the limit order book of GLOBEX futures. *J. Trading*, 2013, **8**, 68–95.
- CME, GCC Product Resources. 2019a. Available online at: <https://www.cmegroup.com/confluence/display/EPICSANDBOX/GCC+Product+Resources> (accessed 15 July 2019).
- CME, Market by Order (MBO). 2019b. Available online at: <https://www.cmegroup.com/education/market-by-order-mbo.html> (accessed 15 July 2019).
- CME, MDP 3.0 - Market by Order - Book Management. 2019c. Available online at: <https://www.cmegroup.com/confluence/display/EPICSANDBOX/MDP+3.0+-+Market+by+Order+-+Book+Management> (accessed 15 July 2019).
- CME, MDP 3.0 - Trade Summary. 2019d. Available online at: <https://www.cmegroup.com/confluence/display/EPICSANDBOX/MDP+3.0+-+Trade+Summary> (accessed 15 July 2019).
- Fleming, M., Mizrahi, B. and Nguyen, G., The microstructure of a US treasury ECN: The brokerTec platform. *J. Financ. Markets*, 2018, **40**, 2–22.
- Frey, S. and Sandås, P., The impact of iceberg orders in limit order books. *Quart. J. Finance*, 2017, **7**, 1750007.
- Hautsch, N. and Huang, R., A statistical model for detecting hidden liquidity. 2010. Available online at: https://www.wiwi.hu-berlin.de/de/forschung/irtg/lvb/research/veranstaltungen/Hejnice2010/talks2010/Ruihong_Huang.
- Kalbfleisch, J. and Prentice, R., *The Statistical Analysis of Failure Time Data (Wiley Series in Probability and Statistics)*, 2nd ed., 2002 (Wiley-Interscience: New York).
- Kaplan, E.L. and Meier, P., Nonparametric estimation from incomplete Observations. *J. Am. Stat. Assoc.*, 1958, **53**, 457–481.
- Mangili, F., Benavoli, A., de Campos, C.P. and Zaffalon, M., Reliable survival analysis based on the Dirichlet process. *Biom. J.*, 2015, **57**, 1002–1019.
- Moro, E., Vicente, J., Moyano, L.G., Gerig, A., Farmer, J.D., Vaglica, G., Lillo, F. and Mantegna, R.N., Market impact and trading profile of hidden orders in stock markets. *Phys. Rev. E. Stat. Nonlin. Soft. Matter. Phys.*, 2009, **80**, 066102.
- R Core Team, R: A Language and Environment for Statistical Computing. 2019. Available online at: <https://www.R-project.org/>,