CIRRELT
1971 – 2021

CIRRELT-2021-36

Reducing Transaction Costs using Intraday Forecasts of Limit Order Book Slopes

Chahid Ahabchane
Tolga Cenesizoglu
Gunnar Grass
Sanjay Dominik Jena

September 2021

Bureau de Montréal
Université de Montréal
C.P. 6 128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Reducing Transaction Costs using Intraday Forecasts of Limit Order Book Slopes

Chahid Ahabchane[1,2], Tolga Cenesizoglu[2,*], Gunnar Grass[2], Sanjay Dominik Jena[1,3]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
2. Department of Finance, HEC Montréal
3. Department of Analytics, Operations and Information Technology, School of Management, ESG UQAM

Abstract. Market participants who need to trade a significant number of securities within a given period can face high transaction costs. In this paper, we document how improvements in intraday liquidity forecasts can help to reduce total transaction costs. We compare various approaches, including autoregressive and machine learning models, to forecast intraday transaction costs computed on comprehensive ultra-high-frequency limit order book data for a sample of NYSE stocks from 2002 to 2012. Our results indicate that improved liquidity forecasts can significantly decrease total transaction costs. In particular, simple models capturing seasonality in market liquidity tend to outperform alternative models.

Keywords: Best order execution, ultra high-frequency data, time series models, Machine learning models

Acknowledgements. The authors acknowledge the financial support of Institut de valorisation des données (IVADO) and HEC Montréal.

---

* Corresponding author: tolga.cenesizoglu@hec.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2021

1 Introduction

Buying or selling large number of shares of a given stock over a given period of time in financial markets is a common quest, faced by traders on a daily basis. For example, passive funds need to regularly trade large numbers of shares to minimize their tracking errors. The costs associated with the execution of large orders over a certain period of time can be substantial, especially for stocks and/or periods with limited supply of liquidity. For example, Lesmond et al. (2004) find that returns on momentum strategies, which are known to be economically and statistically significant and cannot be accounted for by risk-based explanations, disappear when one takes into account transaction costs faced when implementing these strategies. Trading large number of shares such as to minimize transaction costs is therefore a recurrent challenge.

In an effort to limit transaction costs, large orders are commonly split into smaller trades, sometimes referred to as "packages" (see, e.g., Chan and Lakonishok, 1995) that are executed over the given period of time. The question how to optimally select the size and timing of these packages so as to minimize overall transaction costs is known as the optimal execution problem. This problem has attracted renewed attention in the academic literature and has been analyzed, among others, by Almgren and Chriss (1999, 2001), Brunnermeier and Pedersen (2005), Huberman and Stanzl (2005), Almgren and Lorenz (2007), Schoeneborn and Schied (2009), Carlin et al. (2007), Schied et al. (2008, 2010) and Obizhaeva and Wang (2013).

Bertsimas and Lo (1998) define best execution as the choice of package sizes (traded at regular intervals over a given period of time) to minimize the expected cost of execution. They solve the optimal execution problem under the assumption that the law of motion for the underlying asset's price has two components: a random walk component in the absence of a given trade and a price impact that is a linear function of trade size. In this mathematical formulation, the price impact function is assumed to be time invariant, i.e., it does not vary over time.

However, it is well known that liquidity supply exhibits strong variation over time (see, e.g., Chordia et al., 2001; Kempf and Mayston, 2008; Lorenz and Osterrieder, 2008). More recent articles relax this assumption. Obizhaeva and Wang (2013) show the importance of liquidity dynamics in determining optimal trading strategies. Specifically, they demonstrate that the optimal trading strategy is a combination of large and small trades, contrasting the naive approach of splitting the total order into equal-sized packages as suggested in earlier literature. Building on the model of Obizhaeva and Wang (2013), Fruth et al. (2014) analyze optimal trading strategies when the liquidity is changing deterministically. They characterize optimal strategies in terms of trading and waiting periods and find that it might even be optimal not to trade during periods of low liquidity depending on the expected future liquidity.

All of the above-mentioned articles analyze the best execution problem theoretically, using various assumptions about liquidity supply and dynamics. To the best of our knowledge, there is no paper that analyzes the problem empirically.

In this paper, we address this gap in the literature and analyze this problem from an empirical standpoint by quantifying its economic importance using historical data on transaction costs. Our empirical approach is motivated by the recent theoretical literature, but differs from it in one important aspect. Most papers make assumptions on the price dynamics and/or price impact functions and obtain the optimal trading strategies by solving the resulting dynamic optimization problem. Of course, some of these underlying assumptions are rather limiting and might not always hold in reality.

In contrast, we examine the dynamics of high-frequency transaction costs using a comprehensive panel data and treat the optimal execution problem as a forecasting problem with static optimization. Specifically, we have in mind a trader who can trade every five minutes and needs to trade a specific number of shares by the end of a given trading day. We obtain optimal package sizes for this day by minimizing the expected total transaction costs at the beginning of this day based on forecasts of transaction costs.

Our forecasting approach to the best execution problem requires measures of intraday transaction costs. In this paper, we follow Cenesizoglu and Grass (2018) and measure buy and sell side transaction costs separately as the slope of the corresponding side of the limit order book. To be more precise, we use millisecond time-stamped snapshots of the limit order book for stocks traded on the New York Stock Exchange (NYSE) from the Thomson Returns Tick History (TRTH) database, now called Refinitiv Tick History. For each snapshot of the limit order book, we compute a measure of the slope of the limit order book called Marginal Cost of Immediacy (MCI). We define MCI as the transaction cost of immediately buying (selling) all shares in the first ten levels of the ask (bid) side of the limit order book via market order, scaled by dollar quantity. MCI_A (MCI_B) measures the marginal cost of buying (selling) an additional \$1,000 of a stock in basis points and can also be thought of as the price impacts of a market buy (sell) order as a function of its size in dollar terms.

We aggregate snapshot-by-snapshot MCI measures by computing their averages for each five-minute interval between 9:30 and 16:00. We thus use these five-minute MCI measures as our estimate of buy and sell side transaction costs. This measure has several advantages compared to others proposed in the literature (such as the buy and sell side lambdas of Brennan et al. (2013)). It can be computed for any desired interval and for ask and bid sides separately. Furthermore, it is observed and does not need to be estimated, which is especially useful when modeling and forecasting intraday transaction costs.

As the main determinants of liquidity exhibit seasonality at various frequencies, so do transaction costs. Among others, high volatility at market opening - when overnight information is included in prices - translates into high transaction costs in the morning. Similarly, increases in information asymmetry ahead of earnings announcements decrease market liquidity. We first analyze the seasonalities in MCI_A and MCI_B at intraday, weekly, and quarterly frequencies before modeling and forecasting them. We find that transaction costs exhibit strong intraday and intraquarter but not intraweek seasonalities. The transaction costs tend to be higher in the morning and before earnings announcements, in line with our expectations.

We consider two approaches to capture seasonality patterns. The first, which we refer to as 22-day avg., captures the intraday seasonality in a simple fashion. It forecasts the transaction cost for a given five-minute interval, day and stock as its average over the same five-minute interval and stock over the preceding 22 trading days. The second model improves on the first model to also capture intra-week and intra-quarter seasonalities in addition to intra-day seasonalities. In this second model, which we refer to as Adj. 22-day avg, we regress five-minute transaction costs for a given day and stock on the corresponding averages from the first model computed over the preceding 22 days including the day in question. The forecasts for five-minute transaction costs for the next day are then given by the fitted values from this regression. This second model accounts for seasonality at different frequencies while reducing the noise associated by simply using the last day transaction costs as our forecasts.

We then consider different autoregressive models, one with 1 lag and another with 78 lags (which corresponds to the number of five-minute intervals in a trading day). For each of these models, we consider two variants. In the first variant, we directly model and forecast intraday transaction costs without modelling or forecasting the seasonal component. The models of the first variant, referred to as $AR(1)$ and $AR(78)$, assume that the dynamics of the AR model captures, to a certain extent, the seasonal component. In the models for the second variant, referred to as $SAR(1)$ and $SAR(78)$, we remove the seasonal component from intraday transaction costs and use the AR models to model and forecast the non-seasonal component. To obtain our forecasts for the transaction costs. we add the forecast for the seasonal component to that of the non-seasonal component. Finally, we also consider two state-of-the-art recurrent neural networks, namely Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). LSTM networks are a distinct type of Recurrent Neural Networks, which are able to learn long-term dependencies. They are thus very popular for working with sequential data such as time-series data. GRU is similar to an LSTM but has a less complicated structure with fewer parameters.

We estimate/train all AR/ML models recursively using a rolling window of 22 days intraday observations for MCI_A and MCI_B separately. We do not use any data from the day we are forecasting in the sense that our forecasts are not dynamic but rather static. In other words, we assume that the trader obtains her forecasts at the beginning of the trading day and do not update her forecasts as she observes transaction costs during the day. She can, of course, obtain better forecasts if she considers updating her forecasts and parameters as she observes data throughout the trading day. Furthermore, we also implicitly assume that the limit order book is resilient in the sense that any transaction that a trader executes does not impact prices in the subsequent five-minute interval. These assumptions in turn imply that our results provide a lower bound in terms of forecasting ability of a trader.

Given that we need good forecasts of intra-day transaction costs to choose the optimal package sizes, we first focus on the statistical performance of each model in forecasting intra-day transaction costs based on Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). The forecast based on the averages over the last 22 days, which is our benchmark model, performs relatively well for such a simple model. This rather good performance of a simple model suggests that the seasonal component plays an important role in the intraday time variation of transaction costs. More importantly, the proposed adjustment to this simple benchmark to capture daily and weekly time variation in intraday seasonality performs better than all other models considered. Specifically, the RMSE and MAPE of this model has not only the lowest mean and median across stocks and trading days, but are also much less dispersed as measured by its standard deviation. The MAPE of this model has a median of 29.22% showing that it captures more than 70% of the absolute variation in intraday MCI_A. The AR(1) model is a close second best, but its performance exhibits much more variation across stocks and days. The two machine learning models, LSTM and GRU perform relatively poorly mostly due to the high number of parameters to estimate and relatively short training sample (22 days times 78 intraday observations) that we chose to be comparable to other competing models.

Having shown that we can predict intraday transaction costs especially well using models capturing its seasonality, we now turn our attention to the economic value of these predictions for traders with large orders to execute. We measure the economic performance of a given forecasting model based on the resulting total realized transaction costs, using the optimal package sizes implied by the out-of-sample forecasts from that model. We compare the performances of different models to a hypothetical scenario where the trader has perfect knowledge. We also consider two values for the total amount to be transacted, a fixed amount of \$100M and a proportional amount based on market capitalization. Our conclusions are very similar regardless of whether we consider a fixed or a proportional amount. Here, we summarize those based on trading a fixed amount.

The naive approach of trading the same amount every five minutes results in total buying and selling costs with a median of \$1.60M and \$1.76M for \$100M traded, respectively. These numbers are respectively 29% and 25% percent higher than the unattainable perfect knowledge case. In other words, if we ignore intraday seasonality and potential predictability of transaction costs, one can do only about 30% worse than the perfect knowledge case by trading the same amount every five minutes.

Among all the other models considered, the simple model of 22-day Avg. delivers the best forecasts in terms of both total buying and selling costs. Specifically, the median total cost of buying (selling) \$100M of a stock based on our forecast from the model of 22-day Avg. is \$1.38M (\$1.59M). These numbers are only 11% and 13% higher than the median buying and selling costs from the unattainable perfect knowledge case. More importantly, they represent a significant improvement compared to the naive approach. The adjusted 22-day average and AR(1) are close second best performing models, based on median total buying and selling costs. There are no other models that result in lower transaction costs than the naive approach for both buying and selling a fixed or a proportional amount.

We lastly examine how trading costs vary with size in the cross-section. Not surprisingly, trading a fixed amount of \$100M is substantially more costly for small stocks than for large ones. The benefits of forecasting intraday trading costs accurately are accordingly largest for small stocks. In contrast, dollar trading costs increase drastically in size when trading a proportional amount of 0.69% of a stock's market capitalization.

Our paper makes several important contributions to the academic literature. First, to the best of our knowledge, we are among the first to approach the best execution problem from a forecasting perspective. Second, we state the non-linear optimization problem that identifies the optimal package sizes and time periods based on such forecasts, and we provide a closed-form formula to solve this problem. We show that better forecasts can provide economic benefits for a trader by significantly decreasing total transaction costs. Third, we examine the intraday and daily dynamics and seasonalities of transaction costs. Fourth, we are among the first to model and forecast these intraday transaction costs. Fifth, we show that simple forecasting models that capture intraday and lower frequency seasonalities perform better than more advanced time series and machine learning models in forecasting high frequency transaction costs in limit order books. Finally, to ensure practical relevance of our empirical results, all our experiments have been carried out on 11 years of the extensive NYSE data set.

Our paper also has important practical implications. First of all, our results provide important insights about the dynamics of transaction costs to traders. Second, our approach can be easily automatized so that discretionary traders can simply focus on the total amount and the direction to trade and leave the optimal trade execution to an algorithm based on our forecasting approach. Last but not least, our approach is easily scalable in the sense that it can be rapidly implemented across a large number of stocks and can provide an edge to traders in today's extremely fast financial markets.

The rest of the paper is organized as follows. Section 2 presents the data and how we compute intraday transaction costs from this data. Section 3 presents different models used for forecasting intraday transaction costs. Section 4 presents performances of different models in an out-of-sample forecasting exercise. Section 5 formulates the best execution problem and shows how it can be approached from a forecasting perspective.

2 Data Description

2.1 Measure of Transaction Costs

Our objective is to understand how a trader that needs to either buy or sell a specific quantity of stocks by the end of a given trading day should split this quantity into multiple market orders to minimize transaction costs. To carry out such an analysis, we require a measure of transaction costs that depends on the trading quantity. While there are many such measures such as Kyle's lambda (Kyle, 1985), most of these measures need to be estimated. This becomes an issue when the trader needs a measure of transaction costs that is available at intraday frequency for each trading day. Furthermore, it is also crucial for the trader to measure, model and forecast buying and selling transaction costs separately as they can sometimes behave quite differently from each other. We therefore use a measure for buy and sell side transaction costs proposed by Cenesizoglu and Grass (2018), the Marginal Cost of Immediacy (MCI). This measure is based on limit order book snapshots and can be computed at any frequency required, therefore removing the need for estimations.

Cenesizoglu and Grass (2018) define MCI as the transaction cost of immediately buying (selling) all shares in the first ten levels of the ask (bid) side of the limit order book via market order, scaled by dollar quantity. MCI measures the marginal cost of buying (selling) an additional \$1,000 in basis points and is ultimately a measure of the slope of the ask (bid) side of the limit order book.

We briefly describe how MCI can be computed for a given snapshot s of the ask side on a given trading day d, $MCI^A_{d,s}$, for a given stock i. $MCI^B_{d,s}$ is defined in a similar fashion. To simplify the notation, we drop the subscript for stock i, but it should be understood that the measures are stock-specific.

For a given snapshot s of the ask side of the limit order book, we first compute the volume-weighted average price of the ask side scaled by the midprice, $VWAPM^{A,L}_{d,s}$, as follows:

$$VWAPM^{A,L}_{d,s} = \ln\left(\frac{VWAP^{A,L}_{d,s}}{M_{d,s}}\right), \tag{1}$$

where

$$VWAP^{A,L}_{d,s} = \frac{Vlm^{A,L}_{d,s}}{\sum_{l=1}^{L} Q^{A,l}_{d,s}} \tag{2}$$

and $Vlm^{A,L}_{d,s} = \sum_{l=1}^{L} P^{A,l}_{d,s} \times Q^{A,l}_{d,s}$ and $P^{A,l}_{d,s}$ and $Q^{A,l}_{d,s}$ are the price and quantity available at the l^{th} level of the ask side for the snapshot s and trading day d. $M_{d,s}$ is the midquote price defined simply as the average of $P^{A,1}_{d,s}$ and $P^{B,1}_{d,s}$. For the bid side, $VWAPM^{B,L}_{d,s}$ is defined analogously.

$VWAPM^{A,L}_{d,s}$ and $VWAPM^{B,L}_{d,s}$ are, respectively, the transaction cost of buying and selling all shares available up to level L in the book, expressed as a log return relative to the midquote price. We obtain $MCI^A_{d,s}$

by dividing $VWAPM^{A,L}_{d,s}$ by the total dollar volume acquired, $Vlm^{A,10}_{d,s}$, as follows:

$$MCI^{A}_{d,s} = \frac{VWAPM^{A,10}_{d,s}}{Vlm^{A,10}_{d,s}}. \tag{3}$$

$MCI^{B}_{d,s}$ is defined analogously as follows:

$$MCI^{B}_{d,s} = \frac{-VWAPM^{B,10}_{d,s}}{Vlm^{B,10}_{d,s}}. \tag{4}$$

MCI is a measure of the slope of the limit order book. A higher MCI implies that less liquidity is supplied via the limit order book and thus transaction costs are high. We measure $MCI^{A}_{d,s}$ and $MCI^{B}_{d,s}$ in basis points (bp) per \$1,000. This provides a simple interpretation of the MCI measures. For example, an $MCI^{A}_{d,s} = 0.2$ means that a trader who wants to buy an additional \$1,000 worth of shares via a market order and facing the snapshot s on trading day d will incur a transaction cost 0.2 bp, i.e., 0.002%. However, if she decides to buy \$1,000,000, she will face a transaction cost of 2.0%.

2.2 Data Set

Our limit order book data is from the Market Depth files of the Thomson Returns Tick History (TRTH) database, now called Refinitiv Tick History. While it includes data for several exchanges, we restrict our analysis to stocks traded on the New York Stock Exchange (NYSE).

Each snapshot has a millisecond time stamp and includes prices and quantities for the first ten levels of both bid and ask sides of the limit order book. The TRTH data do not include hidden orders and comprises only NYSE limit orders and no limit orders from other markets. Our transaction cost measures based on the TRTH data might therefore overstate the actual transaction costs faced by a trader who has access to limit orders from other markets.

NYSE began making this level-two LOB data available to market participants outside the trading floor in January 2002. Our sample period starts in January 2002 and ends in December 2012. We focus on an 11-year period due to sheer size of data.

2.3 Data Processing

Our initial data set includes 52.44 billion LOB snapshots for the years 2002-2012. We follow the steps described in Cenesizoglu and Grass (2018) to initially process this data. Specifically, we remove any snapshot for which the price or volume information is either missing or is equal to zero for any of ten levels of the ask or bid side, as well as any snapshot with the best ask price lower than the best bid price, Level 1 (Level 10) bid-ask spreads above 25% (250%), midquote prices below \$1 or above \$1000. We discard snapshots for which prices do not increase monotonically from level 10 of the bid side to level 10 of the ask side and snapshots outside regular trading hours.

We also delete all securities other than ordinary common shares and any security for which we are unable to find a match in the Center for Research in Security Prices (CRSP) files, using the same matching procedure as Cenesizoglu and Grass (2018). Finally, we exclude data for any firm-day with fewer than 100 snapshots in the

TRTH data or less than 100 trades in the matched TAQ data set, as well as any observations with incomplete CRSP or Compustat data to obtain a sample including 31.19 billion observations reported for 2,103 stocks over 2,740 trading days and 3.37 million stock-day observations.

We compute the $MCI^A_{d,s}$ and $MCI^B_{d,s}$ for each snapshot s, trading day d and stock i (subscript dropped) in our sample. For a given stock, we compute the average buying and selling transaction costs for five-minute interval t on trading day d, $MCI^A_{d,t}$ and $MCI^B_{d,t}$, as the equal-weighted average of the snapshots in that five-minute interval, i.e. $MCI^A_{d,t} = \frac{1}{N^A_{d,t}} \sum_{s=1}^{N^A_{d,t}} MCI^A_{d,s}$ and $MCI^B_{d,t} = \frac{1}{N^B_{d,t}} \sum_{s=1}^{N^B_{d,t}} MCI^B_{d,s}$, where $N^A_{d,t}$ and $N^B_{d,t}$ are respectively the number of snapshots of the ask and bid sides available in five-minute interval t on trading day d. $MCI^A_{d,t}$ and $MCI^B_{d,t}$ are the variables of interest, which we model and forecast. If a given stock has less than 78 observations on a given trading day, we remove this stock-day pair from our final sample.

2.4 Descriptive Statistics

Table 1 presents summary statistics for $MCI^A_{d,t}$ and $MCI^B_{d,t}$ (multiplied by 10,000,000 for ease of presentation) across stocks and trading days in our sample. The median $MCI^A_{d,t}$ is 0.156 basis points, suggesting that a trader who wants to instantaneously buy \$1,000 worth of stock would pay 0.156 basis points in transaction costs. The median $MCI^B_{d,t}$ is slightly higher at 0.174 basis points suggesting that selling via a market order is slightly more expensive in our sample than buying via a market order. Means are much higher than the medians suggesting that the distribution is skewed to the right. This can also be seen in the distributional statistics. Furthermore, the standard deviations are multiple times as high as the average values, suggesting a high degree heterogeneity in terms of transaction costs across stocks and over time.

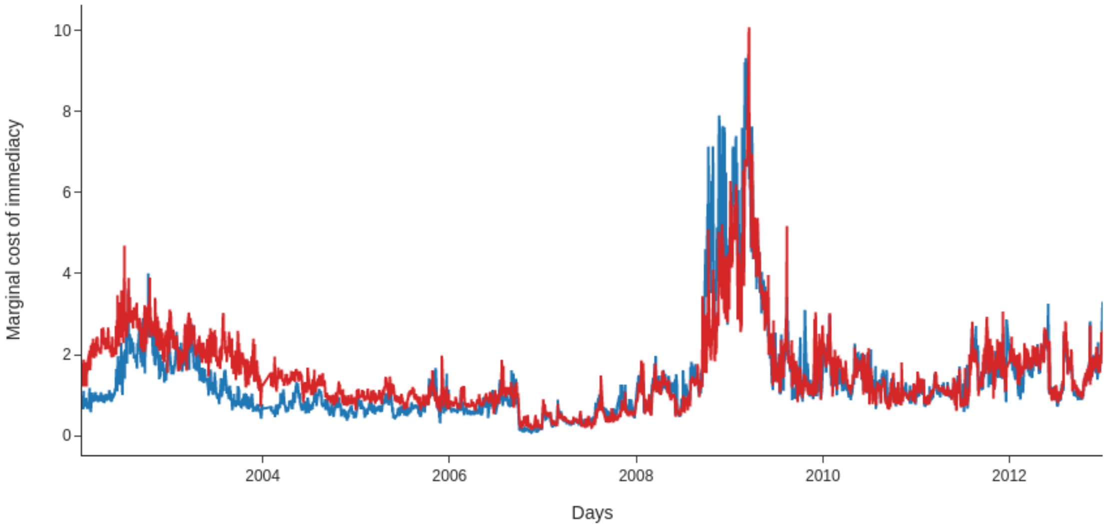
Table 1: Descriptive statistics of MCI

	MCI_A	MCI_B
Mean	1.456	1.627
Median	0.156	0.174
Std. Dev.	8.686	9.625
25% Percentile	0.048	0.052
75% Percentile	0.624	0.728
Nb of Obs	125939000	125939000

Note: This table presents summary statistics for 5-minute transaction costs MCI_A and MCI_B multiplied by 10^7, which can be interpreted as transactions cost in basis points for trading 1000 \$. Our sample period is between January 2002 and December 2012.

Figure 1 presents the time series of $MCI^A_{d,t}$ and $MCI^B_{d,t}$ averaged over all stocks. Average transaction costs are high at the beginning of our sample period with $MCI^B_{d,t}$ slightly higher than $MCI^A_{d,t}$. They then gradually decrease until the Global Financial Crisis, when they reach their highest values. The transaction costs decline following the financial crisis but remain higher than what they were before the financial crisis.

Figure 1: Time Series of the Daily averages of MCI_A and MCI_B



Note: This figure presents the time series of $MCI_{d,t}^{A}$ (blue line) and $MCI_{d,t}^{B}$ (red line) averaged over all stocks in our sample on trading days between January 2002 and December 2012.

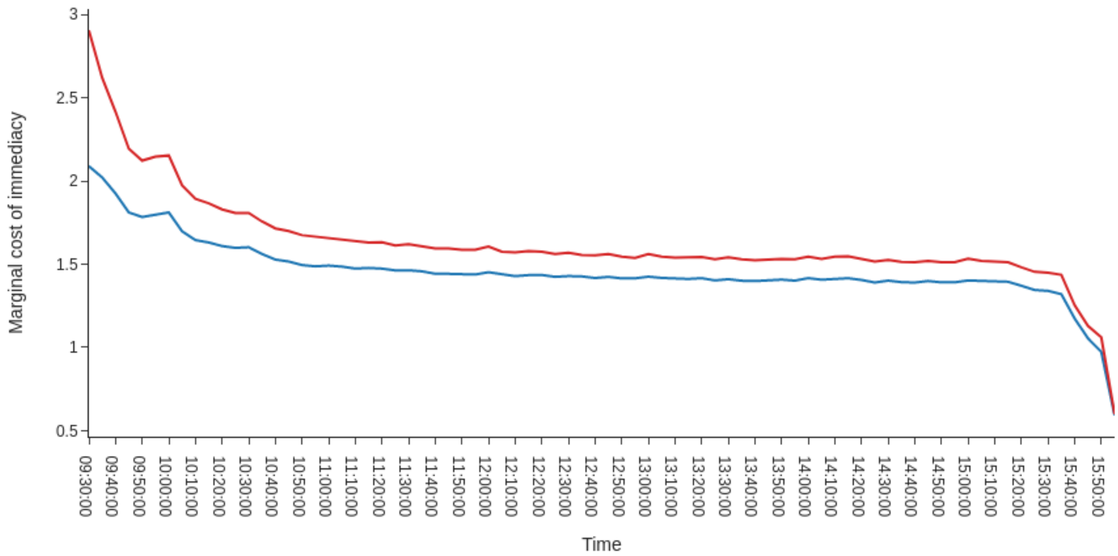
2.5 Seasonality

Transaction costs are driven by various factors including information asymmetry and volatility. As these exhibit seasonality over time, so do transaction costs. In this section, we document such seasonality patterns at daily and quarterly frequency.

During a trading day, bid-ask spreads are known to start high and drop sharply within the first hour of trading. Among others, this is due to the fact that news announcements made after the previous close are incorporated into stock prices when market opens, boosting volatility. Not surprisingly, we observe a similar pattern for our measure of transaction costs, MCI.

Figure 2 shows average transaction costs over all stocks and trading days for each five-minute interval between 9:30 and 16:00. The figure shows a clear intra-day pattern in transaction costs. Both MCI measures are high at the beginning of the day and decrease gradually until 11:00. They then remain practically constant until one hour before closing when they start to decrease rapidly. Although not reported here, this seasonal pattern exhibits significant variation across stocks and trading days. All else equal, the intraday seasonality in transaction costs is more pronounced when the level of transaction costs is higher. As mentioned above and as we will discuss in more detail at the end of this section, transaction costs exhibit intraquarter seasonality in relation to the quarterly earnings announcement cycle. As such, the intraday seasonality of transaction costs also depends on the intraquarter seasonality of transaction costs.

Figure 2: Intraday seasonality in MCI_A and MCI_B



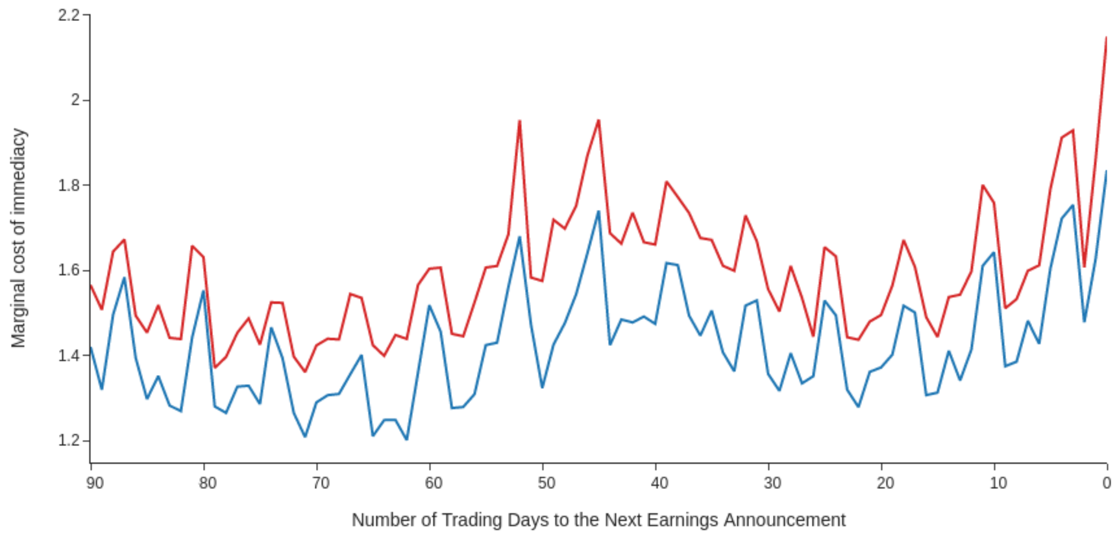
Note: This figure shows the intraday seasonality of ask and bid side transaction costs. More specifically, it plots average $MCI^A_{d,t}$ (blue line) and $MCI^B_{d,t}$ (red line) values computed across all stock-days for five-minute interval during trading hours between 9:30 and 16:00 EST.

The decrease in transaction costs throughout the trading day is substantial. Specifically, MCI_A and MCI_B are around 3 and 2 basis points, respectively, at the beginning of the day and they both decrease to around 0.6 basis points by the end of the trading day. This represents almost an 80% decrease for MCI_A and 70% decrease for MCI_B. Given that our objective is to model and forecast intraday transaction costs, it is very important to capture this intraday variation in transaction costs.

Over a quarter, transaction costs are impacted by changes in information asymmetry over the earnings cycle. Just after companies publish their earnings data – together with other fundamentals such as sales forecasts, changes in management, new supply partners – information asymmetry is relatively low. As companies do not update investors every day, information asymmetry starts accumulating and peaks just prior to the next earnings announcement. As information asymmetry is a key determinant of bid-ask spreads, this seasonality in information asymmetry over a fiscal quarter translates into a similar seasonal pattern in MCI. The sharpest spikes in transaction costs can typically be observed at fiscal year end, when investors anticipate even more information to be released than ahead of the other three quarters.

Figure 3 plots average ask and bid side MCI as a function of the number of calendar days remaining until the next quarterly earnings announcement. MCI averages are computed across all stock-days with the same number of days remaining until the next announcement. The graph shows a clear pattern in line with the dynamics of information asymmetry described above. Trading costs are relatively low in between the weeks surrounding the earnings announcements. They begin to increase about a week before the next earnings announcement. They reach their highest level, on average, right after the earnings announcement and decrease gradually over the two weeks following the earnings announcement. Average MCI_A values increase from about 1.4 to about 1.8 right before earnings announcements. Similarly, average MCI_B values increase from about 1.55 to about 2.2 right before earnings announcements.

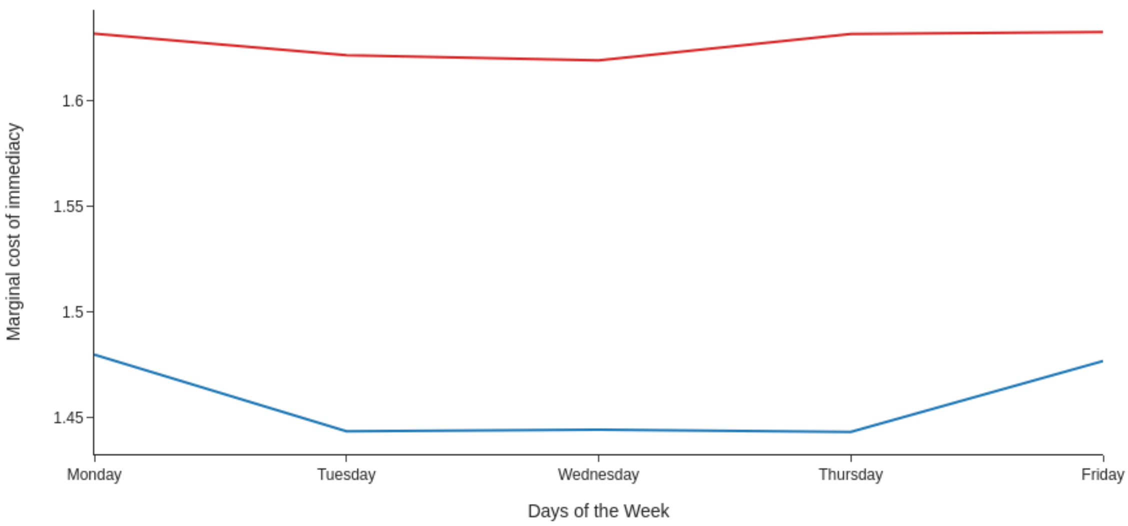
Figure 3: Quarterly seasonality in MCI_A and MCI_B



Note: This figure shows how daily averages MCI_A (blue line) and MCI_B (red line) change as a function of the number of days remaining to the next earnings announcement. Averages are computed across all stock-days with the same number of days remaining before the next earnings announcement. Zero indicates the earnings announcement day.

Seasonal patterns observed within a day and within a fiscal quarter are economically large. In contrast, MCI does not exhibit a large seasonality within a week, as shown in Figure 4. While ask and bid side MCI slightly increase towards the end of the week, the variation in average MCI values within a week is a small fraction of the variation within a day and within a quarter.

Figure 4: Weekly seasonality in MCI_A and MCI_B



Note: The figure shows how average values of $MCI^A_{d,t}$ (blue line) and $MCI^B_{d,t}$ (red line) evolve over the week. The averages are computed over all stocks.

Given the strength of the observed seasonality patterns, we conclude that accounting for seasonality at intraday and intraquarter level is essential when forecasting liquidity. In the following, we propose simple approaches to do so.

3 Forecasting Models

In order to be able to devise efficient trading strategies that minimize the overall transaction costs, we need to predict intraday transaction costs as accurately as possible. In this section, we propose several approaches to predict the intraday time series of transaction costs. A particular challenge in this context is that transaction costs exhibit seasonality at multiple intervals, as documented in the previous section. We thus propose several approaches to explicitly account for these seasonalities. Table 2 summarizes the different models used and its acronyms:

Table 2: Acronyms for Different Forecasting Models

Model Acronym	**Description**
22-day Avg.	Average of $MCI_{d,t}$ over the preceding 22 days.
Adj. 22-day Avg.	Regression of $MCI_{d,t}$ on the the average of $MCI_{d,t}$ over the preceding 22 days.
$AR(1)$	Autoregressive model with 1 lag for $MCI_{d,t}$ estimated over the last 22 days.
$SAR(1)$	Seasonality-adjusted autoregressive model with 1 lag of $MCI_{d,t}$ estimated over the last 22 days.
$AR(78)$	Autoregressive model with 78 lags for $MCI_{d,t}$ estimated over the last 22 days.
$SAR(78)$	Seasonality-adjusted autoregressive model with 78 lag of $MCI_{d,t}$ estimated over the last 22 days.
LSTM	Long Short-Term Memory model for $MCI_{d,t}$ trained using data over the last 22 days.
GRU	Gated Recurrent Units model for $MCI_{d,t}$ trained using data over the last 22 days.

Note: This table presents the acronyms used for different forecasting models in the paper.

Note that, while some of the approaches may potentially benefit from accounting for exogenous variables (such as trading price and volume), in this work, we restrict our analysis to the performance of the models that can be obtained exclusively using information on transaction costs and the corresponding time stamps.

In all the models, our objective is to predict $MCI^A_{d+1,t}$ and $MCI^B_{d+1,t}$ for $t = 1, 2, \dots, 78$, using information available up to and including trading day d. While these predictions will be computed separately for the ask and bid sides, in the following, we will simply refer to these forecasts to as $MCI_{d+1,t}$.

We introduce the seasonality component and the related predictive models in Section 3.1. We then propose auto-regressive models in Section 3.2 and models based on Recurrent Neural Networks in Section 3.3.

3.1 Accounting for Seasonality

As discussed above, MCI exhibits strong intraday seasonality, which in turn depends on the quarterly seasonality. Accounting for these patterns can significantly increase the accuracy of our predictions.

A simple prediction of tomorrow's intraday time series of the MCI is simply today's intraday time series of MCI. Using today's 78 MCI values at the five-minute frequency as a forecast for tomorrow's pattern would not only account for intraday seasonality but also for the quarterly seasonality described before. Given the proximity of today to tomorrow, today's MCI has to be a better predictor of tomorrow's MCI than the MCI

observed yesterday or the day before – all else equal. However, MCI values at five-minute intervals tend to be rather noisy. To reduce the noise, we use average values computed over the last 22 trading days and suggest alternative ways to account for intraquarter seasonality.

The 22-day Average Model.

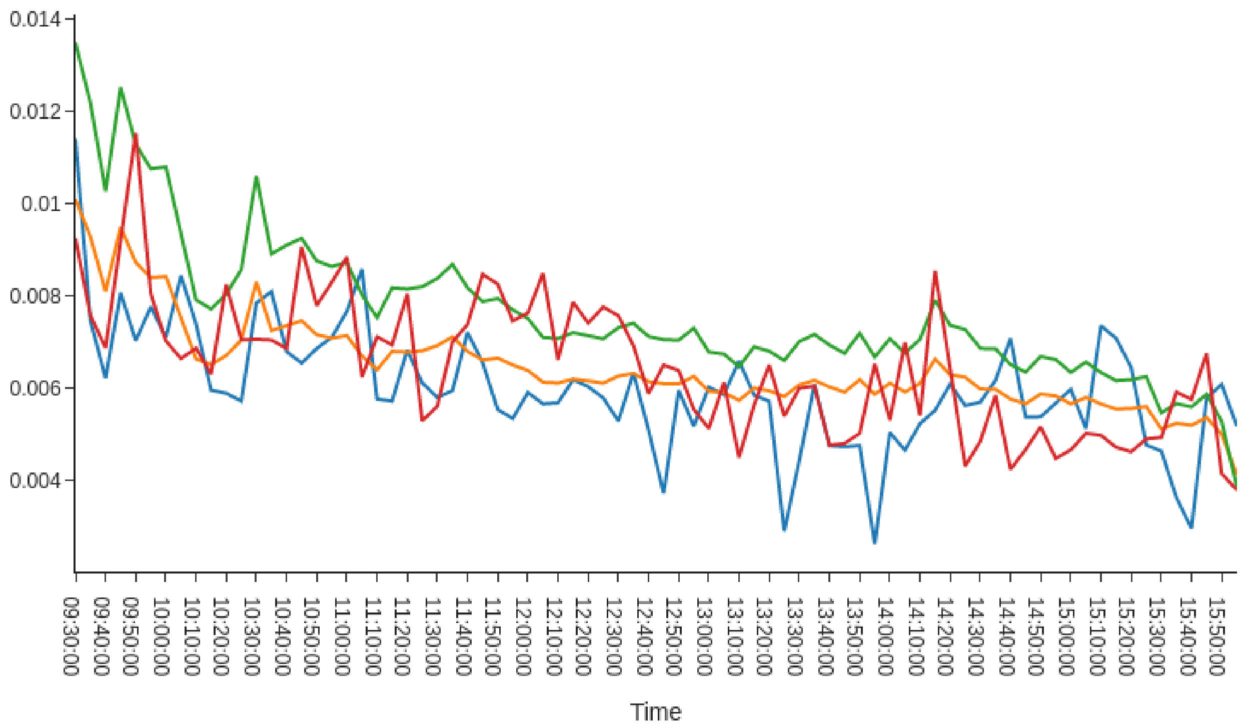
The Adjusted 22-day Average Model. While the previous model averages out the volatility over several days, it is insensitive to cycles other than daily or monthly seasonality. A detailed analysis of our data suggested that the daily variation in intraday seasonality pattern (see Figure 2) strongly depends on the earnings announcement cycles of a stock. To capture the variation in the intraday seasonality patterns over a fiscal quarter, our second seasonality-based model (in the following referred to as the *Adj. 22-day Avg.*) regresses five-minute transaction costs over the current day d (i.e., $MCI_{d,t}$) on the averages over the last 22 days (i.e., $\overline{MCI}_{d,t}$):

$$MCI_{d,t} = \alpha + \beta\overline{MCI}_{d,t} + \varepsilon_t. \tag{5}$$

We then obtain the forecasts for the intraday transaction costs over the following trading day $d+1$ as the fitted values, i.e., $\widehat{MCI}_{d+1,t} = \hat{\alpha} + \hat{\beta}\overline{MCI}_{d,t}$, where $\hat{\alpha}$ and $\hat{\beta}$ denote the estimated coefficients of the regression in Equation (5).

Figure 5 visualizes the forecasts from the 22-day average and the adjusted 22-day average models, along with the realized intraday transaction costs for the IBM stock on 28-03-2012 and 29-03-2012. It is easy to see the importance of the adjustment, which produces a forecast much closer to the realized intraday transaction costs on 29-03-2012.

Figure 5: Accounting for Seasonality Example



Note: This figure provides an example of how different models used for accounting for intraday and intraquarter seasonality perform on a specific day for a specific stock. Our objective in this example is to forecast the realized intraday transaction costs for IBM on 29-03-2012 (blue line). The red line presents the realized intraday transaction costs for IBM on the previous day 28-03-2012. The green and orange lines present respecively the forecasts from the 22-day average and adjusted 22-day average models obtained at the end of the previous trading day on 28-03-2012.

3.2 Autoregressive models

The first approach presented above is based on the average of the last 22 trading days. As such, it does not account for variations between days including potential trends in the data over recent days. The second approach uses a linear regression to fit the intraday curve of the last available day to the 22-day average curve, and therefore incorporates the most recent trend to forecast the transaction costs for the next day. In this section, we present autoregressive models that aim at incorporating more specific information of the last training day (or potentially several days). Instead of using the daily average, these autoregressive models estimate the correlation of the transaction costs of a specific five-minute interval with those of the k preceding five-minute intervals, i.e. k lags, by means of a linear combination.

We consider autoregressive models which differ in terms of the number of lags and how we treat the seasonal component. Specifically, we consider different number of lags between 1 and 78 (which corresponds to up to one complete trading day). For the sake of brevity, we only present results for models with $k = 1$ and with $k = 78$ lags (since values in between did not improve the forecasting results). For each of those, we consider two model variants.

Simple autoregressive AR(k) models. In the first variant, we directly model and forecast intraday transaction costs without modelling or forecasting the seasonal component. For a given trading day d and five-minute interval t, the autoregressive model of order k for $k \leq 78$, also referred to as *AR(k)* is estimated as:

$$MCI_{d,t} = \begin{cases} \beta_0 + \sum_{i=1}^{t-1} \beta_i MCI_{d,t-i} + \sum_{i=0}^{k-t} \beta_{t+i} MCI_{d-1,78-i} + \varepsilon_{d,t} & \text{if } t \leq k \\ \beta_0 + \sum_{i=1}^{k} \beta_i MCI_{d,t-i} + \varepsilon_{d,t} & \text{if } t > k, \end{cases} \tag{6}$$

where β_i for $i = 0, 1, \ldots, k$ are the parameters to be estimated. In particular, β_0 is the constant offset, while the other coefficients represent the correlation with the last k lags, and $\varepsilon_{d,t}$ is the error term. As one can observe, an $AR(k)$ autoregressive model bases its forecasts exclusively on the k previous five-minute intervals. Here, the model is expected to implicitly capture, to a certain extent, the seasonal component.

Under these assumptions, a forecast for time interval t for $t = 1, 2, \ldots, 78$ at day $d+1$ from the AR(k) model with $k \leq 78$ is then obtained by computing:

$$\widehat{MCI}_{d+1,t} = \begin{cases} \hat{\beta}_0 + \sum_{i=1}^{t-1} \hat{\beta}_i \widehat{MCI}_{d+1,t-i} + \sum_{i=0}^{k-t} \hat{\beta}_{t+i} MCI_{d,78-i} & \text{if } t \leq k \\ \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i \widehat{MCI}_{d+1,t-i} & \text{if } t > k, \end{cases} \tag{7}$$

where $\hat{\beta}_i$ for $i = 0, 1, \ldots, k$ are the constant offset and the coefficients of the $AR(k)$ model as estimated in Equation (6). It is easy to see from the above equations that the forecasts for day $d+1$ are static forecasts in the sense that they do not incorporate the real values from day $d+1$, but only their predictions. To be more precise, $\widehat{MCI}_{d+1,t}$ does not depend on $MCI_{d+1,t'}$ for $t' < t$ and is computed using forecasts of the previous five-minute intervals in the same day, i.e. $\widehat{MCI}_{d+1,t'}$ for $t' < t$ and $d+1$, or the real observed transaction costs $MCI_{d,t}$ for from the previous day d.

Seasonality-adjusted autoregressive SAR(k) models. In the second model variant, we first remove the seasonal component from intraday transaction costs and then use autoregressive models to estimate and forecast the non-seasonal component $\widetilde{MCI}_{d+1,t} = MCI_{d+1,t} - \overline{MCI}_{d+1,t}$. Specifically, we estimate the model in Equation (6) by replacing $MCI_{d,t}$ with $\widetilde{MCI}_{d,t}$. We obtain the forecasts of $\widetilde{MCI}_{d+1,t}$, denoted by $\widehat{\widetilde{MCI}}_{d,t}$, using Equation (7). The final forecasts of $MCI_{d+1,t}$, i.e. $\widehat{MCI}_{d+1,t}$, are then computed by adding back the seasonal component to the forecasts for the non-seasonal components: $\widehat{MCI}_{d+1,t} = \widehat{\widetilde{MCI}}_{d+1,t} + \overline{MCI}_{d+1,t}$.

Estimation of different AR models were performed on the Cedar cluster provided by Compute Canada (implying the parallel use of at most 10 CPUs of 2.1 Ghz and a total maximum of 20Gb of RAM), using statsmodels (Seabold and Perktold, 2010).

3.3 Recurrent Neural Networks based Models

Long Short-Term Memory (LSTM). Long Short-Term Memory networks (Hochreiter and Schmidhuber, 1997) are an improved architecture of Recurrent Neural Networks (RNN) designed to model sequenced data such as time series. LSTMs were proposed to overcome the difficulties of RNNs to capture long-term dependencies, given that, for RNNs, the gradients of those dependencies tend to converge to zero, making gradient-based optimization method struggle (Chung et al., 2014).

We use a multivariate LSTM time series with input vectors $x_t = \left[MCI_{t,1} \; MCI_{t,2} \; \ldots \; MCI_{t,78}\right]^{\mathrm{T}}$, where t

refers to a previous day (in our experiments, we have used the previous 22 days). The long-short-term memory cell uses three mechanisms in order to control memory at each LSTM neuron: the input gate, the forget gate and the output gate. These gates allow the network to decide which information to retain and which to forget. The forget gate vector f_t decides which information to forget using the output vector h_{t-1} of the previous LSTM neuron and the input vector x_t (Equation (8)). The input gate i_t decides which information to update (Equation (9)) combined with the new memory content $\tilde{c}_t$ (Equation (10)), where $\circ$ represents the element-wise matrix multiplication (Hadamard product). The new memory cell c_t (cell state vector) is computed as the sum of the new memory content $\tilde{c}_t$ (cell input vector) and the existing memory cell c_{t-1} multiplied by f_t. Finally, the output h_t (Equation (13)) is computed using the output gate vector o_t (Equation (11)) to decide the amount of memory content to share multiplied by the hyperbolic tangent of cell state c_t (Equation (12)). Specifically:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{8}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{9}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{10}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{11}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{12}$$

$$h_t = o_t \circ \tanh(c_t), \tag{13}$$

where σ is the sigmoid function, tanh is the hyperbolic tangent function, and W, U, b are the weight matrices and bias vector, respectively, which are estimated while the model is trained.

Gated Recurrent Units (GRU). Gated Recurrent Units (Cho et al., 2014) are similar to LSTMs, but have fewer parameters. As the LSTM, it uses input vector $x_t = \begin{bmatrix} MCI_{t,1} & MCI_{t,2} & \dots & MCI_{t,78} \end{bmatrix}^{\mathrm{T}}$, where t refers to a previous day with 78 entries, one for each five-minute interval. GRUs use two mechanisms to control memory: an update gate and a reset gate. The update gate vector u_t decides which information is passed along the next state using the output vector h_{t-1} of the previous GRU unit and the input vector x_t (Equation (14)). The reset gate vector r_t decides which information should be neglected (Equation (15)). The current memory content $\hat{h}_t$ (output activation vector) is computed using the previous output vector h_{t-1} and the reset gate r_t (Equation (16)). Finally, the ouput vector h_t is computed using the current memory content $\hat{h}_t$, the previous output vector h_{t-1} and the update gate vector u_t. Specifically:

$$u_t = \sigma(W_u x_t + U_u h_{t-1} + b_u) \tag{14}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{15}$$

$$\hat{h}_t = \tanh(W_h x_t + U_h (r_t \circ h_{t-1}) + b_h) \tag{16}$$

$$h_t = (1 - u_t) \circ h_{t-1} + u_t \circ \hat{h}_t, \tag{17}$$

where, again, σ is the sigmoid function, tanh is the hyperbolic tangent function, and W, U, b are the weight matrices and bias vector, respectively, which are estimated while the model is trained.

Hyper-parameter selection. The design of the above LSTM and GRU networks are determined by several hyper-parameters, which are typically tuned to the application context and data in order to ensure a stable predictive performance. A grid search over all different combinations of such parameters is computationally prohibitive. We therefore carried out a random search (Bergstra and Bengio, 2012) over the following hyper-parameters (Table 3), where the final selection used in our models is indicated in bold:

Table 3: Hyper-Parameter Values

Hyper-Parameter	**Values**
Learning Rate	**0.1**; 0.01; 0.001
Number of Epochs	1; 5; **10**; 20; 50; 100
Batch size	1; 12; **78**; 156
Hidden Layers	1; **2**
Hidden Units,	12; **78**; 156; 312
Activations Functions	**Tanh**; Relu; Sigmoid; Selu
Dropout	0.0; 0.1; 0.2; **0.3**; 0.4; 0.5

Time series modeling, training and model evaluation. The representation of time series may have a major impact on the predictive performance of LSTM and GRU (Iwok and Okpe, 2016). Throughout our experiments, we have modelled the LSTM and GRU as both a univariate time series (using information of the previous day only) and a multivariate time series (using several previous days). We focus on the latter, given that these models provided superior results. To be specific, we consider a total of H previous days. Each day d is represented as an input variable $\boldsymbol{MCI}_d^H$ and the number H of previous days to be considered, where $\boldsymbol{MCI}_d^H = \left[\boldsymbol{MCI}_{d-H}\ \boldsymbol{MCI}_{d-H+1}\ \ldots\ \boldsymbol{MCI}_d\right]$ and $\boldsymbol{MCI}_d = \left[MCI_{d,1}\ MCI_{d,2}\ \ldots\ MCI_{d,78}\right]^{\mathrm{T}}$.

The models used in this paper have been trained using data from the 22 previous days, i.e. we set $H = 22$. To evaluate the predictive performance of machine learning methods, splitting the data set into training, test and validation sets via cross validation would typically be appropriate. However, when dealing with times series, one has to respect the temporal order of the data and the requirement of having sequential data in each of those sets. Therefore, other evaluation approaches such as nested cross-validation methods or rolling-origin evaluation (Tashman, 2000) have been proposed. In the latter, the training set first contains a sequence at the beginning of the time series, and the validation and test sets are subsequent data points. In the next iteration, the test set is added to the training set, and the new test and validation sets are again the subsequent data points. Given that such expanding training windows can become rather large, such methods may be computationally demanding. We therefore adopt a sliding training window approach, in which the size of the training set remains fixed. A graphical illustration of both approaches can be found in Figures 7 and 8 in Appendix B. All hyper-parameters previously introduced are selected such that the performance on the validation sets is best. The final model is then blindly applied to the test sets to evaluate the model performance. A detailed pseudo-code of the training and prediction process of the LSTM and GRU can be found in Algorithm 1 in Appendix B. All LSTM and GRU models were trained on the Cedar cluster provided by Compute Canada (as described in the previous section), using Keras (Chollet, 2015) with Tensorflow (Abadi et al., 2015).

4 Forecasting Results

In this section, we present the statistical performance of different models in forecasting intraday transaction costs. Before discussing different statistical performance metrics, several remarks are in order regarding the estimation/training of different forecasting models. As previously mentioned, all models are estimated recursively using a 22-day rolling window of intraday observations for $MCI^A_{d,t}$ and $MCI^B_{d,t}$ separately. Furthermore, we update the parameter estimates and/or retrain the model at the end of each trading day. More importantly, we do not use any data from the day we are forecasting in the sense that our forecasts are static and not dynamic. In other words, we assume that the trader obtains her forecasts at the beginning of the trading day and does not update these forecasts even though she observes transaction costs during the day. In turn, the static nature of these forecasts implies that our results provide a lower bound of the forecasting ability of a trader. That is, a trader may obtain better forecasting results if she considers dynamic forecasts and/or re-estimates/re-trains the model throughout a given trading day as she observes transaction costs during that day.

To evaluate the statistical performance of the models, we consider two standard performance metrics, namely RMSE and MAPE, defined as follows:

$$RMSE_{d+1} = \sqrt{\frac{\sum_{t=1}^{78}(MCI_{d+1,t} - \widehat{MCI}_{d+1,t})^2}{78}} \tag{18}$$

$$MAPE_{d+1} = \frac{1}{78}\sum_{t=1}^{78}\left|\frac{MCI_{d+1,t} - \widehat{MCI}_{d+1,t}}{MCI_{d+1,t}}\right|. \tag{19}$$

4.1 Results for All Stocks

Table 4 presents the summary statistics for RMSE and MAPE across all stocks and five-minute intervals for different models in forecasting the intraday ask side transaction costs (MCI_A) in panel (a) and bid side transaction costs (MCI_B) in panel (b).

Table 4: Performance of Different Models in Forecasting MCI_A and MCI_B

Panel (a)

	RMSE		MAPE	
Model	Median	IQR	Median	IQR
22-day Avg.	0.079	0.229	32.74	35.05
Adj. 22-day Avg.	0.060	0.224	28.41	28.56
$AR(1)$	0.059	0.253	29.23	29.91
$SAR(1)$	0.090	0.316	47.62	52.56
$AR(78)$	0.075	0.295	39.09	43.50
$SAR(78)$	0.072	0.257	38.09	36.61
LSTM	0.115	0.460	40.45	63.55
GRU	0.120	0.506	43.02	73.19

Panel (b)

	RMSE		MAPE	
Model	Median	IQR	Median	IQR
22-day Avg.	0.066	0.239	33.33	36.54
Adj. 22-day Avg.	0.060	0.200	29.22	29.99
$AR(1)$	0.063	0.204	29.62	30.37
$SAR(1)$	0.089	0.280	44.98	46.87
$AR(78)$	0.079	0.274	40.41	44.85
$SAR(78)$	0.071	0.229	37.25	33.90
LSTM	0.069	0.245	33.82	34.13
GRU	0.071	0.256	35.92	42.12

Note: This table presents summary statistics for Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) of different models in forecasting MCI_A in panel (a) and MCI_B in panel (b). The IQR is the interquartile range computed as the difference between the 75% and 25% quantiles of the distribution of RMSE or MAPE. The summary statistics are computed over all stock-day pairs in our sample between January 2002 and December 2012.

The forecast based on the averages over the last 22 days, which is our benchmark model, performs relatively well for such a simple model. To be more precise, it has a median MAPE of 33.33% for MCI_B and 32.74% for MCI_A suggesting that it explains 70% of the variation in intraday transaction costs. As mentioned above, this relatively good performance of a simple model suggests the seasonal component plays an important role in the intraday time variation of transaction costs.

More importantly, the proposed adjustment to this simple benchmark to capture daily time variation in intraday seasonality performs better than all other models considered. Specifically, the RMSE and MAPE of the model not only have the lowest median but are also much less dispersed as measured by their standard deviations. The MAPE of this model has a median of 29.22% showing that more than 70% of the absolute variation in intraday transaction costs is captured by this model.

The AR(1) model is a close second best with RMSE and MAPE close to the best model, i.e the Adj. 22-day Avg. It has a median MAPE of 29.62% compared to a MAPE of 29.22% for the Adj. 22-day Avg. However, its RMSE and MAPE have higher standard deviations, suggesting that its performance exhibits much more variation across stocks, five-minute intervals and trading days. On the other hand, the SAR(1) model, which models the seasonal component separately, performs worse than the AR(1) model itself. The difference in their performances is not negligible. For example, the SAR(1) model has a median MAPE of around 45% compared to 30% for the AR(1) model.

Increasing the number of lags in the AR model makes the performance worse for the simple AR models. To be more precise, AR(78) performs much worse than AR(1). This is mostly due to the fact that forecasts from the AR(78) model depend closely on the transaction costs from the day before and there is too much variation in the daily transaction costs. On the other hand, the static forecasts from the AR(1) model converge to its mean relatively quickly and behave like the forecasts from the 22-day Avg. The SAR(78) model performs relatively better than the AR(78) model.

Finally, the two advanced machine learning models, LSTM and GRU, perform relatively poorly. When predicting MCI_A, the median RMSE of these two models are higher than all other models considered. Their performance improves when we consider predicting MCI_B. For example, their median MAPEs are respectively 33.82% and 35.92% which are better than SAR(1), AR(78) and SAR(78) and not much worse than other competing models. These results suggest that advanced machine learning models can be useful in forecasting the bid side transaction costs. That said, the advanced machine learning models exhibit much larger standard deviation when predicting both MCI_A and MCI_B suggesting that they are less consistent in terms of their performances compared to simple models such as averages or AR(1) model. The poor performance of these more advanced models is due to the large number of parameters to be estimated and our choice of using the same number of trading days (i.e., 22 days) as other models to train these machine learning models.

4.2 Results for Stocks with Different Market Capitalizations

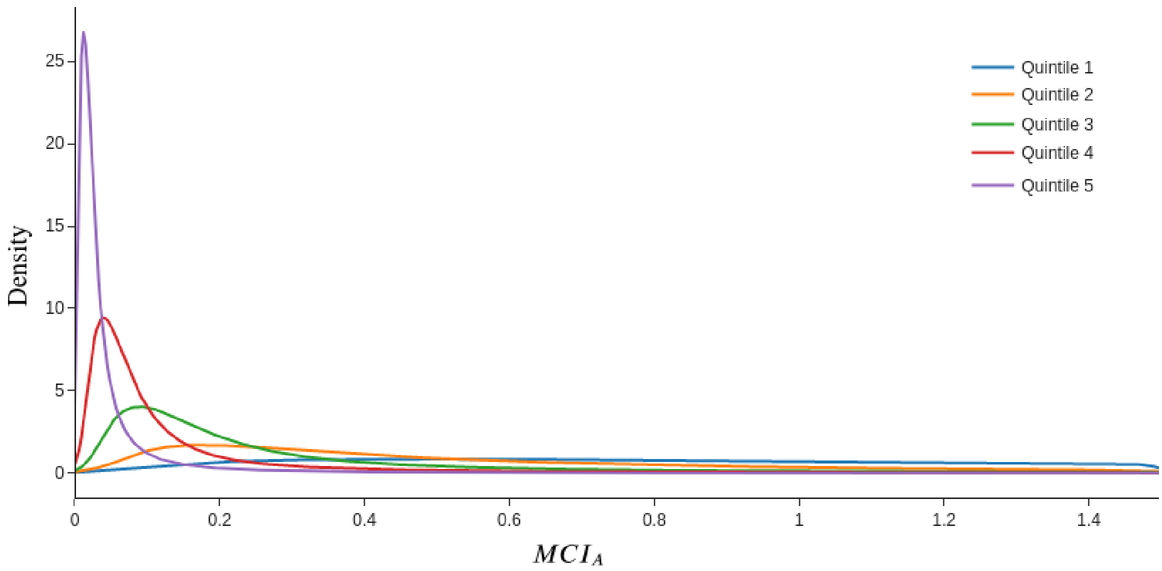
Our conclusions of the performances of different models so far are based on averages over all stocks and trading days in our sample. Transaction costs exhibit significant variation not only over time (see Figure 2), but also over stocks. In this subsection, we analyze the performance of different models in forecasting intraday transaction costs when we distinguish between stocks with different market capitalizations.

As it is well known, market capitalization is closely related to the liquidity of a stock and the transaction costs faced by a trader trading this stock. Large cap stocks tend to be more liquid and thus cheaper to trade while small-cap stocks are less liquid and more expensive to trade. To this end, we group the stocks in our sample into five categories based on their market capitalizations. To be more precise, we compute the market capitalization of each stock in our sample for a given day $d-1$. We sort stocks based on their market capitalizations and group them into five quintiles from stocks with smallest 20% market capitalizations to the largest 20%. We then compute the distribution of intraday transaction costs over all trading days and stocks in that group. Figure 6 presents the distribution of MCI_A and MCI_B for different market capitalization, respectively. The results are in line with the findings of the previous literature on the liquidity of stocks with different market capitalizations.

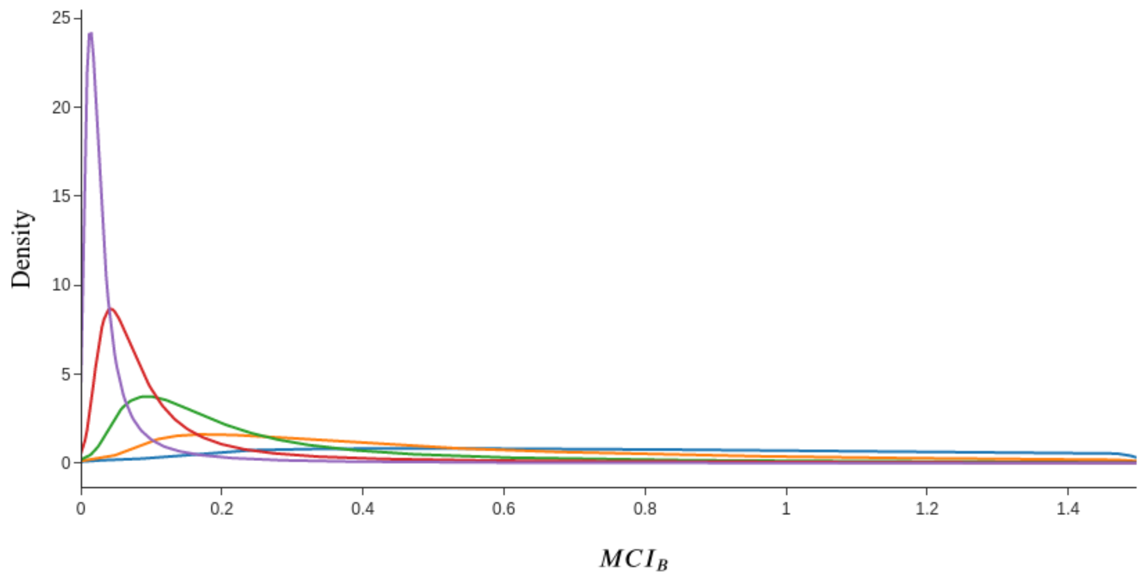
Specifically, both MCI_A and MCI_B have the lowest mean for largest stocks and decrease monotonically with decreasing market capitalization. Furthermore, the distribution of MCI_A and MCI_B have the lowest dispersion for largest stocks while their dispersion increases as we consider stocks with smaller market capitalizations.

Figure 6: Distribution of MCI_A and MCI_B for Stocks with Different Market Capitalizations

Panel (a)



Panel (b)



Note: This figure shows the distribution of MCI_A in panel (a) and MCI_B in panel (b) for different size quintiles. We group the stocks in our sample into five categories based on their market capitalization at day $d-1$. Quintile 1 includes the smallest stocks, quintile 5 the largest. The x-axis shows the values of MCI and y-axis shows the corresponding density.

Overall, these findings suggest that the distributions of MCI_A and MCI_B change significantly with market capitalizations. Hence, different forecasting models might have different performances for stocks with different market capitalizations. To investigate the impact of market capitalization on model performance, we compute the summary statistics for different performance measures across quintiles of market size. Table 5 summarizes these results for both performance measures. The overall tendency of predictive performance among the models is similar regardless of the market capitalization category considered and similar to our main sets of results in Table 4. To be more precise, we find that the adjusted 22-day average is the best performing model for both MCI_A and MCI_B and for most market capitalization quintiles. The only exception to this is when we predict MCI_A for stocks with largest market capitalization, where AR(1) slightly outperforms the adjusted 22-day average based on median MAPE. AR(1) is also very close second best for all other size quintiles and for both MCI_A and MCI_B. The benchmark model, namely the simple 22-day average, also performs relatively well, although always beaten by both the adjusted 22-day average and AR(1) models. All other more advanced time series or machine learning models perform much worse than the two best performing models. This in turn suggests that these models do not provide much value in terms of statistical power for forecasting intraday transaction costs, even when we focus on stocks with different capitalizations.

The results in Table 5 also show that the median MAPEs for largest stocks are almost always lower than those for the small stocks regardless of the forecasting model. For example, the median MAPE for the 22-day Avg. model decreases monotonically with market capitalization. For other models, the median MAPE exhibits more like a smirk shape, where it is high for small stocks, low for mid cap stocks and then slightly increases for large stocks. These results, in turn, suggest that it is easier to forecast transaction costs for mid cap and large cap stocks than for small cap stocks.

Table 5: MAPE of Different Models in forecasting MCI_A and MCI_B for Stocks with Different Market Capitalizations

Panel (a)

Model	Small Cap	Quintile 2	Quintile 3	Quintile 4	Large Cap
22-day Avg.	36.84	33.93	33.58	31.32	31.67
Adj. 22-day Avg.	31.09	28.60	28.61	27.24	28.20
$AR(1)$	32.36	29.65	29.43	27.86	29.01
$SAR(1)$	45.47	44.37	47.02	47.69	49.93
$AR(78)$	43.61	39.41	39.33	37.22	38.64
$SAR(78)$	36.81	36.02	37.49	37.65	39.97
LSTM	43.77	40.8	41.69	38.88	39.63
GRU	45.43	43.27	43.52	42.06	42.48

Panel (b)

Model	Small Cap	Quintile 2	Quintile 3	Quintile 4	Large Cap
22-day Avg.	38.24	34.62	33.76	31.48	32.56
Adj. 22-day Avg.	31.62	29.33	29.04	27.80	29.56
$AR(1)$	32.68	29.88	29.45	28.10	29.79
$SAR(1)$	44.95	43.23	44.37	44.04	46.64
$AR(78)$	45.25	40.99	40.23	38.25	40.44
$SAR(78)$	36.97	35.68	36.29	36.33	39.23
LSTM	45.11	41.76	40.65	38.32	40.69
GRU	46.92	44.31	43.07	41.62	45.08

Note: This table presents the median MAPE for different models in forecasting MCI_A in panel (a) and MCI_B in panel (b). Small Cap is the stocks with lowest 20% market capitalizations on the day the forecasts are obtained. Other quintiles are defined similarly.

5 Best Order Execution

Having shown that we can predict intraday transaction costs especially well when accounting for its seasonality, we now turn our attention to the economic value of these predictions for traders of large positions. We have in mind a trader who needs to trade a significant dollar amount of a given stock in a given trading day. This problem is faced by traders on a daily basis to fulfill the buy or sell requests of their investors by the end of the trading day. Predicting intraday transaction costs can have value to traders because it enables them to reduce transaction costs by trading more at times when trading is less expensive.

In this paper, we break down the trading day into five-minute intervals. We assume that the trader's performance is measured based only on how well she minimizes her expected total transaction costs and not based on any other objective such as minimizing her tracking error. Thus, the trader needs to minimize her expected total transaction costs by choosing the amount to trade every five minutes so that she trades a total dollar amount of A by the end of the trading day. This problem can be expressed as follows:

$$\min_{D_1,\dots,D_{78}} \quad E_0[\sum_{t=1}^{78} MCI_{d,t}{D_t}^2] = \sum_{t=1}^{78} E_0[MCI_{d,t}]{D_t}^2 \tag{8}$$

$$s.t. \quad \sum_{t=1}^{78} D_t = A \tag{9}$$

$$D_t \geq 0, \tag{10}$$

where $E_0[\cdot]$ denotes the expectations at time 0, i.e. the beginning of the trading day before observing any transaction costs of day d. A trader who needs to sell would have to use intraday buy side transaction costs, i.e. MCI_B, in the above optimization problem while a trader who needs to buy would have to use intraday ask side transaction costs, i.e. MCI_A.

By replacing the expectation of transaction costs $E_0[MCI_{d,t}]$ with the forecasts from a given model, i.e. $\widehat{MCI}_{d,t}$, we show in Appendix A that the optimal trading amount for each five-minute interval t is given as follows:

$$D_t^* = A \times \frac{1/\widehat{MCI}_{d,t}}{\sum_{t=1}^{78} 1/\widehat{MCI}_{d,t}}. \tag{11}$$

The intuitive way to think about this solution is as follows: The trader would trade an amount during the five-minute interval with the lowest transaction cost until the marginal cost of trading an additional dollar in that five-minute interval is equal to the marginal cost of trading an additional dollar in the five-minute interval with second-lowest transaction costs. Proceeding in this fashion, it is easy to see that the trader will allocate the amount to be traded to each five-minute interval such that the marginal costs of trading an additional dollar in each five-minute interval is equal. This gives the solution in Equation (11), where the optimal amount to trade is, not surprisingly, inversely proportional to the transaction costs.

Given a set of forecasts for intraday transaction costs, the trader chooses a set of dollar amounts to trade every five minutes throughout the trading day based on the Equation (11). We then compute the total transaction costs faced by the trader based on her optimal allocation decisions as follows:

$$\text{Cost} = \sum_{t=1}^{78} MCI_{d,t}{D_t^*}^2, \tag{12}$$

where $MCI_{d,t}$ is, as before, the realized (observed) transaction cost for the t^{th} five minute interval for $t = 1, 2, \dots, 78$ in trading day d. Note that trading costs increase quadratically with the amount traded. For instance, trading \$100M instead of \$10M will increase transaction costs 100 folds.

In the following, we compare the economic value, i.e., the reduction in trading costs, of a forecasting model based on this out-of-sample realized transaction cost. We therefore argue that the forecasting model which delivers the minimum total transaction cost is the most appropriate model to use. In addition to the Avg. 22-day forecast, we consider another benchmark model for this trading exercise. Specifically, we assume that the trader does not have information about the intraday transaction cost and allocates the same amount to trade in each five-minute interval, i.e. $D_t = A/78$ for $s = 1, 2, \dots, 78$. This is a rather simple benchmark, but still a realistic one since some traders might not have the time or tools to perform an analysis of intraday transaction costs for each stock they trade. We refer to this approach as the naive approach (*Naive App.*). We also compute the transaction costs assuming that the trader has perfect information about the upcoming day's transaction costs ex-ante, which we refer to as Perfect Knowledge Case (*Perfect Know.*). While this case is obviously unrealistic, it provides a lower bound on the possible level of total trading cost.

We define the total amount to be traded within a day in two ways. First, we simply use a fixed amount of 100 M$ for all stock-days. This allows us to analyze how each model performs when the same amount is traded for each stock-day pair. Of course, in relative terms, 100 M$ represents a large trading volume for a small stock with low liquidity but a small trading volume for a large stock with high liquidity.

To overcome this issue, we also use an amount that accounts for variations in trading volume across stock-days. We observe that across all stock-days in our sample, the median ratio of trading volume to market capitalization equals 0.639%. In other words, 0.639% of a stocks market capitalization is traded on a typical day. We compute the second amount to be traded on a given stock-day as 0.639% multiplied by the market capitalization of that stock at the end of the previous day.

Table 6 presents the summary statistics for the total trading costs in million dollars based on different forecasting models for a trader who needs to buy (i.e. facing the ask side transaction costs MCI_A) in panel (a) and a trader who needs to sell (i.e. facing the ask side transaction costs MCI_B). The numbers in the columns titled "Fixed Amount" can also be interpreted as percentage transaction costs because they represent the total transaction costs in million dollars for \$100M traded. For instance, a trading cost of \$1.24M represents 1.24% of \$100M.

Table 6: Summary Statistics for Different Order Placement Strategies under Alternative Scenarios

Panel (a)

	Fixed Amount		Proportional Amount	
	Median	IQR	Median	IQR
Perfect Know.	1.24	3.91	5.64	17.02
Naïve App.	1.60	5.20	7.12	23.43
22-day Avg.	1.38	4.56	6.26	19.56
Adj. 22-day Avg.	1.77	5.35	6.52	21.39
$AR(1)$	1.44	5.15	6.71	22.72
$SAR(1)$	1.81	6.33	8.17	28.67
$AR(78)$	2.13	11.78	9.86	58.09
$SAR(78)$	1.60	6.44	7.60	29.53
LSTM	1.90	5.84	8.81	27.56
GRU	1.93	6.04	8.99	28.42

Panel (b)

	Fixed Amount		Proportional Amount	
	Median	IQR	Median	IQR
Perfect Know.	1.41	4.79	6.37	17.72
Naïve App.	1.76	5.99	7.93	23.67
22-day Avg.	1.59	5.33	7.18	20.92
Adj. 22-day Avg.	1.64	5.46	7.41	21.94
$AR(1)$	1.68	5.52	7.76	24.33
$SAR(1)$	2.00	6.66	9.07	27.85
$AR(78)$	2.68	7.42	71.10	67.06
$SAR(78)$	1.88	6.08	31.90	28.62
LSTM	1.74	5.86	7.97	24.24
GRU	1.80	6.09	8.21	25.47

Note: This table presents summary statistics in million dollars (M \$) for order placement strategies based on different forecasting models for buying (i.e. facing the transaction costs on the ask side MCI_A) in panel (a) and selling (i.e. facing the transaction costs on the bid side MCI_B). Numbers under "Fixed Amount" represent total transaction costs for trading a fixed amount of 100 M\$ worth of a stock. These numbers can e interpreted as percentage transaction costs because they represent total transaction costs in million dollars for 100 M\$traded. Numbers under "Proportional Amount" represent total transaction costs for trading an amount proportional to the market capitalization of a stock. The proportional amount to be traded is computed on a given stock-day as 0.639% multiplied by the market capitalization of that stock at the end of the previous day. IQR is the interquartile range computed as the difference between the 75% and 25% quantiles of the distribution.

We start our discussion with results in panel (a). The median lower bound for the costs of buying a fixed or a proportional amount based on the Perfect Knowledge Case are \$1.24M and \$5.64M, respectively. These numbers are a theoretical lower boundary and total buying costs based on other models will be at least as high by definition. We will use the theoretical minimum as a reference point and discuss how much higher the trading costs of each model are with respect to this unattainable benchmark.

The naive approach of buying the same amount every five minutes results in total buying costs with a median of \$1.60M and an interquantile range (IQR) of \$5.2 M. These numbers are respectively 29% and 33% higher than the unattainable perfect knowledge case. These results are important since it provides a benchmark against which to measure the performance of other models. In other words, if we ignore intraday seasonality and potential predictability of transaction costs, one can do only about 30% worse than the perfect knowledge case. Among all the other models considered, the simple model of 22-day Avg. yields the best forecasts in terms of median total buying costs for both fixed and proportional amounts. Specifically, the median total cost of buying a fixed amount of \$100M worth a stock and an amount proportional to the market capitalization of a stock based on the forecast from the model of 22-day Avg. are respectively \$1.38M and \$6.26M. These numbers are only about 11% higher than the corresponding median buying costs based on the unattainable perfect knowledge case. More importantly, they represent a significant improvement compared to the naive approach. Furthermore, 22-day Avg. also has the lowest dispersion in terms of total buying costs for both fixed and proportional amounts. This is turn signifies that this model provides not only lower median total trading costs but also does so more consistently than any other forecasting model.

AR(1) is the second best-performing model based on median total buying cost of \$1.44M (16% higher than the Perfect Knowledge Case) for a fixed amount \$6.71M (19% higher than the Perfect Knowledge Case) for a proportional amount. Both of these numbers are lower than the corresponding total buying costs based on the naive approach. Furthermore, it also delivers lower dispersion in total trading costs than the naive approach. More importantly, it is the only other model that performs better than the naive approach for both fixed and proportional amounts.

All other models result in median total buying costs that are higher than the naive approach for either the fixed or proportional amount or both. Among these other models, one model's performance requires further attention. Specifically, the adjusted 22-day average, which is among the best-performing models in terms of predictive accuracy, here performs worse than not only the simple 22-day average, but also the naive approach when we consider trading a fixed amount. It performs better than the naive approach only when we consider buying an amount proportional to a stock's market capitalization.

This relatively poor economic performance of a statistically well-performing model is a well-known issue in the forecasting literature and is due to the differences in loss functions. In the statistical exercise, we evaluate performance based on the RMSE loss function, which is an increasing function of the forecast error made forecasting MCI. In the trading exercise considered in this section, we evaluate performance based on the total buying cost which is an increasing function of the forecast error made in forecasting the reciprocal MCI, i.e. $1/MCI$. In other words, a model may have a high predictive accuracy on average. However, if the time intervals with low transaction costs are not correctly identified (or if time intervals with high transaction costs are falsely predicted to have low transaction costs), such model may not perform well in our trading exercise.

The results for total selling costs presented in panel (b) are very similar to the results for buying presented in panel (a), with only few differences. The total selling costs are slightly higher than the corresponding buying costs. This is due to the fact that the MCI_B faced by a trader who wants to sell is, on average, higher than MCI_A faced by a trader who wants to buy in our sample. The 22-day Avg. model is still the model that delivers the lowest median trading costs for both fixed and proportional amounts. In contrast to the results in panel (a), the Adj. 22-day Avg. model performs better than AR(1) when we consider selling instead of buying. That said, AR(1) still performs better than the naive approach. Among all the other models, there is no model that performs better than the naive approach for both fixed and proportional amounts.[1]

Table 7 provides insights about how the economic value of forecasting varies in the cross-section. It shows the trading costs for the various forecasting approaches by size quintile. Panel (a) shows ask side costs (the cost of buying) in million dollars. The columns on the left labelled "Fixed Amount" show the cost of trading \$100M. Not surprisingly, transaction costs decrease drastically in size. In the benchmark case of "perfect knowledge" buying this amount implies extremely high median trading costs of 23.3% for the smallest stocks, 2.57% for mid-sized firms and 0.33% for the largest firms. The benefits of forecasting intraday trading costs accurately are accordingly largest for the smallest firms. Trading costs of the worst performing model (AR(78)) of 38.71% are 12.91% or \$12.91M higher than those of the forecasting model yielding the lowest cost of buying (22-day Avg.), which equal 25.90%. For the largest firms, that difference amounts to a mere 0.11%, or \$110,000.

Panel (b) shows these results for the bid side of the limit order book. While trading costs tend to be lower on the bid side than on the ask side, all of the cross-sectional patterns described previously can also be observed on the bid side.

[1] The LSTM model provides a median selling cost lower than the naive approach only when we consider selling a fixed amount of \$100M worth of a stock.

Table 7: Summary Statistics for Different Order Placement Strategies under Alternative Scenarios for Stocks with Different Market Capitalizations

Panel (a)

	Fixed Amount					Proportional Amount				
	Small Cap	Quintile 2	Quintile 3	Quintile 4	Large Cap	Small Cap	Quintile 2	Quintile 3	Quintile 4	Large Cap
Perfect Know.	23.30	6.45	2.57	1.14	0.33	1.25	2.35	4.02	7.97	45.35
Naïve App.	27.60	7.91	3.23	1.45	0.42	1.48	2.87	5.03	10.04	59.95
22-day Avg.	25.90	7.18	2.84	1.25	0.36	1.38	2.60	4.44	8.69	47.74
Adj. 22-day Avg.	26.92	7.40	2.91	1.27	0.37	1.43	2.67	4.55	8.85	51.81
$AR(1)$	27.92	8.13	3.22	1.37	0.38	1.49	2.82	4.77	8.96	49.58
$SAR(1)$	31.75	9.68	3.95	1.72	0.47	1.68	3.33	5.87	11.11	64.84
$AR(78)$	38.71	10.66	4.10	1.70	0.47	1.99	3.60	6.09	10.96	69.98
$SAR(78)$	29.83	8.91	3.57	1.53	0.42	1.57	3.06	5.31	9.98	57.52
LSTM	28.15	7.89	3.12	1.35	0.40	1.52	2.83	4.88	9.35	56.75
GRU	28.65	8.00	3.17	1.37	0.40	1.55	2.88	4.93	9.49	57.47

Panel (b)

	Fixed Amount					Proportional Amount				
	Small Cap	Quintile 2	Quintile 3	Quintile 4	Large Cap	Small Cap	Quintile 2	Quintile 3	Quintile 4	Large Cap
Perfect Know.	22.41	5.98	2.36	1.02	0.30	1.22	2.17	3.69	7.16	41.24
Naïve App.	26.37	7.18	2.87	1.26	0.38	1.42	2.60	4.49	8.74	52.71
22-day Avg.	24.88	6.68	2.61	1.12	0.34	1.34	2.41	4.09	7.85	46.42
Adj. 22-day Avg.	25.84	6.88	2.68	1.15	0.34	1.41	2.48	4.20	8.01	47.86
$AR(1)$	27.34	7.14	2.74	1.17	0.35	1.47	2.57	4.30	8.11	48.56
$SAR(1)$	30.37	8.29	3.27	1.42	0.42	1.62	2.98	5.13	9.78	59.85
$AR(78)$	41.24	9.83	3.63	1.49	0.45	2.07	3.46	5.68	10.26	73.63
$SAR(78)$	28.82	7.67	2.99	1.28	0.38	1.53	2.75	4.68	8.88	54.89
LSTM	27.16	7.23	2.83	1.23	0.37	1.48	2.60	4.41	8.52	52.35
GRU	27.82	7.43	2.91	1.25	0.38	1.51	2.66	4.54	8.71	54.07

Note: This table presents median total buying costs (i.e. facing MCI_A as transaction costs) in panel (a) and median total selling costs (i.e. facing MCI_B as transaction costs) in panel (b) in million dollars (M \$) for each size quintile. The columns under the heading "Fixed Columns" present the results for trading a fixed amount of 100 M\$. The columns under the heading "Proportional Amount" present the results for trading an amount proportional to the total turnover of a given stock. Small Cap is the stocks with lowest 20% market capitalizations on the day the forecasts are obtained. Other quintiles are defined similarly.

6 Conclusion

How to trade a large amount of a given stock over a given period of time is an important issue for traders in today's financial markets. The costs associated with executing large orders over a certain period of time can be substantial, especially for stocks and/or periods with limited supply of liquidity. This so-called best execution problem has attracted the attention of academics and practitioners alike as it has become clear that the standard approach of splitting large orders into equal-sized trades is often suboptimal. Most existing papers analyze the best execution problem theoretically. This theoretical literature needs to make assumptions about liquidity supply and its dynamics.

In this paper, we approach the best execution problem from a forecasting perspective. Specifically, we obtain the optimal amount to trade every five minutes during a trading day by minimizing the expected total transaction costs at the beginning of this day based on forecasts of transaction costs. To quantify the economic importance of this approach, we measure intraday transaction costs based on ultra high frequency limit order book data for NYSE stocks between 2002 and 2012. We document that intraday transaction costs exhibit significant seasonalities not only within a given day but also over the quarterly earnings cycle. We find that a simple moving average model and its adjusted version can forecast intraday transaction costs better than advanced autoregressive and machine learning models. More importantly, these simple models deliver significantly lower total transaction costs than advanced autoregressive and machine learning models as well as the naive approach of splitting the total order into equal-sized orders. More precisely, the median total cost of buying (selling) \$100M of a stock based on our forecast from the moving average model is \$1.38M (\$1.59M). These numbers are only 11% and 13% higher than the median buying and selling costs from the unattainable perfect knowledge case where the trader has perfect foresight of intraday transaction costs. They also represent a significant improvement over the naive approach which results in total buying and selling costs that are respectively 29% and 25% percent higher than the unattainable perfect knowledge case. Finally, we report that the cost of trading a fixed dollar amount decrease dramatically in market capitalization and document how the benefits of forecasting accurately vary with size.

In our approach, we make some simplifying assumptions. For example, we assume that the trader obtains her forecasts at the beginning of the trading day and do not update her forecasts as she observes transaction costs during the day. She can, of course, obtain better forecasts if she considers updating her forecasts and parameters as she observes data throughout the trading day. Furthermore, we restrict our analysis to the performance of the models that can be obtained exclusively using information on transaction costs and the corresponding time stamps. Some of the approaches may potentially benefit from accounting for exogenous variables (such as trading price and volume). These assumptions imply that our results provide a lower bound for the economic benefits of our approach. One can, of course, relax these assumptions to obtain better forecasts and decrease transaction costs further. We leave this to future research.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Almgren, R. and Chriss, N. (1999). Value under liquidation. *Risk*, 12(12):61–63.

Almgren, R. and Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–40.

Almgren, R. and Lorenz, J. (2007). Adaptive arrival price. *Journal of Trading*, 1:59–66.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305.

Bertsimas, D. and Lo, A. W. (1998). Optimal control of execution costs. *Journal of Financial Markets*, 1(1):1–50.

Brennan, M., Huh, S.-W., and Subrahmanyam, A. (2013). An analysis of the Amihud illiquidity premium. *Review of Asset Pricing Studies*, 3(1):133–176.

Brunnermeier, M. K. and Pedersen, L. H. (2005). Predatory trading. *Journal of Finance*, 60(4):1825–1863.

Carlin, B. I., Lobo, M. S., and Viswanathan, S. (2007). Episodic liquidity crises: Cooperative and predatory trading. *Journal of Finance*, 62(5):2235–2274.

Cenesizoglu, T. and Grass, G. (2018). Bid- and ask-side liquidity in the NYSE limit order book. *Journal of Financial Markets*, 38:14–38.

Chan, L. K. C. and Lakonishok, J. (1995). The behavior of stock prices around institutional trades. *Journal of Finance*, 50(4):1147–1174.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Chollet, F. (2015). Keras. https://github.com/fchollet/keras.

Chordia, T., Roll, R., and Subrahmanyam, A. (2001). Market liquidity and trading activity. *Journal of Finance*, 56(2):501–530.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Fruth, A., Schöneborn, T., and Urusov, M. (2014). Optimal trade execution and price manipulation in order books with time-varying liquidity. *Mathematical Finance*, 24(4):651–695.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huberman, G. and Stanzl, W. (2005). Optimal liquidity trading. *Review of Finance*, 9:165–200.

Iwok, I. and Okpe, A. (2016). A comparative study between univariate and multivariate linear stationary time series models. *American Journal of Mathematics and Statistics*, 6(5):203–212.

Kempf, A. and Mayston, D. (2008). Liquidity commonality beyond best prices. *Journal of Financial Research*, 31(1):25–40.

Kyle, A. S. (1985). Continuous auctions and insider trading. *Econometrica*, 53(6):1315–1335.

Lesmond, D. A., Schill, M. J., and Zhou, C. (2004). The illusory nature of momentum profits. *Journal of Financial Economics*, 71(2):349–380.

Lorenz, J. and Osterrieder, J. (2008). Simulation of a limit order driven market. *Journal of Trading*, 4(1):23–30.

Obizhaeva, A. A. and Wang, J. (2013). Optimal trading strategy and supply/demand dynamics. *Journal of Financial Markets*, 16(1):1–32.

Schied, A., Schöneborn, T., and Bank, D. (2008). Optimal basket liquidation with finite time horizon for CARA investors. *Preprint, TU Berlin*.

Schied, A., Schöneborn, T., and Tehranchi, M. (2010). Optimal basket liquidation for CARA investors is deterministic. *Applied Mathematical Finance*, 17(6):471–489.

Schoeneborn, T. and Schied, A. (2009). Liquidation in the face of adversity: Stealth versus sunshine trading. *SSRN Electronic Journal*.

Seabold, S. and Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, 16(4):437–450.

A Proof of the Optimal Trading Quantities

One replace $E_0[MCI_{d,t}]$ in the optimization problem described by Equations 8-10 with its corresponding static forecast $\widehat{MCI}_{d,t}$ from a given model. The optimization problem in Equations 8-10 can thus be rewritten as follows:

$$\min_{D_1,\dots,D_{78}} \quad \sum_{t=1}^{78} \widehat{MCI}_{d,t} {D_t}^2$$
$$s.t. \quad \sum_{t=1}^{78} D_t = A$$
$$D_t \geq 0$$

The Lagrangian of this optimization problem is given as follows:

$$\mathcal{L}(\mathbf{D}, \lambda) = \sum_{t=1}^{78} \widehat{MCI}_{d,t} {D_t}^2 - \lambda(\sum_{t=1}^{78} D_t - A)$$

where $\mathbf{D} = [D_1, D_2, \dots, D_78]^T$ is the vector of quantities to be traded.

Taking the derivative of the Lagrangian with respect to D_t and setting it equal to zero yields:

$$D_t^* = \frac{\lambda}{2\widehat{MCI}_{d,t}} \tag{20}$$

for $t = 1, 2, \dots, 78$. Plugging in the optimal quantities D_t^* in the equality constraint in Equation 9 yields:

$$\lambda = \frac{A}{\sum_{t=1}^{78} 1/2\widehat{MCI}_{d,t}}$$

Finally, plugging in the solution for λ in Equation 20 yields the optimal quantity in Equation 11.

B Implementation details for LSTM and GRU models

The pseudo-code below describes the training and prediction process for the Long-Short-Term-Memory and Gated Recurrent Unit models.

Algorithm 1: Rolling Multivariate LSTM and GRU

```
Input: MCI_A and MCI_B for all stocks from 2002 to 2012
Scale the dataset between 0 and 1
Reshape dataset as a matrix R^{78×N}
for S in Stocks do
    for i in [H:N] do
        Train set ← Dataset[][i:H+i-1]
        for j in [0:H-1] do
            input(Train set)← Train set[][j-H:j]
            output(Train set)← Train set[][j+1]
        end
        Validation set← Dataset[][H+i]
        Test set← Dataset[][H+i+1]
        Perform the Random search on the Validation set using hyperparameters from table 5
        Define BestModel as the Best Model found on the Random search
        Compute the prediction based on Validation set using BestModel
        for p in [0:78] do
            if Predictions[p]< min Train set[p][] then
                Predictions[p]← min Train set[p][]
            end
        end
        Rescale the predictions
        Compute the RMSE and MAPE
    end
end
Output: Predictions, RMSE and MAPE
```

Figure 7: Nested Cross-validation

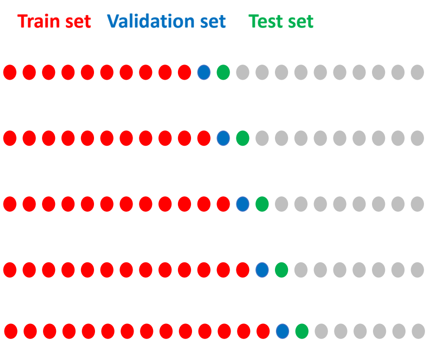


Figure 8: Sliding window Cross-validation