Characterizing the Informativity of Level II Book Data for High Frequency Trading

Logan B. Nielsen
Department of Computer Science
Brigham Young University
Provo, U.S.A
lbendtlynielsen@gmail.com

Sean Warnick
Department of Computer Science
Brigham Young University
Provo, U.S.A
sean@cs.byu.edu

Scott S. Condie
Department of Economics
Brigham Young University
Provo, U.S.A
scott_condie@byu.edu

***Abstract*—High-Frequency Trading (HFT) algorithms are automated feedback systems that interact with markets to maximize investment returns. These systems can process varying resolutions of market information at any given time, with Level I information providing basic equity data—primarily its price—and Level II information offering the complete order book for that equity at a specific moment. This paper presents a study on the application of Recurrent Neural Network (RNN) models to predict the spread of the DOW Industrial 30 index traded on NASDAQ, using both Level I and Level II data as inputs.**

The results suggest that the use of raw Level II data, without preprocessing, does not enhance spread prediction. This finding implies that HFT algorithms should not directly utilize raw Level II information; instead, computational resources should be allocated towards transforming Level II data into a less noisy signal if it is to improve trading performance. Alternatively, when computational resources are limited, HFT algorithms may benefit from omitting Level II data.

***Index Terms*—High Frequency Trading, Data Informativity, Recurrent Neural Network, Spread Prediction, Limit Order Book**

I. Related Works and Introduction

In the past, equities markets were handled by humans on exchange floors who would take and fulfill orders. With the transition to digital systems, a new regime of trading emerged: high-frequency trading (HFT). HFT uses low-latency algorithms to profit from market states that are ephemeral to non-HFT trading strategies. The NASDAQ estimates that 50% of stock trading volume in the United States is driven by HFT [1]. A new form of arbitrage, called latency arbitrage, emerged during this time. This form of arbitrage occurs when there is a latency between exchanges of a price change. For example, if the price of an S&P 500 future contract changes significantly in Chicago, this triggers a race around the world to capture stale quotes of assets highly correlated with the S&P 500. At the turn of the 21st century, these races were measured in milliseconds, but are now measured in microseconds and nanoseconds [2].

As HFT has become prominent in markets, many HFT trading strategies have been developed including spread capturing, market-neutral arbitrage, short-term momentum strategies, and cross-market arbitrage [3]. Much work has been done on the effect of HFT in markets. A review of approximately 100 papers on the role of HFT in markets is provided in [4], including the effect of HFT on volatility, transaction costs, liquidity provision and consumption, price discovery, flash crashes, and profitability. HFT has also been studied extensively through a control-theoretic perspective (see Figure 1). Work in this area includes optimal selling rules investigated in [5], optimal pair trading is studied in [6] and [7], and a transaction-level price model is tested in [8].

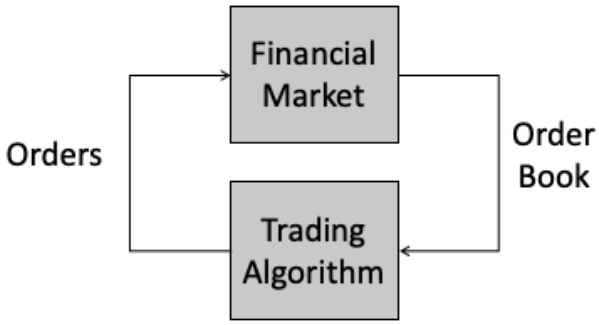

Fig. 1. Automated trading algorithms use information about the order book for each equity to make decisions about new orders to place; well-placed orders result in profitable executions. Nevertheless, two resolutions of order book information are available. Level II information is the full book, while Level I information is a minimal subset essentially just describing the current price of the equity. While it may seem intuitive that more information would be better for making investment decisions, the results of this study suggest that Level II information is no more informative than Level I information when the trading algorithm is making trades faster than 100 milliseconds and may decrease model performance if not preprocessed strategically.

In finance, Artificial Neural Networks (ANNs) have been used for various applications including estimating important financial theoretic concepts such as estimating the structure of the stochastic discount factor [9] and detecting statistical arbitrage [10]. Outside of the field of financial theory, deep reinforcement learning applications in automatic stock trading algorithms are also a prominent research topic [11] [12].

While HFT has been investigated by control theorists and ANNs have been utilized in financial theory and automated trading algorithms, little has been done with ANNs in mea-

suring the the informativity of HFT data. Here we compare the performance of recurrent neural network (RNN) models trained on Level I order book data to models trained on Level II order book data in predicting the bid-ask spread for DOW stocks traded on the NASDAQ. We use messaging data provided by the NASDAQ to reconstruct the order book and compare how these models perform when used to predict spread at varying distances into the future (from 1 millisecond to 10 seconds). This provides a characterization of how informative Level II information is, relative to Level I information, for predicting the bid-ask spread at different moments in the future.

II. Stock Exchange Data

Stocks and other assets such as cryptocurrencies are bought and sold on exchanges. Two of the most popular U.S. equities exchanges include the world's first electronic exchange, the National Association of Securities Dealers Automated Quotations (NASDAQ), as well as the New York Stock Exchange (NYSE). A popular cryptocurrency exchange is provided by Coinbase and was originally launched as the Global Digital Asset Exchange (GDAX) but was renamed to Coinbase Pro.

Exchanges act as brokers for exchanging assets between many clients, including individual and institutional investors. On these exchanges, trades are done electronically and require interested participants to send electronic messages to the exchange. The exchange processes these messages and matches buyers and sellers using a matching engine. Sellers and buyers subscribe to output channels provided by the exchange to know the state of the market and their orders. Information received through these channels are referred to as *messages*. There are various types of messages that can be sent to exchanges and propagated to all interested parties. There are core operations that, while their representation may vary exchange to exchange, are fundamental to exchanges which we call: *bid*, *ask*, *buy*, and *sell*.

A. Messaging Data

While there are many forms of financial data that investors use to inform their trades, such as historical price and volume data, these data are ultimately the result of aggregated messaging data. For example, price can be considered the average of the highest bid and lowest ask while trading volume is the total value of all sells and purchases. Thus, messaging data is the most granular form of exchange data available to an investor.

An example of what some messaging data that may look like on an exchange is provided in Table I, where rows represent message data. Here, the first message represents a request to sell 100 shares of Google stock at $450. Since this is a request for selling, this order will go on the *ask* side of the order book (see II-B). Because this order is added to the book instead of consuming orders that are already on it, this is a *liquidity providing event* or *market making order*. As this is an ask instead of a sell this order is not fulfilled right away. This seller must wait for an interested buyer.

The second message in the table represents a buy request for 133 shares of Yahoo stock. This execution request is on the ask side of the book and will be filled immediately by the exchange's matching engine. Not all these purchases will necessarily occur at the same price. The exchange will provide the investor with the lowest priced 133 shares of Yahoo stock. This removes orders on the book, so these types of orders are referred to as *liquidity taking events* or *market taking* orders.

The final row in the table represents a message from a seller that is willing to purchase 520 shares of Exxon Mobil stock at $112. This type of order is referred to as a bid and is the buy side analog of an ask. It is a liquidity providing event that will begin to be fulfilled once $112 is the best price offered to sellers.

TABLE I
Level I Data Example

Type	Time	Order ID	Side	Price	Shares	Ticker
A	6505	38776	S	450	100	GOOG
E	7574	0	S	-	133	Y
B	8120	57914	B	112	520	XOM

B. The Limit Order Book (LOB)

The *limit order book* (LOB) is a collection of all unfulfilled *bid* and *ask* messages at a particular point in time. It can be determined by aggregating all the messaging data for the day up to the point in time for which the book is desired. A simple order book at a specific moment in time is shown in Figure 2. A representation of the Order Book throughout an entire day is shown in Figure 3.

Important features of the book include the *highest bid*, *lowest ask*, and the *spread* (the difference between the lowest ask and highest bid). The spread is also an important feature for traders because it indicates how much value can be captured by being a liquidity provider. In our simple order book, if a trader were to put a single share for purchase at $4 and request to purchase a single share at $6 and have both orders fulfilled, then the traders position on net would be unchanged but the value in the trader's account would increase by $2.

Level I quote data consists of the highest bid and lowest ask prices, as well as the depth (size) of the bid and ask. Table II provides an example of Level I data.

TABLE II
Level I Data Example

Time	Bid	Ask	B0	A0
103,561	15	16	10	15
103,565	15	16	5	15
105,123	15	17	5	25

At time 103,561 the bid is at $0.15 with a depth of 10 shares and the ask is at 16 cents with a depth of 15 shares. After 4 units of time elapsed, 5 shares were either sold at

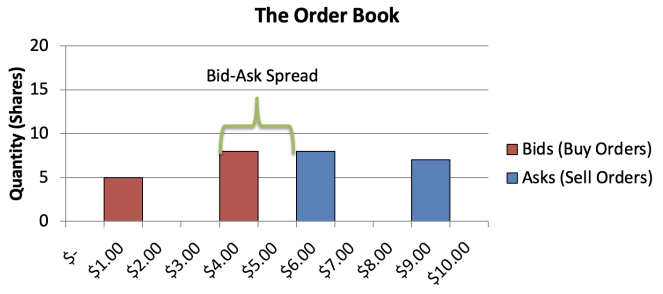

Fig. 2. Cross-Sectional Order Book Visualization. The red bars represent unfulfilled bids while the blue bars represent unfulfilled asks. Because the difference between the highest price a market maker is willing to pay is $4 and the lowest price a market maker is willing to pay is $6, the spread is said to be $2. On the NASDAQ, the smallest the spread of a security can be is $0.01. In economic theory, when the spread is not the smallest possible value there is market uncertainty about what the price of the asset should be. This is intuitive because one would expect to be able to buy a good and instantaneously resell it for the same price.

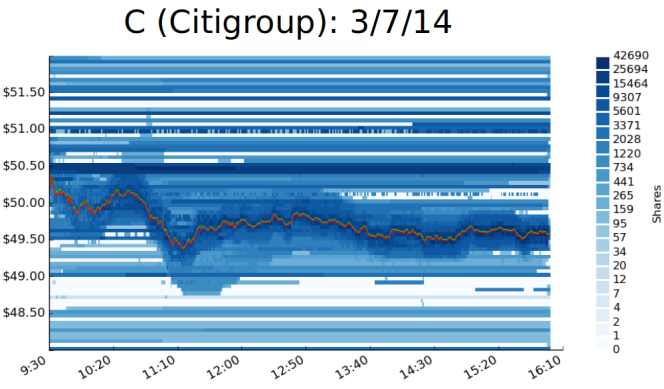

Fig. 3. Time Series Order Book Visualization. The red line represents the bid, the green line represents the ask, and the shades of blue represent the depths at various prices.

$0.15 or 5 shares being offered at $0.15 were cancelled. Some time later, 15 shares were either canceled from being sold at $0.16 or at least 15 shares were purchased. Assuming this event was a market order, with this data we do not know if that execution walked the book because we don't have access to the depth of the book at $0.17 *prior* to the execution.

Level II quote data is a superset of Level I data. In addition to Level I data, it includes depths and prices of the next 10 best bid and offer prices. Table III illustrates the difference between Level I and Level II information. For clarity, we only include the depth at $0.01 below the current bid and $0.01 above the current ask.

This pseudo Level II trader has more information than the Level I trader. Like the Level I trader, she sees at 103,565 that 5 bids at $0.15 were either cancelled or filled. Unlike the Level I trader, she sees a 3 unit increase in demand for the stock at time 104,372 at $0.14. Furthermore, she knows that the price moving event at 105,123 was caused by a market order that walked the book. In particular, this

TABLE III
PSEUDO LEVEL II DATA EXAMPLE

Time	Bid	Ask	B0	A0	B1	A1
103,561	15	16	10	15	17	28
103,565	15	16	5	15	17	28
104,372	15	16	5	15	20	28
105,123	15	17	5	25	20	30

execution consumed all the available stocks at $0.16 and 3 of stocks available at $0.17. A trader with access to Level II data has more information over the same interval of time compared to a trader who only has access to Level I data.

III. EXPERIMENTAL DESIGN

A. Objective

The goal of our study is to quantify the statistical power gained by predicting the spread of a security some time in the future when using Level II data instead of Level I data. This problem is, at its core, an identification problem. A randomized control trial (RCT) is the gold standard to identify the causal effect of a treatment [13]. We use the principles behind an RCT to conduct a paired experiment. In this experiment we keep as many factors as possible consistent while training a pair of comparable model on Level I and Level II data from the Nasdaq. We decided to use the Dow Jones Industrial Average (DJIA), stocks that form a broad-based index [14], because it is untenable to train models for all stocks on all trading days.

Sampling involved randomly selecting three contiguous trading days from the DJIA across the 2018 calendar year. The stocks that were in the DJIA in 2018 are in Table IV. Note that in 2018 General Electric was replaced by Walgreens Boots Alliance. We include both in our dataset.

B. Problem Formulation

Given some security, let s be the spread, and X^1 and X^2 denote Level I and Level II data, respectively, over some fixed time interval. Our goal is to approximate function $f(X)$ in Equation 1.

$$s_{t+\Delta} = f(X_t) + \epsilon_t \tag{1}$$

Delta (Δ) is a fixed amount of time in the future and ϵ_t is noise in the signal. Figure 4 provides a visualization for this problem.

A metric is needed to determine how well the function $f(X_t)$ approximates $s_{t+\Delta}$. A standard metric for model performance is mean squared error (MSE) given in equation 2, where $\hat{f}(\cdot)$ is an approximation of $f(X_t)$.

$$\text{MSE} = \sum_{i=1}^{N} (s_{t+\Delta} - \hat{f}(X_t))^2 / N \tag{2}$$

It is conceivable that the distribution of ϵ for each stock could differ significantly based on volatility of the stock's spread. To normalize the performance of the models, we use

TABLE IV
STOCKS IN THE DOW JONES INDUSTRIAL AVERAGE (DJIA) IN 2018.

Company	Ticker
3M	MMM
American Express	AXP
Apple	AAPL
Boeing	BA
Caterpillar	CAT
Cisco Systems	CSCO
Chevron	CVX
Coca-Cola	KO
DowDuPont	DWDP
Exxon Mobil	XOM
General Electric	GE
Goldman Sachs	GS
Home Depot	HD
IBM	IBM
Intel	INTC
Johnson & Johnson	JNJ
JPMorgan Chase	JPM
McDonald's	MCD
Merck	MRK
Microsoft	MSFT
Nike	NKE
Pfizer	PFE
Procter & Gamble	PG
Travelers	TRV
United Technologies	UTX
UnitedHealth Group	UNH
Verizon Communications	VZ
Visa	V
Walgreens Boots Alliance	WBA
Walmart	WMT
Walt Disney	DIS

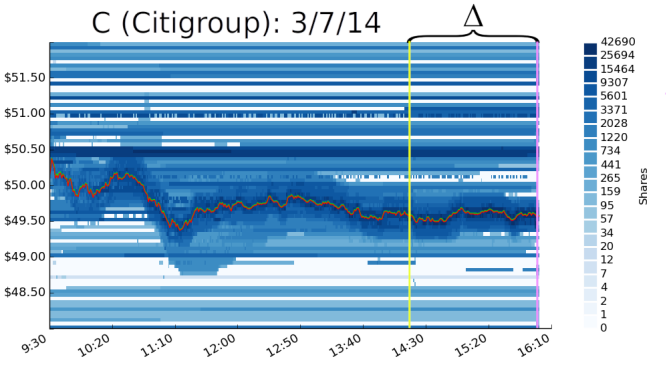

Fig. 4. Objective function visualization. In this figure 14:30 is when the last known message was received and the purple line is the point at which we're predicting the spread.

R-squared as our normalized comparison metric. The formula for R-squared is given in equation 3.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(s_{t+\Delta,i} - \hat{f}(X_{t,i}) \right)^2}{\sum_{i=1}^{N} \left(s_{t+\Delta,i} - \bar{s}_{t+\Delta,i} \right)^2} \tag{3}$$

The usefulness of Level I and Level II data may be associated with a time component. For this reason, we train Level I and Level II models for varying deltas. In particular, we do this for deltas set at 1 millisecond, 10 milliseconds, 100 millisecond, 1 second, and 10 seconds.

C. Comparable Design Matrices

In order to generate comparable $\hat{f}$ models, the design matrices X_1 and X_2 must be comparable. Consider again Tables II and III. These tables cover the same time period; however, the pseudo Level II table (Table III) has one more entry than the Level I table (Table III). If we restricted both tables to consist of the most recent 3 rows, then the input to the Level I model would span a greater time period than the input to the Level II model. The Level I design matrix would include data back to time 103,561 while the Level II design matrix would only include data back to time 103,565.

This example reveals a fundamental tradeoff in constructing comparable design matrices containing Level I or Level II data. If the design matrices represent the same number of events, then the Level II design matrix may be shorter than the Level I design matrix. On the other hand, if the design matrix is constructed to account for a fixed interval of time, then the Level II design matrix may have more rows than the Level I matrix.

Orders closer to the bid and ask prices are more likely to execute than orders further away from the bid and ask prices. Since orders close to the bid and the ask are more likely to execute, it is reasonable to assume that traders who place these orders are more likely to be sincere about their desire to buy or sell the asset at the given price. Conversely, orders further away from the bid and the ask are less likely to execute. This presents malicious traders the opportunity to attempt to manipulate the stock's price by placing insincere orders in hopes of sending false signals of demand to other traders and their algorithms. As Level I data only consists of orders that are close to the bid and the ask, these data are less likely to be manipulated than the orders further from the bid and the ask contained in Level II data. Thus, Level II data is more susceptible to noisy signals sent by bad actors than Level I data. If the design matrices for Level I and Level II data are required to be the same length (have the same number of rows), then the Level II model, with its design matrix that does not contain information as far into the past, would be unable to compete with the Level I model even if Level II data does provide more information. This reasoning forms the basis for why we chose to use a fixed time interval in constructing design matrices for Level I and Level II models instead requiring the same number of rows.

As messaging data occurs frequently, we construct a design matrix that spans 30 seconds from the time of prediction. To illustrate, assume that the current time for the model is 1:52:05pm, then all messages that changed Level I data from 1:51:35pm to 1:52:05pm would be included in the design matrix. If the delta for this model was 1 second, then the model would be predicting the spread at 1:51:36pm. When sampling the training data, we did not sample random times of the day. Instead, we randomly sampled an event from the day and used that timestamp to construct the design

matrix and target spread. We sampled in this way due to the assumption that more HFT traders feed new data into their models upon arrival instead of running their models at fixed time intervals. In other words, we assume event driven trading.

D. Comparable Model Architectures

Core to generating comparable performance metrics for Level I and Level II data is the architecture of the model used to approximate $f(\cdot)$. Given that X_1 and X_2 may differ in length, this imposes a restriction on what the input layer to the model may be. Specifically, as the sequence length can differ dramatically between samples, we chose a model that is agnostic to number of rows in the input matrix. We place a feedforward network at the head of the model to transform X_1 and X_2 to have the same number of columns so that the architectures of the models can be identical after this point. A detailed description of the model architecture is provided in Figure 5.

This approach is not without possible errors. Suppose there is more signal in Y_2 than in Y_1, but that the hidden and output layers are the same dimensionality. If the model is saturated by the signal in Y_1 and incapable of *learning* more, than the model fit on Y_2 may be able to achieve higher performance than the first model. Now assume that the model is not fully saturated after fitting on Y_1, then this model has potential to overfit on Y_1, hindering the model's performance.

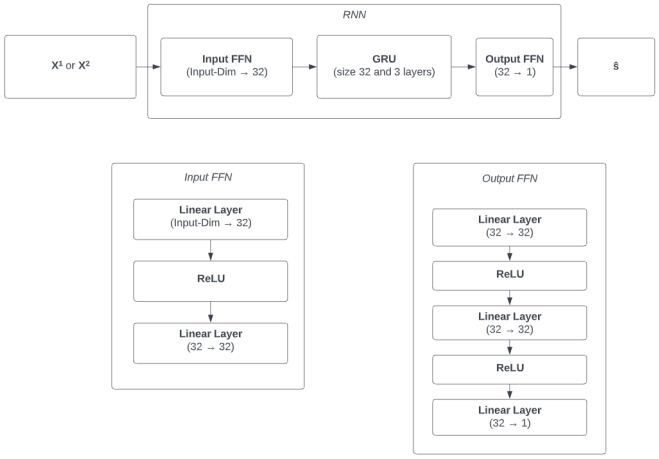

Fig. 5. Model Architecture Flowchart

E. Train, Validation, and Test Set Construction

The dataset used in this study consists of all messaging data on the NASDAQ exchange of all stocks that were part of the DJIA in 2018 (listed in Table IV). In 2018, there were 248 full trading days.

To construct the train, validation, and test sets, we used contiguous sets of three days. We required three contiguous training days to increase the amount of training data available to each model. These days had to be contiguous to combat non-stationarity in the data. We assumed that high frequency traders likely refit their models on non-trading days, so using contiguous days helps guard against this non-stationarity. After imposing these constraints, there were 91 unique sets of contiguous training days in 2018.

For each stock we assigned a delta. We then randomly selected one of these unique sets of three days. For each of the three days, we excluded the first and last hour of the trading day as these hours are more subject to anomalous behavior. We then divided the data into four equal sized sets. Two of these partitions were randomly assigned to be incorporated into the training set, one of the remaining partitions was assigned to the validation set and the final partition was allocated to the test set. For each of these setups, we trained a Level I model and a Level II model. We refer to the stock, with its train, validation, test sets, model type, and delta, as an *experiment*.

We repeated this process twice for replication and averaged their results. This guards against inflated statistical power that comes from duplicate data. In all, we had 5 different deltas, 31 stocks, 2 models, and a replication factor of 2 for a total of 620 experiments.

F. Convergence Criteria and Test Model Selection

Convergence criteria is important because it determines how much *learning* a model can do as well as how much overfitting may occur. This is especially critical in our objective of generating two models where the only difference between them is the use of Level I or Level II data.

The training set is used to tune the models parameters, the validation set is used to determine if overfitting has occurred, and the test set is used to generate the model's performance. The convergence criteria is defined by the following rules:

1) A variable we call patience is set to 4.
2) After an epoch of training, the model's performance is gauged by calculating the MSE on the validation set.
3) If the observed MSE is the lowest MSE seen so far, the patience variable is reset to 4. Else, patience is decreased by 1.
4) If patience has dropped twice, without model improvement, the learning rate is reduced by 90%.
5) Steps 2-4 are repeated until patience reaches 0.

To guard against testing on an overfit model, the model with the lowest validation MSE is evaluated on the test set.

After gathering the R-squared values for all our models and averaging the replicated results, we ran a simple cross-sectional regression of R-squared on our *IsLevel2* dummy variable from Section III-B.

IV. Results

The coefficient recovered on *IsLevel2* is negative for each cross-sectional regression, as seen in Table V. However, the effect does not become statistically significant until 1000 ms (1 second) is reached. Surprisingly, Level II data provides

the largest and most statistically significant decline in performance compared to Level I data 10 seconds from the last observed change to the book.

A visualization of the experiment results and statistics of interest is shown in Figure 6.

TABLE V
R-SQUARED REGRESSION RESULTS

Delta (ms)	1	10	100	1000	10000
Intercept	0.0947	.0087	0.0443	0.0124	-0.0305
	(0.000)	(0.000)	(0.002)	(0.235)	(0.014)
IsLevel2	-0.0150	-0.0272	-0.0182	-0.0438	-0.0410
	(0.583)	(0.187)	(0.360)	(0.004)	(0.019)

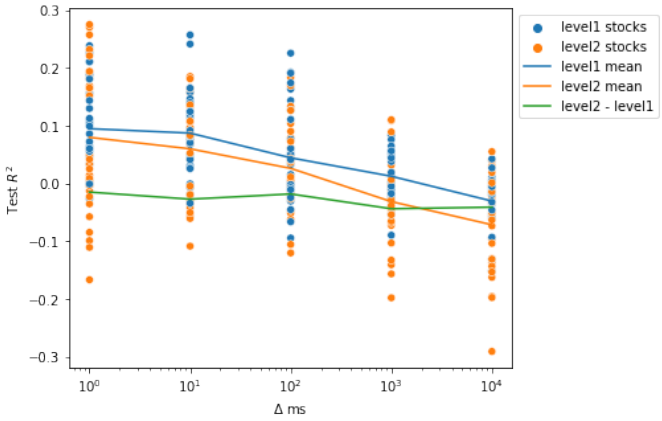

Fig. 6. Level 1 data consistently performs better than level 2 data and the gap in performance gradually widens as delta increases. Models that incorporate level 2 data must do so strategically as a naive incorporation of this data can decrease model performance.

V. CONCLUSION

This suggests that while Level II data is a superset of Level I data, more extensive preprocessing must be undertaken than was conducted in this study if it is to enhance predictive performance. The results of our research indicate that using richer information sets to train a Recurrent Neural Network (RNN) does not necessarily guarantee improved model performance. In comparison to Level I data, employing Level II data in an RNN architecture offers no additional benefit in predicting the spread in a High-Frequency Trading (HFT) environment.

In this study, we examined the value of Level II stock data relative to Level I stock data. Our findings suggest that Level II data may not offer added predictive power over Level I data. While these effects are not statistically significant in the ultra-short term (less than 100 milliseconds), over more extended predictive periods (longer than 1 second), Level II data hampers predictive power for forecasting the bid-ask spread, when provided in the manner described in this paper.

This observation implies that in high-speed trading environments, where latency sensitivity and computational resource limitations are factors, HFT algorithms could benefit from excluding Level II data. Alternatively, additional preprocessing may be necessary to transform Level II data into a format that yields favorable trading outcomes using an RNN.

High-frequency traders might achieve more success by relying on messaging data at the bid and ask prices. In contrast, lower-frequency traders may want to undertake more preprocessing than was performed in this study if the book shape is to be incorporated into an HFT model.

While this study focused on characterizing the informativity of Level II book data for predicting the spread at various future moments, utilizing this information in an automated algorithm would also require forecasting the midpoint of the spread. We envision future work to conduct a similar study, aimed at characterizing the informativity of Level II book data for predicting the midpoint of the bid-ask spread at various future moments. This would complete the effort required to design effective automated trading algorithms. Potential data smoothing operations, such as exponentially weighted averages and simple averages, could also be examined to determine how these transformations might enhance signal extraction from the data.

REFERENCES

[1] NASDAQ. (2018) High frequency trading (hft). [Online]. Available: https://www.nasdaq.com/glossary/h/high-frequency-trading

[2] M. Aquilina, E. Budish, and P. O'Neill, "Quantifying the high-frequency trading "arms race"," *The Quarterly Journal of Economics*, vol. 137, pp. 493–564, 2021.

[3] P. Gomber and M. Haferkorn, "High-frequency-trading," *Business & Information Systems Engineering*, vol. 5, pp. 97–99, April 2013.

[4] G. P. M. Virgilio, "High-frequency trading: a literature review," *Financial Markets and Portfolio Management*, pp. 1–26, 2019.

[5] Q. Zhang, "Stock trading: An optimal selling rule," *SIAM J. Control. Optim.*, vol. 40, pp. 64–87, 2001.

[6] J. Tie, H. Zhang, and Q. Zhang, "An optimal strategy for pairs trading under geometric brownian motions," *Journal of Optimization Theory and Applications*, vol. 179, November 2018.

[7] A. Deshpande and B. R. Barmish, "A generalization of the robust positive expectation theorem for stock trading via feedback control," in *2018 European Control Conference (ECC)*, 2018, pp. 514–520.

[8] B. R. Barmish, J. A. Primbs, and S. Warnick, "On feedforward stock trading control using a new transaction level price trend model," *IEEE Transactions on Automatic Control*, vol. 67, no. 2, pp. 902–909, 2021.

[9] L. Chen, M. Pelger, and J. Zhu, "Deep learning in asset pricing," *Management Science, forthcoming*, 2019.

[10] J. Guijarro-Ordonez, M. Pelger, and G. Zanotti, "Deep learning statistical arbitrage," *ERN: Efficient Market Hypothesis Models (Topic)*, 2021.

[11] T. Kabbani and E. Duman, "Deep reinforcement learning approach for trading automation in the stock market," *IEEE Access*, vol. 10, pp. 93 564–93 574, 2022. [Online]. Available: https://doi.org/10.1109%2Faccess.2022.3203697

[12] J. Zou, J. Lou, B. Wang, and S. Liu, "A novel deep reinforcement learning based automated stock trading system using cascaded lstm networks," *ArXiv*, vol. abs/2212.02721, 2022.

[13] J. Angrist, "Empirical strategies in economics: Illuminating the path from cause to effect," *Econometrica*, vol. 90, no. 6, pp. 2509–2539, November 2022.

[14] F. Jareño and L. Negrut, "Us stock market and macroeconomic factors," *Journal of Applied Business Research*, vol. 32, pp. 325–340, 2015.