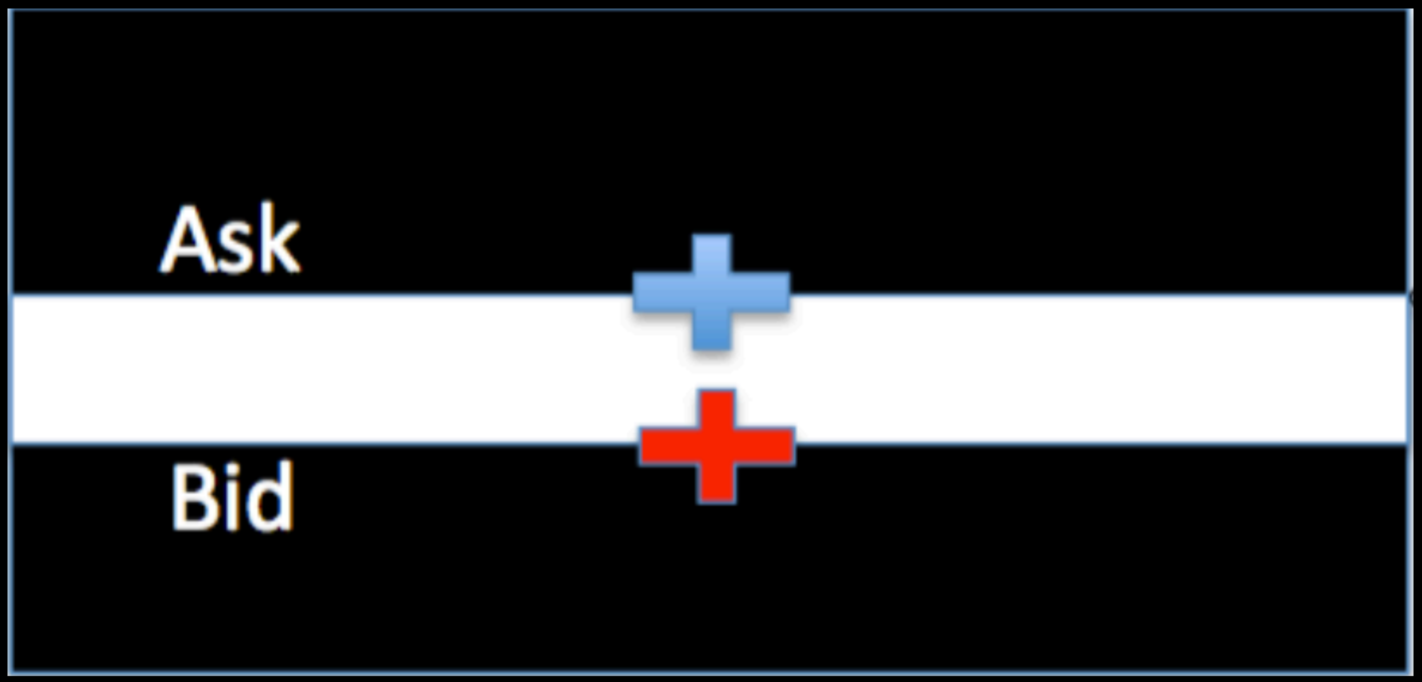
> Shall we play a game?

[1] HFT Market Making
[2] HFT Market Making with ML
[3] HFT Market Making with Reinforcement ML

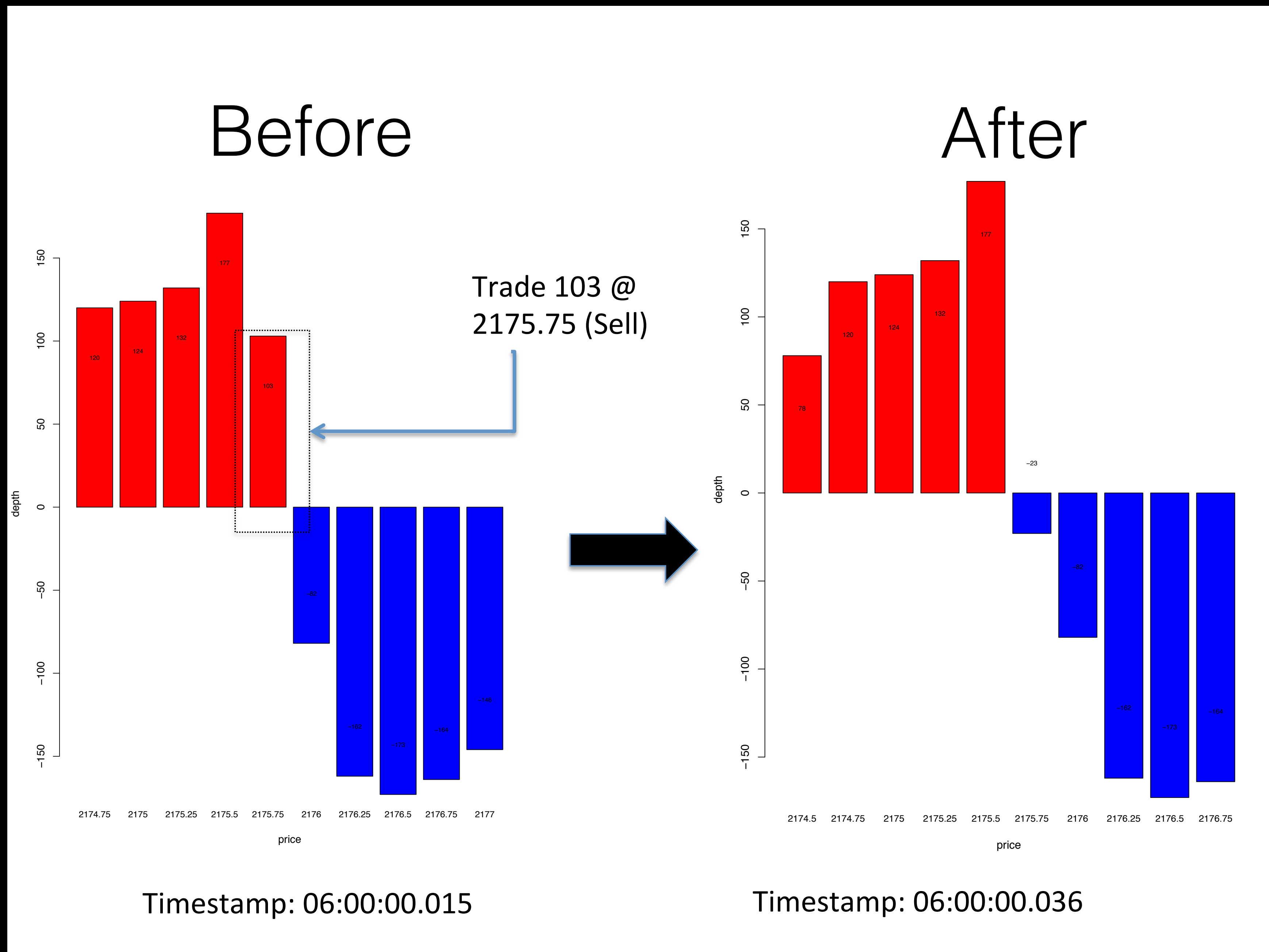
>1

> Where to place limit orders?



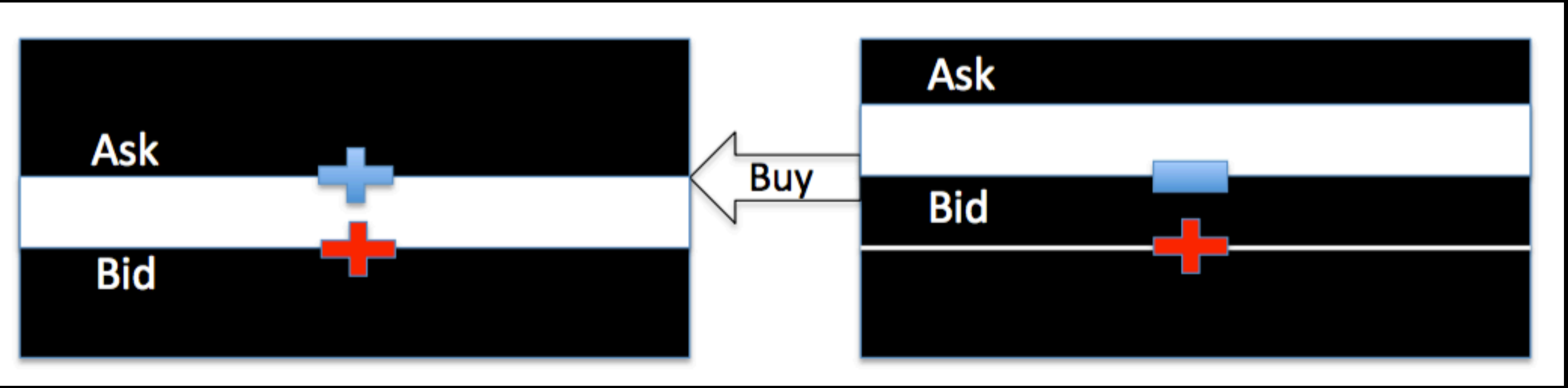
Goal: capture the bid-ask spread

A Limit Order Book



:::no_md_entries - num_groups: 1
::::md_entry_px: 9847.0 md_entry_size: None (2147483647) security_id: 607185 [ZQM8] rpt_seq: 27
trading_reference_date: 16771 settl_price_type: Rounded (4) md_update_action: New (0) md_entry_type: Settlement Price (6)
:packet - timestamp: 2015-12-02 14:15:35.121671 sequence_number: 19448892 sending_time: 1449094535121606602
::MDIncrementalRefreshDailyStatistics - transact_time: 1449094535121000000 match_event_indicator: EndOfEvent (128)

> place bid @ level 1
> place ask @ level 1



☒Marker orders move the book
☒Queue position is determining factor in a fill (i.e. success)
☒Need low latency programmers and FPGAs to win this game

> menu

> Shall we play a game?

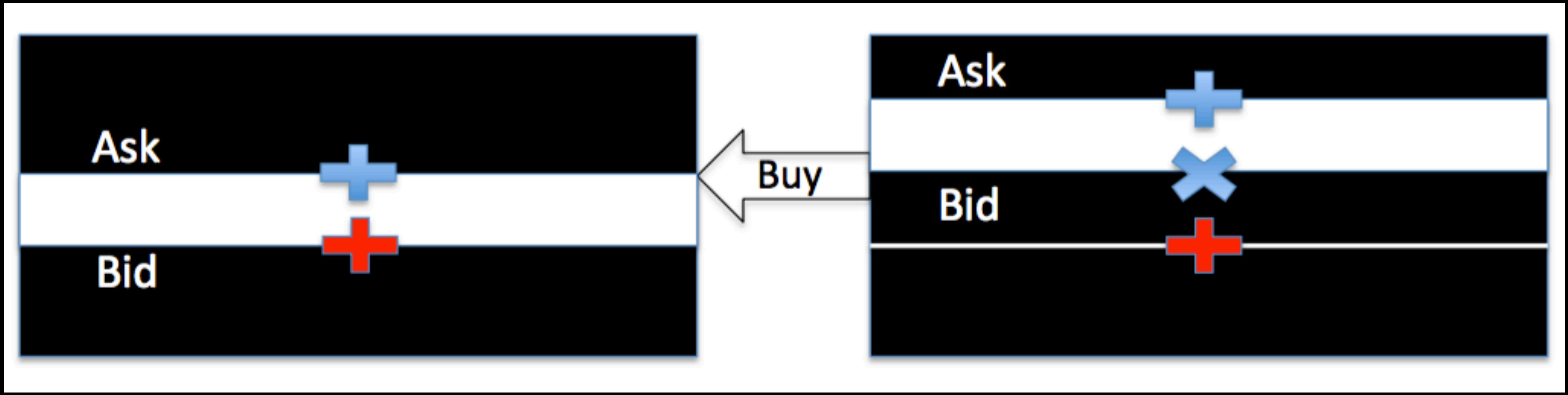
[1] HFT Market Making
[2] HFT Market Making with ML
[3] HFT Market Making with Reinforcement ML

>2

> place bid @ level 1
> place ask @ level 1

. . .

> cancel and replace ask @ level 2



☒Need to predict ahead

Preliminaries: Problem Formulation

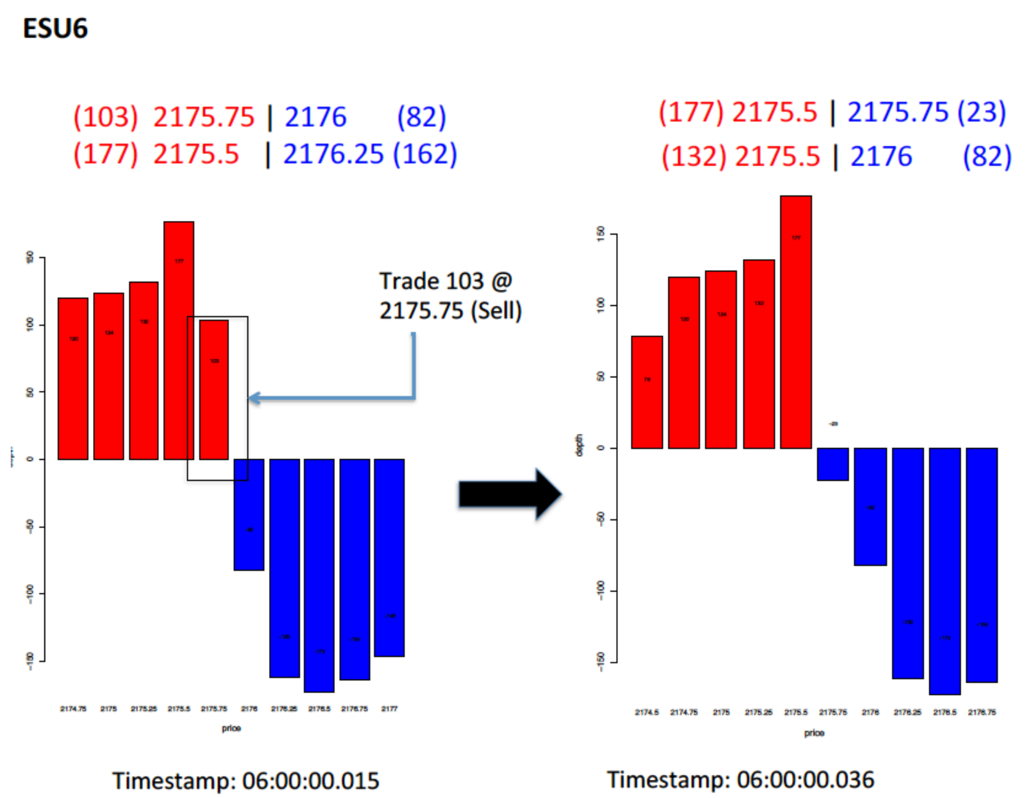


Figure: An exemplary sequence of limit order book updates in the ES futures market (ESU6) is shown before and after the arrival of a sell market order.

time	$\mathcal{X}_t^1$	M_t^s	$L_t^{a,1}$
t_0	$(2175.75, 2176.0, 103, 82)$	0	0
t_1	$(2175.5, 2176.0, 177, 82)$	103	0
t_2	$(2175.5, 2175.75, 177, 23)$	0	23

Table: The limit order book of ESU6 before and after the arrival of the sell aggressor.

Notation

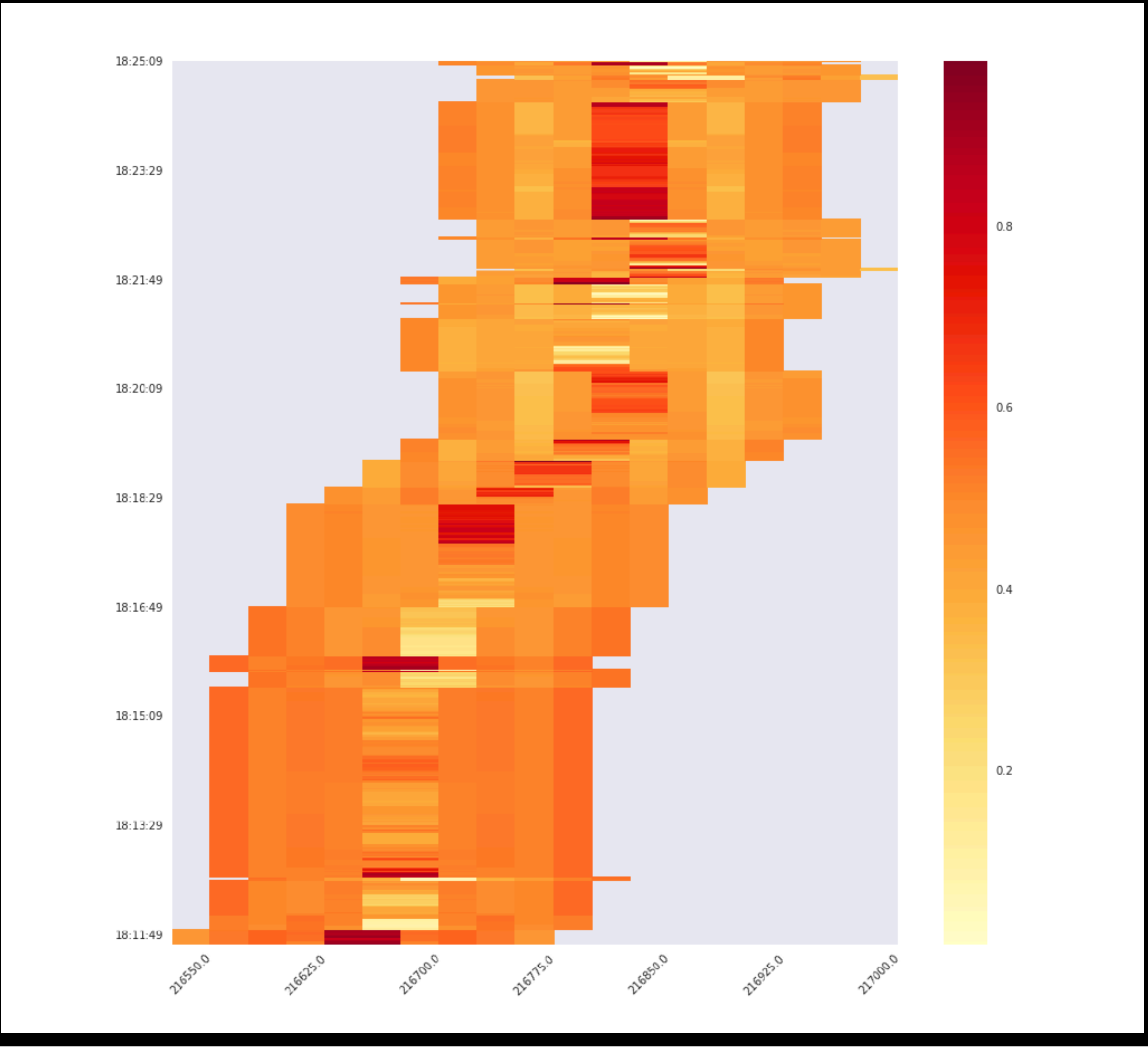
- LOB state: $\mathcal{X}_t := (\mathbf{s}_t^b, \mathbf{s}_t^a, \mathbf{q}_t^b, \mathbf{q}_t^a)$
- Limit orders:
  $\mathbf{L}_t^b := (L_t^{b,1}, \ldots, L_t^{b,n})$,
  $\mathbf{L}_t^a := (L_t^{a,1}, \ldots, L_t^{a,n})$
- Market orders: M_t^b and M_t^s ('aggressors')
- Cancelations:
  $\mathbf{C}_t^b := (C_t^{b,1}, \ldots, C_t^{b,n})$,
  $\mathbf{C}_t^a := (C_t^{a,1}, \ldots, C_t^{a,n})$

Exemplary sequence of events

		Ω_T^1						
time	$\mathcal{X}_0^1$	M_T^s	M_T^b	$C_T^{b,1}$	$C_T^{a,1}$	$L_T^{b,1}$	$L_T^{a,1}$	$\mathcal{X}_t^1$
t_0^-	$(2175.75, 2176.0, 102, 82)$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$(2175.75, 2176.0, 102, 82)$
t_0	$(2175.75, 2176.0, 102, 82)$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{1\}$	$\{\}$	$(2175.75, 2176.0, 103, 82)$
t_1	$(2175.75, 2176.0, 102, 82)$	$\{103\}$	$\{\}$	$\{\}$	$\{\}$	$\{1\}$	$\{\}$	$(2175.5, 2176.0, 177, 82)$
t_2	$(2175.75, 2176.0, 102, 82)$	$\{103\}$	$\{\}$	$\{\}$	$\{\}$	$\{1\}$	$\{23\}$	$(2175.5, 2175.75, 177, 23)$

Table: *The state of the top of the top-of-the-book $\mathcal{X}_t^1$ is updated by data*

Visualizing the flow of information in the limit order book*



*Dixon, Polson and Sokolov, Deep Learning for Spatio-Temporal Modeling: Dynamic Traffic Flows and High Frequency Trading, Applied Stochastic Methods in Business and Industry, 2018.

Timestamp	$s_t^{b,1}$	$s_t^{b,2}$	...	$q_t^{b,1}$	$q_t^{b,2}$	...	$s_t^{a,1}$	$s_t^{a,2}$	...	$q_t^{a,1}$	$q_t^{a,2}$	...	Y_t
06:00:00.015	2175.75	2175.5	...	103	177	...	2176	2176.25	...	82	162	...	-1
06:00:00.036	2175.5	2175.25	...	177	132	...	2175.75	2176	...	23	82	...	0

The Price Impact of Order Flow

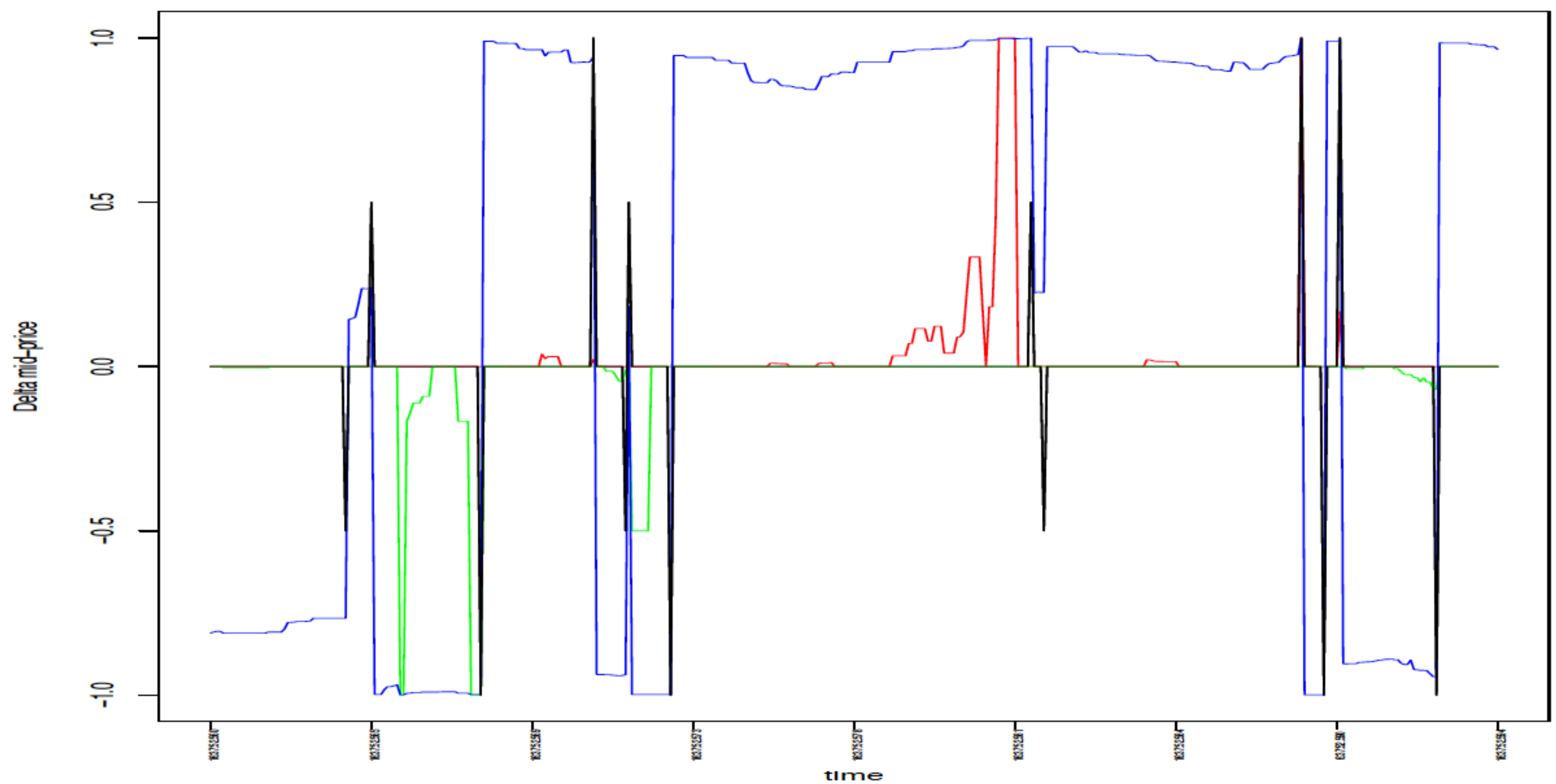
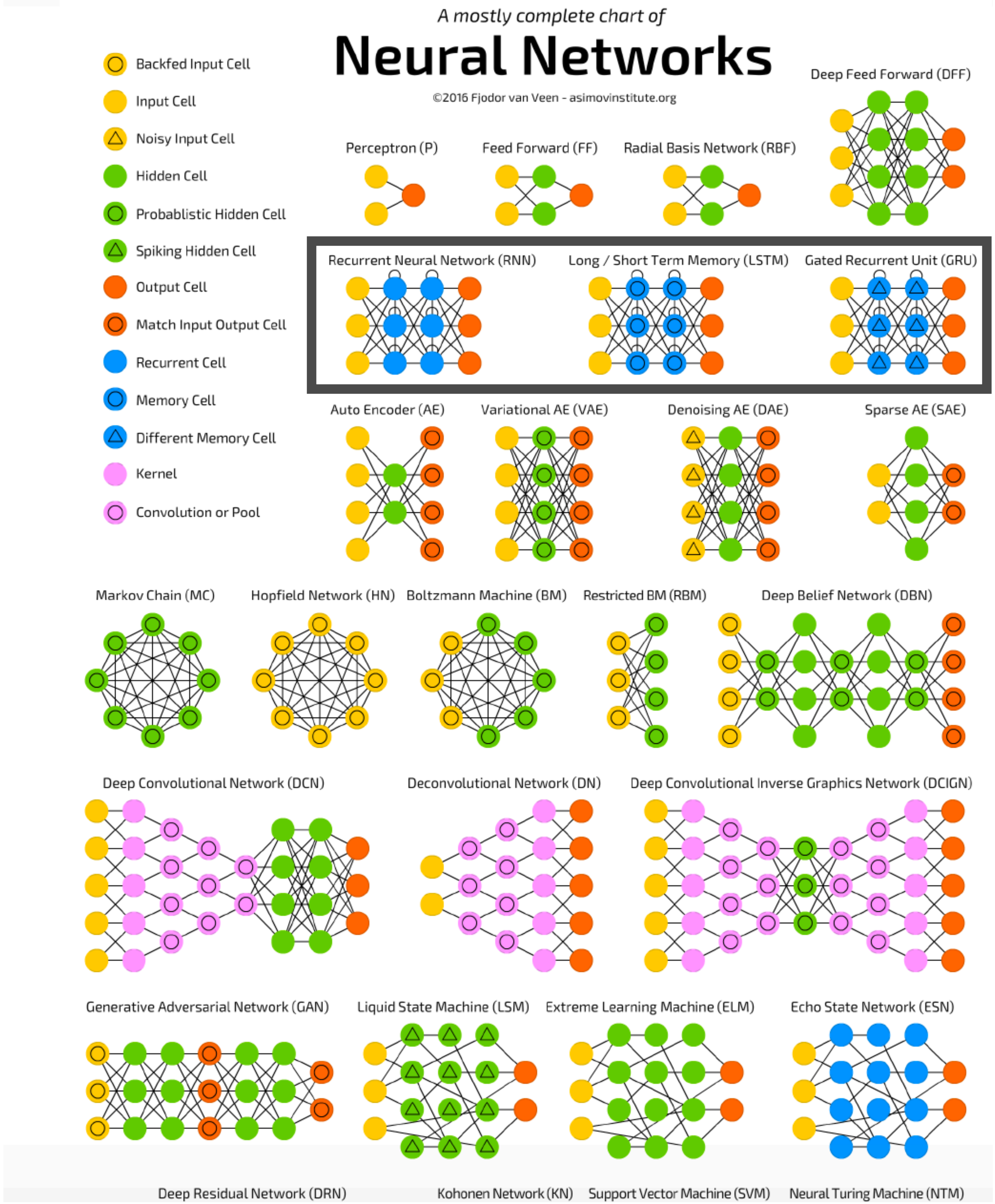


Figure: The black line represents the observed change in mid-price over a 34 milli-second period from 16:37:52.560 to 16:37:52.594. The liquidity imbalance (blue), scaled here to the $[-1, 1]$ interval, although useful in predicting the direction of the next occurring price change, is generally a poor choice for predicting when the price change will occur. The order flow is a better predictor of next-event price movement, although is difficult to interpret when either of the buy (red) and sell order flows (green) are small.


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

Prediction Problem

- The response is

$$Y_t = \Delta p^t_{t+h} \tag{1}$$

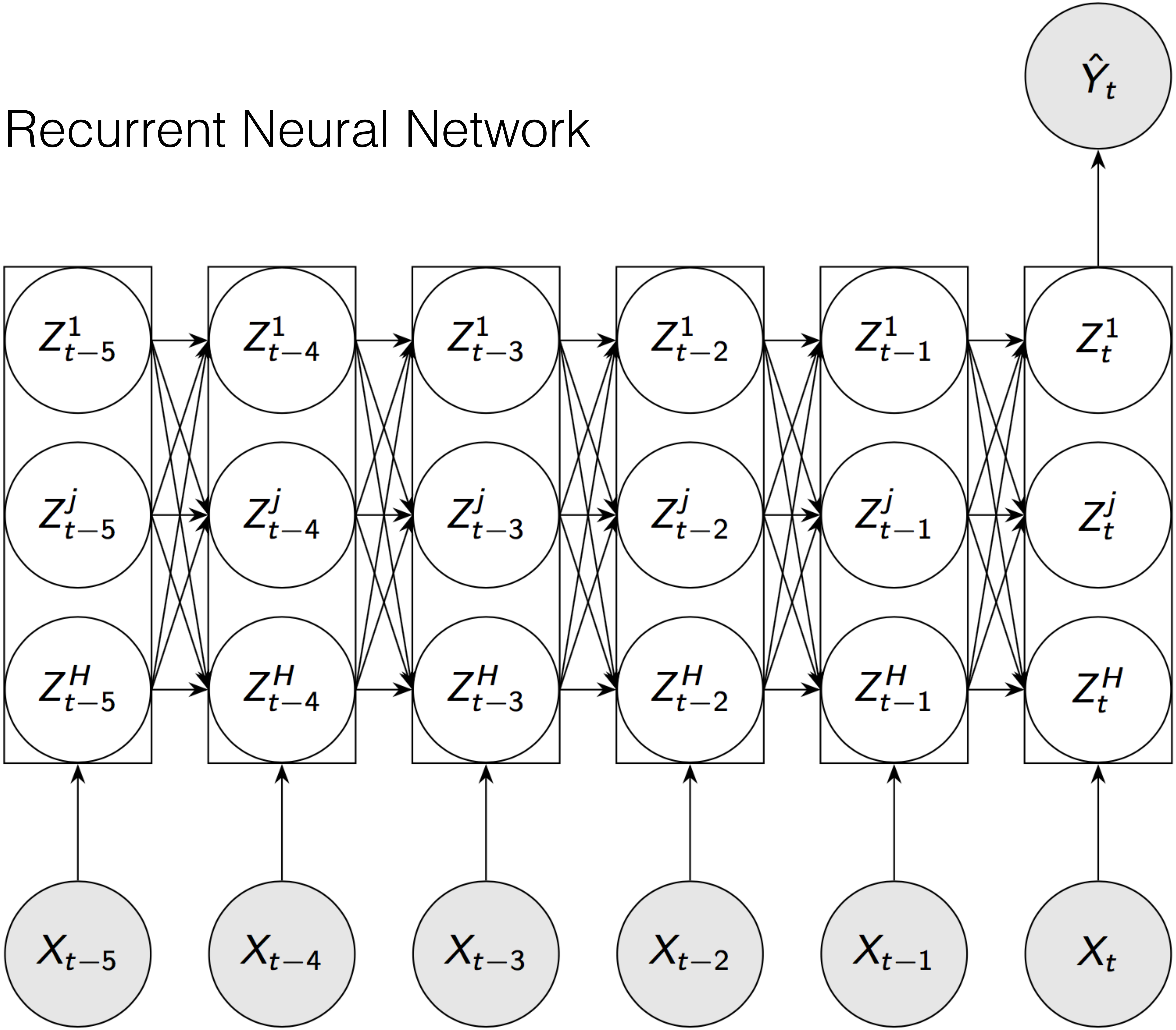
- Δp^t_{t+h} is the forecast of discrete mid-price changes from time t to $t+h$, given measurement of the predictors up to time t.
- The predictors are embedded

$$x = x^t = \text{vec}\begin{pmatrix} x_{1,t-k} & \dots & x_{1,t} \\ \vdots & & \vdots \\ x_{n,t-k} & \dots & x_{n,t} \end{pmatrix} \tag{2}$$

- n is the number of quoted price levels, k is the number of lagged observations, and $x_{i,t} \in [0,1]$ is the relative depth, representing liquidity imbalance, at quote level i:

$$x_{i,t} = \frac{q^{a,i}_t}{q^{a,i}_t + q^{b,i}_t}. \tag{3}$$

Recurrent Neural Network



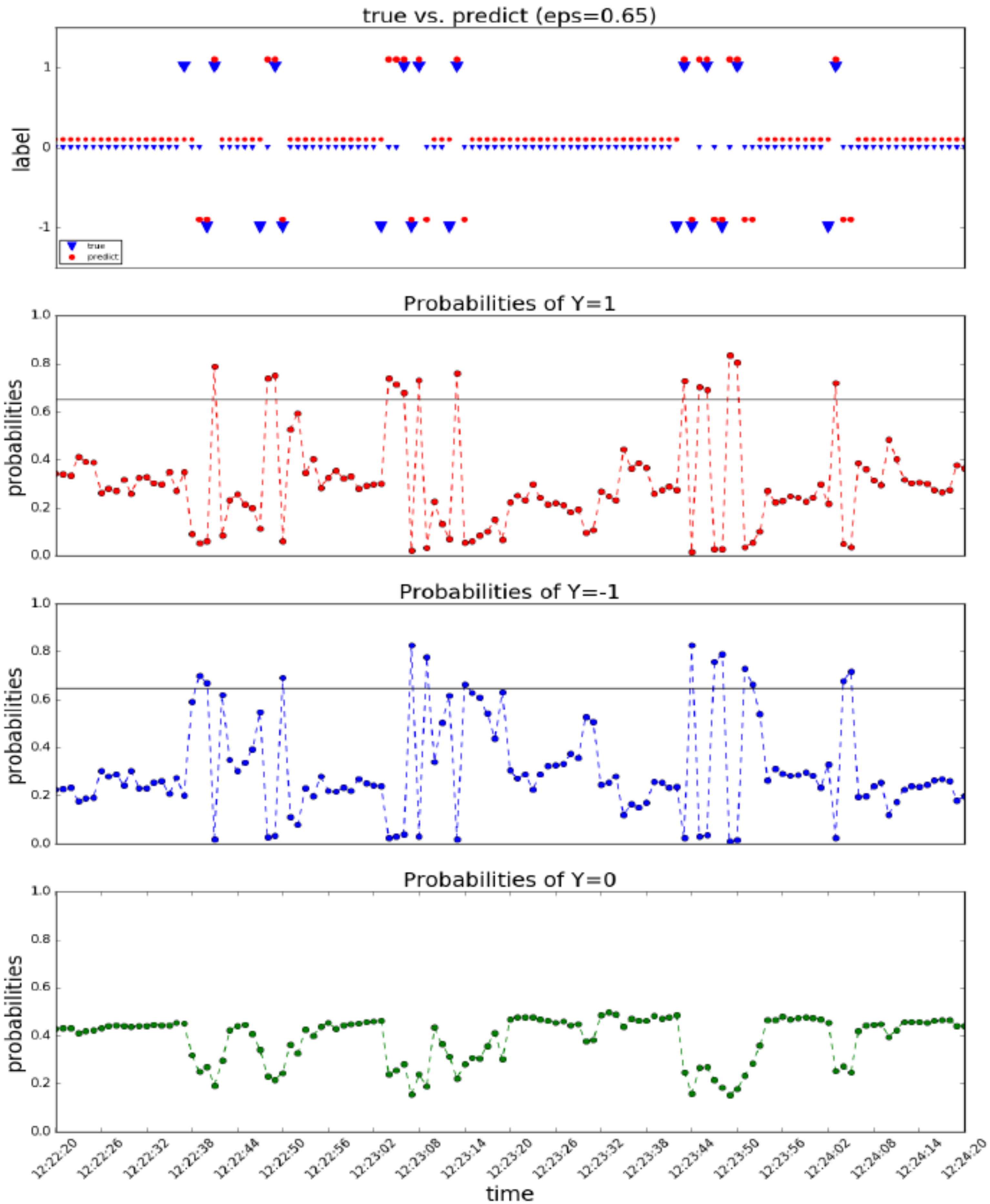
RNNs are Non-parametric Autoregressive Models

- If $W_h = U_h = \phi, |\phi| < 1$, $W_y = 1$ and $b_h = b_y = 0$.
- Then we can show that $\hat{Y}_t = F(\mathcal{X}_{t-1})$ is a zero-drift auto-regressive, $AR(p)$, model with geometrically decaying weights:

$$
\begin{aligned}
Z_{t-p} &= \phi Y_{t-p}, \\
Z_{t-p+1} &= \phi(Z_{t-p} + Y_{t-p+1}), \\
\dots &= \dots \\
Z_{t-1} &= \phi(Z_{t-2} + Y_{t-1}), \\
Y_t &= Z_{t-1},
\end{aligned}
$$

where

$$
\hat{Y}_t = (\phi L + \phi L^2 + \dots + \phi^p L^p)[Y_t].
$$



true vs. predict (eps=0.65)
label
true
predict
Probabilities of Y=1
probabilities
Probabilities of Y=-1
probabilities
Probabilities of Y=0
probabilities
time

Predictive Performance of RNN

Features	Method	$\hat{Y}=-1$ precision	$\hat{Y}=-1$ recall	$\hat{Y}=-1$ f1	$\hat{Y}=0$ precision	$\hat{Y}=0$ recall	$\hat{Y}=0$ f1	$\hat{Y}=1$ precision	$\hat{Y}=1$ recall	$\hat{Y}=1$ f1
Liquidity Imbalance	Logistic	0.010	0.603	0.019	0.995	0.620	0.764	0.013	0.588	0.025
	Kalman Filter	0.051	0.540	0.093	0.998	0.682	0.810	0.055	0.557	0.100
	RNN	0.037	0.636	0.070	0.996	0.673	0.803	0.040	0.613	0.075
Order Flow	Logistic	0.042	0.711	0.079	0.991	0.590	0.740	0.047	0.688	0.088
	Kalman Filter	0.068	0.594	0.122	0.996	0.615	0.751	0.071	0.661	0.128
	RNN	0.064	0.739	0.118	0.995	0.701	0.823	0.066	0.728	0.121
Spatio-temporal	Elastic Net	0.063	0.754	0.116	0.986	0.483	0.649	0.058	0.815	0.108
	RNN	0.084	0.788	0.153	0.999	0.729	0.843	0.075	0.818	0.137
	FFWD NN	0.066	0.758	0.121	0.999	0.657	0.795	0.065	0.796	0.120
	White Noise	0.004	0.333	0.007	0.993	0.333	0.499	0.003	0.333	0.007

Intraday Trading Activity

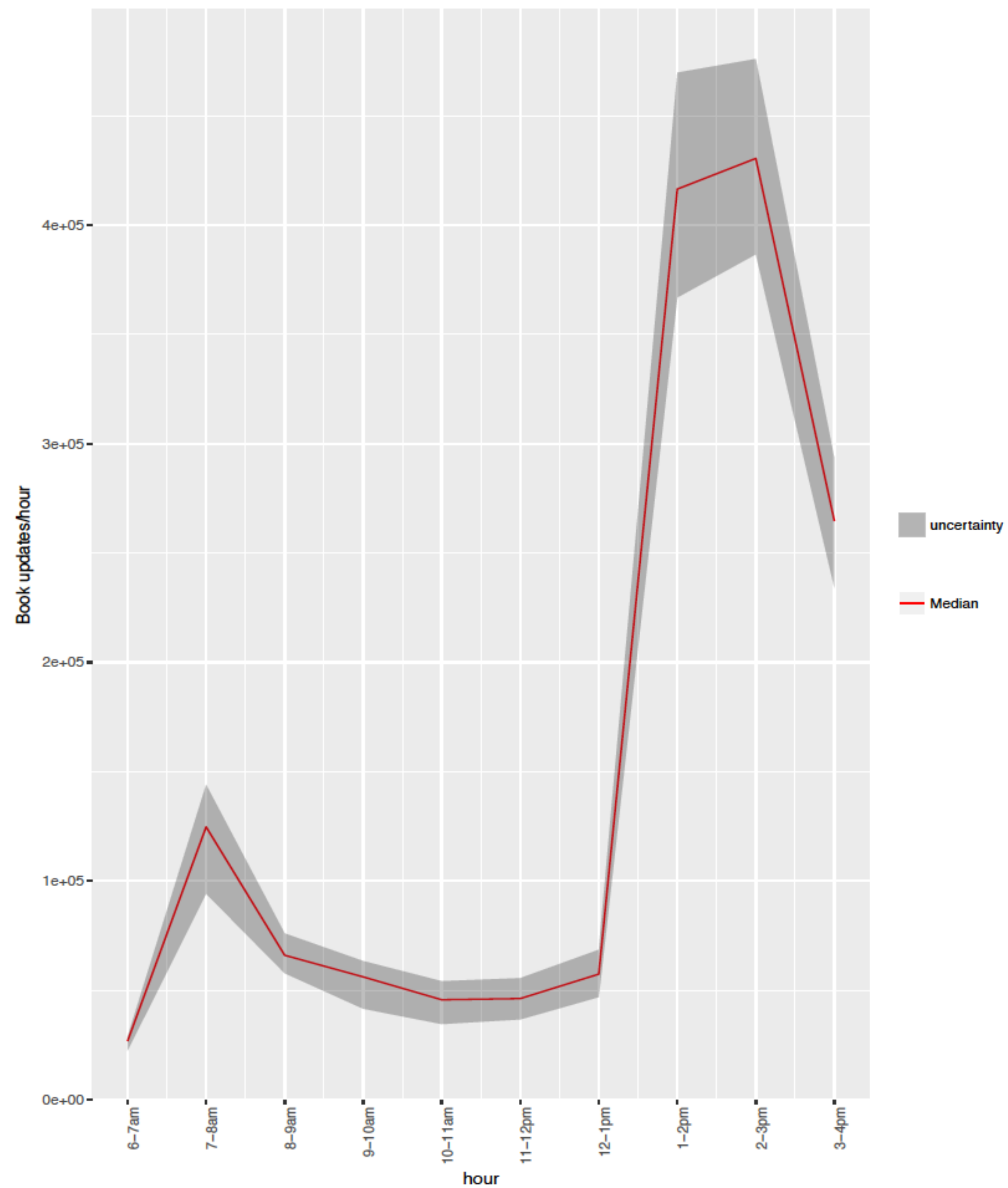
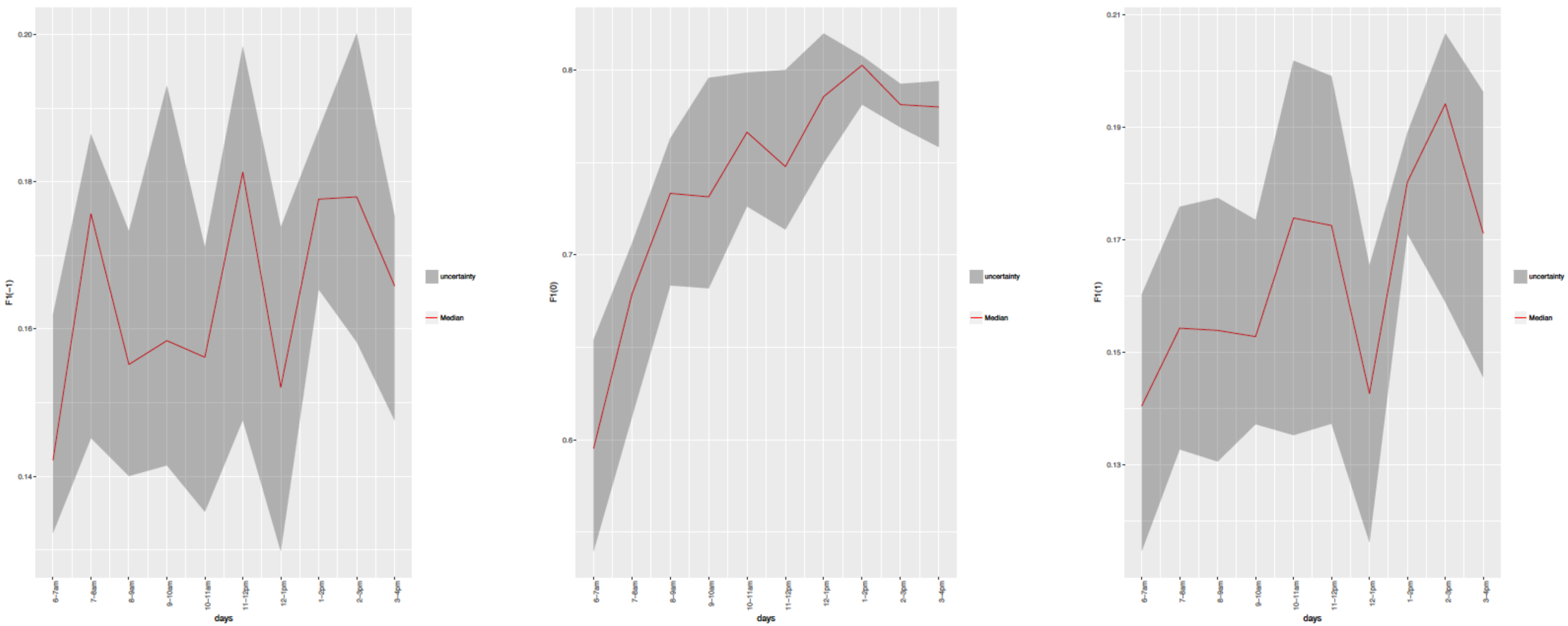


Figure: *The hourly limit order book rates of ESU6 are shown by time of day. A surge of quote adjustment and trading activity is consistently observed between the hours of 7-8am CST and 3-4pm CST.*

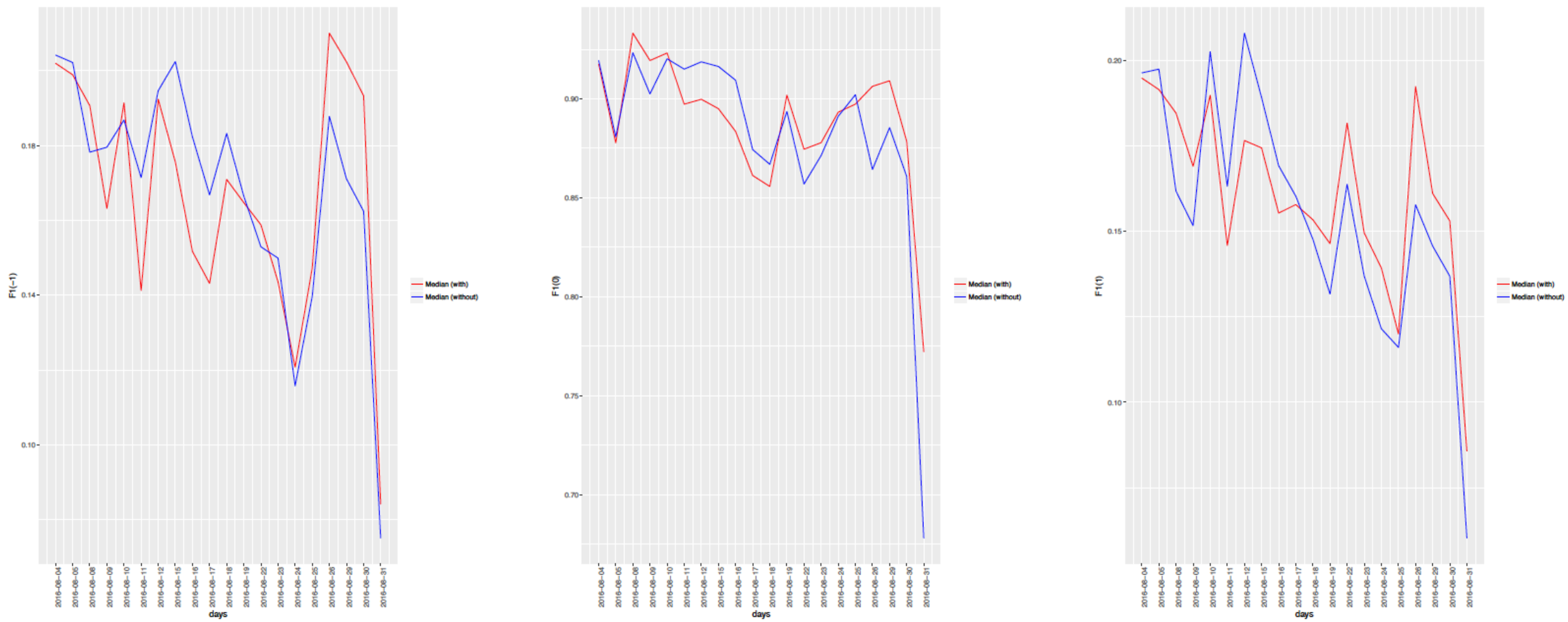
Intraday Predictive Performance



(a) F1 score of $\hat{Y} = -1$ (b) F1 score of $\hat{Y} = 0$ (c) F1 score of $\hat{Y} = 1$.

Figure: *The intra-day F1 scores are shown for (left) downward, (middle) neutral, or (right) upward next price movement prediction.*

Performance Comparison with Daily Retraining



(a) F1 score of $\hat{Y} = -1$ (b) F1 score of $\hat{Y} = 0$ (c) F1 score of $\hat{Y} = 1$.

Figure: *The F1 scores over the calendar month, with (red) and without (blue) daily retraining of the RNN, are shown for (left) downward, (middle) neutral, or (right) upward next price movement prediction.*

Sensitivity to Latency

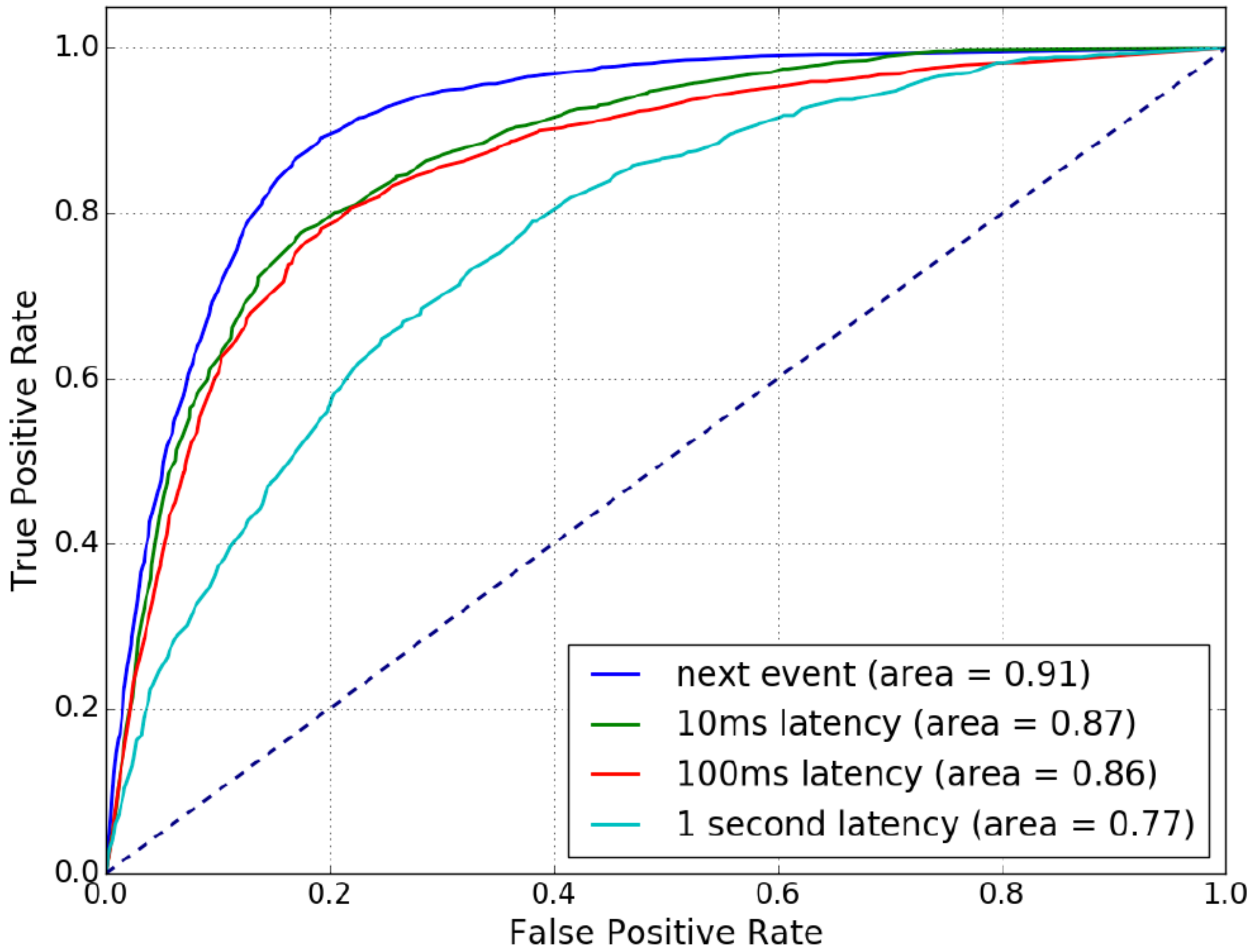


Figure: *The Receiver Operator Characteristic (ROC) curves of a binary RNN classifier over varying prediction horizons. In practice, the prediction horizon should be chosen to adequately account for latency between the trade execution platform and the exchange.*

> play game

Market Making Strategy

A market making strategy is the pair $\mathcal{L}_t := (\mathcal{L}_t^a, \mathcal{L}_t^b)$ representing the quoting of a bid and ask at time t.

$$\mathcal{L}^a(\hat{Y}_0) \begin{cases} \{0, L\}, & \hat{Y}_0 = 1, \\ \{L, 0\}, & \hat{Y}_0 = 0, \\ \{L, 0\}, & \hat{Y}_0 = -1. \end{cases} \qquad \mathcal{L}^b(\hat{Y}_0) \begin{cases} \{L, 0\}, & \hat{Y}_0 = 1, \\ \{L, 0\}, & \hat{Y}_0 = 0, \\ \{0, L\}, & \hat{Y}_0 = -1. \end{cases}$$

Did our order get filled?

Trade-to-Book Ratio

$$\mathrm{R}_t(L_0^{b,j}; \mathcal{D}_\tau^{b,j}, \omega) = \frac{M_t^s}{Q_0^{b,j} - \left(\sum_{u \in \mathbf{t}^s} M_u^s + \omega \sum_{i=1}^{j} \sum_{u \in \mathbf{t}^{c,i}} C_u^{b,i} - \sum_{i=1}^{j} \sum_{t \in \mathbf{t}^{b,i}} 1_{\{\phi_{u,u} < \phi_{u,t_0}\}} L_u^{b,i}\right)}$$

- $Q_0^{b,j} := \sum_{i=1}^{j} q_{t_0}^{b,i}$ is the sum of the depths of the queue at time t_0 up to the j^{th} bid level
- $\sum_{u \in \mathbf{t}^s} M_u^s$ are the sell market orders arriving at times $\mathbf{t}_s$;
- $\sum_{u \in \mathbf{t}^{c,i}} C_u^{b,i}$ are the level i bid orders cancelled at times $\mathbf{t}^{c,i}$;
- $1_{\{\phi_{u,u} < \phi_{u,t_0}\}}$ is an indicator function returning unity if a subsequent limit order placed at time u has higher queue priority than the time t_0 reference limit order; and
- $\omega \in [0,1]$ is an unknown cancellation parameter which denotes the proportion of cancellations of orders with higher queue priority than the reference limit order over the interval τ.

Need to simulate the exchange's matching engine

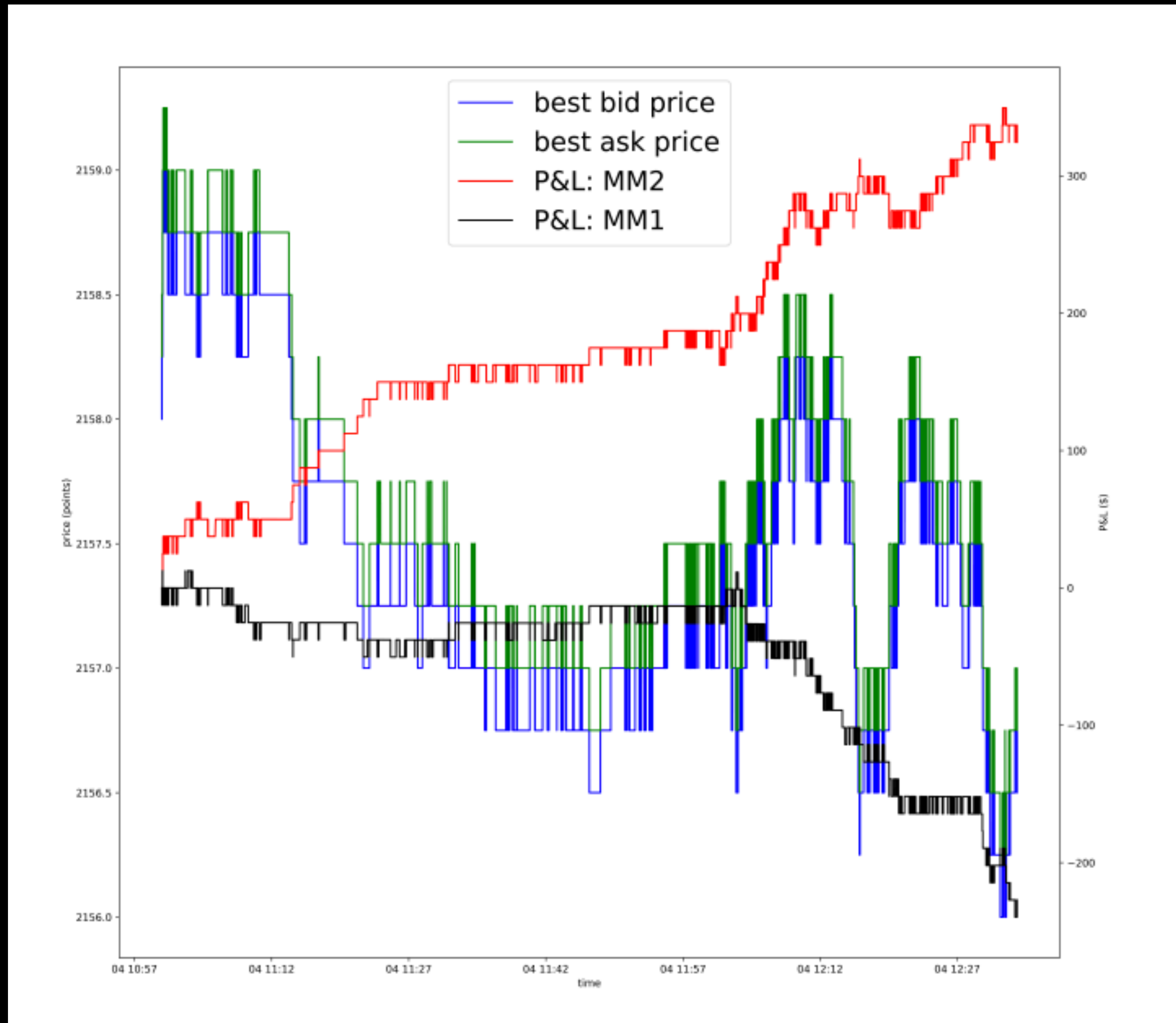
Example: FIFO Market

1. Suppose at time t_0^- the queue depth at the best bid is 50. The largest order has size 20.
2. The reference limit order to buy 50 contracts at the best bid level is received by the exchange at time t_0.
3. A market sell order of size 25 arrives in $(t_0, t]$.
4. The best bid for 20 is cancelled in $(t_0, t]$.
5. The queue position of the reference order consequently advances so that there are 5 contracts ahead of it.

If a new sell market order of size 10 arrives at time t then its trade-to-book ratio, with respect to the reference limit order, has the value

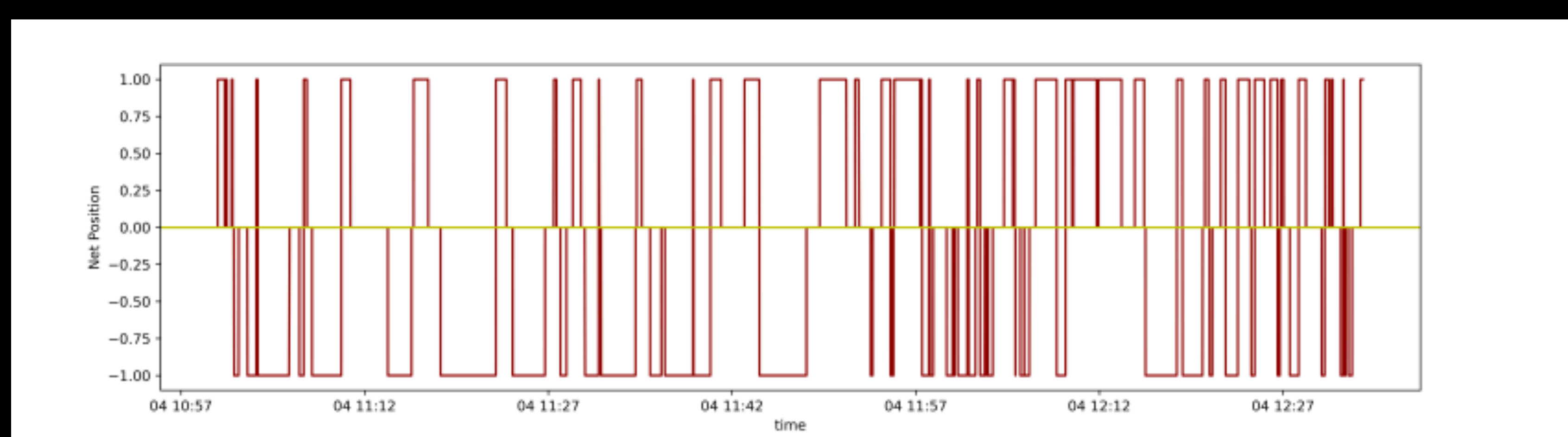
$$\mathcal{R}_t(50; \mathcal{D}^1_\tau, 1) = \frac{10}{50 + 50 - (25 + 1 \cdot 20 + 0)} = 2/11 \quad \text{(partial fill)}$$

Price

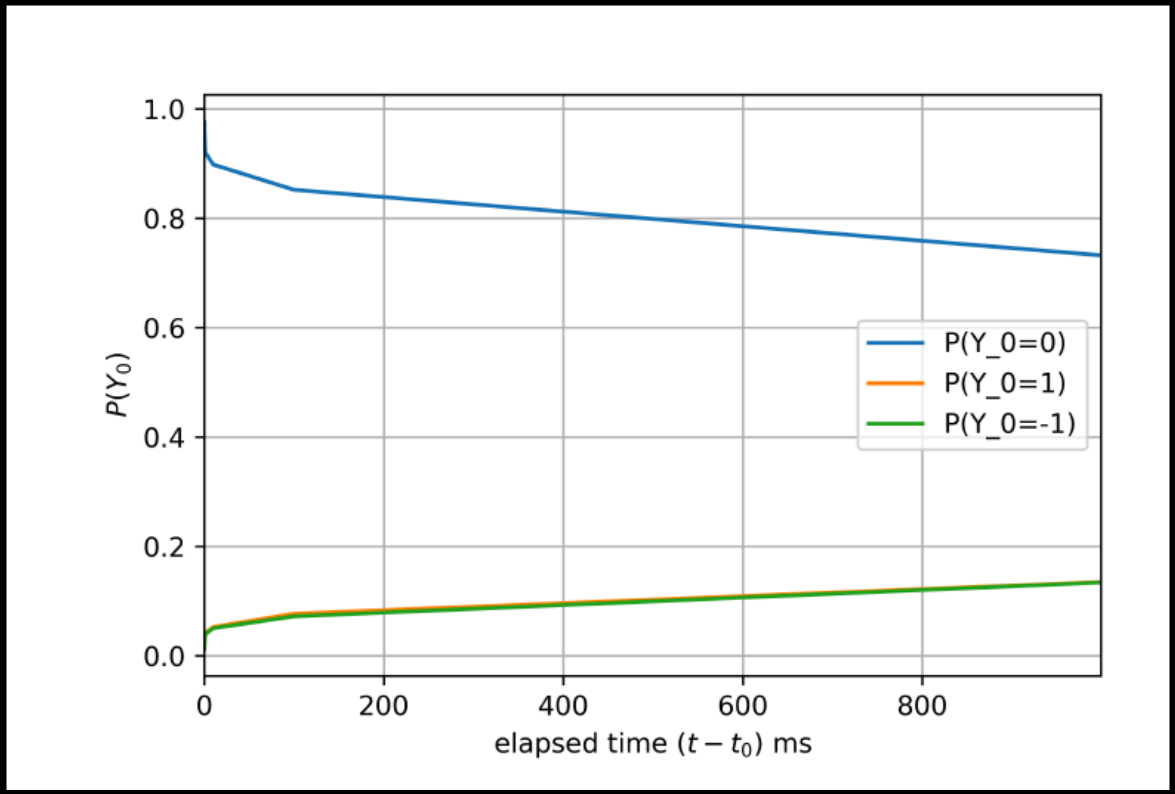


Profit & Loss

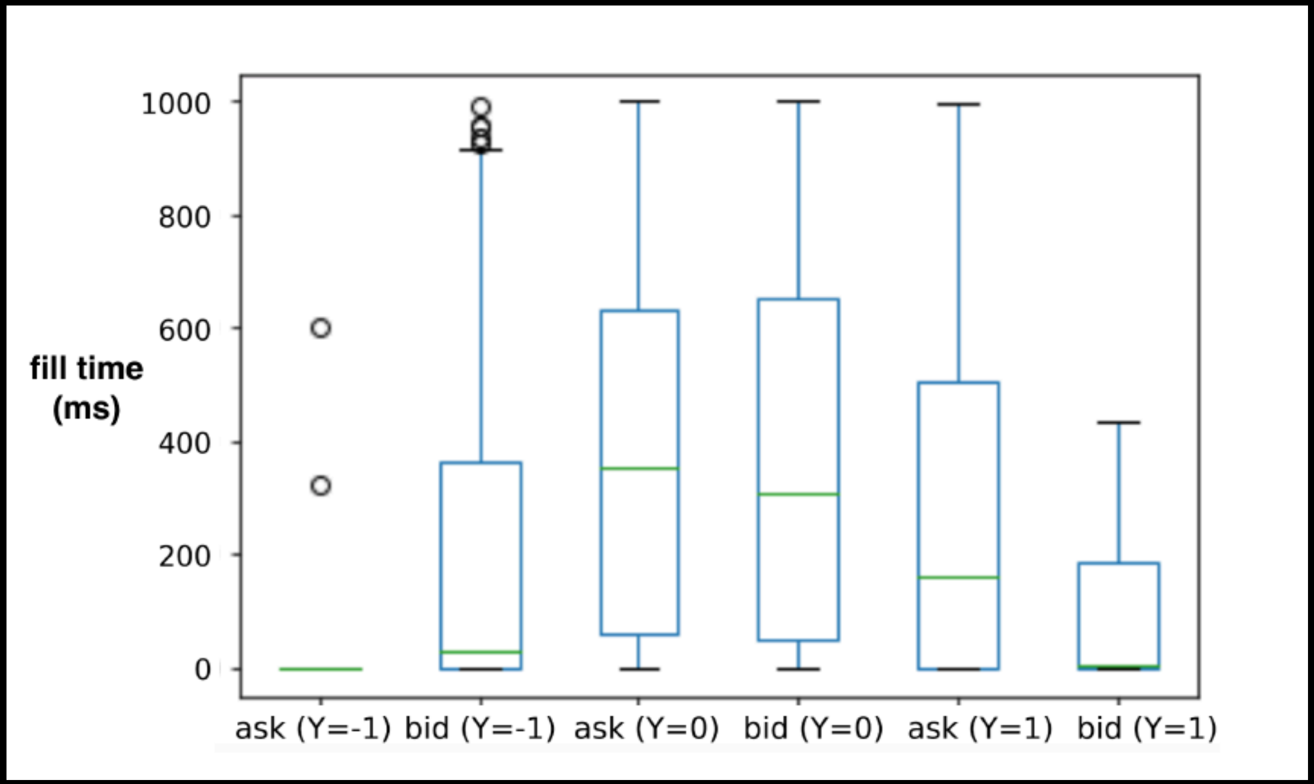
Inventory



Why don't we have the ultimate money printing machine?



State



Fill Time

Supervised machine learning performance ≠ Strategy performance

Fill Diagnostics

MM1	y		
	-1	0	1
$P(Y_0 = y)$	0.134	0.732	0.134
$P(R_t^{a,0} \geq 1 \mid Y_0 = y)$	0.074	0.67	0.581
$P(R_t^{b,0} \geq 1 \mid Y_0 = y)$	0.563	0.615	0.107
$P(Z = 1 \mid Y_0 = y)$	0.042	0.412	0.062
$P(Z = 0 \mid Y_0 = y)$	0.554	0.461	0.563

MM2	y		
	-1	0	1
$P(Y_0 = y)$	0.134	0.732	0.134
$P(R_t^{a,1} \geq 1 \mid Y_0 = y)$	0.007	0.511	0
$P(R_t^{a,2} \geq 1 \mid Y_0 = y)$	0	0	$\boxed{0.421}$
$P(R_t^{b,1} \geq 1 \mid Y_0 = y)$	0	0.504	0.011
$P(R_t^{b,2} \geq 1 \mid Y_0 = y)$	$\boxed{0.403}$	0	0

Table: The estimated empirical price movement probabilities, quote fill probabilities and spread fill probabilities conditioned on the movement of the true state over a forecasting horizon of $t = h = 1s$. Each column shows the corresponding conditional probabilities for each value of Y_0.

Hard to get filled when placing at higher levels

Spread captured

$P(Z_t = 1 \mid Y_0 = y_i, \hat{Y}_0 = \hat{y}_j)$			
	$\hat{y}$		
y	-1	0	1
-1	0.003	0	0
0	0	0.258	0
1	0	0	0.005

Spread not-captured

$P(Z_t = 0 \mid Y_0 = y_i, \hat{Y}_0 = \hat{y}_j)$			
	$\hat{y}$		
y	-1	0	1
-1	0.404,	0.003	0
0	0.305	0.5	0.504
1	0	0.011	0.423

Table: *The probability that the spread is filled (left) and the probability of adverse selection (right) using the MM2 strategy conditioned on the true movement Y_0 and the prediction $\hat{Y}_0$.*

Trade Information Matrices*

Level 1	$\hat{Y}_0 = y$		
	-1	0	1
-1	0.014	0.014	0.014
0	3.324	3.324	3.324
1	0.045	0.045	0.045

Level 2	$\hat{Y}_0 = y$		
	-1	0	1
-1	0.604	0	0
0	2.457	2.225	4.059
1	0	0	0.640

*Dixon, A High Frequency Trade Execution Model for Supervised Learning, High Frequency, 2018.

Game 1 or Game 2?

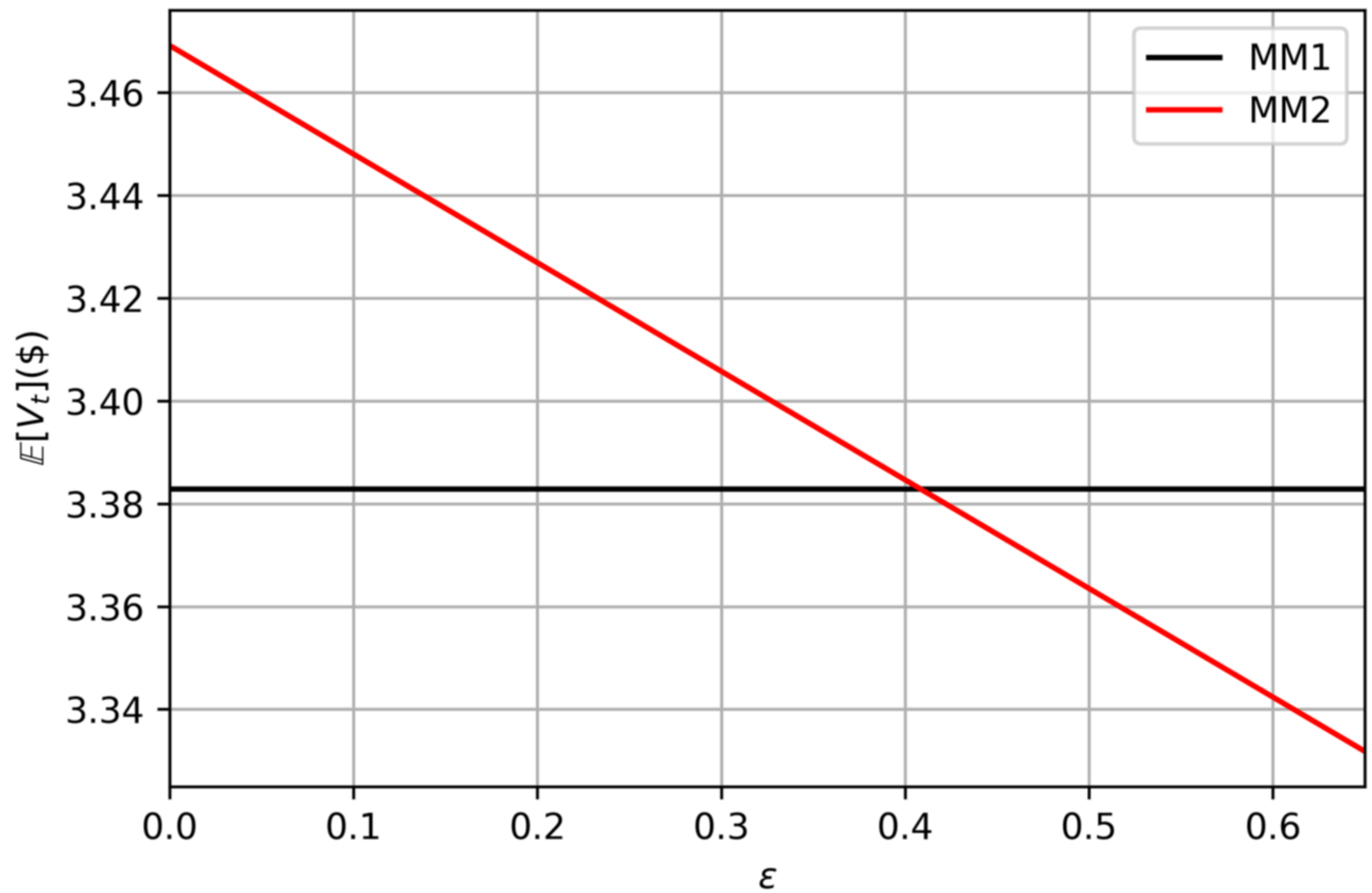


Figure: *This figure compares the expected P&L of the two market market strategies as a function of error ϵ in the confusion matrix. It is observed that the expected realized profit from MM2 (red) linearly decays with ϵ, to the extent that it can become less profitable than the baseline strategy MM1 (black).*

Summary

☒ Our research shows that predicting price movements with machine learning can provide an alternative source of alpha for market makers who currently rely solely on speed.

☒ Introduced `Trade Information Matrices` to translate machine accuracy to profit.

☒ Inclusion of different feature sets and predictors will always vary across market markers => Even with the same ML tools, extremely unlikely to have a zero-sum game.

☒ Reinforcement learning with execution policies are the next step.

Email: mdixon7@stuart.iit.edu
Homepage: https://stuart.iit.edu/faculty/matthew-dixon

> Shall we play a game?

[1] HFT Market Making
[2] HFT Market Making with Supervised ML
[3] HFT Market Making with Reinforcement ML

> 3..