Algorithmic Trading with Machine Learning

Dissertation

zur Erlangung des Doktorgrades
der Wirtschafts- und Sozialwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen

vorgelegt von

Christopher Felder
aus Herrenberg

Tübingen
2023

Tag der mündlichen Prüfung:	12. Juli 2023
Dekan:	Professor Dr. Ansgar Thiel
1. Gutachter:	Professor Dr. Ted Azarmi
2. Gutachter:	Professor Dr. Stefan Mayer

Contents

Article 1

Article 2

Article 3

List of Figures

Synopsis

Article 1

Article 2

Article 3

List of Tables

Synopsis

Article 1

Article 2

Article 3

1 Synopsis

1.1 Introduction and Motivation

In 1966, the ticker tape of the New York Stock Exchange displayed the first electronically processed stock quote (Kennedy, 2017; Currie et al., 2022). This breakthrough was made possible by electronic card readers that allowed price data to pass onto the tape in seconds instead of minutes, as was previously common under human intermediaries (OTA, 1990; Kennedy, 2017). In the decades that followed, technological innovation drove the automation and scaling of exchange infrastructure to ever-higher trade volumes, making securities trading possible without direct human interaction by the mid-1990s (MacKenzie, 2018). The automation of trading platforms has meant that market places are no longer coordinated by "gazes and hand signals" on bustling trading floors, but by an electronic infrastructure that connects "masses of anonymous investors" through "nimble algorithms, sophisticated computer processors, hacked routers, and specialized telecommunication systems" (Pardo-Guerra, 2019; Beverungen, 2019), transforming trading floors into deserted high tech hubs (see Figure 1.1).

Spawned by this evolution as a new type of trading, algorithmic trading today accounts for over 70% of stocks trade volume (Cartea and Jaimungal, 2013). Algorithmic trading describes the use of mathematical models and computer algorithms to automate trading decisions, order submission, and order management after submission (Hendershott et al., 2011; Kirilenko and Lo, 2013). The prevalence of algorithmic trading stems from advantages of algorithms over humans in securities trading: Algorithms have no feelings, are therefore not prone to investor behavior such as overreactions (De Bondt and Thaler, 1985) or biased expectations (De Bondt, 1993), are likely to reduce human-caused errors, and can tackle more complex tasks (Dixon et al., 2020). Advancing algorithmization and the speeding up of trading to higher frequencies requires today's exchanges to use massive data centers that transmit tens of millions of messages between market participants every second (MacKenzie, 2018; Exegy, 2023).

In the past two decades, algorithmic trading has been dominated primarily by deterministic, rule-based models (FSB, 2017). Beverungen (2019) refers to these trading algorithms as "dumb" algorithms because the trade decision is always the same given the same state of input

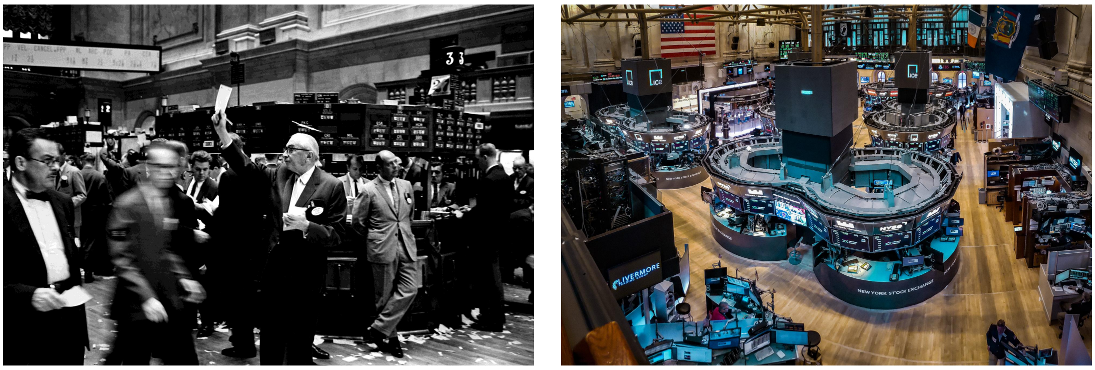

Figure 1.1: The trading floor of the New York Stock Exchange in 1963 (left) and 2022 (right). Copyright Thomas O'Halloran (left) (O'Halloran, 1963) and Tobias Deml (right) (Deml, 2022).

values, usually consisting of price, order book, or other market data. The gradual advent of machine learning techniques in algorithmic trading in the last decade (OECD, 2021) heralded the transition to "smart" trading algorithms that solve problems dynamically outside static models and adapt to market environments. Algorithmic trading is a "natural playground for machine learning" (Dixon et al., 2020) as financial markets are dynamic and trade decisions are data-driven. Moreover, the increasing availability of computing and data storage capacity provides ideal conditions for further advances in algorithmic trading with machine learning.

However, despite impressive experimental results for both directional trading and market making, machine learning does not yet dominate algorithmic trading (ESMA, 2023; FSB, 2017), and practitioners doubt that artificial intelligence fully takes over trade execution in the near future (Beverungen and Lange, 2018). While most firms have integrated machine learning tools into their trading algorithms (ESMA, 2023), final trade execution still falls to either humans or rule-based models. Quantitative hedge funds typically use machine learning for improving investment decisions with humans overseeing trade execution (Ashta and Herrmann, 2021). Banks and non-bank market makers still build most of their algorithmic trading around transparent rule-based models and use machine learning as one part of multi-layer execution processes inside of traditional deterministic algorithms (FMSB, 2020). Therefore, even though machine learning has made inroads into algorithmic trading, it has not yet revolutionized securities trading like, for example, the launch of computer trading has.

Why do practitioners hesitate to adopt autonomous machine learning algorithms for trade execution? Pozen and Ruane (2019) point out that while machine learning can help reduce human bias, humans still need to understand what the machine has learned to "prevent new biases from creeping in". Ray Dalio, founder of the world's largest hedge fund Bridgewater Associates, stresses that trading with machine learning algorithms requires a deep understanding of what the model has learned and how that will play out in the face of changing market conditions (Dalio, 2017). Thus, hazards to trading arise if we cannot discern whether the algorithms learn anything that keeps trading consistently profitable as market conditions change, and whether their understanding of trading is equivalent to that of a human trader.

Correspondingly, practitioners have so far relied on the symbiosis of human intelligence and artificial intelligence to verify what the algorithms learn and preserve the mindset of a human trader for unforeseen market conditions. In doing so, firms face a trade-off between high machine learning-based automation with low robustness to changing market conditions versus low automation with high robustness. To overcome this trade-off, a key task for the future is to develop robust machine learning algorithms optimized for trading in all parts of the trade life cycle. Two research questions arise from this task:

1. How can machine learning algorithms be optimized for algorithmic trading?
2. What do these optimized algorithms learn and what is their advantage over traditional rule-based models?

This dissertation presents three articles, each addressing the research questions for a specific part of the trade life cycle (see Figure 1.2). The first article relates to the investment decision process and tailors machine learning algorithms for return prediction to reduce the likelihood of large trading losses. The second article focuses on trade execution at the retail level and introduces an innovative representation of the relationship between buyer and seller overhangs that helps retail brokers optimize order segmentation and improve market quality. The third article is located in trade execution at the wholesale level and integrates a high-frequency trade flow prediction model into limit order trading, allowing market makers to reduce adverse selection cost by preempting price movements.

Overall, these articles contribute to the transition from rule-based to machine learning techniques in algorithmic trading by introducing new machine learning-based trading algorithms that are superior to rule-based models in directional trading as well as in order execution and market making. Specifically, the dissertation makes three contributions: First, it provides methods for modeling the market tailored to firms' individual trading objectives such as loss aversion (article 1), neutralization of open positions within a particular period (article 2), or anticipation of market movements (article 3) using novel target variables and loss functions. Second, it empirically demonstrates that modeling the market tailored to firms' trading objectives mitigates the trade-off between automation and robustness, as machine learning algorithms can better understand and mimic the intentions of the firms' human traders. Third, it shows that moving from rule-based to machine learning-based trading systems can be worthwhile for both directional traders and market makers, as sentiment signals relevant to the firms' trading objectives are captured in a way no human can, and they both benefit equally from improved market quality and reduced adverse market movements.

The remainder of this synopsis is structured as follows: Section 1.2 outlines the structure and concretizes the overall aim of the dissertation. Section 1.3 gives short overviews over each of the three articles. Section 1.4 presents a final conclusion of the dissertation.

1.2 Dissertation Aim and Structure

The goal of the dissertation is to provide machine learning concepts that model financial markets in a trading-optimized manner and eliminate the need for human monitoring of trading algorithms. Given remarkable test results, recent literature evaluates supervised learning algorithms like Recurrent Neural Network (RNN) (Dixon, 2018*b*), especially Long Short-Term Memory (LSTM) (Fischer and Krauss, 2018), as suitable machine learning algorithms for automated trading. However, practitioners currently only use these algorithms as additional components in traditional programs for trade execution rather than as stand-alones, motivated by the desire to control the learning process and protect against changing market conditions. The dissertation therefore aims to shed light on how the market can be optimally modeled for autonomous trade execution based on supervised learning algorithms through novel loss functions and target variables adapted to firms' trading preferences. In doing so, I take an interdisciplinary research approach including recent developments in machine learning, behavioral finance, cryptocurrencies, market microstructure, and quantitative finance.

In order to address the research questions separately for specific components of a trading system, the structure of the dissertation follows the usual trade life cycle of a marketable retail order for publicly listed stocks (SEC, 2013; ESMA, 2023). Figure 1.2 illustrates how the articles are intertwined with the relevant trade execution modalities.

The first article, "*Return Prediction with Deep Learning under Loss Aversion*", is located in the pre-trade phase, i.e., the pre-trade analysis and investment decision (ESMA, 2023), and provides an approach to improve investment decisions for directional trading through loss aversion. Prospect theory (Kahneman and Tversky, 1979) states that people are more sensitive to losses than gains. Therefore, investors may value overprediction differently than underprediction of returns. The article introduces a loss function designed for loss aversion

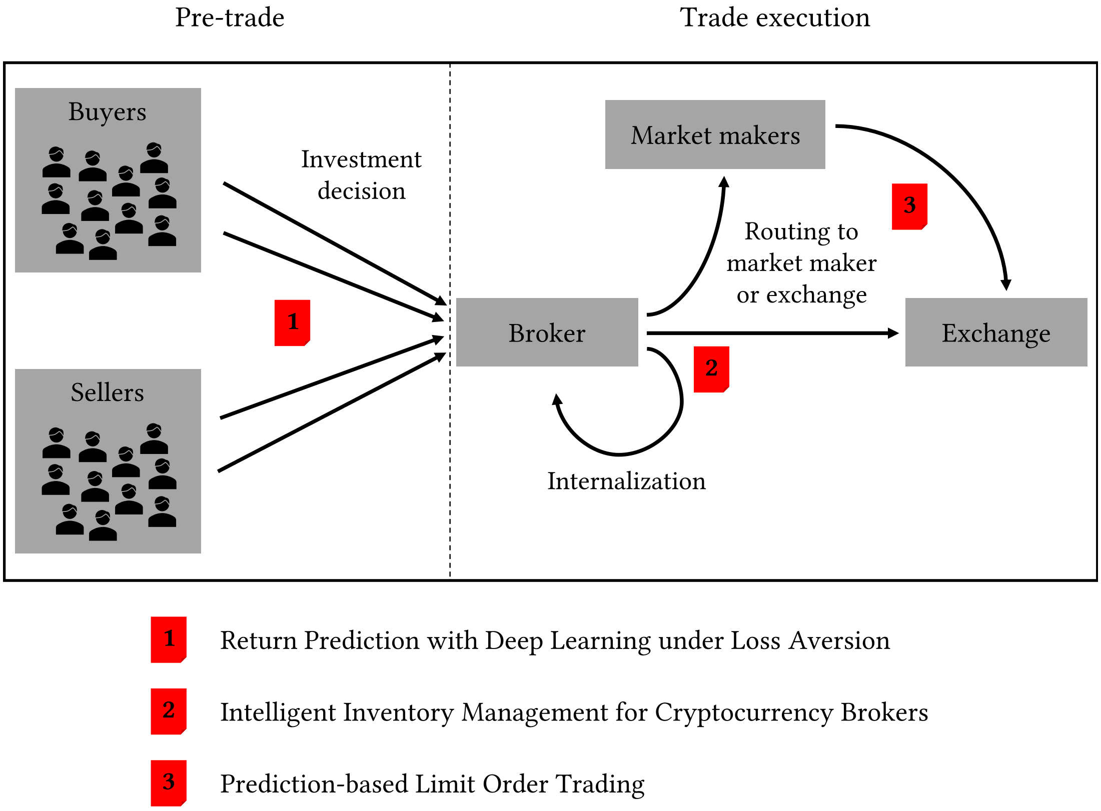
Pre-trade
Trade execution
Buyers
Sellers
Investment decision
1
Market makers
Routing to market maker or exchange
3
Broker
Exchange
2
Internalization
1 Return Prediction with Deep Learning under Loss Aversion
2 Intelligent Inventory Management for Cryptocurrency Brokers
3 Prediction-based Limit Order Trading

Figure 1.2: Dissertation structure based on the trade execution modalities of market orders for publicly listed securities following SEC (2013) and ESMA (2023).

and tests it based on LSTMs for U.S. stock returns between 1990 and 2021. The model learns unique signals, predicts returns more cautiously, and improves profit chances over standard LSTMs and common trend signals. Referring to my research questions, I conclude that investors can model the market in a trading-optimized manner by incorporating their trading objectives such as loss aversion into the learning algorithm, thereby beating both rule-based trend-following and loss-neutral machine learning strategies.

The second article, "*Intelligent Inventory Management for Cryptocurrency Brokers*", is set in the first stage of trade execution, in which the broker segments orders and decides on the execution modality, and introduces a novel representation of buy and sell order flows to optimize this decision process. Basically, a broker can either internalize a market order, i.e., execute it internally against open positions or hold it as an open position for execution against future orders, or route it to an exchange or market maker. This article presents a feedforward neural network, which, upon receiving a customer order, predicts whether future order flow will be sufficient to neutralize the order before its settlement date. An empirical analysis of the German cryptocurrency broker BISON shows that incorporating the predictions into the decision process realizes meaningful cost savings for both investors and brokers over rule-based strategies. I conclude that customizing target variables to brokers' trading objective such as neutralization of positions within a given period contributes to optimizing machine learning algorithms for trade execution and improving market quality over rule-based approaches.

The third article, "*Prediction-based Limit Order Trading*", focuses on the second stage of trade execution, where wholesale market makers execute orders that the broker routes to the exchange or directly to them, and presents a method to optimize the market makers' position in the limit order book using an RNN-based trade flow model. Market makers are firms that continuously quote buy and sell prices and profit from the spread in between through simultaneous trade execution, facing a trade-off between high volume with low margin versus low volume with high margin. I attempt to overcome this trade-off by including predictions for buyer- and seller-initiated trades when submitting limit orders. Tested for two cryptocurrency pairs at the Coinbase exchange, I report higher market making revenues through trade flow-adjusted order prices and lower adverse selection costs through preempted price movements. I infer that it may be worthwhile for market makers to model the market separately for buy and sell trade flow and thus adapt algorithms to the market makers' objective of anticipating changes on the bid and on the ask side of the market.

Table 1.1 provides a condensed overview of the three articles, reporting the overarching research objectives, the data set and the data source used, the research methodology, and the current publication status of each article. Regarding the research methodology, one may notice that all articles rely on supervised learning. Although there is currently a lot of experimentation going on in reinforcement learning in trading (Zhang et al., 2020), I focus on supervised learning, as this is the dominant methodology in practice (ESMA, 2023).

1.3 Overview of Articles

1.3.1 Return Prediction with Deep Learning under Loss Aversion[1]

Christopher Felder and Stefan Mayer

Under prospect theory (Kahneman and Tversky, 1979), people value potential gains and losses asymmetrically when making decisions. As one part of prospect theory, loss aversion (Tversky and Kahneman, 1992) explains that people perceive losses greater than equivalent gains and, e.g., when choosing between a 100% chance to gain \$50 or a 50% chance to gain \$110, tend to choose the certain outcome even though the expected utility of the uncertain outcome is higher. Applied to the return prediction problem, loss aversion implies that investors perceive underpredictions differently than overpredictions of returns and, e.g., prefer an underprediction of their own portfolio return to an overprediction. As standard optimization techniques for return prediction handle prediction losses equally with both over- and underpredictions receiving the same penalty, we reconsider return prediction from a loss-averse perspective and introduce a novel loss function that distinguishes between 'good' and 'bad' losses. If investors buy (short) shares with a predicted positive (negative) return, 'good' losses occur when the predicted positive (negative) return is less (greater) than the realized return, while 'bad' losses classify overprediction (underprediction) of positive (negative) returns.

The objective is to minimize both 'good' and 'bad' prediction losses, with 'good' losses preferred over 'bad' losses. As model framework, we consider a variant of the weighted mean squared error, where the weight is a function of the loss type. Following Tversky and Kahneman (1992), who show that people are twice as sensitive to losses as to gains, we penalize 'bad' losses up to twice as hard as 'good' losses. So in a scenario predicting a 3% return, we

[1] Parts of this work have been published in Felder and Mayer (2022). ©2022 IEEE

Article	Research Objective	Data (Source)	Methodology	Status
1 *Return Prediction with Deep Learning under Loss Aversion* Christopher Felder, Stefan Mayer	· Design a loss function for return prediction with deep learning subject to investors' loss aversion · Empirically examine the impact of loss aversion in return prediction on prediction quality and trading performance · Identify unique loss-averse signals and differences in learning content compared to traditional return prediction models	· Daily closing prices of U.S. stocks from 1990-2021 (Refinitiv)	· Long Short-Term Memory (LSTM) · Linear regression · Investment study	Previous version published in Felder and Mayer (2022)
2 *Intelligent Inventory Management for Cryptocurrency Brokers* Christopher Felder, Johannes Seemüller	· Understand long- and short-term relationships of buy and sell retail order flow · Extract drivers of buy and sell overhangs from order flow and limit order book data · Create a prediction model for order flow imbalance and analyze whether retail brokers can use this model to improve order execution for customers	· Tick-level cryptocurrency retail order flow (BISON App) · High-frequency cryptocurrency OHLCV data (Kraken, Coinbase, Bitstamp)	· Feedforward Neural Network · Penalized logistic regression (Elastic Net) · Trading simulation	Previous version published in Felder and Seemüller (2022)
3 *Prediction-based Limit Order Trading* Christopher Felder	· Derive a theoretical approach for market makers to incorporate trade flow predictions when posting limit orders · Develop a high-frequency prediction model for buyer- and seller-initiated trade flow · Measure the impact of incorporating trade flow predictions into limit order trading on adverse selection cost and market making revenues	· Level II cryptocurrency limit order book data (Coinbase) · Tick-level trade match data (Coinbase)	· Recurrent Neural Network (RNN) · Feedforward Neural Network · Trading simulation	Working paper; presented at the University of Edinburgh's *Economics of Financial Technology Conference 2023*

Table 1.1: Overview of dissertation articles.

would penalize the prediction loss twice as hard when the actual return is 1% than when it is 5%. In doing so, we analyze both a binary weight function that only knows the maximum and minimum penalties, and a sigmoid weight function that continuously distributes weights.

We test our model based on LSTM, a deep learning algorithm proven suitable for return prediction problems (Fischer and Krauss, 2018; Fabbri and Moro, 2018; Borovkova and Tsiamas, 2019; Flori and Regoli, 2021; Chen et al., 2022), and benchmark it against loss-neutral models and short-term reversal signals as suggested by Jiang et al. (2022). Using a test sample of U.S. stock returns between 1990 and 2021, we draw the following conclusions: Loss-averse LSTMs underpredict (overpredict) rather than overpredict (underpredict) positive (negative) returns, and investors are less likely to face situations in which their realized portfolio return is lower than predicted. Portfolio performance improves for prediction time scales between one day and one month, as loss-averse LSTMs extract new, unique signals that contain predictive information superior to reversal signals. Moreover, the loss function contributes positively to portfolio returns throughout changing market sentiments, even after accounting for trading cost and common institutional and retail investor constraints.

In order to identify drivers of the model's superior performance, we compare return predictions of loss-neutral LSTMs with those of loss-averse LSTMs. The loss-averse LSTM delivers more conservative predictions as the distribution of predictions is narrower, i.e., fewer very high and very low predictions occur. In a scenario where standard LSTMs produce more overpredictions than underpredictions, a loss-averse loss function improves the network's approximation of the market. As we find empirical evidence of standard LSTMs' tendency to overpredict (underpredict) positive (negative) returns, we conclude that teaching a model to predict returns more conservatively provides a better approximation of the market.

1.3.2 Intelligent Inventory Management for Cryptocurrency Brokers[2]

Christopher Felder and Johannes Seemüller

Internalization—a practice whereby brokers execute retail orders internally against their own book instead of routing them to an exchange (Grammig and Theissen, 2012)—is the primary execution modality for retail orders in securities trading (Fox et al., 2019; Barardehi et al., 2022; Comerton-Forde et al., 2018). Trading against retail order flow pays off for brokers as retail investors are less informed (Chakravarty, 2001; Linnainmaa, 2010), but can be as profitable for customers since internalized trades typically receive price improvements that "compensate the market for the liquidity lost" through internalization (Kumpan, 2006).

A special type of internalization is internal order matching (Battalio and Loughran, 2008; Challet et al., 2018), sometimes called order netting (Hagerty and McDonald, 1996; Schwartz et al., 2005; Francioni and Schwartz, 2008), where the broker internally matches buy and sell order flow, saving the bid-ask spread without trading against one side of the market. In doing so, the broker stores orders in inventory for which she expects contrary orders in the future and sends all other orders to an exchange. This segmentation is not straightforward, however, as orders arrive asynchronously, and if she cannot internally match an order from inventory until its settlement is due, she must reroute it to an exchange at possibly worse prices. If she had knowledge about the future, she could store only orders actually covered by future order flow in inventory and send all other orders to an exchange, but in practice she does not.

[2] Parts of this work have been published in Felder and Seemüller (2022). © 2022 ACM

Brokers must therefore trade off between high internal matching, with higher profit margins but greater price risk of the inventory, and low internal matching, with lower margins but reduced price risk. To this end, we present a prediction-based model of retail order segmentation for internal order matching. Upon receiving an order that cannot be immediately executed against open positions in inventory, the model predicts whether future order flow will be sufficient to neutralize the order before its settlement date, thereby assisting the broker in deciding whether to hold it in inventory or route it to an exchange for immediate execution.

Using the historical order flow of the German cryptocurrency broker BISON as an example, we first determine the maximum proportion of each order that a broker could have matched internally given a two-day settlement cycle. On average, 85% of retail order volume is followed by contrary order volume within two days. Consequently, retail brokers can execute a maximum of 85% of order volume by internally matching buy and sell order flows. Next, we analyze market and order flow dynamics as a function of that maximum proportion and derive five predictors of internal order matches, which are the order flow imbalance, average order volume, total trade volume, high-low price range, and one-minute price changes.

Based on these predictors, a feedforward neural network correctly predicts around nine out of ten internal order matches, allowing us to internally match three-quarters of order volume. Internally matched orders are largely characterized by small order volume and an average volatile market environment with marginal order flow imbalance. Predictive performance is best during volatile and trading-intensive periods, which pays off disproportionately for brokers and customers, as these periods are associated with large spreads. In a scenario where customers and brokers share the realized cost savings equally, effective spreads lower by 35%. Accordingly, the results support the theory of cost savings, consistent with empirical (Battalio, 1997; Hansch et al., 1999; Battalio et al., 2001; Peterson and Sirri, 2003; Grammig and Theissen, 2012) and theoretical (Battalio and Loughran, 2008; Degryse et al., 2022) studies of execution costs for internalized trades, and thus suggest a potential improvement in market quality.

1.3.3 Prediction-based Limit Order Trading

Christopher Felder

Market makers are firms that continuously offer buy and sell prices for a specific asset on the exchange by placing limit orders on both sides of the order book, thus providing liquidity to the market. The market making business model aims to repeatedly earn the spread between buy and sell quotes by simultaneously buying and selling the asset without accumulating a large net position, and is subject to a trade-off between high trade volume with low margins and low volume with high margins: Wide spreads lead to high margins but less volume, while dealers with tight spreads trade frequently but at unfavorable prices (Guéant, 2017). Market making is associated with several risks, including inventory risk arising from the stochastic behavior of the firm's net position in the traded asset and adverse selection risk arising from limit order trading with informed traders (Frey and Grammig, 2008).

Most limit order models presented so far focus on managing inventory risk (Avellaneda and Stoikov, 2008; Guéant et al., 2013; Cartea and Jaimungal, 2013; Guilbaud and Pham, 2013; Cartea et al., 2014; Bayraktar and Ludkovski, 2014; Guéant, 2017; Ahuja et al., 2017) and build on the assumption that market orders follow a stochastic process. However, there are numerous studies providing empirical evidence of mid price predictability from the limit order book (LOB) (Cao et al., 2009; Zheng et al., 2013; Cont et al., 2014; Kercheval and Zhang, 2015; Tsantekidis et al., 2017; Dixon, 2018*a*; Zhang et al., 2019; Tsantekidis et al., 2020). Inspired by these findings, I present a limit order model that assumes that market order flow is predictable, and helps to both reduce inventory and adverse selection risks and overcome the trade-off between volume and margin by integrating trade flow forecasts into limit order trading.

Based on historical LOB and trade data of cryptocurrency pairs at the Coinbase exchange, I model buyer- and seller-initiated trades separately using RNN and predict how large trades will be during the next five seconds relative to the trades of the previous 24 hours. Next, I adjust limit order prices to these predictions in two ways: First, if the model predicts large (small) trades, I increase (decrease) the depth of orders in the LOB, i.e., post orders deeper (closer) in the LOB. For instance, when predicting large buy orders and small sell orders, I increase both the ask and bid price to maximize the potentially captured spread on the ask side and increase the chance of being consumed on the bid side. Second, I shift the reservation price to protect against the anticipated price movement derived from the imbalance between the predictions for buy and sell trade sizes. Hence, when anticipating a price increase (decline), I shift the reservation price up (down) to protect from being adversely selected by informed buyers (sellers) and participate in the price increase (decline) through long (short) exposure. In analyzing the impact of the two adjustments on P&L, I find that there is an empirical optimum of sensitivity to predictions, stating that one should be twice as sensitive to predictions when adjusting quotes to relative trade sizes than when adjusting them to adverse price movements.

A trading simulation reports improved P&L over the Avellaneda and Stoikov (2008) model, but this improvement also comes with higher risk and depends on market dynamics. As two main contributors to P&L, I identify (1) higher market making revenues due to larger spreads earned especially when the order book is less liquid, and (2) reduced adverse selection costs by preempting adverse price movements, with the cost reduction being proportional to the magnitude of the adverse price movement. Accordingly, the model provides an effective approach to both overcoming the trade-off between volume and margin and reducing adverse selection cost by taking trade flow forecasts into account when deciding whether the market maker can afford a wider spread to increase profitability, while controlling inventory risk.

1.4 Conclusion

Trading mechanisms and exchanges undergo constant transformation along with current technological advances. Following computerization in the past century, the last two decades have been dedicated to quantification, algorithmization, and acceleration of trading. Today, algorithmic trading makes up the majority of securities trading (Cartea and Jaimungal, 2013). The next transformation will be the adoption of algorithmic trading by machine learning (OECD, 2021), fostered by the increasing availability of computing and data storage capacities. Yet, despite convincing experimental results, practitioners have so far been reluctant to implement fully automated machine learning-based trade execution systems, fearing the risk of unforeseen changes in market conditions and relying instead on the symbiosis of humans, rule sets, and machines (Dalio, 2017; Beverungen and Lange, 2018; Pozen and Ruane, 2019).

A key task for the future of algorithmic trading is therefore to develop machine learning algorithms that are optimized for trading purposes along the entire trade life cycle and robust to changing market conditions such that they do not require human monitoring anymore. From this task, I establish two research questions: (1) How can machine learning algorithms be optimized for algorithmic trading? (2) What do these optimized algorithms learn and what is their advantage over traditional rule-based models?

The aim of the dissertation is to provide machine learning concepts for algorithmic trading that model financial markets in a trading-optimized manner, thereby eliminating the need for human or rule supervision. In three articles, I present novel approaches to adapt machine learning algorithms to securities trading that manage the entire trade life cycle—from investment decision to trade execution and settlement—better than rule-based strategies. My approaches focus on tailoring loss functions used during the learning process (article 1), data representations, and target variables (articles 2, 3) to firms' trading objectives.

Each article deals with a specific stage of the trade life cycle. The first article relates to the investment decision stage and shows that incorporating loss aversion into return prediction improves investment decisions compared to a loss-neutral algorithm. The second article focuses on trade execution at the retail level and introduces an innovative representation of the relationship between buyer and seller overhangs that helps retail brokers optimize trade execution and improves market quality. The third article is located in trade execution at the wholesale level and integrates trade flow forecasting into limit order trading, allowing market makers to reduce adverse selection cost over rule-based strategies. Thus, aligning market data representations or loss functions with firms' trading objectives such as (1) loss aversion, (2) neutralization of open positions within a particular period, or (3) anticipation of market movements, can help machine learning algorithms to better adapt to algorithmic trading. The main contributions of my work are methods for modeling the market tailored to firms' individual trading objectives, which allow firms to mitigate the trade-off between automation and robustness, and empirical evidence of their advantages over traditional rule-based trading strategies for both directional traders and market makers.

The dissertation is contributing to the ongoing shift away from rule-based algorithms to machine learning algorithms for algorithmic trading. However, there is still a long way to go before machine learning fully takes over algorithmic trading (Beverungen and Lange, 2018). The transformation may take many years or even decades, and this dissertation can only provide impetus for possible directions that researchers and practitioners can take toward that transformation. A key challenge for financial research will be to reliably evaluate the performance of trading algorithms: Every trade, every order submitted, affects other market participants, which cannot be accounted for in an empirical analysis such as this dissertation, but can only be measured in a live trading environment. The dissertation can therefore be seen as one building block of many to ultimately accomplish the transformation and attain fully automated self-learning trading systems suitable for practice.

In summary, this dissertation develops three machine learning concepts for algorithmic trading: A loss-averse prediction model of the stock market, a dynamic representation of retail order flow imbalance, and a high-frequency model of buyer- and seller-initiated trades. All three articles provide empirical evidence that these concepts could be advantageous over traditional rule-based models for all market participants and market quality. With my work, I hope to advance the transition to intelligent trading algorithms and motivate practitioners and researchers to conduct further experiments on embedding algorithmic trading in machine learning techniques.

References

Ahuja, S., Papanicolaou, G., Ren, W. and Yang, T.-W. (2017), 'Limit order trading with a mean reverting reference price', *Risk and Decision Analysis* **6**(2), 121–136. doi:10.3233/RDA-160118

Ashta, A. and Herrmann, H. (2021), 'Artificial intelligence and fintech: An overview of opportunities and risks for banking, investments, and microfinance', *Strategic Change* **30**(3), 211–222. doi:10.1002/jsc.2404

Avellaneda, M. and Stoikov, S. (2008), 'High-frequency trading in a limit order book', *Quantitative Finance* **8**(3), 217–224. doi:10.1080/14697680701381228

Barardehi, Y. H., Bernhardt, D., Da, Z. and Warachka, M. (2022), Institutional Liquidity Demand and the Internalization of Retail Order Flow: The Tail Does Not Wag the Dog, The Warwick Economics Research Paper Series (TWERPS), University of Warwick, Department of Economics. Available from https://warwick.ac.uk/fac/soc/economics/research/workingpapers/2022/twerp_1394_-_bernhardt.pdf.

Battalio, R. H. (1997), 'Third Market Broker-Dealers: Cost Competitors or Cream Skimmers?', *Journal of Finance* **52**(1), 341–352. doi:10.1111/j.1540-6261.1997.tb03819.x

Battalio, R. H. and Loughran, T. (2008), 'Does payment for order flow to your broker help or hurt you?', *Journal of Business Ethics* **80**(1), 37–44. doi:10.1007/s10551-007-9445-x

Battalio, R., Jennings, R. and Selway, J. (2001), 'The Relationship Among Market-Making Revenue, Payment for Order Flow, and Trading Costs for Market Orders', *Journal of Financial Services Research* **19**(1), 39–56. doi:10.1023/A:1011181317099

Bayraktar, E. and Ludkovski, M. (2014), 'Liquidation in limit order books with controlled intensity', *Mathematical Finance* **24**(4), 627–650. doi:10.1111/j.1467-9965.2012.00529.x

Beverungen, A. (2019), Algorithmic Trading, Artificial Intelligence and the Politics of Cognition, *in* A. Sudmann, ed., 'The Democratization of Artificial Intelligence', transcript Verlag, Bielefeld, pp. 77–94. doi:10.1515/9783839447192-005

Beverungen, A. and Lange, A.-C. (2018), 'Cognition in High-Frequency Trading: The Costs of Consciousness and the Limits of Automation', *Theory, Culture & Society* **35**(6), 75–95. doi:10.1177/0263276418758906

Borovkova, S. and Tsiamas, I. (2019), 'An ensemble of LSTM neural networks for high-frequency stock market classification', *Journal of Forecasting* **38**(6), 600–619. doi:10.1002/for.2585

Cao, C., Hansch, O. and Wang, X. (2009), 'The information content of an open limit-order book', *Journal of Futures Markets: Futures, Options, and Other Derivative Products* **29**(1), 16–41. doi:10.1002/fut.20334

Cartea, Á. and Jaimungal, S. (2013), 'Modelling asset prices for algorithmic and high-frequency trading', *Applied Mathematical Finance* **20**(6), 512–547. doi:10.1080/1350486X.2013.771515

Cartea, Á., Jaimungal, S. and Ricci, J. (2014), 'Buy low, sell high: A high frequency trading perspective', *SIAM Journal on Financial Mathematics* **5**(1), 415–444. doi:10.1137/130911196

Chakravarty, S. (2001), 'Stealth-trading: Which traders' trades move stock prices?', *Journal of Financial Economics* **61**(2), 289–307. doi:10.1016/S0304-405X(01)00063-0

Challet, D., Chicheportiche, R., Lallouache, M. and Kassibrakis, S. (2018), 'Statistically validated lead-lag networks and inventory prediction in the foreign exchange market', *Advances in Complex Systems* **21**(08), 1850019. doi:10.1142/S0219525918500194

Chen, L., Pelger, M. and Zhu, J. (2022), 'Deep Learning in Asset Pricing', *Management Science* **forthcoming**. doi:10.1287/mnsc.2023.4695

Comerton-Forde, C., Malinova, K. and Park, A. (2018), 'Regulating dark trading: Order flow segmentation and market quality', *Journal of Financial Economics* **130**(2), 347–366. doi:10.1016/j.jfineco.2018.07.002

Cont, R., Kukanov, A. and Stoikov, S. (2014), 'The price impact of order book events', *Journal of Financial Econometrics* **12**(1), 47–88. doi:10.1093/jjfinec/nbt003

Currie, W. L., Seddon, J. J. and Van Vliet, B. (2022), 'From decision optimization to satisficing: Regulation of automated trading in the US financial markets', *Information & Management* **59**(8), 103721. doi:10.1016/j.im.2022.103721

Dalio, R. (2017), 'Billionaire Ray Dalio Reveals His 'Principles' for Success in Life and Investing'. Interview by *TheStreet*. Avaialbe from https://www.thestreet.com/investing/billionaire-ray-dalio-reveals-his-principles-for-success-in-life-and-investing-14309949.

De Bondt, W. F. M. (1993), 'Betting on trends: Intuitive forecasts of financial risk and return', *International Journal of Forecasting* **9**, 355–371. doi:10.1016/0169-2070(93)90030-Q

De Bondt, W. F. M. and Thaler, R. (1985), 'Does the stock market overreact?', *Journal of Finance* **40**(3), 793–805. doi:10.2307/2327804

Degryse, H., Van Achter, M. and Wuyts, G. (2022), 'Plumbing of securities markets: The impact of post-trade fees on trading and welfare', *Management Science* **68**(1), 635–653. doi:10.1287/mnsc.2020.3880

Deml, T. (2022), 'Personal work'. Available from https://commons.wikimedia.org/wiki/File:Gaming-Wall-Street_BTS_Prodigium-266.jpg.

Dixon, M. (2018*a*), 'A high-frequency trade execution model for supervised learning', *High Frequency* **1**(1), 32–52. doi:10.1002/hf2.10016

Dixon, M. (2018*b*), 'Sequence classification of the limit order book using recurrent neural networks', *Journal of Computational Science* **24**, 277–286. doi:10.1016/j.jocs.2017.08.018

Dixon, M., Halperin, I. and Bilokon, P. (2020), *Machine Learning in Finance: From Theory to Practice*, Springer International Publishing. doi:10.1007/978-3-030-41068-1

ESMA (European Securities and Markets Authority) (2023), Artificial intelligence in EU securities markets, *in* 'ESMA TRV Risk Analysis'. Available from https://www.esma.europa.eu/sites/default/files/library/ESMA50-164-6247-AI_in_securities_markets.pdf.

Exegy (2023), 'MarketDataPeaks'. Real-time visualization of aggregated market data message rates across United States exchanges. Available from https://marketdatapeaks.net/.

Fabbri, M. and Moro, G. (2018), Dow Jones Trading with Deep Learning: The Unreasonable Effectiveness of Recurrent Neural Networks, *in* 'Proceedings of the 7th International Conference on Data Science, Technology and Applications', DATA 2018, SciTePress, pp. 142–153. doi:10.5220/0006922101420153

Felder, C. and Mayer, S. (2022), Customized Stock Return Prediction with Deep Learning, *in* '2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)', pp. 1–8. doi:10.1109/CIFEr52523.2022.9776177

Felder, C. and Seemüller, J. (2022), Intelligent Inventory Management for Cryptocurrency Brokers, *in* 'Proceedings of the Third ACM International Conference on AI in Finance', ICAIF '22, Association for Computing Machinery, New York, NY, USA, pp. 1–8. doi:10.1145/3533271.3561661

Fischer, T. and Krauss, C. (2018), 'Deep learning with long short-term memory networks for financial market predictions', *European Journal of Operational Research* **270**(2), 654–669. doi:10.1016/j.ejor.2017.11.054

Flori, A. and Regoli, D. (2021), 'Revealing Pairs-trading opportunities with long short-term memory networks', *European Journal of Operational Research* **295**(2), 772–791. doi:10.1016/j.ejor.2021.03.009

FMSB (FICC Markets Standards Board Limited) (2020), Emerging themes and challenges in algorithmic trading and machine learning, *in* 'Spotlight Review'. Available from https://fmsb.com/wp-content/uploads/2020/04/FMSB-Spotlight-Review-%E2%80%98Emerging-themes-and-challenges-in-algorithmic-trading-and-machine-learning%E2%80%99.pdf.

Fox, M. B., Glosten, L. and Rauterberg, G. (2019), *The New Stock Market: Law, Economics, and Policy*, Columbia University Press, New York Chichester, West Sussex. doi:10.7312/fox-18196

Francioni, R. and Schwartz, R. (2008), Dialog with Reto Francioni, *in* R. A. Schwartz, A. Colaninno and J. A. Byrne, eds, 'Competition in a Consolidating Environment', Springer US, Boston, MA, pp. 1–5. doi:10.1007/978-0-387-75943-2_1

Frey, S. and Grammig, J. (2008), Liquidity supply and adverse selection in a pure limit order book market, *in* 'High Frequency Financial Econometrics', Springer, pp. 83–109. doi:10.1007/978-3-7908-1992-2_5

FSB (Financial Stability Board) (2017), 'Artificial intelligence and machine learning in financial services: Market developments and financial stability implications'. Available from https://www.fsb.org/wp-content/uploads/P011117.pdf.

Grammig, J. and Theissen, E. (2012), 'Is Best Really BETTER? Internalization of Orders in an Open Limit Order Book', *Schmalenbach Business Review (sbr)* **64**(2), 82–100. doi:10.1007/BF03396891

Guéant, O. (2017), 'Optimal market making', *Applied Mathematical Finance* **24**(2), 112–154. doi:10.1080/1350486X.2017.1342552

Guéant, O., Lehalle, C.-A. and Fernandez-Tapia, J. (2013), 'Dealing with the inventory risk: a solution to the market making problem', *Mathematics and financial economics* **7**(4), 477–507. doi:10.1007/s11579-012-0087-0

Guilbaud, F. and Pham, H. (2013), 'Optimal high-frequency trading with limit and market orders', *Quantitative Finance* **13**(1), 79–94. doi:10.1080/14697688.2012.708779

Hagerty, K. and McDonald, R. L. (1996), Brokerage, Market Fragmentation and Securities Market Regulation, *in* A. W. Lo, ed., 'The Industrial Organization and Regulation of the Securities Industry', National Bureau of Economic Research Conference Report series, University of Chicago Press, pp. 35–56. Available from https://www.nber.org/system/files/chapters/c8101/c8101.pdf.

Hansch, O., Naik, N. Y. and Viswanathan, S. (1999), 'Preferencing, Internalization, Best Execution, and Dealer Profits', *Journal of Finance* **54**(5), 1799–1828. doi:10.1111/0022-1082.00167

Hendershott, T., Jones, C. M. and Menkveld, A. J. (2011), 'Does Algorithmic Trading Improve Liquidity?', *Journal of Finance* **66**(1), 1–33. doi:10.1111/j.1540-6261.2010.01624.x

Jiang, J., Kelly, B. T. and Xiu, D. (2022), '(Re-) Imag (in) ing Price Trends', *Journal of Finance* **forthcoming**. Available at SSRN 3756587. doi:10.2139/ssrn.3756587

Kahneman, D. and Tversky, A. (1979), 'Prospect Theory: An Analysis of Decision under Risk', *Econometrica* **47**(2), 263–291. doi:10.2307/1914185

Kennedy, D. (2017), 'The machine in the market: Computers and the infrastructure of price at the New York Stock Exchange, 1965–1975', *Social Studies of Science* **47**(6), 888–917. doi:10.1177/0306312717739367

Kercheval, A. N. and Zhang, Y. (2015), ‘Modelling high-frequency limit order book dynamics with support vector machines’, *Quantitative Finance* **15**(8), 1315–1329. doi:10.1080/14697688.2015.1032546

Kirilenko, A. A. and Lo, A. W. (2013), ‘Moore’s Law versus Murphy’s Law: Algorithmic Trading and Its Discontents’, *Journal of Economic Perspectives* **27**(2), 51–72. doi:10.1257/jep.27.2.51

Kumpan, C. (2006), ‘Carrot and Stick – The EU’s Response to New Securities Trading Systems’, *European Company and Financial Law Review* **3**(4), 383–407. doi:10.1515/ECFR.2006.017

Linnainmaa, J. (2010), ‘Do limit orders alter inferences about investor performance and behavior?’, *Journal of Finance* **65**(4), 1473–1506. doi:10.1111/j.1540-6261.2010.01576.x

MacKenzie, D. (2018), ‘Material Signals: A Historical Sociology of High-Frequency Trading’, *American Journal of Sociology* **123**(6), 1635–1683. doi:10.1086/697318

OECD (Organisation for Economic Co-operation and Development) (2021), ‘Artificial Intelligence, Machine Learning and Big Data in Finance: Opportunities, Challenges, and Implications for Policy Makers’. Available from https://www.oecd.org/finance/financial-markets/Artificial-intelligence-machine-learning-big-data-in-finance.pdf.

O’Halloran, T. (1963), ‘Library of Congress/Science Photo Library’. Available from https://www.loc.gov/pictures/item/2003673967/.

OTA (U.S. Congress, Office of Technology Assessment) (1990), *Electronic Bulls & Bears: U.S. Securities Markets & Information Technology, OTA-CIT-469*, U.S. Government Printing Office, Washington DC. Avalaible from https://ota.fas.org/reports/9015.pdf.

Pardo-Guerra, J. P. (2019), *Automating Finance: Infrastructures, Engineers, and the Making of Electronic Markets*, Cambridge University Press. doi:10.1017/9781108677585

Peterson, M. A. and Sirri, E. R. (2003), ‘Order Preferencing and Market Quality on U.S. Equity Exchanges’, *Review of Financial Studies* **16**(2), 385–415. doi:10.1093/rfs/hhg009

Pozen, R. and Ruane, J. (2019), ‘What machine learning will mean for asset managers’, *Harvard Business Review* **12/3/2019**. Available from https://hbr.org/2019/12/what-machine-learning-will-mean-for-asset-managers.

Schwartz, R., Cormack, M., Domowitz, I., Federspiel, F., Francioni, R., Mogavero, D. and Ryan, M. (2005), How Best to Integrate the Order Flow, *in* R. A. Schwartz, J. A. Byrne and A. Colaninno, eds, ‘Coping with Institutional Order Flow’, Springer US, Boston, MA, pp. 59–83. doi:10.1007/0-387-25881-7_3

SEC (U.S. Securities and Exchange Commission) (2013), Trade Execution: What Every Investor Should Know, *in* ‘Investor Publications’. Available from https://www.sec.gov/reportspubs/investor-publications/investorpubstradexec.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M. and Iosifidis, A. (2017), Forecasting stock prices from the limit order book using convolutional neural networks, *in* ‘2017 IEEE 19th conference on business informatics (CBI)’, Vol. 1, IEEE, pp. 7–12. doi:10.1109/CBI.2017.23

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M. and Iosifidis, A. (2020), ‘Using deep learning for price prediction by exploiting stationary limit order book features’, *Applied Soft Computing* **93**, 106401. doi:10.1016/j.asoc.2020.106401

Tversky, A. and Kahneman, D. (1992), ‘Advances in prospect theory: Cumulative representation of uncertainty’, *Journal of Risk and uncertainty* **5**(4), 297–323. doi:10.1007/BF00122574

Zhang, Z., Zohren, S. and Roberts, S. (2019), ‘DeepLOB: Deep convolutional neural networks for limit order books', *IEEE Transactions on Signal Processing* **67**(11), 3001–3012. doi:10.1109/TSP.2019.2907260

Zhang, Z., Zohren, S. and Roberts, S. (2020), 'Deep Reinforcement Learning for Trading', *Journal of Financial Data Science* **2**(2), 25–40. doi:10.3905/jfds.2020.1.030

Zheng, B., Moulines, E. and Abergel, F. (2013), 'Price jump prediction in limit order book', *Journal of Mathematical Finance* **3**(2), 242–255. doi:10.4236/jmf.2013.32024

2 Return Prediction with Deep Learning under Loss Aversion[3]

Christopher Felder and Stefan Mayer

Abstract

In finance, researchers typically use standard loss functions such as mean squared error when training artificial neural networks for return prediction. However, prospect theory suggests that people are more sensitive to losses than gains. Thus, investors might assess overprediction differently than underprediction of portfolio returns. We present a loss function designed for loss-averse investors and test it based on Long Short-Term Memory (LSTM) models. Our model learns unique signals, predicts returns more cautiously, and improves profit chances over the standard LSTM and reversal signals. Daily and weekly revised portfolios achieve on average five percentage points higher annualized returns. We show that our loss function is robust to market sentiment and beneficial in nonlinear optimization.

Keywords: Time series analysis, Loss function, Return prediction, Prospect theory.

JEL classification: C45, C55, C58, G11, G12, G17.

[3] A previous version of this article has been published in Felder and Mayer (2022). Relevant paragraphs, tables, and figures are reprinted with permission. ©2022 IEEE

2.1 Introduction

An extensive literature examines trend-based return predictability and identifies numerous predictors widely used in practice (Sullivan et al., 1999; Bajgrowicz and Scaillet, 2012; Jiang et al., 2022). In recent years, dynamic machine learning approaches have revolutionized technical analysis. One powerful deep learning tool well suited for time series problems is the Long Short-Term Memory (LSTM) proven successful in many return prediction problems, see Fischer and Krauss (2018), Fabbri and Moro (2018), Borovkova and Tsiamas (2019), Chen et al. (2022), Flori and Regoli (2021)—to name a few.

We reconsider return prediction with LSTM from a behavioral perspective and link the optimization problem to prospect theory (Kahneman and Tversky, 1979). Under prospect theory, people assess potential losses and gains asymmetrically when making decisions, contradicting with the expected-utility framework. One feature of prospect theory is loss aversion (Tversky and Kahneman, 1992), explaining that people perceive losses greater than equivalent gains and, e.g., tend to reject 50:50 bets to win $110 or lose $100. Correspondingly, an investor predicting a portfolio return of 5% would be severely disappointed if the actual return was 3%, but would be satisfied if the actual return was 7%. Standard optimization techniques handle those two prediction errors equally, and both the positive and negative errors receive the same penalty, even though the investor financially suffers from only one error.

We propose a loss function distinguishing between 'good' and 'bad' prediction losses. 'Good' losses occur when the predicted portfolio return is lower than the realized return, 'bad' losses describe the opposite case. The economic objective is to minimize both 'good' and 'bad' prediction losses, with 'good' losses preferred over 'bad' losses. Our framework is based on a weighted mean squared error, where the weight is a function of the loss type. Motivated by Tversky and Kahneman (1992), who show that people are twice as sensitive to losses as they are to gains, we allow the weight to penalize 'bad' losses up to twice as hard as 'good' losses. Applied to our previous example, we would penalize the prediction error twice as hard when the realized return is 3% than when it is 7%. We discuss a binary weight function that only knows the maximum and minimum penalties, and a sigmoid weight function that distributes weights continuously between the maximum and minimum penalties.

We benchmark our new approach with a standard LSTM as well as short-term reversal, one of the most popular and robust trend-based predictors proven competitive with deep learning (Jiang et al., 2022), whose signals are probably similar to LSTM signals (Fischer and Krauss, 2018; Chen et al., 2022; Gu et al., 2020). We arrive at five findings: First, the loss-averse LSTMs underpredict positive returns rather than overpredict them. Investors are less likely to face situations in which the realized portfolio return is lower than the predicted return. Second, we report improved portfolio performance when tested for different prediction horizons. Third, in addition to well-known reversal signals, we extract new unique signals that contain superior predictive information. Fourth, an investor benefits from the loss functions in the nonlinear environment of the LSTM. Fifth, our loss function contributes positively to portfolio returns throughout changing market sentiments.

The remainder of this article proceeds as follows. Section 2.2 introduces to LSTM networks, Section 2.3 discusses intersections between reversal and LSTM signals, Section 2.4 embeds our research in the current state of literature, and Section 2.5 derives a trading-customized loss function. In Section 2.6 we discuss our empirical results for daily U.S. stock returns, whereas Section 2.7 reviews the benefits of the loss function and Section 2.8 concludes.

2.2 LSTM networks

2.2.1 The Long Short-Term Memory architecture

The LSTM[4] is an artificial recurrent neural network architecture introduced by Hochreiter and Schmidhuber (1997). While traditional feedforward neural networks are unable to extract temporally encoded information from time series data, recurrent neural networks use time components based on feedback connections with neurons of the same or previous layers (Rumelhart et al., 1986*a*,*b*). A hidden state transfers information across time steps and facilitates stepwise processing of entire sequences to connect information between time steps. However, Bengio et al. (1994) and Hochreiter (1998) discuss that conventional recurrent networks suffer from short-term memory in practice. The longer the time gap between the occurrence of an information and when we need it, the less able recurrent networks are to remember and link the information, thereby forgetting information far in the past and focusing on the most recent information (see also Section 4.4.2). Technically, the problem arises from the fact that gradients used to update the network parameters shrink over time and do not contribute to learning anymore (Hochreiter, 1998; Hochreiter et al., 2001).

LSTMs overcome these deficits by using two different states of memory: The cell state s serves as a long-term memory and carries all information learned in the past when processing a sequence. The hidden state h serves as short-term memory and carries the cell state adapted to the information from the recent steps. Instead of updating all parameters at each step of the sequence, multiple layers acting as gates protect the cell state from unnecessary updates. Thus, information far in the past survives, but relevant new information still can remove or update cell knowledge. A gate is a mechanism to filter information before passing it to the state, consisting of a logistic layer that determines which information to pass and a multiplication operation with the vector of information. A value of 0 (1) indicates that we should forget (keep) all of the component's information, whereas values between 0 and 1 indicate that we should keep specific parts of the information.

An LSTM cell has three gates: The forget gate f decides which information from the previous cell state to forget, the input gate i updates the cell state with new information, and the output gate o generates the new hidden state. Figure 2.1 illustrates an LSTM cell at time t, where the inputs to the cell are the vector of cell states s_{t-1}, the vector of hidden states h_{t-1}, and the vector of elements of the current input sequence x_t.

Each of the gates processes the input information one after another. First, the forget gate processes vectors h_{t-1} and x_t through a logistic layer σ and determines which components of s_{t-1} to remove. The output of the forget gate at time step t is a vector f_t that contains activation values for each element in s_{t-1}. f_t is defined by

$$f_t = \sigma\left(w_{f,x} \cdot x_t + w_{f,h} \cdot h_{t-1} + b_f\right) \tag{2.1}$$

where w_f depicts the weight matrix and b_f depicts the bias vector. When later computing the new cell state in equation (2.4), we use element-wise multiplication ($\circ$) of f_t by s_{t-1} to activate those elements in the cell state s_{t-1} that the layer decides to remove.

Second, the input gate uses the weight matrix w_i and the bias vector b_i to control that only the necessary information from h_{t-1} and x_t will pass to the cell state. The output of the input

[4] Section 2.2.1 describes the functionality of LSTMs following Olah (2015), Fischer and Krauss (2018), Aggarwal et al. (2018), Graves (2013), Brownlee (2017), Karpathy (2015), Zhao et al. (2017), and Sherstinsky (2020).

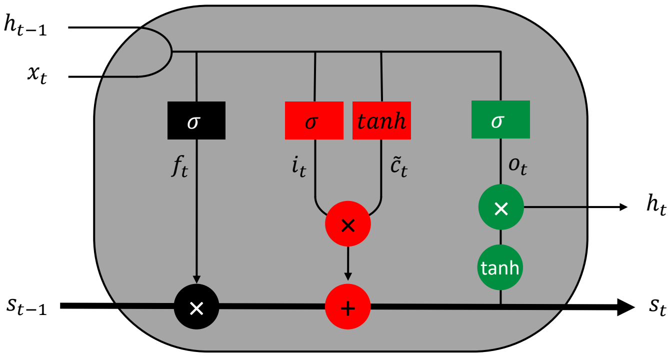

Figure 2.1: Structure of an LSTM cell following Olah (2015) with cell state s, hidden state h, forget gate f (black), input gate i (red), and output gate o (green) at time t. Rectangles denote neural network layers, circles are element-wise operations.

gate at time step t is a vector of activation values, i_t, that is defined by

$$i_t = \sigma\left(w_{i,x} \cdot x_t + w_{i,h} \cdot h_{t-1} + b_i\right) \tag{2.2}$$

and contains values between 0 and 1 that activate elements of h_{t-1} and x_t according to their relevance for the cell state. In a parallel operation, a *tanh* layer computes the information from the input vectors h_{t-1} and x_t that we potentially could add to the cell state. Accordingly, we calculate the vector of these candidate values $\tilde{c}_t$ at time step t by

$$\tilde{c}_t = tanh\left(w_{\tilde{c},x} \cdot x_t + w_{\tilde{c},h} \cdot h_{t-1} + b_{\tilde{c}}\right) \tag{2.3}$$

where the weight matrix $w_{\tilde{c}}$ and the bias vector $b_{\tilde{c}}$ highlight the important information in h_{t-1} and x_t when adding to the cell state. In contrast to the sigmoid function, *tanh* scales the input information to a range between −1 and 1, thereby avoiding that signals disappear when multiplied by zero. Now, the new cell state s_t equals the sum of $\tilde{c}_t$ scaled by the input gate's activation values i_t and the information from s_{t-1} that survived the forget gate:

$$s_t = f_t \circ s_{t-1} + i_t \circ \tilde{c}_t. \tag{2.4}$$

Third, the output gate o manages the information flow to the hidden state through the weight matrix w_o and the bias vector b_o. Analogous to equations (2.1) and (2.2), we calculate the activation values of the output gate, o_t, by

$$o_t = \sigma\left(w_{o,x} \cdot x_t + w_{o,h} \cdot h_{t-1} + b_o\right) \tag{2.5}$$

determining which information from the inputs h_{t-1} and x_t to pass to the new hidden state h_t. Finally, we compute the LSTM layer's new hidden state output by

$$h_t = o_t \circ tanh\left(s_t\right) \tag{2.6}$$

where a *tanh* operation for the current cell state s_t ensures that no important information gets lost. Thus, the new hidden state output is the cell state vector s_t filtered by the important short-term information carried by h_{t-1} and x_t.

2.2.2 Network configuration

We propose an LSTM network that processes one-day returns on stock j and one-day returns on the market portfolio m. Our approximation to the market portfolio is an equally weighted portfolio holding the S&P 500 and the Nasdaq Composite. We follow Fischer and Krauss (2018) and calculate the one-day return on asset j on day t based on the closing price p by

$$r_{j,t} = \frac{p_{j,t}}{p_{j,t-1}} - 1. \tag{2.7}$$

Figure 2.2 illustrates the architecture of our network, comprising an input layer with two features ($r_{j,t}$, $r_{m,t}$) and 60 days time dimension, one single hidden layer with 32 LSTM cells and a one-node dense output layer ($r_{j,t}$), which is similar to Fischer and Krauss (2018) and Flori and Regoli (2021). The time dimension varies by researchers from five to 60 days in Jiang et al. (2022) up to one year in Flori and Regoli (2021). We choose a sequence length of 60 days, accounting for the fact that LSTMs retrieve the most important information from recent time steps. At market close on day t, the network processes the sequence $[r_{j,t-59}, r_{m,t-59}, r_{j,t-58}, r_{m,t-58}, ..., r_{j,t}, r_{m,t}]$ and predicts the next one-day stock return, $r_{j,t+1}$. Strictly speaking, it is not possible to consider the closing price to predict the next one-day return and simultaneously buy or sell at the same closing price. Accordingly, we consider the closing price as an approximation of the last market price prior to market close used to deliver a prediction.

We regularize training in three ways. First, we prohibit cell states from communicating across training batches. In a stateful LSTM, memory transfers information between sequences of different batches so that states persist for a full training epoch. Our stateless configuration allows states to transfer information between sequences of the same batch, but resets memory after one batch. Second, we allow the network to shuffle samples within batches to ensure generalizability across batches. Batches thus contain randomly picked sequences from different shares at different time points. Third, we apply model averaging using dropout (Srivastava et al., 2014). Dropout randomly ignores a certain fraction of neurons in hidden

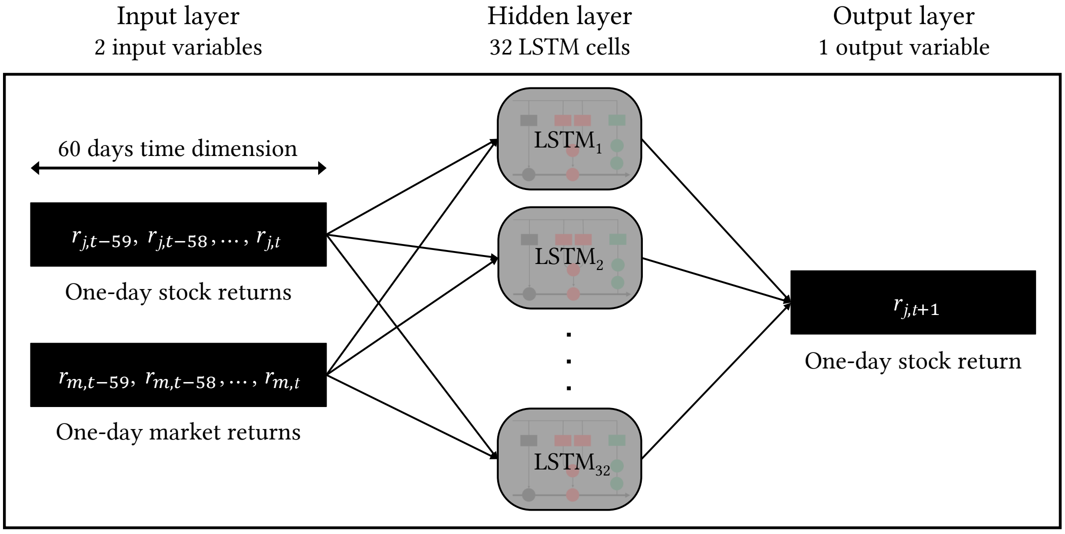

Figure 2.2: LSTM network architecture comprising two input variables (r_j, r_m) with 60 days time dimension, one single hidden layer with 32 LSTM cells, and a one-node dense output layer.

and visible layers. Motivated by Fischer and Krauss (2018) who implement a dropout ratio of 10% and Jiang et al. (2022) who use 50% dropout, we use a dropout ratio of 30%. For the optimization, we use stochastic gradient descent with an initial learning rate of 0.001 and a batch size of 8. The activation function on the hidden units is the hyperbolic tangent. Similar to Fischer and Krauss (2018), we train our models using `keras` (Chollet et al., 2015) and Google's `TensorFlow` (Abadi et al., 2015). The training runs for 50 epochs.

2.3 Short-term reversal signals

A well-known trend phenomenon is short-term reversal. Stocks with relatively poor (strong) performance in a previous short-term period, e.g., one month, are likely to produce positive (negative) abnormal returns in the following period (Fama, 1965; Jegadeesh, 1990; Lehmann, 1990; Jegadeesh and Titman, 1993). According to Flori and Regoli (2021), among the most common explanations are investor behavioral bias leading to over- or underreaction to news events (Jegadeesh and Titman, 1995; Subrahmanyam, 2005; Lehmann, 1990), and size (Lo and MacKinlay, 1990; Boudoukh et al., 1994) and liquidity effects (Avramov et al., 2006; Chordia et al., 2002), which foster reversal patterns for small, illiquid shares. Research on reversal phenomena assumes that past performances may affect investors' future trading behavior and thus contain predictive signals for future returns, thereby contradicting with the efficient market hypothesis by Fama (1970), which states that returns are stochastic and unpredictable.

Many studies document a close relationship between reversal signals and LSTM signals. In Fischer and Krauss (2018), LSTMs buy (sell) stocks that previously exhibited sharp declining (increasing) returns. Other researchers employ LSTMs utilizing reversal patterns (Flori and Regoli, 2021) and point out that LSTMs extract the most important information from short-term reversal signals (Chen et al., 2022; Gu et al., 2020). Besides, there are also studies claiming that signals have less in common (Guijarro-Ordonez et al., 2021). Another well-documented trend signal is momentum (Jegadeesh and Titman, 1993). Since LSTMs do not extract momentum signals (Fischer and Krauss, 2018) and momentum portfolio returns are less competitive with LSTMs (Jiang et al., 2022), we choose to limit our financial benchmark to reversal.

We follow Jiang et al. (2022) and benchmark our model against month reversal (MR) and week reversal (WR), which consider the returns achieved in the last month and week, respectively. Additionally, we include the midweek reversal (MWR) that focuses on returns achieved in the last two days. We calculate returns following equation (2.7), replacing the reference price $p_{j,t-1}$ with the closing price one month (MR), one week (WR) or two trading days (MWR) ago. A reversal strategy buys (sells) the shares with the most negative (positive) returns in the previous period.

2.4 Related work

We build our methodology on Fischer and Krauss (2018) and Jiang et al. (2022), both of which have analyzed a U.S. stock sample similar to ours. While Fischer and Krauss (2018) focus on return classification and compare LSTMs to memory-free methods, Jiang et al. (2022) use convolutional neural networks and document improvements in prediction quality and portfolio performance over WR and MR. Our motivation stems from Avramov et al. (2023), who discuss that the return prediction power of LSTMs weakens when taking into account investor constraints, such as excluding micro caps, distressed firms or firms with poor credit ratings.

Literature widely debates the reliability of trend analysis. While Sullivan et al. (1999) and Bajgrowicz and Scaillet (2012) doubt significant benefit of technical trading rules, Brock et al. (1992) and Lo et al. (2000) present evidence for predictive information in some trading signals (Jiang et al., 2022). Although the invention of LSTM dates back more than two decades, with the increasing availability of computational power and data, financial research on trend analysis has discovered LSTM-driven approaches to model time series data only a few years ago (Chen et al., 2015). The literature discusses return predictions with LSTMs that disentangle technical trading indicators (Bao et al., 2017; Sang and Di Pierro, 2019), process high-frequency data (Borovkova and Tsiamas, 2019; Nelson et al., 2017), improve prediction quality over ARIMA or feedforward neural networks (Fabbri and Moro, 2018; Siami-Namini et al., 2018), detect buy and sell signals from market data (Fischer and Krauss, 2018; Troiano et al., 2018), model the stochastic discount factor and explore economic state processes (Chen et al., 2022), trace reversal patterns in pairs trading (Flori and Regoli, 2021), and include additional layers with attention mechanisms (Qiu et al., 2020).

Besides, our work addresses studies of prospect theory and loss aversion in financial market modeling. To date, behavioral research studies investigate the impact of loss aversion on return predictability (Barberis et al., 2001), the predictive power of prospect theory values for stock returns (Barberis et al., 2016), asset prices and trade volume under prospect theory (Li and Yang, 2013), implications of loss aversion for asset prices (Yang, 2019), and the level of loss aversion implied by historical stock returns (Berkelaar et al., 2004).

We contribute to both research disciplines with a novel approach that links the optimization problem of return prediction to the behavioral characteristics of trading. Recent literature debates approaches that regard the optimization problem from a market perspective, e.g., by weighting prediction errors by market capitalization (Gu et al., 2020) or by the length of the observation period (Chen et al., 2022). We show that considering the optimization problem from a behavioral perspective provides investors with more cautious return predictions and thus adapts to investors' loss aversion. Thus, although not explicitly addressing this question, our work also comes in the wake of recent debates about whether investors *are* loss-averse at all or whether loss aversion under prospect theory is a result of sampling biases in favor of individuals with high cognitive ability (Chapman et al., 2022).

2.5 A loss-averse loss function

As part of the optimization algorithm, the loss function used to train a deep neural network plays a pivotal role: The loss of a prediction determines by how much the network adjusts the weights to reduce the loss of the next prediction. Large losses are associated with large adjustments, small losses are associated with small adjustments. Our goal is to make the loss function distinguish between positive and negative prediction errors and force the network adjust weights more when realized returns are lower than predicted.

In practice, where investors can both buy and short shares, there are four cases of prediction losses. We assume an investor who buys shares with a positive return prediction ($\hat{r}_{j,t} > 0$) and short-sells shares with a negative return prediction ($\hat{r}_{j,t} < 0$). The prediction error is $\hat{r}_{j,t} - r_{j,t}$. Figure 2.3 illustrates the change in returns on portfolio p, $r_{p,t}$, as a function of the loss type for long and short investments. Prediction errors in quadrants II and IV are associated with increasing portfolio returns: In the case of a long (short) position, investors would be happy to see a negative (positive) prediction error because the portfolio return increases unexpect-

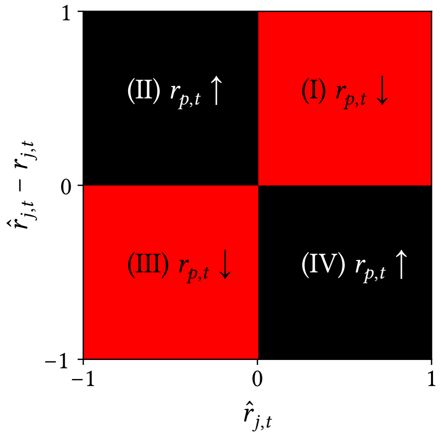

Figure 2.3: Change in portfolio returns $r_{p,t}$ by prediction loss type in the range $[-1, 1]$. Quadrants I and II (III and IV) represent a positive (negative) return prediction error. Red (black) quadrants indicate decreased (increased) portfolio returns $r_{p,t}$ in response to 'bad' ('good') prediction errors. © 2022 IEEE

edly, thereby financially benefiting from this 'good' prediction error. In quadrants I and III, the investor faces unexpectedly lower portfolio returns ('bad' errors) resulting from a return overprediction (underprediction) for a long (short) position.

We use the term 'unexpected portfolio loss' to describe a situation in which the realized portfolio return is less than the predicted portfolio return, but can still be positive. For example, a predicted return of 5% with a realized return of 3% means an unexpected loss of 2%. An 'unexpected portfolio gain' describes a situation where the realized return is greater than the predicted return, i.e., the realized return on the portfolio is larger than predicted.

Next, we design a framework for a loss function capable to consider prediction losses according to Figure 2.3 under prospect theory and loss aversion. The most common loss function for regression problems is the mean squared error (MSE). The MSE of a set of $n \cdot z$ return predictions for n shares and z time steps is

$$MSE = \frac{1}{nz} \sum_{j=1}^{n} \sum_{t=1}^{z} \left(\hat{r}_{j,t} - r_{j,t}\right)^2 . \tag{2.8}$$

Accordingly, MSE treats any prediction loss equally, regardless of the sign. In order to weight prediction errors according to their loss type, we propose a weighted MSE ($WMSE$) in the style of Gu et al. (2020). We define $WMSE$ by

$$WMSE = \frac{1}{nz} \sum_{j=1}^{n} \sum_{t=1}^{z} w\left(\hat{r}_{j,t}, r_{j,t}\right) \left(\hat{r}_{j,t} - r_{j,t}\right)^2 \tag{2.9}$$

where the weight $w\left(\hat{r}_{j,t}, r_{j,t}\right)$ is a function of the loss type determined by $\hat{r}_{j,t}$ and $r_{j,t}$. To comply with the properties of loss aversion (Kahneman and Tversky, 1979), $w\left(\hat{r}_{j,t}, r_{j,t}\right)$ must satisfy at least two requirements for $WMSE$: First, $WMSE$ should penalize small 'bad' errors heavier than small 'good' errors. Second, large 'bad' errors should receive the maximum weight, and large 'good' errors should receive the minimum weight.

Next, we configure the weight function according to prospect theory (Kahneman and Tversky, 1979; Tversky and Kahneman, 1992). As outlined by Yang (2019), Tversky and Kahneman

(1992) propose that people assess gains and losses by the value function

$$v(\rho) = \begin{cases} \rho & \text{for} \quad \rho \geq 0 \\ \lambda\rho & \text{for} \quad \rho < 0 \end{cases} \tag{2.10}$$

where $\lambda > 1$ controls for the degree of loss aversion and $\rho \geq 0$ ($\rho < 0$) indicates gains (losses). The larger λ, the larger the sensitivity to losses. Tversky and Kahneman (1992) show that, empirically, people are more sensitive to losses than gains by a factor of two ($\lambda \approx 2$). We follow Tversky and Kahneman (1992) and propose to weight prediction errors that lead to unexpected portfolio losses twice as much as an error that leads to unexpected portfolio gains. Hence, the numerical range for the minimum and maximum weights is $w\left(\hat{r}_{j,t}, r_{j,t}\right) \in [1, 2]$.

Analogous to $v(\rho)$, we consider a binary weight distribution in the first step. A binary weight function forces the model to distinguish exclusively between 'bad' and 'good' errors. Small negative errors are penalized in the same way as large negative errors, and the difference in penalty between small positive and small negative errors equals the difference for large positive versus large negative errors.

We define the binary weight function $w^{bin}\left(\hat{r}_{j,t}, r_{j,t}\right)$ by

$$w^{bin}\left(\hat{r}_{j,t}, r_{j,t}\right) = \begin{cases} 1.5 + sgn\left(\hat{r}_{j,t}\right)(0.5) & \text{for} \quad \hat{r}_{j,t} - r_{j,t} \geq 0 \\ 1.5 - sgn\left(\hat{r}_{j,t}\right)(0.5) & \text{for} \quad \hat{r}_{j,t} - r_{j,t} < 0 \end{cases} \tag{2.11}$$

where $sgn\left(\hat{r}_{j,t}\right)$ is the signum function of the predicted return $\hat{r}_{j,t}$ that determines the type of investment, taking -1 (resp. 1) for $\hat{r}_{j,t} < 0$ (resp. $\hat{r}_{j,t} > 0$) and 0 for $\hat{r}_{j,t} = 0$. If $\hat{r}_{j,t} \neq 0$, $w^{bin}\left(\hat{r}_{j,t}, r_{j,t}\right)$ can only take 1 or 2. Thus, if the investor buys the security ($\hat{r}_{j,t} > 0$), the penalty for a positive prediction error is $1.5 + 0.5 = 2$ and for negative prediction errors $1.5 - 0.5 = 1$. If the investor short-sells the security, the penalty for a positive prediction error is $1.5 - 0.5 = 1$.

Inspired by Kahneman and Tversky (1979), who present a value function that is concave for gains and convex for losses, we additionally consider a continuous weight function, providing a more economical understanding as small prediction losses have less impact on the portfolio performance than large losses and thus may receive a smaller penalty. The model learns that a large loss is worse than a small loss and penalizes very small positive losses similarly with very small negative losses. We define the continuous weight function $w^{log}\left(\hat{r}_{j,t}, r_{j,t}\right)$ by

$$w^{log}\left(\hat{r}_{j,t}, r_{j,t}\right) = \frac{1}{1 + e^{-sgn\left(\hat{r}_{j,t}\right)\left(\hat{r}_{j,t} - r_{j,t}\right)\cdot 100}} + 1. \tag{2.12}$$

Since the prediction loss usually is a small percentage, we multiply it by 100 to accelerate the impact of the weighting. Figure 2.4 illustrates the weights for prediction losses in the range $[-5\%, 5\%]$. For example, if the investor is long, a prediction loss of 2% (−2%) receives a weight of 1.88 (1.12), while a relatively small loss of 0.1% (−0.1%) receives a weight of 1.52 (1.48).

One might wonder how investors' preference for prediction errors would affect a model's prediction quality. Our loss function aims to deliver more conservative return predictions, i.e., to narrow the distribution of return predictions. In a scenario where overpredictions (underpredictions) exceed underpredictions (overpredictions) for positive (negative) returns, using the loss-averse loss function instead could improve the network's approximation of the market over the standard LSTM. As Table 2.1 and Figures 2.5 and 2.9 present empirical evidence that standard LSTMs indeed overpredict (underpredict) positive (negative) returns on average, we conclude that teaching a network to predict returns more conservatively can provide a better approximation of the market.

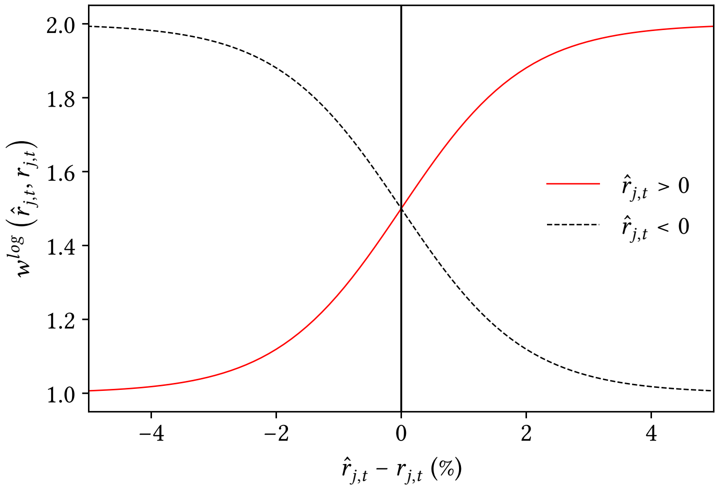

Figure 2.4: Logistic weight function $w^{log}\left(\hat{r}_{j,t}, r_{j,t}\right)$ for prediction errors $\hat{r}_{j,t} - r_{j,t}$ between −5% and 5%. $\hat{r}_{j,t} > 0$ depicts a long investment (red), $\hat{r}_{j,t} < 0$ depicts a short investment (black).

2.6 Prediction of U.S. stock returns

2.6.1 Data

We exploit the Refinitiv database[5] for daily adjusted closing prices of U.S. stocks primarily listed on NYSE or Nasdaq between 1990 and 2021. The set includes 9,289 shares from NYSE and 6,841 shares from Nasdaq. Since we address our work to private investors, we require a minimum market cap of $50 million to ensure liquidity and moderate transaction costs. As proposed by LeCun et al. (2012), we standardize returns by subtracting the cross-sectional mean of returns and dividing by the standard deviation prior to training the network.

We follow Jiang et al. (2022) and split the data set into a shorter period for training and validation (1990 to 2004), with validation accounting for the last quarter of the period, and a longer period for testing (2005 to 2021). Training a model once on the training data and then testing it on the test data makes the model more general compared to retraining parameters multiple times over rolling windows as in Fischer and Krauss (2018), but at the same time may result in lower performance as the model cannot adapt as easily to short-term dynamics (Jiang et al., 2022). An implicit assumption therefore is that the chosen test period is a suitable test period for what the model has learned during the training period. We perform the data split before sequencing the time series periods, i.e., no element of a sequence in one data set appears in a sequence of another data set. The validation set serves to calibrate the network's hyperparameters such as the number of LSTM cells or the batch size (see Section 2.2.2).

2.6.2 Prediction quality

In order to assess the behavior of the network for long and short investments, we split the test set into $\hat{r}_{j,t} \leq 0$ and $\hat{r}_{j,t} > 0$ groups. Table 2.1 reports the average prediction errors and the share of negative prediction errors for each group. LSTM^{bin} (LSTM^{log}) denotes the

[5] We download price data from Refinitiv by using the Refinitv Eikon software package, see https://www.refinitiv.com/en/products/eikon-trading-software.

Model	$\hat{r}_{j,t} \leq 0$	$\hat{r}_{j,t} > 0$	All
Average prediction error (%)			
LSTM^{bin}	−0.01	−0.07	−0.04
LSTM^{log}	0.08	0.03	0.05
LSTM^{mse}	−0.02	0.18	0.08
Share of negative prediction errors			
LSTM^{bin}	0.50	0.51	0.50
LSTM^{log}	0.48	0.49	0.49
LSTM^{mse}	0.49	0.47	0.48

Table 2.1: Prediction error metrics sorted by model and sign of the predicted return. The prediction error is the difference between the predicted and realized return ($\hat{r}_{j,t} - r_{j,t}$). © 2022 IEEE

LSTM trained with the $WMSE$ using the binary (logistic) weight function, whereas LSTM^{mse} represents the standard LSTM trained with the MSE.

In the loss-averse LSTMs, the average prediction errors are greater in the short position than in the long position (−0.01 > −0.07 and 0.08 > 0.03), i.e., the models favor 'good' losses when buying shares over 'bad' losses when selling shares. Moreover, the proportion of unexpected gains on long positions exceeds the proportion of unexpected losses on short positions (0.51 > 0.50 and 0.49 > 0.48). In contrast, the average error of the MSE model is greater in the long position than in the short position, resulting in larger unexpected losses on both types of investments, and investors face relatively more 'bad' than 'good' losses.

As aggregated ratios do not tell us about the actual distribution of prediction errors, we next examine the relationship between prediction errors and $\hat{r}_{j,t}$. To this end, Figure 2.5 illustrates average prediction errors as a function of $\hat{r}_{j,t}$ percentile groups, with percentile group 1 (100) containing the 1% lowest (highest) $\hat{r}_{j,t}$. The left chart in Figure 2.5 transfers the theoretical model from Figure 2.3 with exemplary values into a function of return predictions to illustrate what the extreme scenarios—a prediction model with only 'bad' or with only 'good' losses—would look like. Assumed that the median return prediction is zero, i.e., predictions are positive in the upper half of predictions and negative in the lower half, a prediction model with only 'bad' ('good') losses generates only positive (negative) prediction errors for positive return predictions and only negative (positive) errors for negative return predictions.

The right chart in Figure 2.5 demonstrates the error distribution of our empirical results and shows that the loss-averse LSTMs largely squeeze prediction errors into quadrants II and IV, and the 'bad' errors in quadrants I and III are small compared to the 'good' errors in quadrants II and IV. The negative slope indicates the preference for 'good' errors in both types of investments. The MSE model has 'bad' errors in about three quarters of $\hat{r}_{j,t}$ percentile groups. In particular, strong buy and strong sell classifications pose significant unexpected losses. If compared with the left chart in Figure 2.5, the loss-averse models behave most similarly to a prediction model with only 'good' losses, whereas the MSE model is similar to a prediction model with only 'bad' losses. In Appendix 2.A, we show that the total sum of unexpected losses in all unexpected movements is at least eight times larger in the MSE model than in the loss-averse strategies, which is due to the fact that the 'bad' losses in the loss-averse strategies are relatively small compared to the 'bad' losses in the MSE strategy.

Next, we convert our prediction task into a classification problem with buy ($\hat{r}_{j,t} > 0$) and

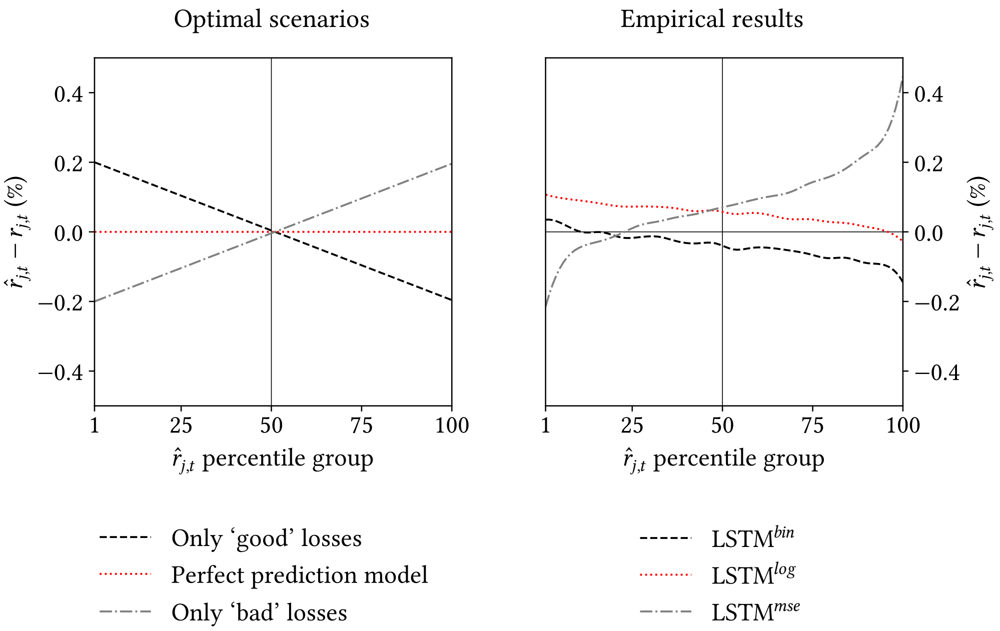

Figure 2.5: Prediction errors as a function of $\hat{r}_{j,t}$ percentile groups. The left chart illustrates a prediction model with only `good' losses, a prediction model with only `bad' losses, and a perfect prediction model based on theoretical values. The right chart presents our empirical results. © 2022 IEEE

sell ($\hat{r}_{j,t} < 0$) signals. Accordingly, reversals classify shares with preceding negative (positive) returns as buy (sell) class. Following the confusion matrix illustrated in Table 2.2, a true positive (negative) classification is a buy (sell) classification followed by a positive (negative) realized return, whereas a false positive (negative) classification is a buy (sell) classification followed by a negative (positive) realized return.

Our prediction quality measure is the prediction accuracy ACC, which determines the proportion of true classifications in all classifications by

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.13}$$

where TP (TN) indicates true positive (negative) classifications and FP (FN) indicates false positive (negative) classifications (see Table 2.2).

Table 2.3 reports the prediction accuracy of the models. All models correctly classify more than 50% of returns. The mean accuracy across all models is 52%, largely consistent with Jiang et al. (2022). Our loss functions lead to a marginal improvement over the standard models[6].

[6] It is important to note here that minor changes in accuracy are scalable for investors and can already provide financial benefit. For example, in Jiang et al. (2022), each 0.5% increase in accuracy for monthly signals results in

Realized return	Buy ($\hat{r}_{j,t} > 0$)	Sell ($\hat{r}_{j,t} < 0$)
$r_{j,t} > 0$	True positive (TP)	False negative (FN)
$r_{j,t} < 0$	False positive (FP)	True negative (TN)

Table 2.2: Confusion matrix for the directional return classification problem.

Model	*ACC*	Correlation
LSTM^{bin}	52.24	4.49
LSTM^{log}	52.29	4.58
LSTM^{mse}	52.17	4.35
MR	51.93	3.87
WR	52.16	4.33
MWR	52.20	4.41

Table 2.3: Prediction accuracy and correlation. Correlation is the cross-sectional Pearson correlation between return classifications and subsequent realized returns. MR denotes the month reversal strategy, WR denotes week reversal strategy, and MWR denotes the midweek reversal strategy. All numbers denote in percentage. © 2022 IEEE

In the right column, we also determine the cross-sectional correlation of signals and realized returns. Most signals have a correlation of about 4% with subsequent realized returns, which is slightly higher than in Jiang et al. (2022).

Finally, we follow Fischer and Krauss (2018) and employ the Diebold and Mariano (1995) test, finding empirical evidence against the null that the difference between predictions of the loss-averse LSTMs and the benchmark models is zero (all p-values < 0.05). Even when adjusting the critical p-value to 0.00625 to control for multiple testing (Bonferroni correction), we reject the null in seven out of eight tests. We conclude that our proposed loss function provides investors with superior forecasts.

2.6.3 Decile portfolio analysis

Next, we follow Gu et al. (2020), Flori and Regoli (2021), Chen et al. (2022), Jiang et al. (2022), and many other researchers and analyze portfolio performance based on signal deciles, where signal decile 1 (10) comprises the lowest (highest) 10% of return predictions. Similarly, reversal strategies assign the 10% shares with the most positive (negative) preceding returns to decile 1 (10). Accordingly, when correctly assigned, decile 1 has the lowest realized return and decile 10 has the highest realized return. Figure 2.6 illustrates realized daily returns and annualized standard deviations as functions of $\hat{r}_{j,t}$ deciles.

Similar to Jiang et al. (2022), realized returns are slightly negative in the first signal decile and positive in all other deciles, with an overall positive slope from decile 1 to decile 10. While the loss-averse LSTMs show the steepest increases at the edge deciles that flatten out around the median, the other models show steady to slightly increasing returns in the lower percentiles and reflect a rather softly linear increase. Thus, our loss-averse LSTMs are most superior in predicting large negative and large positive returns, whereas the differences in signal decile 10 are relatively small compared to those in decile 1. Besides, the volatility curves are U-shaped, consistent with Jiang et al. (2022). Edge portfolios are more risky compared to portfolios with very small returns. The skewness towards the upper deciles indicates that the portfolios with large positive returns are riskier than the portfolios with large negative returns. Accordingly, Appendix 2.C illustrates that Sharpe Ratios (Sharpe, 1994) are similar across strategies, with a marginal lead of the loss-averse LSTMs in the edge portfolios. Decile 10 produces a Sharpe Ratio of up to 0.84, which is slightly less than in Jiang et al. (2022).

a 0.1 increase in Sharpe Ratio.

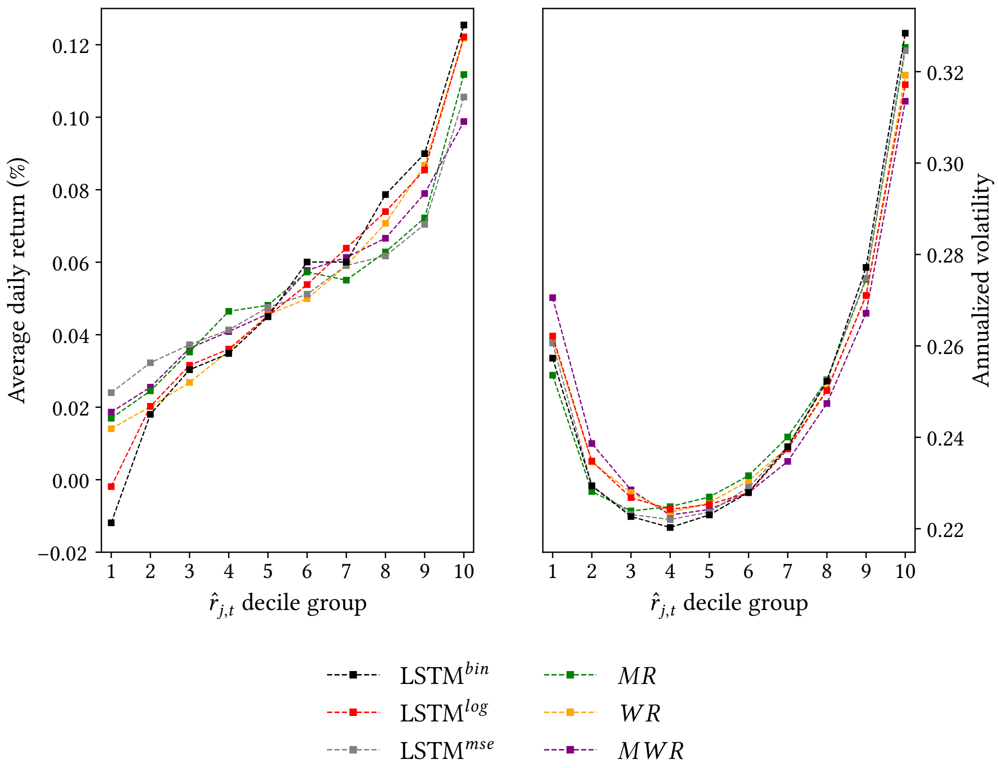

Figure 2.6: Average realized daily return (left) and annualized standard deviation (right) as functions of signal ($\hat{r}_{j,t}$) deciles. Decile 1 (10) comprises the lowest (highest) 10% of return predictions. The illustration style follows Jiang et al. (2022). Note that for decile 10 in the left-hand chart, LSTM^{log} covers WR.

2.6.4 Long-short spread returns

Long-short spread returns are a widely used metric to measure the financial utility of return forecasts (Fischer and Krauss, 2018; Jiang et al., 2022). We calculate long-short spread returns as the difference between the equal-weighted realized returns of the highest 5% return predictions and the realized returns of the lowest 5% predictions. Table 2.4 shows the average long-short spread returns before transaction costs and their risk characteristics. In addition, we determine the statistical difference between the long-short spread returns and the returns of a Buy & Hold strategy, which represents a long position in all shares in the dataset.

Table 2.4 illustrates that the loss-averse LSTMs provide investors with statistically significant long-short spread returns and superior Sharpe Ratios. For example, the binary LSTM model generates an annualized return of 50% before transaction costs and a Sharpe Ratio of 1.6, beating all reversal strategies, whereas the MSE model ranks in roughly the same performance category as the week reversal. Overall, week reversal returns appear to be most competitive with LSTM returns, consistent with Jiang et al. (2022). The risk metrics suggest that the distributions of realized returns of the loss-averse LSTM may be advantageous compared to the MSE model and some reversal models. For example, the VaR1% of the binary model is 7% smaller than that of the MSE model. In terms of maximum drawdown, investors do not benefit from the loss-averse LSTM models. The standard deviation is similar across all models, resulting in higher Sharpe Ratios for the LSTMs given higher non-adjusted returns.

Model	**LSTM^{bin}**	**LSTM^{log}**	**LSTM^{mse}**	***MR***	***WR***	***MWR***
Av. daily return (%)	0.17***	0.15***	0.14**	0.10	0.13*	0.10
Annualized return	0.50	0.43	0.42	0.28	0.41	0.27
Sharpe Ratio	1.61	1.42	1.33	0.90	1.31	0.93
Standard deviation	0.24	0.23	0.24	0.25	0.24	0.22
Max. drawdown	−0.50	−0.57	−0.46	−0.53	−0.35	−0.55
VaR1% (%)	−3.66	−3.61	−3.94	−4.00	−3.89	−3.57
CVaR1% (%)	−4.92	−5.20	−5.42	−5.61	−5.35	−5.10

Table 2.4: Average realized long-short spread returns between the highest 5% and lowest 5% return predictions and their risk characteristics. Daily returns accompanied by ***, **, * are significantly different from a Buy & Hold strategy at the 1%, 5%, 10% level (t-test). Returns disregard transaction cost.

Next, we explore long-short spread returns along the time axis. Figure 2.7 plots the average daily long-short spread returns for each year of the test period. With few exceptions, the loss-averse LSTMs (black and red markers) consistently achieve higher average daily returns than the *MSE* model (gray) and some reversal models. Reversals and LSTMs follow similar patterns and tend to perform well (poorly) relative to the market represented by the Buy & Hold strategy when others also perform well (poorly). It is also noticeable that a large

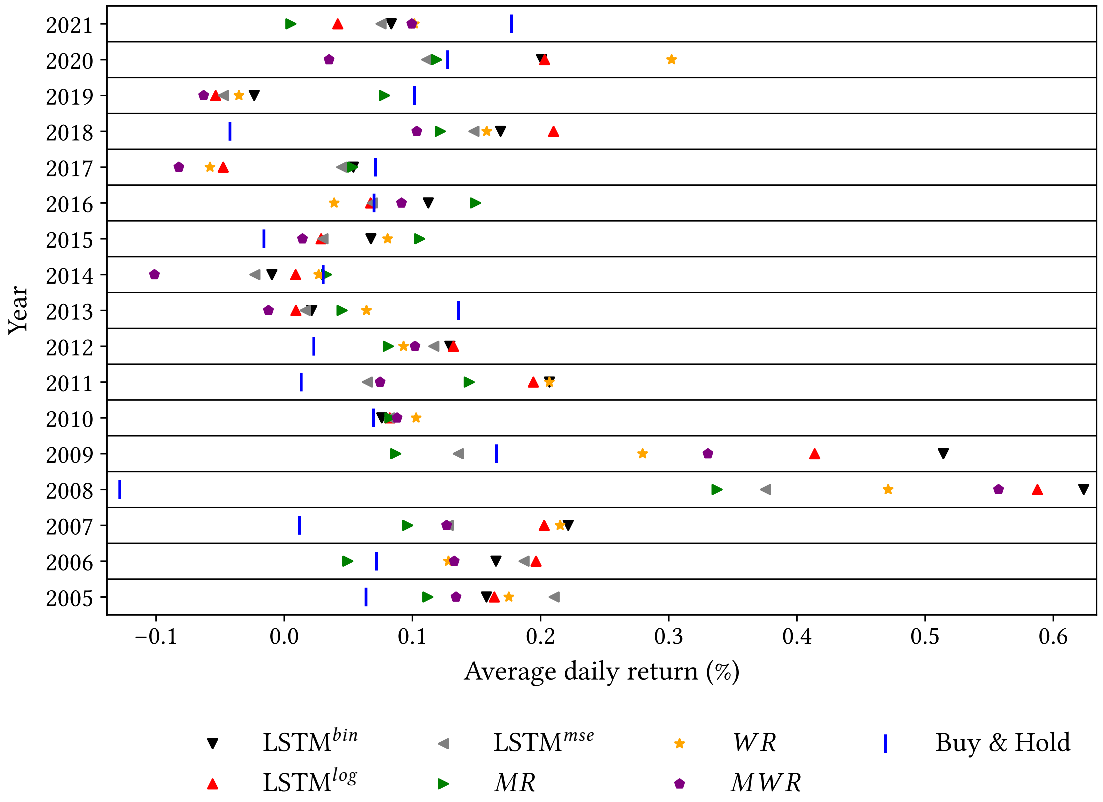

Figure 2.7: Average daily realized long-short spread returns between the highest 5% and lowest 5% by year. Returns are before transaction cost.

portion of the returns generated occurs in years associated with high market volatility, e.g., 2008 and 2009. One explanation could be that LSTMs follow reversal patterns (discussed in Section 2.7.1) that are likely to be most pronounced during volatile periods with large over- and underreactions by investors. If compared with the Buy & Hold strategy, we find that LSTM and reversal strategies perform best (worst) during falling (rising) markets.

2.6.5 Investment case study

Our practice test discusses how beneficial our models are for private investors and whether the loss function actually delivers on its promise to protect investors from unexpected losses. Fama and French (2010) argue that active management pays off only when the portfolio *constantly* outperforms the market *after* costs. In our investment study, we assume a private investor who uses an online broker to trade shares every day at market close and pays \$0.50 per trade and \$0.004 per share traded. With most popular online brokers, private investors can go both long and short in a matter of seconds at the same cost. We assume that the investor buys the 15 shares with the highest return predictions and shorts the 15 shares with the lowest return predictions, with the shares being arbitrarily divisible. Similar to Section 2.6.4, we compare our models to a Buy & Hold portfolio that buys shares when they first trade and sells them on the last trading day.

Table 2.5 shows the average annualized returns and Sharpe Ratios after transaction costs for an initial investment of \$20,000. The investor allocates the money either equally or by market capitalization (in parentheses) across the selected shares. The last column gives the average number of trades per day, where replacing one portfolio constituent triggers two trades, i.e., the maximum number of trades per day is 60. Compared to the benchmark models, the loss-averse LSTMs contribute positively to portfolio performance and generate beneficial Sharpe Ratios and returns. For example, the logistic LSTM model delivers an equally weighted Sharpe Ratio of 1.25. While the binary LSTM incurs an average of 39.2 trades each day and a cost of $\$0.50 \cdot 39.2 = \$19.6$ plus a smaller fee per share traded, the logistic LSTM model benefits from less trading activity and generates a higher annual return. Similarly, the MSE model suffers from frequent trading and performs worse than the other LSTM models. In contrast, the monthly reversal strategy, for example, incurs on average \$7.4 in transaction costs per day. When allocating funds by market capitalization, portfolio returns decrease, which leads us to conclude that the profitability of our trading strategies largely depends on small stocks.

Model	**Annualized return**	**Sharpe Ratio**	**Trades per day**
LSTM^{bin}	0.62 (0.59)	1.23 (0.88)	39.2
LSTM^{log}	0.64 (0.53)	1.25 (0.80)	32.7
LSTM^{mse}	0.57 (0.50)	1.06 (0.78)	39.8
MR	0.60 (0.38)	1.15 (0.58)	14.8
WR	0.59 (0.52)	1.16 (0.79)	26.2
MWR	0.25 (0.09)	0.56 (0.13)	40.5
Buy & Hold	0.13 (0.13)	0.41 (0.49)	0.3

Table 2.5: Equal-weighted (value-weighted) absolute returns after transaction costs on an initial investment of \$20,000. We model an investor who buys the 15 shares with the highest and shorts the 15 shares with the lowest return predictions. The right column reports the average number of trades per day.

Hence, as discussed by Avramov et al. (2023), our LSTM and reversal signals indeed seem to favor small caps over large caps, which could negatively impact the profit chances of our models in practice if investors are restricted in trading small caps.

Next, we focus on the protection from unexpected portfolio losses. To this end, we calculate the predicted portfolio return from the sum of the predictions for all 30 selected shares. So, for example, if we predict a -5% return for the 15 shares we short and a 5% return for the 15 shares we buy, the equally weighted portfolio return prediction is 10%. The left panel in Figure 2.8 shows the fraction of days on which the investor faces unexpected portfolio gains (black) out of all trading days, with the dashed gray line marking the half of all 3,850 trading days of the test sample. While both loss-averse LSTMs predict too large portfolio returns (red) only on around 1,800 days, *MSE* investors experience unexpected portfolio losses on two-thirds of all trading days. Thus, our loss functions reduce the number of trading days with unexpected portfolio losses by about 30%, fulfilling our idea of protecting investors from situations where the realized return is lower than the predicted return.

The right panel in Figure 2.8 illustrates the average loss or profit for positive and negative prediction errors, respectively. The loss-averse models earn about 1.8% return on top of the predicted return if the prediction is too low, and loose about 1.6% if the prediction is too high. For *MSE* investors, on the other hand, average unexpected losses exceed average unexpected gains ($2.11\% > 1.28\%$). Relating these results to the left-hand chart, the average unexpected gain is $1.76\% \cdot 0.54 - 1.62\% \cdot 0.46 = 0.23\%$ in the binary model, and -1.03% in the *MSE* model. In other words, the *MSE* investor earns on average 1.03% less daily return than predicted, while the binary model investor earns on average 0.23% more daily return than predicted.

This observation is consistent with the density plots of portfolio return prediction losses in Figure 2.9, whose maximums are reached for negative prediction losses in the case of the loss-averse LSTMs and for positive prediction losses in the case of the *MSE* model. While the loss-averse LSTMs produce mostly negative losses, the *MSE* model rather overpredicts portfolio returns. This, in turn, helps explain why loss-averse loss functions capture return

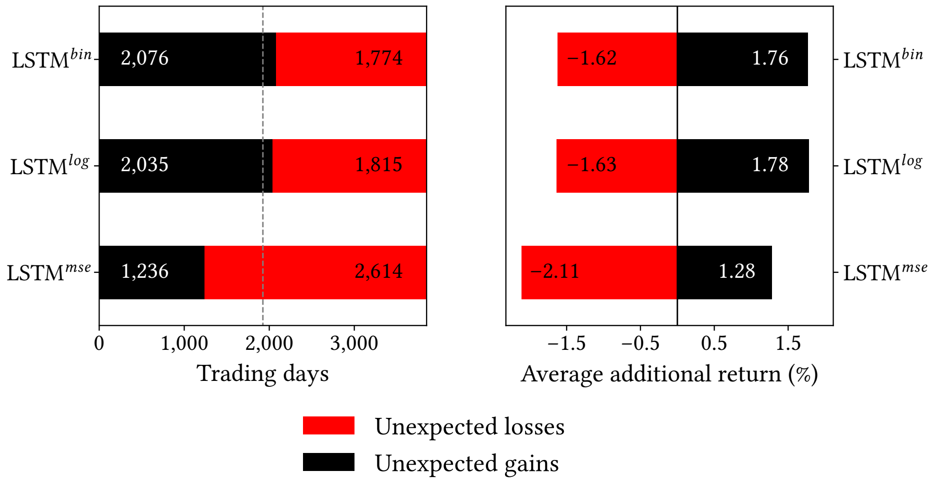

Figure 2.8: Left: Trading days with recorded unexpected portfolio gains (black) and trading days with unexpected portfolio losses (red). The dashed gray line marks the half of all trading days. Right: Average change in portfolio value for unexpected portfolio gains and losses, respectively. © 2022 IEEE

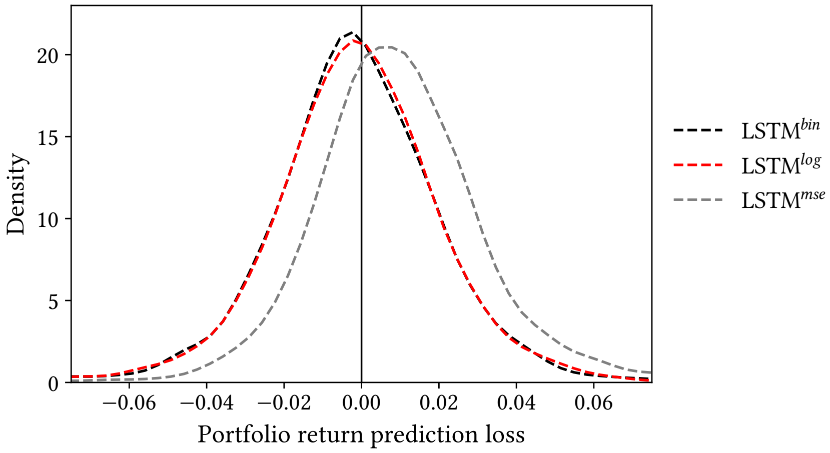

Figure 2.9: Estimated probability density function of daily portfolio return prediction losses in the range $[-0.075, 0.075]$ based on Gaussian kernels and bandwidth selection method by Scott (1992).

patterns more successfully: Standard loss functions tend to overpredict returns disproportionately and may predict returns too optimistically. Accordingly, training a network to predict returns more conservatively leads to better capture of returns, reducing the number of realized portfolio losses and increasing realized portfolio returns.

2.6.6 Long-horizon predictions

There are two reasons why we should test our model for longer prediction and holding periods. First, the fractal markets hypothesis states that returns exhibit self-repeating patterns when viewed at different time scales (Cont, 2005; Mandelbrot, 2013; Jiang et al., 2022). Second, one may argue that our case study is not relevant for every private investor due to the frequent trading. Thus, we follow Jiang et al. (2022) and test our models at weekly and monthly prediction scales and assume a private investor who uses the predictions to rebalance the portfolio once per week or once per month, respectively. To this end, we consider closing prices on the last day of a period (week or month) and calculate the returns achieved during that period following equation (2.7). Accordingly, the LSTM processes only weekly (monthly) returns when predicting weekly (monthly) returns. We convert the sequence length of 60 trading days to 12 weeks in the weekly model and to 3 months in the monthly model. Apart from that, the network configuration is identical to the daily model defined in Section 2.2.2.

Table 2.6 reports the average annualized returns and Sharpe Ratios after transaction cost determined following the investment case study from Section 2.6.5. As prediction periods grow, the advantage of LSTMs over reversal strategies increases, e.g., the returns on the weekly revised loss-averse LSTM portfolio are about three times larger than the returns on the reversal portfolios. One reason is that, compared to the daily model, less trading and thus lower transaction costs improve trading performance, making weekly-revised LSTM portfolios superior to their daily-revised counterparts in Table 2.5. For instance, even if the strategy replaces all portfolio constituents once a week, the trade frequency is still much lower than that from Table 2.5. On average, investors using loss-averse LSTMs would outperform reversal strategies and standard LSTMs on either daily, weekly or monthly prediction scales, whereas the lead of loss-averse LSTMs over standard LSTMs decreases for longer time scales.

Model	**Weekly rebalancing**		**Monthly rebalancing**	
	Annualized return	Sharpe Ratio	Annualized return	Sharpe Ratio
LSTM^{bin}	0.71	1.40	0.41	0.88
LSTM^{log}	0.73	1.54	0.66	1.20
LSTM^{mse}	0.65	1.24	0.65	1.16
WR (MR)	0.23	0.53	0.29	0.14

Table 2.6: Long-short spread returns on long-horizon portfolios. Returns are equal-weighted absolute returns after transaction costs on an initial investment of $20,000 calculated according to Section 2.6.5. Due to the lower data frequency, our benchmark models are the week (month) reversal for the weekly (monthly) rebalancing trading strategy. © 2022 IEEE

2.6.7 Robustness to market sentiment

Patterns in financial time series captured by LSTMs may reflect market sentiment. One question that arises is whether our models perform equally well in different market environments or exhibit systematic differences. For example, it would be possible that a particular strategy systematically performs better in a specific market phase because it can better capture the characteristics of that market phase. As a measure of market sentiment, we develop 32 sentiment classes representing all possible combinations of signs of the market return during the last five trading days before a prediction. The first (last) sentiment class represents five consecutive positive (negative) market returns, the second class represents four consecutive positive followed by a negative market return, and so on. Higher sentiment classes are associated with more negative market returns in the last five trading days. In more detail, Appendix 2.B illustrates the market return sequences of each market sentiment class.

Figure 2.10 plots the average daily long-short spread returns calculated following Section 2.6.4 as a function of the market sentiment class. From left to right, market sentiment changes from very positive (first class) to very negative (last class). The map illustrates that each prediction model has its own strengths and weaknesses in different situations. On average, how-

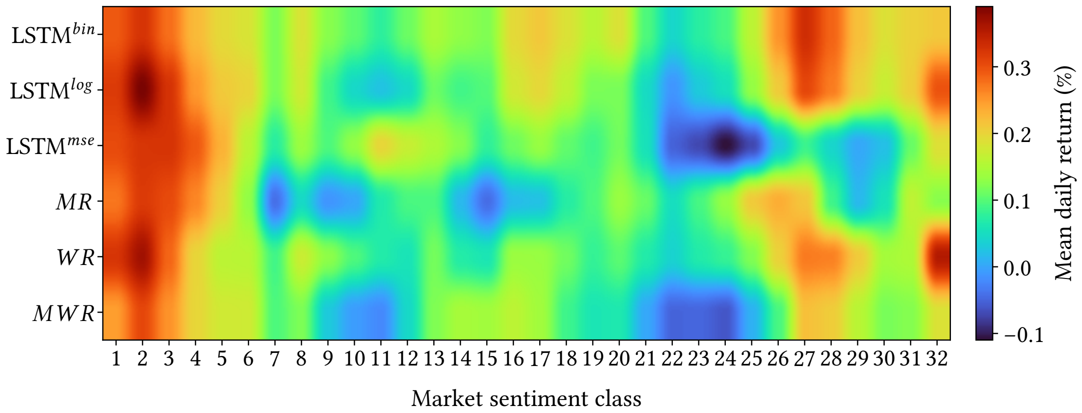

Figure 2.10: Average realized long-short spread returns between the highest 5% and lowest 5% return predictions before transaction cost as functions of the market sentiment. Market sentiment class 1 (32) represents five consecutive positive (negative) market returns.

ever, the models perform slightly better in the right and left parts of the map. This area groups combinations with four or more positive and negative returns, respectively. We conclude that, consistent with Section 4.6.1, the prediction models perform well when market sentiment is either clearly positive or clearly negative. Besides, the returns on the loss-averse LSTM portfolios outperform the benchmark models in the majority of sentiment classes, indicating the robustness of the loss functions to market sentiment.

One way to balance the strengths and weaknesses of multiple prediction models conditional on market sentiment is to combine the models into an ensemble that selects predictions based on the current market sentiment. The literature presents various approaches of ensemble learning in return prediction (Krauss et al., 2017; Sun et al., 2018; Borovkova and Tsiamas, 2019; Gu et al., 2020; Chen et al., 2022). Similar to Borovkova and Tsiamas (2019), whose ensemble weights predictions based on recent model performance, we consider an ensemble that picks prediction models based on their historical performance in the current market sentiment. For example, if the market return has been positive for five consecutive trading days, the ensemble picks the prediction of the model with the best past performance in the first market sentiment class. Appendix 2.D outlines the learning content of the ensemble, emphasizing that it considers prediction models in roughly equal proportions. Using ensemble learning, we achieve 52.40% accuracy and a long-short spread Sharpe Ratio of 2.03, which is 0.1 percentage point and 0.4 higher than the best stand-alone model in Table 2.3 and Table 2.4, respectively.

2.7 Review of the loss function

2.7.1 Reversal imitation or unique learning?

One might wonder why loss-averse LSTMs outperform all other approaches. We try to shed light on this question by looking into how unique the signals of the loss-averse LSTMs are compared to reversal signals. To date, the literature comes to mixed conclusions about whether trend signals captured by deep learning algorithms are similar to reversal signals or not: Guijarro-Ordonez et al. (2021) and Jiang et al. (2022) claim that the signals have little in common, whereas Fischer and Krauss (2018) or Chen et al. (2022) report that LSTMs largely adapt to reversal patterns. Motivated by this contrast, we examine the relationship between LSTM and reversal signals in four steps:

First, Figure 2.11 illustrates the average preceding returns over the last two days, last week, and last month for positive and negative return predictions, respectively. The top (bottom) panel focuses on negative (positive) return predictions. For example, if the binary LSTM model delivers a negative return prediction for a particular share, the share has averaged a return of 3% in the past week and a return of slightly less than 5% in the past month, while if its return prediction is positive, the past week and month returns of the share are negative on average. Similar to the reversal strategies, the preceding returns for positive (negative) return predictions of LSTMs are negative (positive) on average. The large distance between the preceding returns of positive return predictions and the corresponding preceding returns of negative return predictions indicates the imitation of reversal signals. This distance is smaller in the *MSE* model than in the loss-averse models, suggesting that reversal imitation is more pronounced in the loss-averse models.

Second, we analyze the correlation between the signal deciles discussed in Figure 2.6. To this end, we determine the cross-sectional Pearson correlation between all signal deciles per strategy. Figure 2.12 illustrates the correlation matrix and indicates that the correlation be-

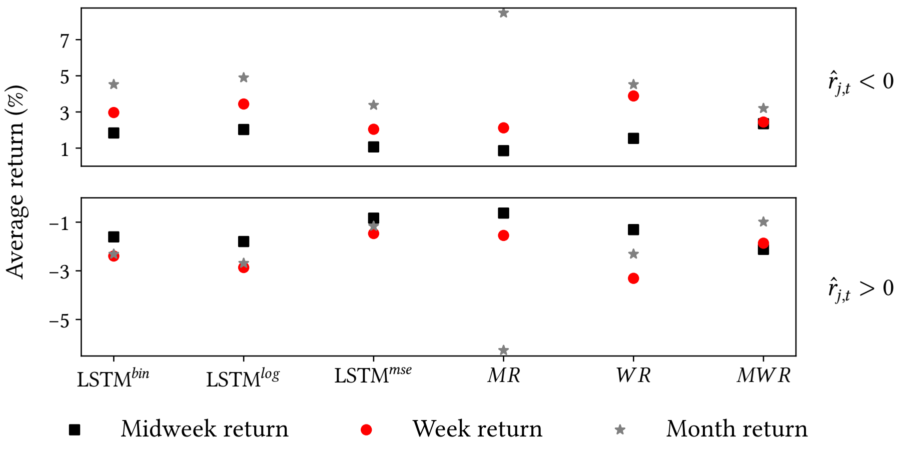

Figure 2.11: Average preceding returns for negative and positive return predictions $\hat{r}_{j,t}$. The upper (bottom) panel shows the average two-day, one-week and one-month return generated prior to a negative (positive) return prediction. All numbers denote in percentage. © 2022 IEEE

tween loss-averse LSTMs and week and midweek reversals is higher than between all other strategies except between the two loss-averse LSTMs themselves. Thus, a higher return prediction by a reversal strategy is likely accompanied by a higher return prediction by the loss-averse models. Since WR and MWR do not have quite as high a correlation with each other, we infer that loss-averse LSTMs extract different signals from week reversal patterns and midweek reversal patterns, respectively. The MSE model, on the other hand, imitates reversals less strongly and extracts more non-reversal signals, consistent with Figure 2.11.

Third, we follow Jiang et al. (2022) and Fischer and Krauss (2018) and examine the relationships between portfolios using linear regression on portfolio returns. To include all predictions, we calculate portfolio returns by the equally weighted spread between the realized returns of all buy signals ($\hat{r} > 0$) and the realized returns of all sell signals ($\hat{r} < 0$). Table 2.7 shows the estimated parameters and the R^2 while controlling for the market risk premium

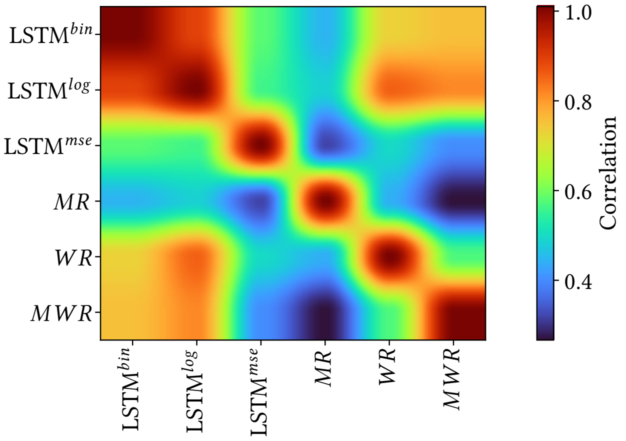

Figure 2.12: Cross-sectional Pearson correlation of signal deciles.

Model	**LSTM**^{bin}	**LSTM**^{log}	**LSTM**^{mse}
Alpha	0.02^{***}	0.02^{***}	0.01^{*}
MRP	0.08^{***}	0.04^{**}	0.09^{***}
MR	0.09^{***}	0.09^{***}	0.05^{***}
WR	0.25^{***}	0.26^{***}	0.25^{***}
MWR	0.32^{***}	0.33^{***}	0.17^{***}
R^2	0.55	0.63	0.28

Table 2.7: Linear regression of LSTM portfolio returns on the market risk premium (MRP) and reversal portfolio returns. Portfolio returns are the daily equally weighted spreads between the realized returns on buy signals and the realized returns on sell signals. Numbers followed by ***, **, and * indicate p-values below 1%, 5%, and 10%. © 2022 IEEE

MRP, which we calculate by subtracting the risk-free rate[7] from the market portfolio defined in Section 2.2.2. The coefficients illustrate that the relationships of LSTM portfolio returns with reversal portfolio returns are stronger than with the returns on the market portfolio. Moreover, LSTM portfolios earn returns on reversal patterns mainly over time horizons of one week and shorter. While the R^2 is larger than in Fischer and Krauss (2018) and Jiang et al. (2022), the association between LSTM and reversal portfolios is similar to Fischer and Krauss (2018). Thus, market and reversal returns explain slightly more than half of the variation in LSTM portfolio returns, but at the same time this means that they also fail to explain less than half of the variation in returns.

Finally, to determine the proportion of unique learning content, we follow Jiang et al. (2022) and compare the R^2 of linear approximations to realized returns with and without controlling for return predictions. To do this, we run four regressions: First, we regress the realized returns on the set of predictors we use to train the LSTM, i.e., 60-days sequences of market and stock returns. In the other three models, we regress realized returns on both the set of predictors and the predictions of one prediction model each. Table 2.8 illustrates the explanatory power of the four approaches: While MSE predictions do not increase the R^2, the predictions generated by the loss-averse LSTMs improve the determination by up to 14%. The higher explanatory content of the realized returns suggests that the loss functions provide unique information, undiscovered by the standard models, that leads to better knowledge.

[7] We model the nominal U.S. treasury yield curve following the Nelson–Siegel–Svensson method (Nelson and Siegel, 1987; Svensson, 1994; Gürkaynak et al., 2007) with considering an investment horizon of one day, consistent with the prediction scale and portfolio revision frequency. We use the beta and tau parameters provided by *Nasdaq Data Link*, see https://data.nasdaq.com/data/FED/PARAMS-us-treasury-beta-and-tau-parameters.

Predictor	R^2 (%)
Market data	1.97
Market data & LSTM^{mse}	1.99
Market data & LSTM^{bin}	2.24
Market data & LSTM^{log}	2.23

Table 2.8: Linear regression of realized returns on the set of predictors used to train the LSTM and defined in Section 2.2.2 (`Market data') with and without controlling for LSTM return predictions. LSTM predictions base on parameters trained during the training period. © 2022 IEEE

Model	Accuracy (%)	Correlation (%)	Annualized return	Sharpe Ratio
LIN^{log}	51.85	3.71	0.10	0.23
LIN^{mse}	51.92	3.84	0.12	0.34

Table 2.9: Prediction quality and long-short spread returns of linear models. Annualized returns are absolute long-short spread returns after transaction costs on an initial investment of $20,000 calculated according to Section 2.6.5, with long in the 15 highest and short in the lowest 15 predictions. © 2022 IEEE

2.7.2 Loss functions in linear approximations to returns

LSTMs are often said to be superior compared to other time series models (Siami-Namini et al., 2018) because of their capability of capturing nonlinearity in the data. However, the question arises whether linear approximations to returns would also improve with our loss-averse loss function. To answer this question, we perform a linear approximation to realized returns minimizing the $WMSE$ on the training and validation period and then use the estimated coefficients to predict returns for the test period.

Table 2.9 shows the prediction quality and equally weighted long-short spread returns after transaction costs on a 15-15 portfolio (see Section 2.6.5). Here, we omit the binary loss function since the coefficients are identical with the MSE model, i.e., all losses are 'bad' losses and receive the same weights. Although the prediction quality competes with LSTMs, the portfolios exhibit a high trading frequency and thus underperform the market with an annualized return of less than 0.13, suggesting poor nominal return predictions at the return distribution tails. Besides, the MSE loss function provides better results in linear approximations than the $WMSE$ loss function. We infer that loss-averse loss functions are beneficial in non-linear optimization, but do not add value in linear optimization. Nevertheless, Appendix 2.D shows that under some market conditions, the linear models deliver trading results superior to the non-linear models, thereby adding value to an ensemble.

2.8 Conclusion

In this paper, we present a novel approach that connects return prediction with prospect theory and loss aversion of investors. To this end, we introduce a loss-averse loss function for return prediction that teaches LSTMs to underpredict rather than overpredict returns. We show that annualized returns on regularly revised portfolios increase by an average of five percentage points over the standard LSTM and reversal strategies.

One limitation of our approach, however, is that both the computing power required for the training and the technical infrastructure for automated trading with connected live market data are not easy to set up for private investors. Yet, our model does not require costly retraining: Once the model is trained and set up, it runs with very little effort.

Overall, our proposed loss function is beneficial in nonlinear optimization and extracts unique signals in a return-predicting LSTM model. Moreover, we find robust performance throughout a wide range of market sentiments.

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 https://www.tensorflow.org/.

Aggarwal, C. C. et al. (2018), 'Neural networks and deep learning', *Springer* **10**(978), 3. doi:10.1007/978-3-319-94463-0

Avramov, D., Cheng, S. and Metzker, L. (2023), 'Machine Learning vs. Economic Restrictions: Evidence from Stock Return Predictability', *Management Science* **69**(5), 2587–2619. doi:10.1287/mnsc.2022.4449

Avramov, D., Chordia, T. and Goyal, A. (2006), 'Liquidity and Autocorrelations in Individual Stock Returns', *Journal of Finance* **61**(5), 2365–2394. doi:10.1111/j.1540-6261.2006.01060.x

Bajgrowicz, P. and Scaillet, O. (2012), 'Technical trading revisited: False discoveries, persistence tests, and transaction costs', *Journal of Financial Economics* **106**(3), 473–491. doi:10.1016/j.jfineco.2012.06.001

Bao, W., Yue, J. and Rao, Y. (2017), 'A deep learning framework for financial time series using stacked autoencoders and long-short term memory', *PLoS ONE* **12**(7). doi:10.1371/journal.pone.0180944

Barberis, N., Huang, M. and Santos, T. (2001), 'Prospect Theory and Asset Prices*', *The Quarterly Journal of Economics* **116**(1), 1–53. doi:10.1162/003355301556310

Barberis, N., Mukherjee, A. and Wang, B. (2016), 'Prospect Theory and Stock Returns: An Empirical Test', *Review of Financial Studies* **29**(11), 3068–3107. doi:10.1093/rfs/hhw049

Bengio, Y., Simard, P. and Frasconi, P. (1994), 'Learning long-term dependencies with gradient descent is difficult', *IEEE Transactions on Neural Networks* **5**(2), 157–166. doi:10.1109/72.279181

Berkelaar, A. B., Kouwenberg, R. and Post, T. (2004), 'Optimal Portfolio Choice under Loss Aversion', *Review of Economics and Statistics* **86**(4), 973–987. doi:10.1162/0034653043125167

Borovkova, S. and Tsiamas, I. (2019), 'An ensemble of LSTM neural networks for high-frequency stock market classification', *Journal of Forecasting* **38**(6), 600–619. doi:10.1002/for.2585

Boudoukh, J., Richardson, M. P. and Whitelaw, R. F. (1994), 'A Tale of Three Schools: Insights on Autocorrelations of Short Horizon Stock Returns', *Review of Financial Studies* **7**, 539–573. doi:10.1093/rfs/7.3.539

Brock, W., Lakonishok, J. and LeBaron, B. (1992), 'Simple technical trading rules and the stochastic properties of stock returns', *Journal of Finance* **47**(5), 1731–1764. doi:10.2307/2328994

Brownlee, J. (2017), *Long short-term memory networks with python: Develop sequence prediction models with deep learning*, Machine Learning Mastery. Available from https://machinelearningmastery.com/lstms-with-python/.

Chapman, J., Snowberg, E., Wang, S. W. and Camerer, C. (2022), Looming large or seeming small? Attitudes towards losses in a representative sample, NBER Working Papers 30243, National Bureau of Economic Research. doi:10.3386/w30243

Chen, K., Zhou, Y. and Dai, F. (2015), A LSTM-based method for stock returns prediction: A case study of China stock market, *in* '2015 IEEE International Conference on Big Data (Big Data)', pp. 2823–2824. doi:10.1109/BigData.2015.7364089

Chen, L., Pelger, M. and Zhu, J. (2022), 'Deep Learning in Asset Pricing', *Management Science* **forthcoming**. doi:10.1287/mnsc.2023.4695

Chollet, F. et al. (2015), 'Keras', www.keras.io. Package available from https://github.com/fchollet/keras.

Chordia, T., Roll, R. and Subrahmanyam, A. (2002), 'Order imbalance, liquidity, and market returns', *Journal of Financial Economics* **65**(1), 111–130. doi:10.1016/S0304-405X(02)00136-8

Cont, R. (2005), Long range dependence in financial markets, *in* J. Lévy-Véhel and E. Lutton, eds, 'Fractals in Engineering', Springer London, London, pp. 159–179. doi:10.1007/1-84628-048-6_11

Diebold, F. X. and Mariano, R. S. (1995), 'Comparing Predictive Accuracy', *Journal of Business & Economic Statistics* **13**(3), 253–263. doi:10.1080/07350015.1995.10524599

Fabbri, M. and Moro, G. (2018), Dow Jones Trading with Deep Learning: The Unreasonable Effectiveness of Recurrent Neural Networks, *in* 'Proceedings of the 7th International Conference on Data Science, Technology and Applications', DATA 2018, SciTePress, pp. 142–153. doi:10.5220/0006922101420153

Fama, E. F. (1965), 'The Behavior of Stock-Market Prices', *Journal of Business* **38**(1), 34–105. doi:10.1086/294743

Fama, E. F. (1970), 'Efficient Capital Markets: A Review of Theory and Empirical Work', *Journal of Finance* **25**(2), 383–417. doi:10.2307/2325486

Fama, E. F. and French, K. R. (2010), 'Luck versus skill in the cross-section of mutual fund returns', *Journal of Finance* **65**(5), 1915–1947. doi:10.1111/j.1540-6261.2010.01598.x

Fischer, T. and Krauss, C. (2018), 'Deep learning with long short-term memory networks for financial market predictions', *European Journal of Operational Research* **270**(2), 654–669. doi:10.1016/j.ejor.2017.11.054

Flori, A. and Regoli, D. (2021), 'Revealing Pairs-trading opportunities with long short-term memory networks', *European Journal of Operational Research* **295**(2), 772–791. doi:10.1016/j.ejor.2021.03.009

Graves, A. (2013), 'Generating sequences with recurrent neural networks'. arXiv preprint, unpublished. doi:10.48550/arxiv.1308.0850

Gu, S., Kelly, B. T. and Xiu, D. (2020), 'Empirical Asset Pricing via Machine Learning', *Review of Financial Studies* **33**(5), 2223–2273. doi:10.1093/rfs/hhaa009

Guijarro-Ordonez, J., Pelger, M. and Zanotti, G. (2021), 'Deep Learning Statistical Arbitrage'. arXiv preprint, unpublished. doi:10.48550/arxiv.2106.04028

Gürkaynak, R. S., Sack, B. and Wright, J. H. (2007), 'The U.S. Treasury yield curve: 1961 to the present', *Journal of Monetary Economics* **54**(8), 2291–2304. doi:10.1016/j.jmoneco.2007.06.029

Hochreiter, S. (1998), 'The vanishing gradient problem during learning recurrent neural nets and problem solutions', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **6**(02), 107–116. doi:10.1142/S0218488598000094

Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J. (2001), Gradient flow in recurrent nets: The difficulty of learning longterm dependencies, *in* 'A Field Guide to Dynamical Recurrent Networks', IEEE Press, pp. 237–243. doi:10.1109/9780470544037

Hochreiter, S. and Schmidhuber, J. (1997), 'Long Short-term Memory', *Neural computation* **9**, 1735–80. doi:10.1162/neco.1997.9.8.1735

Jegadeesh, N. (1990), 'Evidence of Predictable Behavior of Security Returns', *Journal of Finance* **45**(3), 881–898. doi:10.1111/j.1540-6261.1990.tb05110.x

Jegadeesh, N. and Titman, S. (1993), 'Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency', *Journal of Finance* **48**(1), 65–91. doi:10.2307/2328882

Jegadeesh, N. and Titman, S. (1995), 'Overreaction, Delayed Reaction, and Contrarian Profits', *Review of Financial Studies* **8**(4), 973–993. doi:10.1093/rfs/8.4.973

Jiang, J., Kelly, B. T. and Xiu, D. (2022), '(Re-) Imag (in) ing Price Trends', *Journal of Finance* **forthcoming**. Available at SSRN 3756587. doi:10.2139/ssrn.3756587

Kahneman, D. and Tversky, A. (1979), 'Prospect Theory: An Analysis of Decision under Risk', *Econometrica* **47**(2), 263–291. doi:10.2307/1914185

Karpathy, A. (2015), 'The Unreasonable Effectiveness of Recurrent Neural Networks'. Available from https://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Krauss, C., Do, X. A. and Huck, N. (2017), 'Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500', *European Journal of Operational Research* **259**(2), 689–702. doi:10.1016/j.ejor.2016.10.031

LeCun, Y. A., Bottou, L., Orr, G. B. and Müller, K.-R. (2012), Efficient BackProp, *in* 'Neural networks: Tricks of the trade', Springer, pp. 9–48. doi:10.1007/978-3-642-35289-8_3

Lehmann, B. N. (1990), 'Fads, Martingales, and Market Efficiency', *The Quarterly Journal of Economics* **105**(1), 1–28. doi:10.2307/2937816

Li, Y. and Yang, L. (2013), 'Prospect theory, the disposition effect, and asset prices', *Journal of Financial Economics* **107**(3), 715–739. doi:10.1016/j.jfineco.2012.11.002

Lo, A. W. and MacKinlay, A. C. (1990), 'When Are Contrarian Profits Due to Stock Market Overreaction?', *Review of Financial Studies* **3**(2), 175–205. doi:10.1093/rfs/3.2.175

Lo, A. W., Mamaysky, H. and Wang, J. (2000), 'Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation', *Journal of Finance* **55**(4), 1705–1765. doi:10.1111/0022-1082.00265

Mandelbrot, B. B. (2013), *Fractals and Scaling in Finance: Discontinuity, Concentration, Risk. Selecta Volume E*, Springer Science & Business Media, New York. doi:10.1007/978-1-4757-2763-0

Nelson, C. R. and Siegel, A. F. (1987), 'Parsimonious modeling of yield curves', *Journal of business* pp. 473–489. doi:10.1086/296409

Nelson, D. M. Q., Pereira, A. C. M. and de Oliveira, R. A. (2017), Stock market's price movement prediction with LSTM neural networks, *in* '2017 International Joint Conference on Neural Networks (IJCNN)', IEEE, pp. 1419–1426. doi:10.1109/IJCNN.2017.7966019

Olah, C. (2015), 'Understanding LSTM networks'. Available from https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Qiu, J., Wang, B. and Zhou, C. (2020), 'Forecasting stock prices with long-short term memory neural network based on attention mechanism', *PLOS ONE* **15**(1), 1–15. doi:10.1371/journal.pone.0227222

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986*a*), Learning Internal Representations by Error Propagation, *in* J. McClelland and D. Rumelhart, eds, 'Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations', MIT Press, Cambridge, MA, USA, pp. 318–362. Available from https://apps.dtic.mil/dtic/tr/fulltext/u2/a164453.pdf.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986*b*), 'Learning representations by back-propagating errors', *Nature* **323**(6088), 533–536. doi:10.1038/323533a0

Sang, C. and Di Pierro, M. (2019), 'Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network', *Journal of Finance and Data Science* **5**(1), 1–11. doi:10.1016/j.jfds.2018.10.003

Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, Wiley Series in Probability and Statistics, Wiley, New York. doi:10.1002/9780470316849

Sharpe, W. F. (1994), 'The Sharpe Ratio', *Journal of Portfolio Management* **21**(1), 49–58. doi:10.3905/jpm.1994.409501

Sherstinsky, A. (2020), 'Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network', *Physica D: Nonlinear Phenomena* **404**, 132306. doi:10.1016/j.physd.2019.132306

Siami-Namini, S., Tavakoli, N. and Siami Namin, A. (2018), A Comparison of ARIMA and LSTM in Forecasting Time Series, *in* '2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)', pp. 1394–1401. doi:10.1109/ICMLA.2018.00227

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting', *Journal of Machine Learning Research* **15**(56), 1929–1958. Available from https://dl.acm.org/doi/10.5555/2627435.2670313.

Subrahmanyam, A. (2005), 'Distinguishing Between Rationales for Short-Horizon Predictability of Stock Returns', *Financial Review* **40**(1), 11–35. doi:10.1111/j.0732-8516.2005.00091.x

Sullivan, R., Timmermann, A. and White, H. (1999), 'Data-Snooping, Technical Trading Rule Performance, and the Bootstrap', *Journal of Finance* **54**(5), 1647–1691. doi:10.1111/0022-1082.00163

Sun, S., Wei, Y. and Wang, S. (2018), AdaBoost-LSTM Ensemble Learning for Financial Time Series Forecasting, *in* Y. Shi, H. Fu, Y. Tian, V. V. Krzhizhanovskaya, M. H. Lees, J. Dongarra and P. M. A. Sloot, eds, 'Computational Science – ICCS 2018', Springer International Publishing, Cham, pp. 590–597. doi:10.1007/978-3-319-93713-7_55

Svensson, L. (1994), Estimating and Interpreting Forward Interest Rates: Sweden 1992–1994, NBER Working Papers 4871, National Bureau of Economic Research. doi:10.3386/w4871

Troiano, L., Villa, E. M. and Loia, V. (2018), 'Replicating a Trading Strategy by Means of LSTM for Financial Industry Applications', *IEEE Transactions on Industrial Informatics* **14**(7), 3226–3234. doi:10.1109/TII.2018.2811377

Tversky, A. and Kahneman, D. (1992), 'Advances in prospect theory: Cumulative representation of uncertainty', *Journal of Risk and uncertainty* **5**(4), 297–323. doi:10.1007/BF00122574

Yang, L. (2019), 'Loss Aversion in Financial Markets', *Journal of Mechanism and Institution Design* **4**(1), 119–137. doi:10.22574/jmid.2019.11.005

Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y. and Liu, J. (2017), 'Lstm network: a deep learning approach for short-term traffic forecast', *IET Intelligent Transport Systems* **11**(2), 68–75. doi:10.1049/iet-its.2016.0208

Appendix

2.A Share of unexpected gains

Figure 2.13 reports the share of the absolute sum of 'good' prediction losses in the absolute sum of all losses. Due to relatively small 'bad' losses, the loss-averse LSTMs exhibit a large share of unexpected gains, whereas the *MSE* model exposes nine times larger unexpected losses than gains due to relatively large 'bad' prediction losses. In the binary (logistic) model, 85% (73%) of all prediction losses are associated with realized returns larger than predicted.

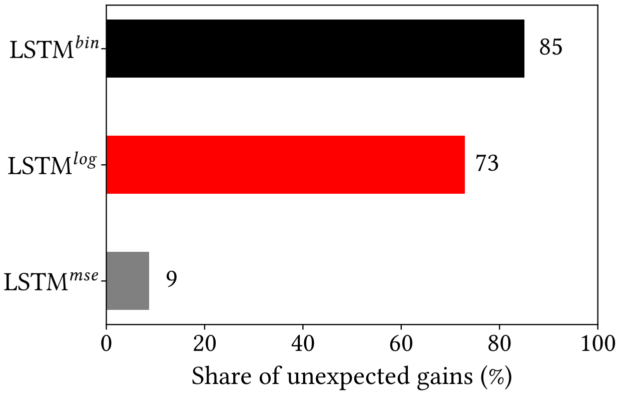

Figure 2.13: Share of unexpected gains in all unexpected movements. Unexpected gains are the area between the loss curve and the x-axis in quadrants II and IV in Figure 2.5 (right chart), unexpected losses are the corresponding area in quadrants I and III.

2.B Market sentiment classes

Figure 2.14 illustrates the market return sequences of each sentiment class used for the robustness check in Section 2.6.7. We define the market sentiment class on day t based on the market returns r_m during the past five trading days, where $t-5$ denotes the day five trading days ago, and $t-1$ denotes the last trading day. For instance, market sentiment class 1 is the most positive sentiment with five consecutive days that generated a positive market return. With increasing classes, market sentiment becomes more negative.

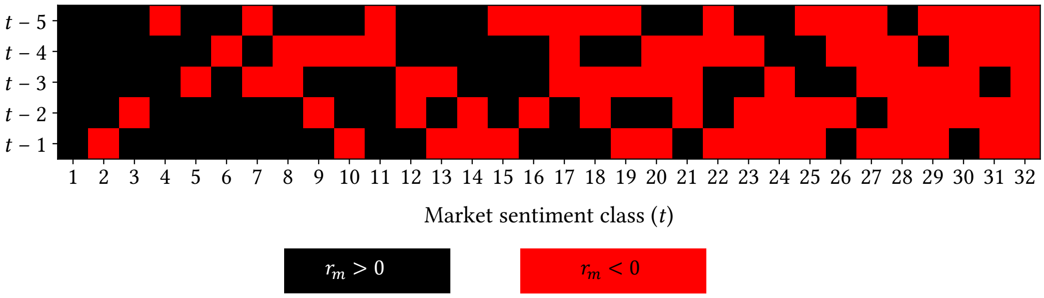

Figure 2.14: Market sentiment classes on day t based on the market returns during the past five trading days. Black (red) dots indicate negative (positive) market return.

2.C Average realized returns and Sharpe Ratios

In complement to Figure 2.6, Table 2.10 reports the average realized daily return and annualized Sharpe Ratio (Sharpe, 1994) by signal ($\hat{r}_{j,t}$) decile. Out of all benchmark models, the week reversal portfolio appears to be most competitive with LSTM in both realized returns and Sharpe Ratios, consistent with Jiang et al. (2022). Both loss-averse LSTMs and the week reversal produce a Sharpe Ratio of more than 0.8 in decile 10, whereas loss-averse LSTMs produce negative Sharpe Ratios and are thus superior to the week reversal in decile 1.

	Decile portfolio									
Model	1	2	3	4	5	6	7	8	9	10
Average realized daily return (%)										
LSTM^{bin}	−0.01	0.02	0.03	0.03	0.04	0.06	0.06	0.08	0.09	0.13
LSTM^{log}	−0.00	0.02	0.03	0.04	0.05	0.05	0.06	0.07	0.09	0.12
LSTM^{mse}	0.00	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.11
MR	0.02	0.02	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.11
WR	0.01	0.02	0.03	0.04	0.05	0.05	0.06	0.07	0.09	0.12
MWR	0.02	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.08	0.10
Annualized Sharpe Ratios										
LSTM^{bin}	−0.22	0.07	0.20	0.25	0.36	0.51	0.49	0.64	0.68	0.83
LSTM^{log}	−0.12	0.09	0.21	0.26	0.36	0.45	0.53	0.60	0.66	0.84
LSTM^{mse}	0.02	0.22	0.28	0.32	0.39	0.42	0.48	0.48	0.52	0.70
MR	0.05	0.14	0.25	0.37	0.39	0.48	0.44	0.49	0.53	0.74
WR	0.02	0.09	0.16	0.26	0.36	0.40	0.48	0.57	0.67	0.83
MWR	0.06	0.14	0.26	0.32	0.37	0.49	0.51	0.54	0.61	0.67

Table 2.10: Equal-weighted average realized returns and Sharpe Ratios by signal decile. Decile 1 (10) contains the shares with the lowest (highest) 10% return predictions.

2.D (Non-)Linear Ensemble learning

We consider an ensemble that comprises all LSTM, reversal and linear (Section 2.7.2) prediction models. Thus, in total, the ensemble comprises eight unique forecast models. To ensure the ensemble sees only training and validation data, we train the ensemble with long-short spread returns on the validation set and then test it on the test set. An ensemble prediction consists of two sequential steps: First, the ensemble defines the market sentiment based on the present sequence of the past five market returns. Second, it selects the prediction model with the best performance in that sentiment during the validation period and uses the selected model to produce a prediction. Figure 2.15 explores what the ensemble learns from the validation set. It is important to keep in mind that Figure 2.10 focuses on test data and thus is not related to the results presented in this section which may differ from Figure 2.10.

The left chart in Figure 2.15 shows the proportion of prediction models selected by the ensemble. LSTMs account for 16, reversals for eleven, and linear models for five sentiment classes, with loss-averse LSTMs accounting for about one third of sentiment classes. However, as market sentiment classes are not equally distributed within the considered period—sentiment classes occur with different frequencies—, Figure 2.15 does not tell us how often the

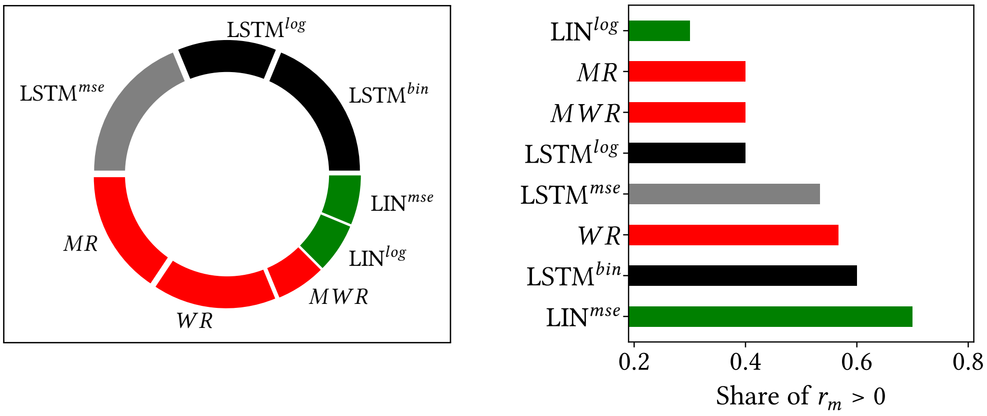

Figure 2.15: Learning content of the ensemble (validation period). The left image illustrates the share of each model in the prediction models picked by the ensemble. The right chart reports the average share of positive market returns in the market class per prediction model if picked by the ensemble. Red (green) indicates reversal (linear) strategies, whereas black (gray) indicates LSTM strategies based on loss-averse (standard) loss functions.

ensemble actually picks a particular model. Thus, if sentiment classes for which the ensemble picks the LSTM occur more frequently, the distribution among prediction models could exhibit a different pattern. The right chart in Figure 2.15 documents the average share of positive market returns in market classes per selected prediction model. For instance, a combination with two positive and three negative signs corresponds to a share of 0.4. Linear models fill the edge cases with the lowest and highest shares, i.e., linear models perform best in market classes with at most one day diverging from a dominant one-week market trend. Apart from that, both reversals and LSTMs exhibit relatively large and small shares.

3 Intelligent Inventory Management for Cryptocurrency Brokers[8]

Christopher Felder and Johannes Seemüller

Abstract

In equity trading, internalization is the predominant execution method for uninformed order flow, allowing retail brokers to realize cost savings and thereby offer price improvements to customers. In cryptocurrency trading, there are doubts as to whether informed and uninformed traders can be distinguished in the same way, leading brokers to seek cost savings through internal order matching instead. Using the historical order flow of the German cryptocurrency broker BISON, we present a prediction-based approach to internal order matching: Upon receiving a customer order, our model forecasts whether future order flow will be sufficient to neutralize the order before the settlement date. With a prediction accuracy of 85%, it enables brokers to match three-quarters of order volume internally, which is three times as much as a traditional rule-based approach, and realize meaningful cost savings, even after accounting for common minimum price improvements.

Keywords: Segmentation, Retail internalization, Cryptocurrency.

JEL classification: C45, C55, C63, D49, G17, G24.

[8] Parts of this article have been published in Felder and Seemüller (2022). Relevant paragraphs, tables, and figures are reprinted with permission. ©2022 ACM

3.1 Introduction

Only few retail stock traders are aware that their orders are typically not executed on exchanges (Comerton-Forde et al., 2018). Instead, firms execute them internally against their own book. Internalization represents the major trading modality for retail order flow in equity trading (Fox et al., 2019; Barardehi et al., 2022) and pays off because retail traders are supposed to not have superior information (Chakravarty, 2001; Linnainmaa, 2010). The technique is sometimes referred to as 'cream skimming' (Easley et al., 1996; Battalio, 1997) as brokers pick the uninformed part of the order flow and route unwanted orders to an exchange (Preece and Rosov, 2014). Since order filling at exchanges is not necessary while brokers are obliged to the best execution principle (Macey and O'Hara, 1997; SEC, 2013), firms typically offer price improvements over the quoted spread for internalized trades. Consequently, "internalization can be profitable both for the customer, who receives the price improvement, and for the broker, who earns a large realized spread" (Grammig and Theissen, 2012).

In cryptocurrency (crypto) trading, given the large retail share in the total trade volume (Bianchi et al., 2022), brokers may have similar incentives for retail internalization. However, while academic literature is confident that retail orders are less informed than institutional orders in equity trading, this is rather unclear in crypto trading. Makarov and Schoar (2020) find that "it is less obvious whether there are traders who are more informed than others and what the nature of the information is". Moreover, Bianchi et al. (2022) argue that the opaqueness of information flow compared to traditional markets and participants' heterogeneous beliefs about cryptocurrencies generate information asymmetry.

As a consequence, trading against the crypto retail order flow is not necessarily profitable, and brokers may seek for alternative off-exchange execution methods to save costs. One method discussed by Battalio and Loughran (2008) and Challet et al. (2018) is to internally match customers' buy and sell orders instead. Internal order matching, sometimes called order netting (Hagerty and McDonald, 1996; Schwartz et al., 2005; Francioni and Schwartz, 2008), allows brokers to save the bid-ask spread without having to trade against customers for their own account. The broker stores orders in inventory for which she expects a contrary order in the future and sends all other orders to the exchange. However, this order segmentation is not straightforward as it depends on the future order flow and traders submit orders asynchronously (Challet et al., 2018). If a broker had knowledge about the future, she could store exclusively orders covered by the future order flow in inventory, and route all other orders to the exchange. In practice, brokers do not have this information, and rule-based frameworks serve to control the market risk of their inventory position: Indeed, if she cannot internally match an order from inventory before its settlement date, she will have to execute it on the exchange at possibly worse prices and pay the full spread (Hagerty and McDonald, 1996).

Inspired from the work of Challet et al. (2018), this article presents a prediction-based model of crypto retail order segmentation for internal order matching. When a broker receives an order that she cannot immediately execute against open orders in inventory, the model predicts whether future order flow will be sufficient to neutralize the order before its settlement date. Using this prediction, the broker can decide whether to hold the order in inventory or route it to an exchange for immediate execution. Based on an order sample from BISON, the crypto retail broker of Börse Stuttgart Group with more than half a million active users and a trade volume of €5.6 billion in 2021, we develop a supervised learning model in two steps: First, we define the target variable *optimal internal matching rate*, which represents the maximum fraction of an order that a broker could have matched internally given a $t + 2$ settlement cycle, and thus is the true case we want to predict. Second, we derive predictors from an analysis of market and order flow dynamics prior to the submission of internally matched orders (positive outcome) and orders routed to the exchange (negative outcome). Our proposed predictors are the buy volume surplus, average order volume, total trade volume, high-low price range, and one-minute price changes. Prior to order submission, each predictor exhibits behavior specific to each of the two outcomes.

We arrive at the following findings: Historically, 85% of order volume is followed by contrary order volume within two days. Consequently, brokers can internally match a maximum of 85% of buy and sell order volume. Using both penalized logistic regression and artificial neural networks, we predict 85% of order flow correctly, allowing brokers to internally match three-quarters of daily order volume. Surprisingly, our models achieve the best prediction results during highly volatile and trading-intensive periods. Since these periods are typically associated with large spreads, internal order matching disproportionately pays off for both brokers and customers. Assumed that customers and brokers share the cost savings equally, prediction-based segmentation leads to a 35% reduction in effective spreads compared to quoted spreads. These results also hold under minimum price improvements such as required by the European Union for equity trading (Lucas, 2020). Our results thus support the theory of cost savings, consistent with empirical (Battalio, 1997; Hansch et al., 1999; Battalio et al., 2001; Peterson and Sirri, 2003; Grammig and Theissen, 2012) and theoretical (Battalio and Loughran, 2008; Degryse et al., 2022) studies of execution costs for internalized trades, and thereby suggest a potential improvement in market quality.

The remainder of the article is structured as follows. Section 3.2 embeds our research in the current state of theoretical and empirical research. Section 3.3 introduces the market structure of internally matching retail brokers, and Section 3.4 presents the data set used. Section 3.5 develops a model for retail order flow prediction. Section 3.6 is devoted to the empirical analysis of prediction-based order internalization, and Section 3.7 concludes.

3.2 Literature and regulation overview

Internalization has attracted the attention of researchers and regulators for about two and a half decades (Malinova, 2012). While there is a widespread agreement in the research literature that internalization may realize cost savings (Battalio, 1997; Hansch et al., 1999; Battalio et al., 2001; Peterson and Sirri, 2003; Battalio and Loughran, 2008; Grammig and Theissen, 2012; Degryse et al., 2022), there are mixed conclusions about market quality under internalization: Opponents of the practice argue that off-exchange trade execution reduces the liquidity of the lit market and thus may deteriorate market quality (Easley et al., 1996; Bessembinder and Kaufman, 1997; Chakravarty and Sarkar, 2002; Preece and Rosov, 2014). The problem is that if uninformed trades do not reach the public market anymore, market makers may cover increasing adverse selection risk through wider spreads. On the other hand, Battalio (1997), Battalio et al. (1997), and Hansch et al. (1999) find no evidence of detriments to traders.

One approach to regulate internalization in a way that allows realizing cost savings while protecting or improving market quality could be to require a minimum price improvement for internalized orders. Under a minimum price improvement, firms are allowed to internalize orders only if they offer a substantially better price than the public market to the trader and thereby reduce effective spreads. Hence, a minimum price improvement can be regarded “as a means to compensate the market for the liquidity lost" due to internalization (Kumpan, 2006). Indeed, Larrymore and Murphy (2009) and Comerton-Forde et al. (2018) show that mandatory price improvements for internalized orders can improve market quality.

With MiFID (Directive 2004/39/EC)[9] and MiFID II (Directive 2014/65/EU)[10], the European Union established a uniform regulation of internalization for equity trading. Accordingly, firms are required to either internalize orders at a minimum price improvement (Lucas, 2020), thereby "generating *economically meaningful* cost savings" (Preece, 2012), or route the orders to an exchange. Hence, internalization is higher among less liquid stocks with more potential for price improvement and lower among liquid stocks (Anolli and Petrella, 2007). As of today, crypto trading is not subject to internalization regulation by MiFID II. However, in the course of the upcoming regulation of the crypto market by MiCA[11], crypto internalizers could face the same or similar conditions as for equity trading in the next few years.

Besides, our work contributes to studies of order netting (Hagerty and McDonald, 1996) and order segmentation (Fleming and Nguyen, 2013; Garriott and Walton, 2018; Brolley and Cimon, 2020). We build our prediction model on study results on the predictive power of retail order flow in the stock market (Kelley and Tetlock, 2013; Boehmer et al., 2021) and in the crypto market (Scaillet et al., 2018; Silantyev, 2019; Ante, 2020). For instance, Boehmer et al. (2021) argue that retail investor behavior can predict future returns, with the predictive power depending on the level of the price improvement. Moreover, our work addresses studies on the drivers of internalization (Anolli and Petrella, 2007; Kwan et al., 2015; Barardehi et al., 2022), with Barardehi et al. (2022) finding that firms' internalization decisions depend on order type, amount of retail order flow and institutional liquidity demand. Finally, we contribute to the analysis of crypto order flow, although we do not attempt to measure informed trading (Wang et al., 2021; Feng et al., 2018), but follow the argument of Makarov and Schoar (2020) that it is unclear what the nature of superior information is in crypto trading. Our model eliminates the need to identify informed traders and thus contributes a meaningful improvement to off-exchange trade execution in crypto trading. The motivation for this article stems from Challet et al. (2018), who present an inventory prediction model for market participants who try to match buy and sell order flows by keeping some orders in their inventory.

3.3 Market structure

3.3.1 Internal order matching of a retail broker

We assume a retail broker who deploys a Request for Quote system for crypto trading as shown in Figure 3.1. When receiving a Request for Quote from a customer, the broker displays the price at which order execution is possible. The displayed price thus depends on how the

[9] Directive 2004/39/EC of the European Parliament and of the Council of 21 April 2004 on markets in financial instruments amending Council Directives 85/611/EEC and 93/6/EEC and Directive 2000/12/EC of the European Parliament and of the Council and repealing Council Directive 93/22/EEC, *Official Journal of the European Union*, L 145, 2004/04/30, p. 1–44. Download link: http://data.europa.eu/eli/dir/2004/39/oj.

[10] Directive 2014/65/EU of the European Parliament and of the Council of 15 May 2014 on markets in financial instruments and amending Directive 2002/92/EC and Directive 2011/61/EU. *Official Journal of the European Union*, L 173, 2014/06/12, p. 349–496. Download link: http://data.europa.eu/eli/dir/2014/65/oj.

[11] Markets in Crypto-Assets (MiCA) is a proposed regulation for European crypto-assets with the title 'Proposal for a regulation of the European Parliament and of the Council on Markets in Crypto-assets, and amending Directive (EU) 2019/1937 COM/2020/593 final'. Download link: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020PC0593.

broker executes the order: When routing the order to an exchange, the broker determines the public price at which immediate trade execution would be possible, i.e., the current volume-weighted best ask (bid) price for buy (sell) orders. When executing the order internally against other customer orders, the broker may offer a better price. We assume that the brokerage market is competitive and brokers have an incentive to offer price improvements on internally matched orders although not yet being obliged to do so (see Section 3.2).

Next, we parameterize that price improvement. Grammig and Theissen (2012) state that the price improvement determines how customer and broker share the cost savings. Under MiFID II, the minimum price improvement is one tick size, i.e., the broker may receive all cost savings beyond one tick. In contrast, Battalio and Loughran (2008) argue that brokers meet best execution only when passing the full monetary benefit to the customers, which,

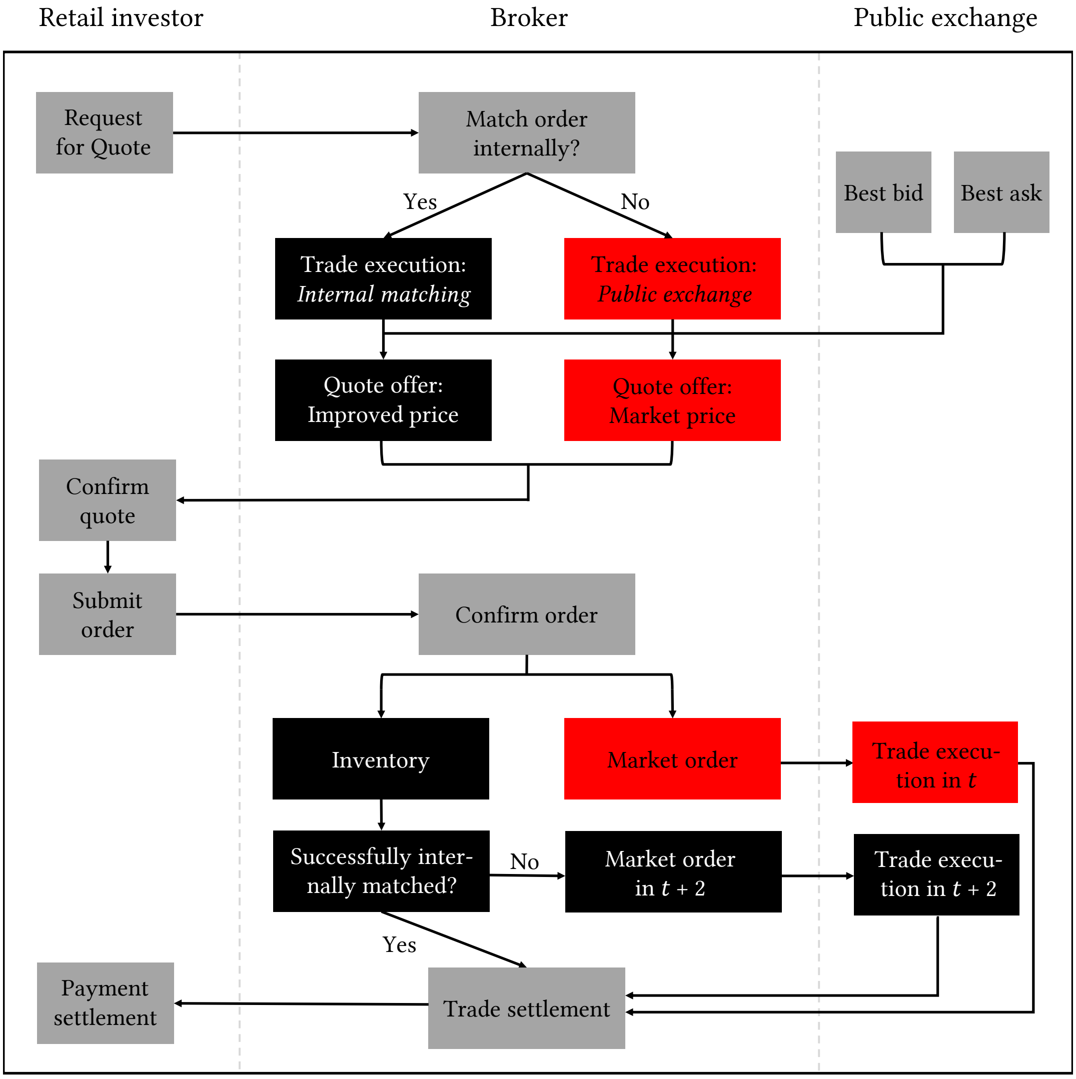

Figure 3.1: Market structure of a retail crypto broker who deploys a Request for Quote system and executes customer orders by internally matching with other customer orders at improved prices (black stream), or by placing market orders at exchanges (red stream). The illustration uses elements of Barardehi et al. (2022), but takes the broker's perspective and focuses on internal order matching.

however, implies that brokers have no monetary incentive to internalize. Thus, we assume that brokers and customers equally share the benefit and trade at the midpoint between the mid price and the quoted price, both saving one-fourth of the quoted spread. For instance, when BTC-EUR trades at €30,000/€30,020, the broker internally matches a buy (sell) order at €30,015 (€30,005), realizing cost savings of €5 for both herself and the customer[12].

By accepting the displayed quote, the customer submits a request for trade. If confirmed by the broker, the trade is legally binding for both sides and no subsequent changes to the execution price are possible. The broker takes the counterpart of the trade, i.e., a buy (sell) order results in a short (long) position of the broker. When executing the order at an exchange, she places a corresponding buy (sell) market order and thereby neutralizes the open position. When internally matching the order instead, she either immediately executes the order against open positions from inventory, thereby neutralizing the position, or stores it in inventory, thereby holding the position and awaiting internal order matches in the future.

Motivated by the standard settlement procedure in European securities trading (Thomadakis, 2022), we assume $t+2$ (48 hours) settlement cycle, i.e., the broker has to transfer the traded amounts between the customer's and her own wallet 48 hours after trade confirmation at the latest. When immediately routing the order to the exchange or executing the order against open positions, trade settlement occurs in t. When the broker cannot fully execute the order against incoming orders within two days, she has to close the position at an exchange two days after, which is detrimental to her revenue as she pays the quoted spread and thus makes a loss of one-fourth of the quoted spread on that trade[13]. For this reason, when internally matching customer orders, older orders in inventory have a higher priority. For instance, if we receive a large buy order while we have several small sell orders in inventory, we start the internal matching procedure with the longest waiting sell order, continue with the second longest waiting order, and so on until we have been able to fully internally match the order. If the order is larger than the total open position, we hold the remaining amount as an open position in inventory to execute against future orders. So in this scenario, our exposure would change from a net long to a net short position.

3.3.2 Order segmentation

Order segmentation describes the practice of picking the part from the order flow which we wish to execute internally. We regard two methods of order segmentation: The traditional size-based segmentation and a novel prediction-based segmentation. In size-based segmentation, illustrated in Figure 3.2, brokers segment orders by order size, which is the standard approach for filtering uninformed trades (Kim and Verrecchia, 1991; Grundy and McNichols, 2015; Shen et al., 2017). Similar to Anolli and Petrella (2007), our proxy for the critical order size is the European Union regulation of Systematic Internalizers that are allowed to internalize orders up to the standard market size defined by the Commission Delegated Regulation

[12] In practice, brokers internally match asynchronously submitted orders, and the displayed quotations in the order book at the time of arrival are usually not identical. For instance, internally matching a buy order about 1 BTC at €40,050 against a sell order about 1 BTC at €40,055 submitted by another customer few seconds later would occur a loss of €5 for the broker. However, Appendix 3.A suggests that, empirically, price differences between orders executed against each other net out in the cross-section and thus have no impact on the broker's revenue.

[13] Besides, closing open positions in $t+2$ bears market risk, as market prices can change in the time between order arrival and final execution. Appendix 3.A shows, however, that these price differences have no impact on our revenue in the cross-section, consistent with the price differences for internal order matches. As a result, the average loss from closing inventory positions is one-fourth of the bid-ask spread.

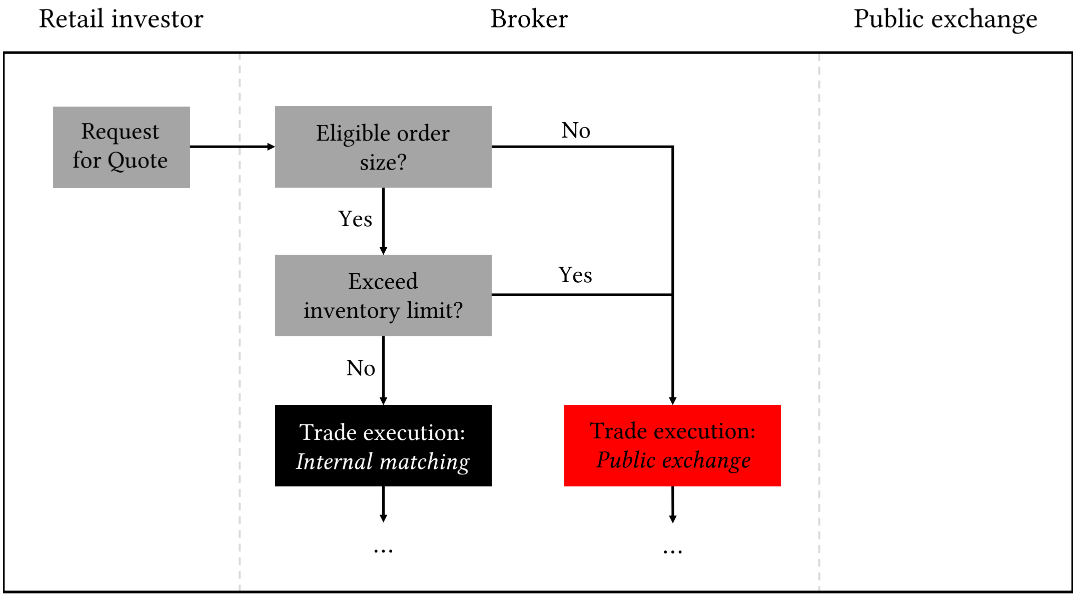

Figure 3.2: Market structure of a retail crypto broker who segments order flow based on order size and routes large orders to the exchange and small orders to inventory.

2017/587[14], amounting to €10, 000 for most securities analyzed by Lucas (2020). Thus, the broker theoretically could route all incoming orders up to €10, 000 volume to inventory.

However, in practice, brokers limit their exposure to market risk by setting a maximum open position (Challet et al., 2018). As this limit typically depends on undisclosed firm-specific risk metrics, we assume a discretionary inventory limit of €100,000, i.e., if a broker is short (long) €98,000 and receives another €5,000 buy (sell) order, she has to route it to an exchange. Accordingly, Figure 3.2 shows that, after checking the order for eligibility under the order size limit and before determining the trade execution modality, we check whether routing the arriving order to inventory would lead to a violation of the inventory limit. If so, we would route the order to the exchange, although it might be of eligible order size. As we choose a discretionary inventory limit, Appendix 3.E examines the relevance of the inventory limit choice for order segmentation and demonstrates that it plays a rather minor role compared to the order size limit. Overall, size-based order segmentation under inventory limit constraints is very straightforward as the broker only needs to check order size and net position to decide whether to internally match the order or route it to the exchange.

In contrast, we present a novel prediction-based segmentation as a way to segment order flow by predicting future order flow from a broker's private order flow data and exchanges' publicly available market data. Figure 3.3 illustrates a broker who determines trade execution

[14] Commission Delegated Regulation (EU) 2017/587 of 14 July 2016 supplementing Regulation (EU) No 600/2014 of the European Parliament and of the Council on markets in financial instruments with regard to regulatory technical standards on transparency requirements for trading venues and investment firms in respect of shares, depositary receipts, exchange-traded funds, certificates and other similar financial instruments and on transaction execution obligations in respect of certain shares on a trading venue or by a systematic internalizer. *Official Journal of the European Union*, L 87, 2017/03/31, p. 387–410. Download link: http://data.europa.eu/eli/reg_del/2017/587/oj.

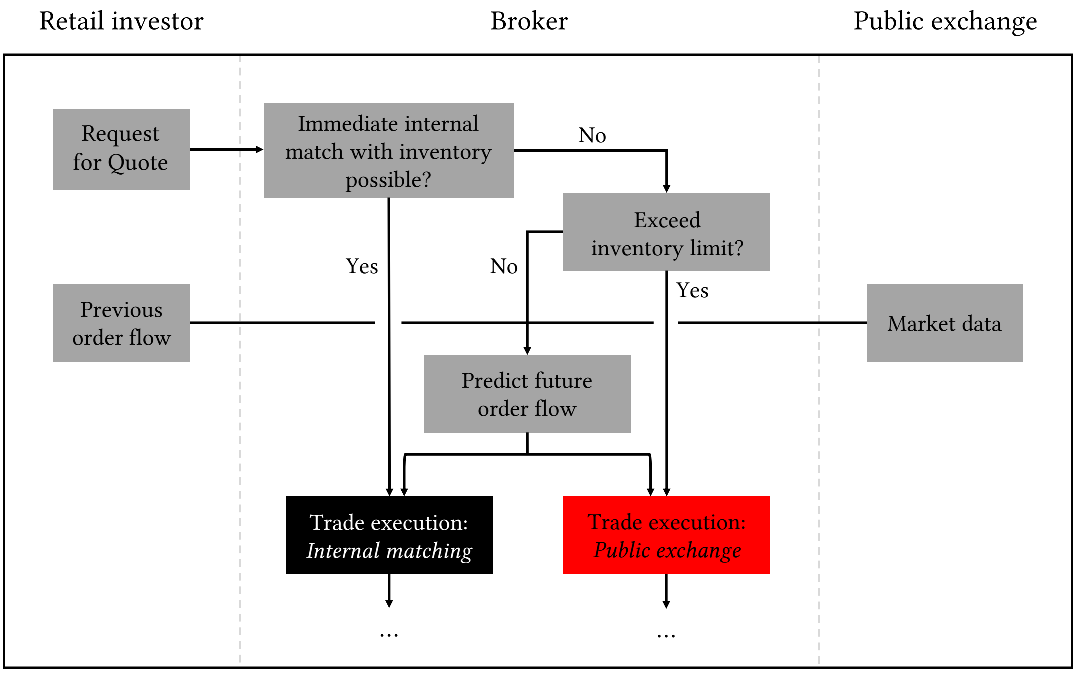
Retail investor
Broker
Public exchange
Request for Quote
Immediate internal match with inventory possible?
No
Exceed inventory limit?
Yes
No
Yes
Previous order flow
Market data
Predict future order flow
Trade execution: *Internal matching*
Trade execution: *Public exchange*
...
...

Figure 3.3: Market structure of a retail crypto broker who segments order flow based on order flow predictions and routes orders either to an exchange or to inventory. We assume that the broker processes private information from previous order flow data and public information from market data to deliver a prediction.

modality according to order flow prediction: When receiving a customer order that the broker cannot immediately execute against open inventory positions, we retrieve market and order flow data and forecast whether the future order flow will neutralize the requested order before the settlement date. If predicted so, the broker offers an improved price and stores the order in inventory unless it violates the inventory limit. Otherwise, the broker offers the public quotes and routes the order to an exchange. Before generating a prediction, the broker must check whether adding the order to the inventory would exceed the inventory limit. Hence, we make a prediction only if we cannot immediately execute the requested order against open positions and if we do not violate our inventory limit. If we can internally match a portion of the requested order against open positions, we make a prediction based on the remaining order amount. Correspondingly, we are also allowed to split requested orders in order to meet the inventory limit: Thus, if we are short €98,000 and receive another €5,000 buy order, we could hold 40% of the position in inventory and route the other 60% to the exchange.

3.4 Data

We gratefully acknowledge the complete order history of the BISON app from Boerse Stuttgart Group and its affiliate Sowa Labs. In the provided data set, each order entry reports the exact time, currency pair, order size and order side. BISON offers Euro-paired trading for Bitcoin Cash (BCH), Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), and Ripple (XRP)[15]. BISON cus-

[15] We exclude Chainlink and Uniswap from the data set since BISON only started trading in late 2021. In late 2022, BISON launched additional currency pairs, but these were not yet part of the dataset (last entry March 30, 2022).

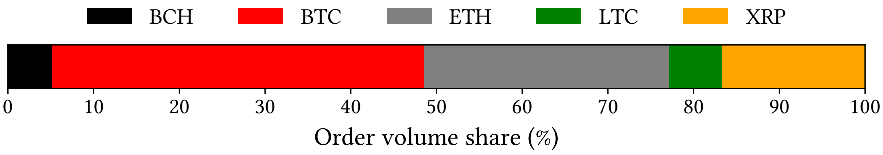

Figure 3.4: Share of order volume by currency pair between July 01, 2020 and March 30, 2022.

tomers can submit market and limit orders, whereas a triggered limit order is converted to a market order. Due to the low trade volume in the early phase of the app, we analyze market orders and triggered limit orders between July 01, 2020 and March 30, 2022. Figure 3.4 illustrates the order volume share by currency. BTC and ETH together account for nearly three quarter of the total order volume, whereas XRP accounts for about 15%.

In addition, we download historical open, high, low, closing price and trade volume (OHLCV) data at one-minute granularity from the crypto exchanges Bitstamp[16], Kraken[17], and Coinbase[18], three popular crypto exchanges also considered by other empirical studies like Alexander et al. (2020), which serve as the broker's reference markets for immediate order execution in our study. As historical limit order book data are not publicly available, we follow Brauneis, Mestel, Riordan and Theissen (2021), who propose to estimate cryptocurrency high-frequency spreads based on the Abdi and Ranaldo (2017) estimator (Brauneis, Mestel and Theissen, 2021). Given the high (h), low (l) and closing (c) price of a time interval t, Abdi and Ranaldo (2017) estimate the bid-ask spread for two adjacent intervals t and $t+1$ by

$$\text{Spread}_t = \sqrt{\max\left[4 \cdot \left(\ln(c_t) - m_t\right)\left(\ln(c_t) - m_{t+1}\right), 0\right]} \tag{3.1}$$

where $m_t = \frac{1}{2} \cdot \left(\ln(h_t) + \ln(l_t)\right)$ is the midpoint between logarithmic high and logarithmic low price of interval t. In the following sections, we report bid-ask spreads as volume-weighted average across exchanges, where the volume weights base on the total trade volume of the respective exchange during that minute. For more details, Appendix 3.C illustrates the historical volume-weighted daily bid-ask spread estimates separated by exchange, suggesting that spreads are similar across all three exchanges.

Lastly, we divide the data set into training and test periods. Following Ji et al. (2019), we distinguish two potential partitioning methods[19]: Sequential partitioning uses the first, e.g., 80% of the period for training and validation and the rest 20% for testing, assuming that dynamics in the data remain constant, e.g., that March 2022 would be suitable to test of what the model has learned from 2020 (see Section 2.6.1). In contrast, n-fold cross validation divides the period into n equal-sized sub-intervals and trains n independent models for each of the sub-intervals, each consisting of a sequentially partitioned train and test period.

Ji et al. (2019) show that for classification problems in crypto trading, n-fold cross-validation

[16] We download historical OHLCV data from Bitstamp via the Bitstamp API (https://www.bitstamp.net/api/).

[17] Kraken offers readily downloadable OHLCV data online at https://support.kraken.com/hc/en-us/articles/360047124832-Downloadable-historical-OHLCVT-Open-High-Low-Close-Volume-Trades-data.

[18] We retrieve historical OHLCV data from Coinbase using `get_product_historic_rates` method of the Public Client of `cbpro` (Paquin, 2020), the Python client for the Coinbase Pro API (https://docs.cloud.coinbase.com/).

[19] Following Ji et al. (2019), another method is random partitioning that randomly picks training and test data from the entire period, i.e., both training and test set comprise data from the entire period. However, for testing our predictions in a trading simulation with internal matching of subsequent orders, we need time sequential predictions for a given period.

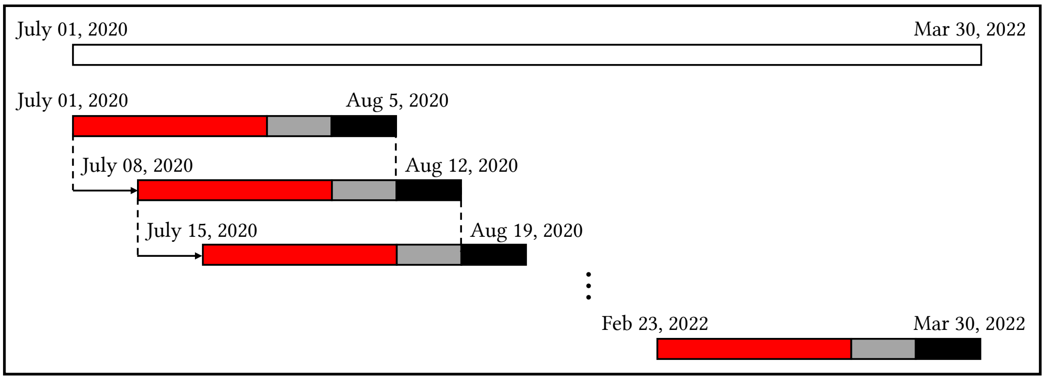

Figure 3.5: Division of the observation period into overlapping five-weeks sub-intervals for n-fold cross validation. Each sub-interval consists of three weeks (60%) training (red), one week (20%) validation (gray) and one week (20%) test (black). The overlap is four weeks.

is the best-suited sequencing method. Moreover, compared to Section 2.6.1, the dynamics of the crypto market differ substantially from traditional markets (Klein et al., 2018), driven by temporary trends due to exploding retail and institutional inflows between 2020 and 2022. Therefore, we follow Ji et al. (2019) and divide the entire period into overlapping sub-intervals of five weeks in length, of which, as Figure 3.5 illustrates, the first three weeks are for training (60%), the fourth week is for validation (20%), and the last week is for testing (20%). The entire period comprises 87 such sub-intervals. The validation sets serve to design the architecture of the prediction model and to regularize the training (see Section 3.5.4). The overlap is four weeks, i.e., we test every week of the entire period with an independently trained model.

3.5 A prediction model for internal order matching

3.5.1 Optimal internal matching rate

The internal matching rate of an order measures the share of the internally matched order volume in its total order volume (Anolli and Petrella, 2007). For instance, if we receive an order about 0.5 BTC and can internally match 0.25 BTC of it, the internal matching rate of that order is 50%. As target variable for our prediction model, we propose the *optimal* internal matching rate of an order. The optimal internal matching rate represents the maximum achievable internal matching rate under $t + 2$ settlement. A broker with complete knowledge of the future would achieve the optimal internal matching rate by holding in inventory all orders for which she receives a contrary order in the future and routing to the exchange only those orders that cannot be neutralized by future order flow. Thus, the optimal internal matching rate represents the true case we wish to predict, as we assume a profit-maximizing broker who aims to internally match as much order volume as possible and route the smallest possible share of orders to the exchange.

We determine the optimal internal matching rate by chronologically routing all orders to inventory, where we either execute them immediately against open positions, or store them and determine whether future orders are sufficient to neutralize the resulting position or not. An internal matching rate of 1 occurs when either open inventory positions at the time of order arrival or subsequent order flow have fully neutralized the order. A rate of 0 indicates

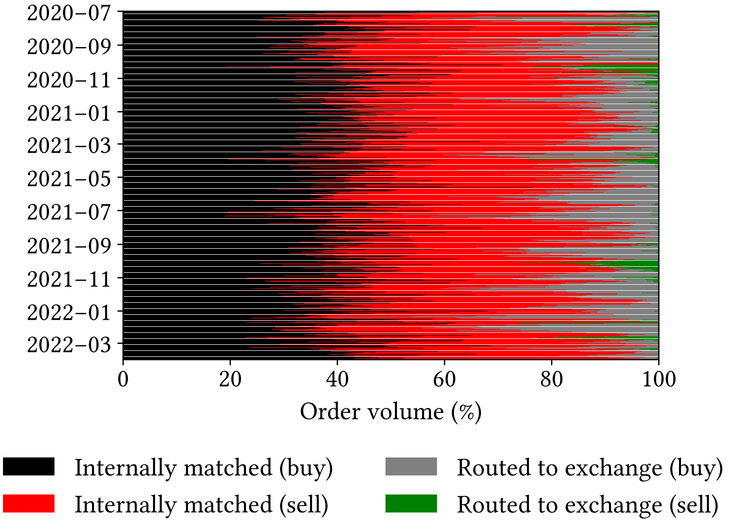

Figure 3.6: Daily volume-weighted optimal internal matching rate of order flow across all currency pairs. We calculate optimal internal matching rate according to Section 3.5.1. The black (red) bars represent internally matched buy (sell) order volume, while the gray (green) bars represent buy (sell) volume routed to the exchange. © 2022 ACM

that we either have the order stored in inventory but future orders could not neutralize it, or we had to pass it to an exchange due to the inventory limit. An internal matching rate of 0.5 may indicate that subsequent orders neutralized half of the order, and we routed the other half to an exchange in $t + 2$. Since internal order matching is possible only within orders of the same currency pair, we carry out the iteration for each pair once.

Figure 3.6 illustrates the daily volume-weighted optimal internal matching rate across all currencies. We determine daily volume-weighted internal matching rates by multiplying each order's volume with its optimal internal matching rate, and then aggregating the resulting optimal internal matching volumes on a day-by-day basis. Finally, we calculate the volume-weighted averages across all currency pairs. The figure illustrates that we can internally match 85% of daily buy and sell order volume on average, i.e., 85% of daily buy (sell) order volume is followed by sell (buy) order volume within 48 hours. Hence, a broker who is willing to internally match as much order volume as possible, theoretically could realize cost savings for 85% of order volume. Orders routed to the exchange are largely buy orders, indicating a buy surplus in the underlying period of the data set. Overall, Figure 3.6 suggests that internal matching seems to be relatively constant throughout time with no general trends.

3.5.2 Market and order flow dynamics before and after order arrival

Next, we wish to understand what are the drivers of internal matching and whether particular market and order flow circumstances drive internal matching. To this end, Figure 3.7 illustrates market and order flow dynamics before and after order arrival for internal order matches and orders routed to the exchange. The upper left plot illustrates the average hourly *buy volume share*, which is the buy order volume during a particular period divided by the total order volume during that period. The x-axis reports the time difference to the order arrival

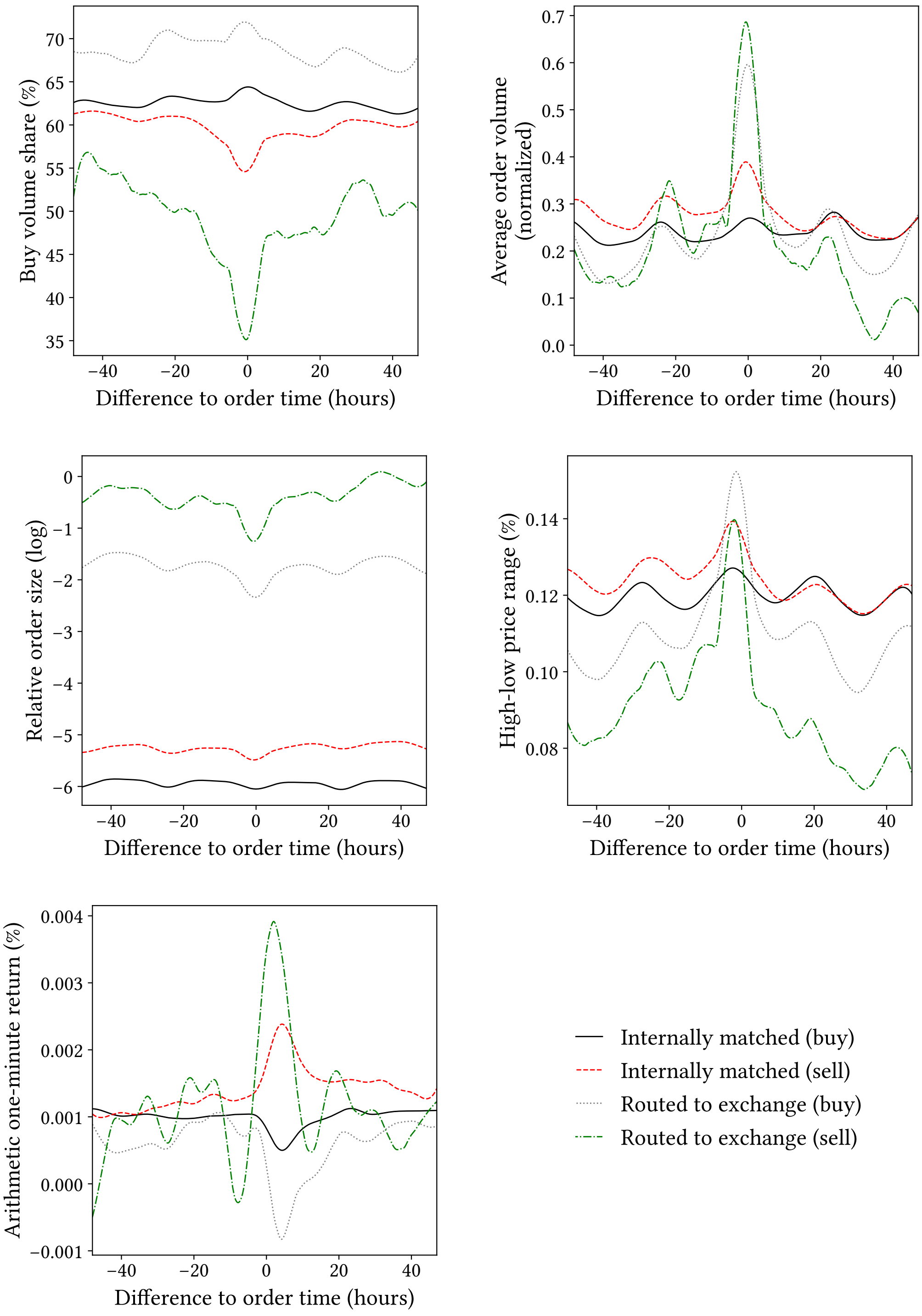

Figure 3.7: Market and order flow dynamics before and after order arrival for internally matched customer orders (black, red) and customer orders uncovered by subsequent order flow (gray, green). Internal order matches base on the optimal internal matching rate following Section 3.5.1. Reported values are equal-weighted averages within each currency pair and volume-weighted averages across pairs. © 2022 ACM

time in hours. For instance, the value reported at −5 hours depicts the average buy volume share between 5 and 6 hours prior to receiving an order. The calculation of the reported value includes two steps: First, for each order of the data set, we aggregate all orders between 5 and 6 hours prior to the order arrival time, and divide the total buy order volume of the aggregated group by the total order volume of that group. Second, we take the equal-weighted average of the calculated ratios across all orders. The chart shows that buy (sell) orders that we could not internally match exhibit relatively high (low) buy volume shares prior and after order arrival, suggesting that the chance that future order flow neutralizes arriving buy (sell) orders may be lower when the current buy volume share is above (below) the cross-sectional mean, i.e., when there is a buyer (seller) overhang, and higher when we received fewer buy (sell) orders in the past.

The upper right plot illustrates the *average order volume*, which measures the total order volume during a period divided by the number of orders during that period. For reasons of data confidentiality, we have normalized the numbers in the range [0, 1]. Orders routed to the exchange exhibit a sharply increasing average order volume shortly before order arrival, suggesting that when customers submit large orders, inventory positions and future order flow cover smaller portions of subsequent orders. The reason is that the internal order matching algorithm follows a waiting time hierarchy (see Section 3.3.1), which gives older open positions a higher priority. As a consequence, future order flow neutralizes older orders first, and the larger they are, the less likely it is that the future order flow will be sufficient to neutralize new orders, regardless of their size.

The center left plot presents the logarithmic *relative order size* of orders, which is the volume of the submitted order relative to the previous total trade volume. For instance, the value reported at −5 hours depicts the order volume divided by the total order volume between five and six hours ago. Orders that we are not able to internally match are relatively large compared to the order flow: The volume of internally matched orders accounts for less than 1% of previous and following hourly order volume on average ($e^{-5} < 1\%$), while the volume of buy and sell orders routed to the exchange averages 18% (= $e^{-1.7}$) and 67% (= $e^{-0.4}$), respectively, of the hourly order flow. The downward breakout of the gray and green graphs around the time of order arrival indicates a large order volume at that time.

The center right chart shows the one-minute percentage *high-low price range*, calculated by dividing the distance between the high and low price in a minute by the high price in that minute. We determine high and low prices as the trade volume-weighted average high and low prices at Kraken, Bitstamp and Coinbase. Compared to internally matched orders, orders routed to the exchange exhibit both decreasing high-low ranges prior and after order arrival and rising high-low ranges around the time of arrival, suggesting that order flow less likely offsets incoming orders when high-low price ranges increase. As high-low ranges can serve as a measurement of volatility (Martens and van Dijk, 2007), we show in Appendix 3.D that standard deviation of returns, total trade volume and bid-ask spreads follow similar patterns. Orders routed to the exchange are accompanied by below-mean volatility before and after order arrival, and sharply increasing volatility around the time of order arrival.

Finally, the bottom chart shows the average *arithmetic one-minute return*, determined by the one-minute percentage change of volume-weighted mid prices at Kraken, Bitstamp and Coinbase. The black and red graphs indicate that slightly decreasing (increasing) returns may increase internal matching rate of buy (sell) orders. Orders routed to the exchange occur in an environment of sharply changing returns, e.g., sell orders see abruptly falling returns before order arrival, followed by strong positive returns at the time of order arrival, potentially indicating a period in which customers sell large volumes when a price increase slows down.

3.5.3 Predictors of internal matches

We define the prediction task as a classification problem with the optimal internal matching rate as target variable, labeled as '1' if it is greater or equal to 50%, and labeled as '0' otherwise. Thus, the model predicts the probability that future order flow neutralizes an incoming order. Next, we derive suitable predictors for that prediction task based on what we have learned from the market and order flow analysis in Section 3.5.2.

Since all variables analyzed in Figure 3.7 show individual behavior for both classes, i.e., orders internally matched and orders routed to the exchange, we propose the differences between the current one-hour averages and the past two-hours, three-hours, six-hours, twelve-hours, one-day and two-days averages of all variables as predictors. For example, if the buy volume share has been 0.4 in the past hour and 0.5 in the past six hours, the six-hours change is -0.1. The chosen time windows serve to incorporate both short-, mid- and long-term trends. Prior to the training, we standardize each predictor separately for each currency by subtracting the mean and dividing by the standard deviation across all sub-intervals (LeCun et al., 2012). With controlling for the order side, order volume, order amount, open inventory position and currency, all combinations yield a total of $5 \cdot 6 + 5 = 35$ predictors.

3.5.4 Prediction methodology

Elastic Net The first prediction methodology[20] that we analyze is logistic regression. Logistic regression minimizes the binary cross-entropy loss, which is

$$L_{\log}\left(y_i, \hat{y}_i\right) = -\frac{1}{n}\sum_{i=0}^{n}\left(y_i \cdot \log\left(\hat{y}_i\right) + \left(1 - y_i\right) \cdot \log\left(1 - \hat{y}_i\right)\right) \tag{3.2}$$

where y_i is the true label and $\hat{y}_i$ is the predicted label of order i, and n is the number of orders. However, as our predictors set is large and predictors are partly correlated or irrelevant, which causes the simple logistic regression to "overfit noise rather than extract signal" (Gu et al., 2020), we consider penalized logistic regression instead. Penalized logistic regression appends a penalty on coefficient estimates to $L_{\log}\left(y_i, \hat{y}_i\right)$ in order to reduce the number of estimated parameters β_j, thereby avoiding noise fit. Ridge regression (Hoerl and Kennard, 1970) penalizes the sum of squared coefficients in order to shrink them towards zero (L_2 regularization). LASSO regression (Tibshirani, 1996) penalizes the sum of the coefficients' absolute values and performs variable selection by shrinking coefficients to exactly zero (L_1 regularization). Elastic Net (Zou and Hastie, 2005) combines L_1 and L_2 regularization based on a mix parameter $\alpha \in [0, 1]$ that determines each penalty's share. The Elastic Net loss function is

$$L_{\text{ENet}}\left(y_i, \hat{y}_i, \alpha, \lambda\right) = L_{\log}\left(y_i, \hat{y}_i\right) + \lambda \sum_{j=1}^{m}\left(\frac{1}{2}\alpha\beta_j^2 + (1 - \alpha)\left|\beta_j\right|\right) \tag{3.3}$$

where λ controls the level of coefficient shrinkage and m equals the number of predictors. We follow Gu et al. (2020) and optimize α and λ based on the volume-weighted prediction

[20] Section 3.5.4 describes prediction methodologies following Gu et al. (2020), Algamal and Lee (2015), Aggarwal et al. (2018), Géron (2018), Nielsen (2015), and Brownlee (2016).

accuracy (see Section 3.6.1) for the validation samples. Our grid includes each 100 values for α ranging from 0.01 to 1.00 and for λ on a logarithmic scale ranging from 0.0001 to 10,000. We train Elastic Net using the linear classifiers with stochastic gradient descent training provided by the `scikit-learn` package (Pedregosa et al., 2011).

Artificial Neural Network The second prediction methodology that we analyze is the Artificial Neural Network (ANN). ANNs are able to learn data structures outside of a static model structure by using multiple layers that extract different levels of characteristics from the input information. Equipped with nonlinear activation functions, ANNs generally can approximate any complex, nonlinear function. We consider feedforward neural networks, which have an input layer of raw predictors, one or more hidden layers that transform the predictors, and an output layer that aggregates the previous layers into a prediction (Gu et al., 2020).

Figure 3.8 illustrates an example of an ANN with eight predictors and two hidden layers. Inspired by how animal brains learn and process information, each layer contains neurons connected with neurons in the next layer by individual synapses. The number of neurons (units) in the first and last layer equals the dimensions of predictors and target variables, respectively. Accordingly, our network's input layer has 35 units and the output layer has one unit. Designing the hidden architecture is not as straightforward: While Csáji et al. (2001) argue that a single hidden layer can approximate any function, Eldan and Shamir (2016) demonstrate that increasing depth can be superior to increasing width. We use the validation sets to test fully connected architectures of up to five hidden layers with dimensions according to the geometric pyramid rule (Masters, 1993), i.e., a three-layer architecture with one hidden layer architecture has $\sqrt{35 \cdot 1} \approx 6$ hidden neurons, the four-layer architecture has $[12, 3]$, the five-layer architecture has $[16, 6, 2]$ hidden neurons, and so on. Finally, we choose the four-layer architecture that achieves the highest volume-weighted accuracy on the validation sets.

Each training step minimizes the cross-entropy loss $L_{\log}\left(y_i, \hat{y}_i\right)$ defined in equation (3.2) using stochastic gradient descent and the Adam algorithm (Kingma and Ba, 2015) with an initial learning rate of 0.001 and a batch size of 16. Stochastic gradient descent improves the generalization of the model (Smith et al., 2020) by introducing randomness into the optimiza-

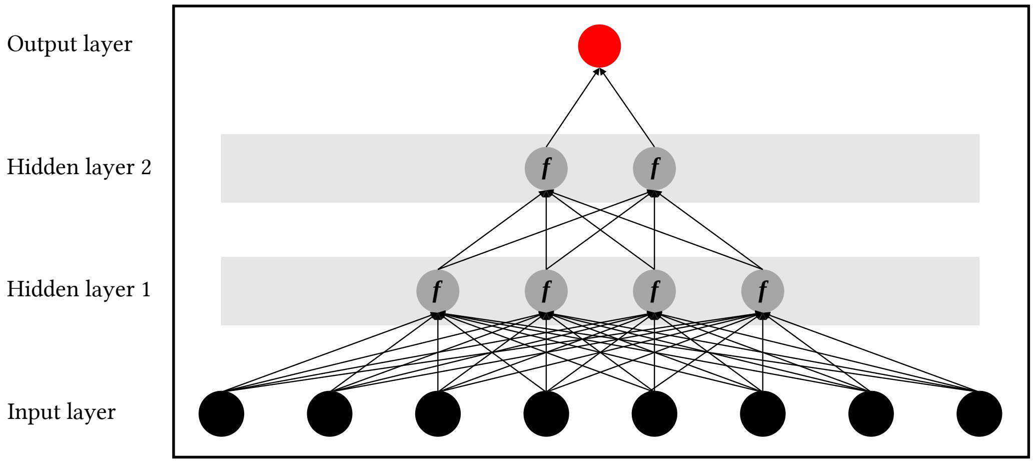

Figure 3.8: Illustrative artificial neural network following Gu et al. (2020) with eight input units, two hidden layers in the style of Masters (1993) with activation functions f, and one output neuron.

tion routine so that repeated optimizations can lead to different local minima, i.e., different model parameterizations. As activation function of the nodes in the hidden layers, we consider the rectified linear unit (ReLu) defined in Nair and Hinton (2010) by

$$\mathrm{ReLu}(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{otherwise.} \end{cases} \tag{3.4}$$

We regularize training in two ways. First, dropout is a way of model averaging by randomly ignoring a certain fraction of neurons in both hidden and visible layers and thereby simulating a large number of different network architectures in one model (Srivastava et al., 2014). As calibrated on the validation sets, we use 10% dropout, i.e., the network randomly ignores every tenth node during each training step. Second, an early stopping rule serves to halt training once the performance on the validation set does not improve for more than two epochs. Like in Section 2.2.2, we train ANN with the `keras` package (Chollet et al., 2015) and Google's `TensorFlow` (Abadi et al., 2015).

3.6 Prediction-based internal order matching

3.6.1 Prediction quality

Given that the positive and negative classes represent order execution through internal matching and routing to the exchange, respectively, a classification is true-positive (TP) if we hold the order in inventory and subsequent order flow actually neutralizes it. If the position remains uncovered until $t+2$ and we have to close it on the exchange, the classification is false positive (FP). If we routed it to an exchange and subsequent order flow would actually not have neutralized it, the classification is true negative (TN). Otherwise, if the order flow had been sufficient to neutralize it, the classification would have been false negative (FN).

Traditional quality measures are accuracy (ACC), true positive rate (TPR), and true negative rate (TPN). ACC determines the proportion of true classifications in all observations, whereas TPR (TNR) determines the ratio between inventory (exchange) classifications and the actual number of orders internally matched (routed to the exchange). In a trading context, however, these measures are insufficient as they weight all observations equally and thus ignore the financially different effects of classifications. For example, misclassifying a large order may have a larger impact on the broker's revenue than misclassifying a small order. We therefore consider order volume-weighted variants of the traditional quality measures instead: According to equation (2.13), we define the order volume-weighted accuracy by

$$ACC^w = \frac{V_{TP} + V_{TN}}{V_{TP} + V_{TN} + V_{FP} + V_{FN}} \tag{3.5}$$

where V_{TP} (V_{TN}) is the total order volume of TP (TN) classifications and V_{FP} (V_{FN}) is the total order volume of FP (FN) classifications. Correspondingly, the order volume-weighted true positive rate TPR^w and true negative rate TNR^w are defined by

$$TPR^w = \frac{V_{TP}}{V_{TP} + V_{FN}} \tag{3.6}$$

$$TNR^w = \frac{V_{TN}}{V_{TN} + V_{FP}} \tag{3.7}$$

We segment orders based on the prediction probability p and the probability threshold th. If $p > th$, we store the order in inventory, and if $p \leq th$, we route it to the exchange. In case of a data set with balanced class distributions, a common value for th is 50%, implying that we store orders in inventory if we predict that future order flow neutralize them with a probability of more than 50%. However, as our data set is imbalanced with relatively more internally matched orders and Zou et al. (2016) point out that "classification with imbalanced class distributions is a major problem in machine learning", we should choose th carefully.

Figure 3.9 analyzes prediction quality for different values of th and reveals the typical trade-off between high TNR^w with low TPR^w vs. low TNR^w with high TPR^w. A question that arises is whether either brokers or customers have individual preferences for a particular prediction quality outcome. For instance, a high TNR^w combined with a low TPR^w implies that the broker has to close more inventory positions on the exchange than she erroneously sends to the exchange. Consequently, the broker prefers a high TPR^w over a high TNR^w. On the other hand, customers save more cost when the broker sends more orders to inventory by accepting a lower TPR^w and thus prefer a high TNR^w. In order to consider both sides' preferences, we propose th for which $TNR^w \approx TPR^w$, which is $th = 97\%$ in the case of Elastic Net and $th = 89\%$ for ANN. In other words, we route orders to inventory if the predicted probability of an internal match for this order is more than 97% or 89%, respectively. These values are well above 50%, reflecting the fact that most orders *are* neutralized by subsequent order flow, so we must be more sensitive to the prediction probability by already routing orders to the exchange with even a small chance that future order flow may be insufficient to neutralize them.

Table 3.1 reports the prediction quality of four order segmentation scenarios: In the scenario 'None', we do not segment order flow at all and route all orders to the exchange. The scenario 'Order size' represents size-based order segmentation, whereas 'Elastic Net' and 'ANN' represent prediction-based order segmentation following Elastic Net and ANN predictions, respectively. The bottom line focuses on the Pearson correlation of optimal internal match-

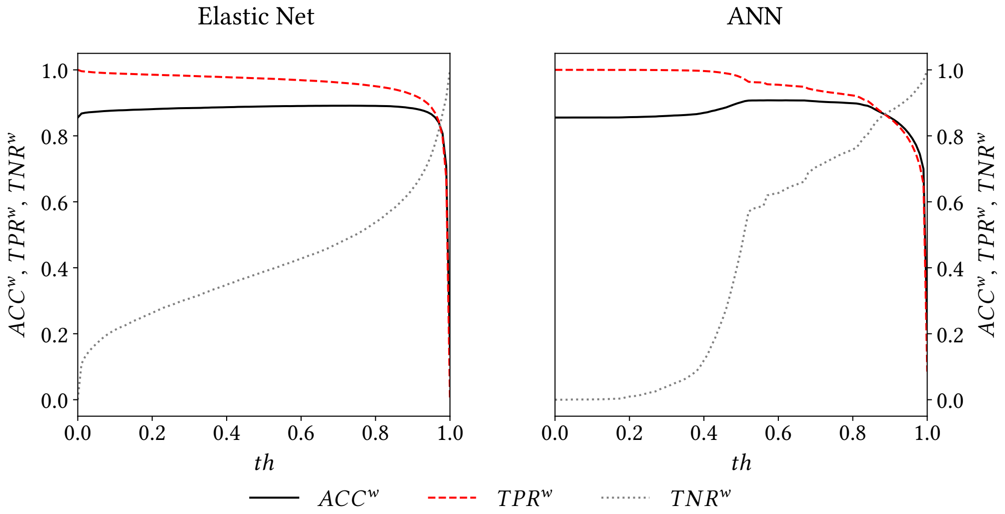

Figure 3.9: Volume-weighted prediction quality as a function of the probability threshold th (validation sets). Reported values use the predictions of Elastic Net (left) and ANN (right). We define prediction quality for each value in the range $th = [0, 1]$ in 0.01 steps. © 2022 ACM

Segmentation	None	Order size	Elastic Net	ANN
ACC^w	0.14	0.40	0.84	0.86
TPR^w	0.00	0.30	0.84	0.86
TNR^w	0.14	0.99	0.82	0.87
Correlation	–	0.13	0.38	0.51

Table 3.1: Volume-weighted prediction quality of the four scenarios `None' (without internal matching), `Order size' (size-based segmentation), `Elastic Net' and `ANN' (prediction-based segmentation). The last row reports the Pearson correlation of optimal internal matching rates and prediction probability. The probability threshold th is 97% for Elastic Net and 89% for ANN. © 2022 ACM

ing rates and prediction probability as a measure of model fit.

Prediction-based order segmentation correctly sends around 85 out of 100 traded Euros to either exchange or inventory, which is more than double the amount of size-based segmentation, with ANN being slightly superior to Elastic Net. Regarding Figure 3.6, this means that we correctly predict 85% out of 85% of potential internal order matches. The large TNR in size-based segmentation accounts for the fact that large orders typically are not neutralized by subsequent order flow (see Figure 3.7). Besides, ANN predictions achieve a correlation of 51% with realized order neutralization, outperforming Elastic Net by more than 10 percentage points. Finally, we implement a Diebold and Mariano (1995) test for the null that the difference between size- and prediction-based classifications is zero. We can reject the null at the 0.0001 significance level for both Elastic Net and ANN, and conclude that the forecasts are significantly different from size-based segmentation, i.e., our prediction models extract more information from the input data than just that we should send large orders to the exchange.

3.6.2 Internal matching rate

Precisely, evaluating our model performance by statistical measures is insufficient as a true classification does not necessarily imply that we are actually able to internally match that order. For instance, when erroneously sending an order to the exchange (false negative), a subsequent order that was empirically executed against that first order, now cannot be internalized anymore. If nevertheless routing that second order to inventory, the order classification would appear as true positive when comparing predicted and true labels, when in fact we cannot execute it internally and have to close it on the exchange. Hence, for both brokers and traders, only the share of successfully internally matched orders is relevant.

To this end, we determine the internal matching rate following order segmentation based on Elastic Net or ANN predictions according to the order execution algorithm from Section 3.3.1. Figure 3.10 illustrates the proportion of internally matched volume by currency across all test periods. The bottom row represents the volume-weighted average of all currencies and shows an internal matching rate of one quarter for size-based segmentation and three quarters for forecast-based segmentation, which is about 10 percentage points below the optimal internal matching rate indicated by green diamonds. Thus, by using our forecast model instead of order size limits, a broker can triple internal matching. Internal matching rates achieved by Elastic Net and ANN are comparable across all currency pairs, whereas size-based segmentation reveals that orders in less liquid currency pairs such as BCH or LTC are smaller and thus generate higher internal matching rates.

Next, we analyze the relationship between internal matching rate and predictors. Figure

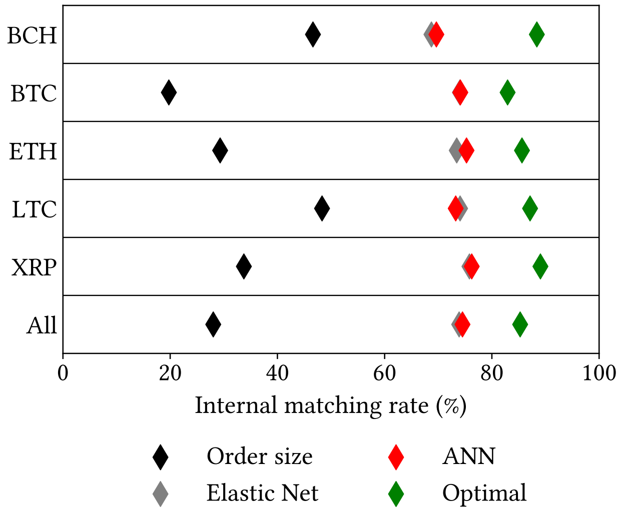

Figure 3.10: Internal matching rate by currency. We determine internal matching rate based on size-based segmentation (black), prediction-based segmentation for Elastic Net predictions (gray) and ANN predictions (red). Green diamonds indicate the optimal internal matching rate from Section 3.5.1. Note that gray and red diamonds are close to each other and sometimes cover each other. © 2022 ACM

3.11 illustrates the volume-weighted internal matching rate for percentile groups of predictors, where percentile group 1 (100) contains the orders in the lowest (highest) 1% of a predictor across all test periods. For instance, if the internal matching rate of ANN is high (red) for the lower 20% of AvOrderVol_24h and low (blue) for the upper 20% of AvOrderVol_24h, this means that ANN achieves relatively large internal matching rates when the average order volume has decreased within the past 24 hours, and relatively small internal matching rates when the average order volume has increased instead.

Size-based segmentation shows characteristic behavior for small orders and negative changes in the average order volume and relative order size. Prediction-based segmentation achieves high (low) internal matching when changes in buy volume share and average order volume are negative (positive), which is consistent with our findings from Figure 3.7 that suggest lower internal matching rate given strong increases in the two variables. Correspondingly, negative changes in the relative order size are also associated with high internal matching rates. Since relative order size has the total trade volume as denominator, negative changes in the relative order size may indicate increasing total trade volume with a higher chance of internal order matches (see Appendix 3.D). Apart from that, however, the models fit less strongly to volatility measured by the high-low price range and to one-minute arithmetic returns. Overall, the size of an order is critical to its internal matching rate, regardless of the segmentation method. Orders executed on the exchange refer to relatively large orders in an above-average volatile market environment accompanied by a buy or sell surplus. When comparing ANN and Elastic Net, we notice that ANN internally matches more large orders than Elastic Net and shows a more pronounced relationship with the buy volume share.

3.6.3 Learning content

In order to examine what our model learns from the provided input data, we follow Jiang et al. (2022) and regress true and predicted labels on the predictors set. In total, we carry out

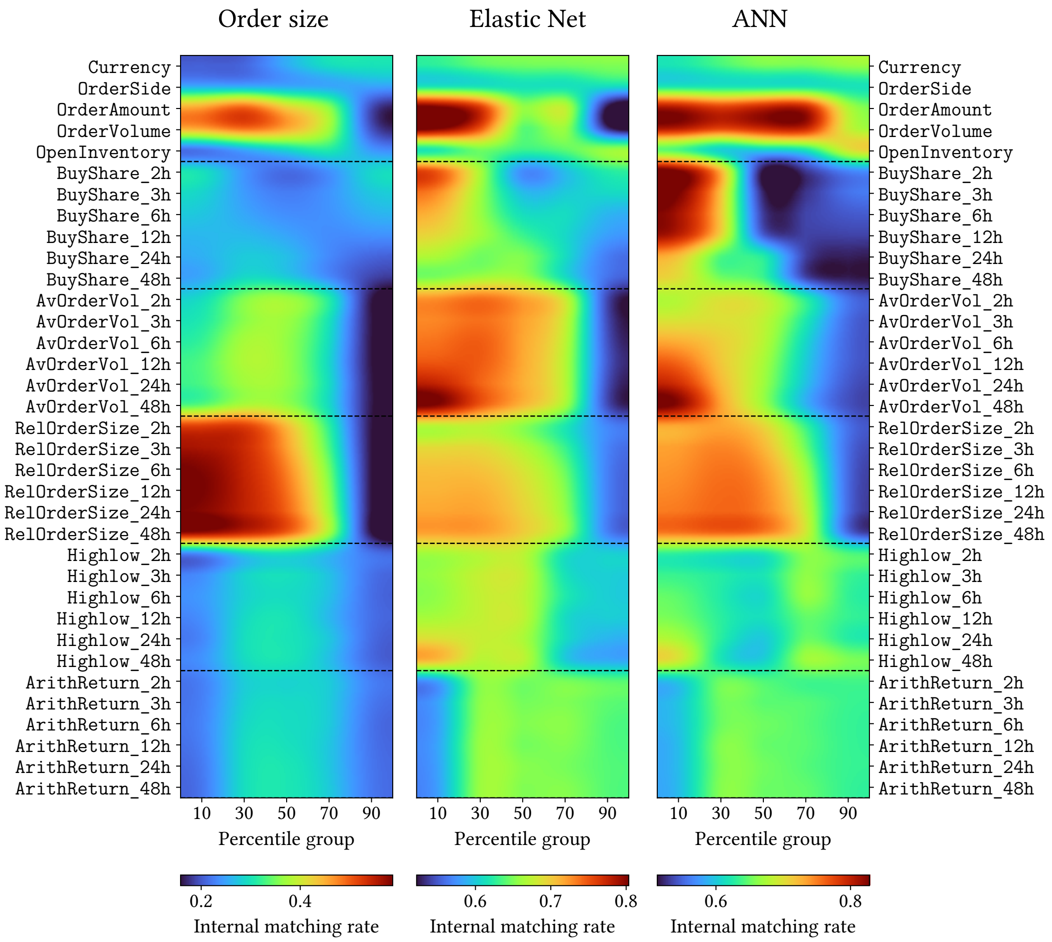

Figure 3.11: Internal matching rate for percentile groups of predictors. Predictors are named according to the considered sequence length, e.g., `AvOrderVol_6h` denotes the change in the average order volume between now and six hours ago. `Currency` represents BCH (groups 1-20), BTC (21-40), ETH (41-60), LTC (61-80) and XRP (81-100). `OrderSide` represents sell (groups 1-50) and buy (51-100) side. Percentile 1 (100) of `OpenInventory` demonstrate the 1% largest short (long) position. © 2022 ACM

four separate logistic regressions, each with the same regressors. Figure 3.12 illustrates the estimated coefficients with black (red) dots indicating (no) statistical significance. A positive (negative) coefficient means that an increase in that variable leads to a higher (lower) probability of order neutralization by future order flow. For instance, since larger orders are more difficult to neutralize than small orders, the coefficients for order volume are negative.

Consistent with Figure 3.11, the signs of the estimated coefficients for order volume, order amount, buy volume share, average order volume and relative order size are all negative. Increasing buy volume share and large order sizes lower the neutralization probability. Coefficients for high-low price range and arithmetic one-minute return are mixed. Regarding the coefficients for relative order size, we notice that long-term changes have more impact on internal matching than short-term changes. Both Elastic Net and ANN learn characteristics from the data similar to the direct approximation to optimal internal matching labels, represented by identical signs for most of the coefficients, although the estimated coefficients are larger for most predictors. The left column emphasizes that the two most relevant predictors for predicting optimal internal matches are order volume and the two-day change in the relative order size. Accordingly, a large (small) order submitted at a time where its two-day relative order size increases (decreases), e.g., due to a smaller (larger) total trade volume during the past two days, have a low (high) probability to be internally matched.

3.6.4 What drives segmentation performance?

This section aims to understand what circumstances make which segmentation approach perform well. For this purpose, we compare the one-day TPR^w and TNR^w with the daily means of the five variables buy volume share, average order volume, relative order size, high-low

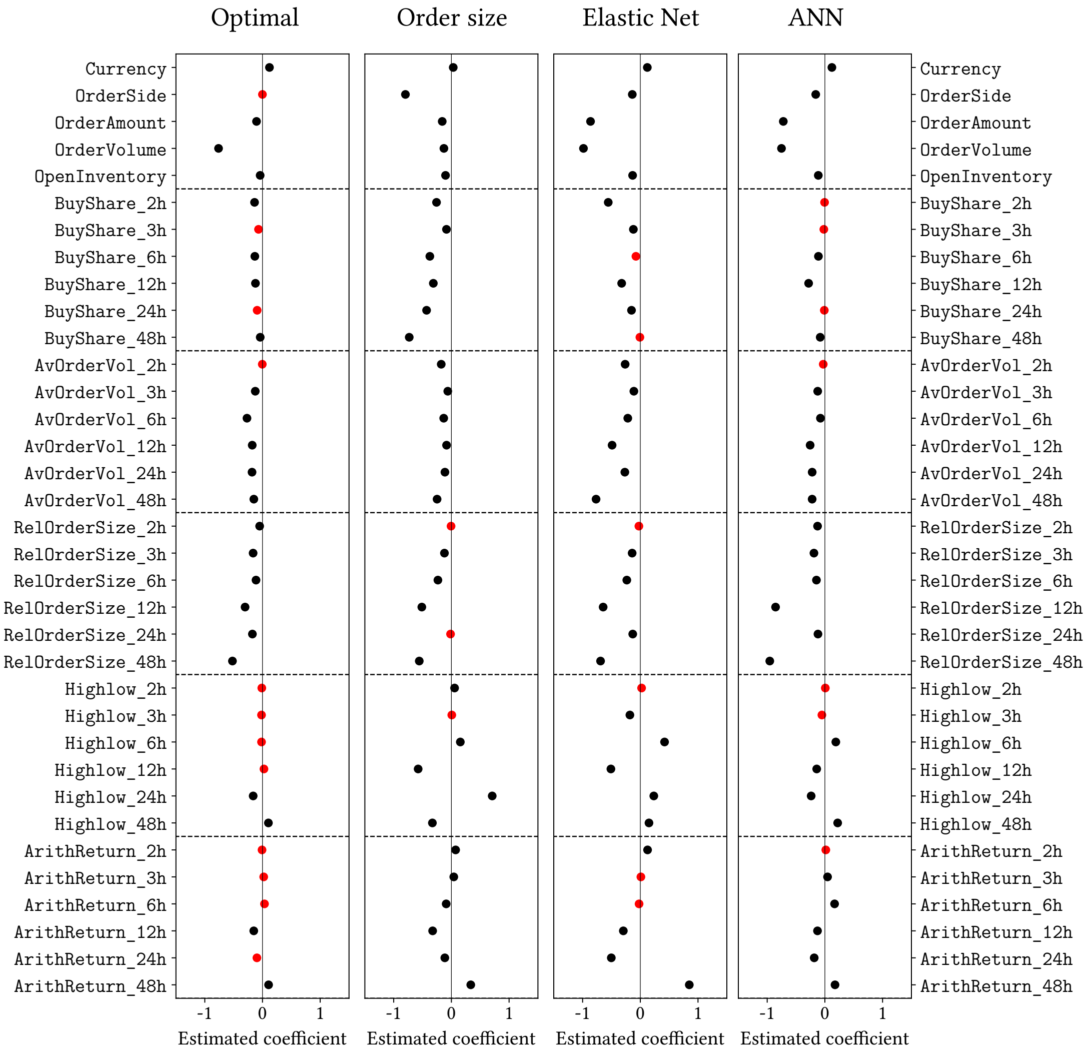

Figure 3.12: Estimated coefficients of logistic regression of optimal internal matching rate labels (first column), size-based and prediction-based labels on predictors during all test periods. Black dots indicate statistical significance at the 0.01 level, red dots indicate no significance. `Currency` is 0 for BCH, 0.25 for BTC, 0.50 for ETH, 0.75 for LTC and 1.00 for XRP. `OrderSide` takes `1' for buy and `0' for sell orders.

price range and arithmetic one-minute return. We group the test periods into ten groups, where decile group 1 covers the 10% days with the lowest TPR^w (TNR^w) and decile group 10 covers the 10% days with the highest TPR^w (TNR^w). Prior to the grouping, we standardize each variable to a mean of zero and a standard deviation of 1. The upper (lower) panel in Figure 3.13 illustrates the daily means of the five variables as functions of one-day TPR^w (TNR^w) decile groups. The charts can be interpreted as follows: For instance, under size-based segmentation, the graph for the average order volume has its maximum (minimum) at the first TPR^w (TNR^w) decile group. This means that the 10% days with the poorest inventory (exchange) classifications are the 10% days with the highest (lowest) average order volume.

Size-based segmentation delivers the best (worst) inventory classifications in a low-volatile (high-volatile) market environment with small (large) daily means of average order volume, relative order size, high-low price range, and buy volume share, associated with negative (positive) returns. Exchange classifications are poor when average order volume is small and volatility is low while prices are rising. In contrast, surprisingly, prediction-based segmentation provides the best inventory decisions in an above-average volatile environment, whereas prediction quality declines in a relatively low-volatile market environment. We cannot iden-

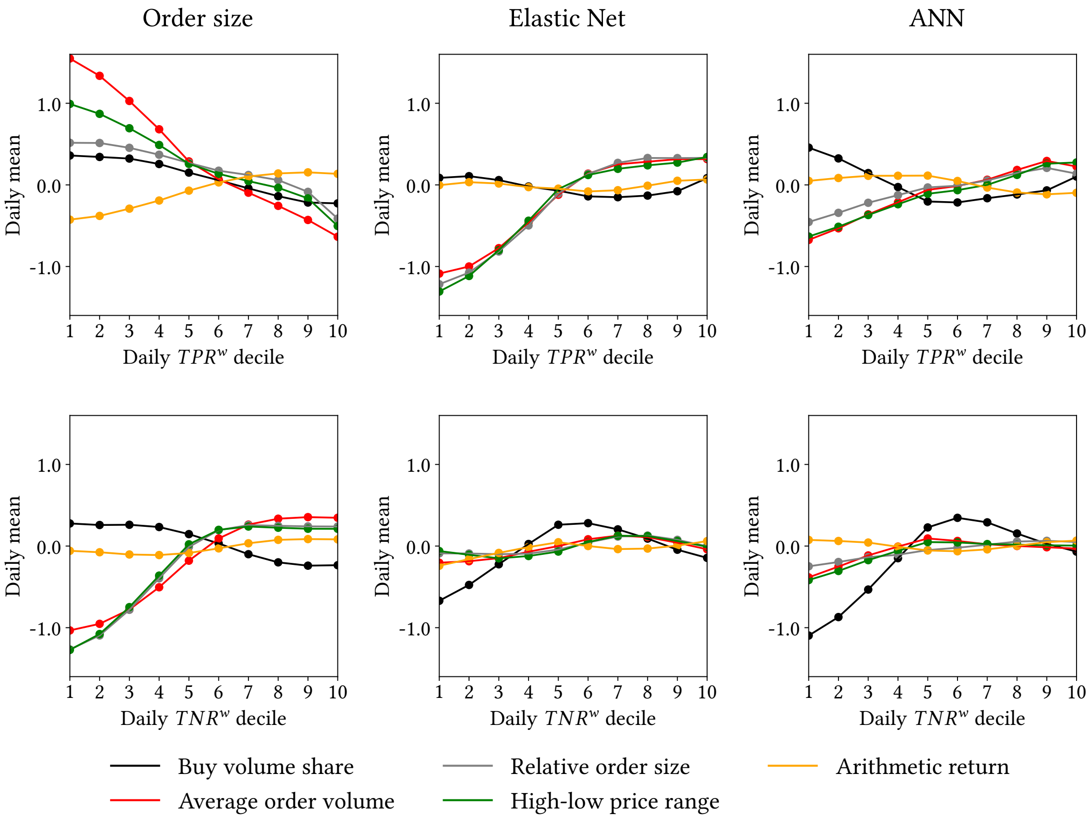

Figure 3.13: Market and order flow dynamics of daily TPR^w and TNR^w decile groups. Numbers report the average daily values of standardized market variables for daily TPR^w (top row) and TNR^w (bottom row) decile groups. The two left plots focus on size-based segmentation, whereas the middle (right) plot focus on prediction-based segmentation following Elastic Net (ANN) predictions. Arithmetic return and high-low price range are the volume-weighted averages from Coinbase, Kraken and Bitstamp. © 2022 ACM

tify a relationship with specific states of buy volume share and price changes. Prediction-based exchange classifications are weak when the buy volume share is low, whereas, apart from that, TNR^w shows no dominant relationship with market volatility or price changes.

In relating these results to our previous results from Figure 3.7 and Figure 3.11, it is first important to distinguish between statistical measures and internal matching, and to keep in mind that, e.g., a large TPR^w does not automatically imply high internal matching. One might wonder why in volatile markets the prediction quality for internal matches increases, while at the same time the chance of internal matching decreases. One possible reason is that we chose a probability threshold th optimized for volatile environments. Since trading volume is highest in volatile market environments, and we parameterized th based on *volume-weighted* quality measures, our parameterization is possibly less suited for low-volatile periods, potentially requiring a less strict probability threshold. We support this argument by testing a prediction model with th parameterized based on equally weighted quality measures, which reports inverse behavior of TPR^w and TNR^w deciles.

3.6.5 Analysis of cost savings

In the absence of internal order matching, investors always trade at the quoted price and thus pay the quoted spread for trade execution. In the presence of internal order matching, investors trade at either the quoted or the improved price. We measure investors' cost savings from internal order matching by the distance between the actual trade price and the quoted price based on the effective spread. We follow Bollen et al. (2004) and calculate the effective spread by (2 · |trade price − mid price|)/mid price, where the mid price is the average of the quoted ask and bid price, and the trade price of an internally matched order is the midpoint between mid price and quoted ask or bid price, respectively (see Section 3.3.1).

Table 3.2 reports the average volume-weighted effective spreads by segmentation scenario. The average quoted spread across all currencies is 0.17%[21], represented by the scenario without internal order matching ('None'). Internal matching by size-based segmentation leads to a 10% decrease in effective spreads to 0.15%, whereas Elastic Net and ANN reduce effective spreads by around 35% to 0.11%. Similar to Anolli and Petrella (2007), cost savings are

[21] In light of the historical bid-ask spreads illustrated in Appendix 3.C, 0.17% appears to be large. Appendix 3.B elaborates that this impression is due to the fact that spreads are larger in volatile market phases with large trade volumes and thus receive a larger weight when calculating volume-weighted spreads.

Currency	None	Order size	Elastic Net	ANN
BCH	0.25	0.20	0.16	0.15
BTC	0.13	0.12	0.08	0.08
ETH	0.17	0.15	0.11	0.10
LTC	0.23	0.18	0.14	0.14
XRP	0.23	0.20	0.14	0.14
All	0.17	0.15	0.11	0.11

Table 3.2: Average volume-weighted quoted ('None') and effective ('Order size', 'Elastic Net', 'ANN') spreads across all test periods. Numbers denote in percentage. We follow Bollen et al. (2004) and weight spreads once according to the volume of each individual trade, and once across exchanges according to the trade volume share of each exchange. © 2022 ACM

larger for less liquid pairs, e.g., 40% for BCH, and smaller for liquid pairs such as BTC and ETH. Thus, on average, retail investors can save more than one third of the quoted spread if their broker uses prediction-based internal order matching and passes half of the cost savings to customers (see Section 3.3.1). Accordingly, the broker also achieves cost savings of one third of the spread, so the total monetary value added is around 70% of the bid-ask spread. Nonetheless, this 70% can be split arbitrarily by the regulator or the participants, e.g., brokers in a competitive environment could provide even greater cost savings to their customers.

Finally, we evaluate whether our results are robust to possible regulatory changes with respect to order internalization in crypto trading. Following Section 3.2, one possible regulation would be that crypto brokers, similar to MiFID II, would only be allowed to internalize orders when meeting a minimum price improvement, and would have to send all orders for which they cannot meet this requirement to the exchange. For our analysis, however, we cannot simply adopt the MiFID II regulation because tick sizes in the crypto market are not as uniformly harmonized as they are under MiFID II (Lucas, 2020). For example, the tick size of BTC on most exchanges is €0.01, even though the price is in the five-digit range.

To this end, we measure price improvement in basis points (bp) of the volume-weighted mid price, i.e., given a minimum price improvement of 1 bp, an order about 1 BTC trading at a mid price of €30,000 must receive a price improvement of at least €3. With regard to MiFID II, a 1 bp price improvement for a stock trading at €100 is €0.01, identical with the MiFID II tick size regime. As we assume that customers and brokers share the cost savings equally, a 1 bp price improvement also implies 1 bp additional gain for the broker.

Figure 3.14 illustrates internal matching as a function of the minimum price improvement. At 1 bp, the broker could internally match 70% of the order volume, but would save relatively more in costs than the customer on most of these orders. The broker could also internally match more than 70% if she accepts that the customer saves relatively more costs. More than 60% of total order volume could receive a price improvement of more than 2.5 bp.

3.7 Conclusion

In this article, we introduce a new approach to internal order matching in crypto trading. To this end, we present an order segmentation methodology based on predicting whether or not

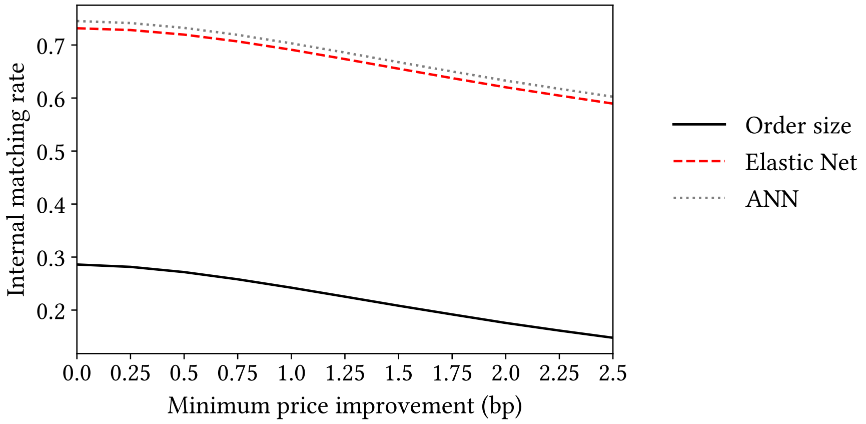

Figure 3.14: Volume-weighted internal matching under minimum price improvements between 0.0 and 2.5 basis points (bp) of the mid price. © 2022 ACM

an incoming order that the broker cannot immediately execute against orders in inventory will be neutralized by future order flow before its settlement date. We show that a broker using our model can triple internally matched volume compared to traditional segmentation, allowing both brokers and traders to realize cost savings of 35% of the quoted spread.

A limitation of our work is that we cannot assess the actual impact of increased internal matching on market quality. Our analyses assume a positive linear relationship between internal matching and cost savings. However, in equity trading, empirical studies question this relationship (Easley et al., 1996; Bessembinder and Kaufman, 1997; Chakravarty and Sarkar, 2002; Preece and Rosov, 2014). Future work should examine whether increased internal matching in crypto markets affects market quality and, if so, whether customers' cost savings could be eroded by widening spreads in the main market.

Overall, our model is robust to market dynamics and provides a data-driven solution for preferenced trading with crypto order flow, that realizes meaningful cost savings both for brokers and traders, even after accounting for common minimum price improvements.

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 https://www.tensorflow.org/.

Abdi, F. and Ranaldo, A. (2017), 'A Simple Estimation of Bid-Ask Spreads from Daily Close, High, and Low Prices', *Review of Financial Studies* **30**(12), 4437–4480. doi:10.1093/rfs/hhx084

Aggarwal, C. C. et al. (2018), 'Neural networks and deep learning', *Springer* **10**(978), 3. doi:10.1007/978-3-319-94463-0

Alexander, C., Choi, J., Park, H. and Sohn, S. (2020), 'BitMEX bitcoin derivatives: Price discovery, informational efficiency, and hedging effectiveness', *Journal of Futures Markets* **40**(1), 23–43. doi:10.1002/fut.22050

Algamal, Z. Y. and Lee, M. H. (2015), 'Applying penalized binary logistic regression with correlation based elastic net for variables selection', *Journal of Modern Applied Statistical Methods* **14**(1), 15. doi:10.22237/jmasm/1430453640

Anolli, M. and Petrella, G. (2007), 'Internalization in European Equity Markets Following the Adoption of the EU MiFID Directive', *Journal of Trading* **2**(2), 77–88. doi:10.3905/jot.2007.682142

Ante, L. (2020), 'Bitcoin transactions, information asymmetry and trading volume', *Quantitative Finance and Economics* **4**(3), 365–381. doi:10.3934/QFE.2020017

Barardehi, Y. H., Bernhardt, D., Da, Z. and Warachka, M. (2022), Institutional Liquidity Demand and the Internalization of Retail Order Flow: The Tail Does Not Wag the Dog, The Warwick Economics Research Paper Series (TWERPS), University of Warwick, Department of Economics. Available from https://warwick.ac.uk/fac/soc/economics/research/working papers/2022/twerp_1394_-_bernhardt.pdf.

Battalio, R., Greene, J. and Jennings, R. (1997), 'Do Competing Specialists and Preferencing Dealers Affect Market Quality?', *Review of Financial Studies* **10**(4), 969–993. doi:10.1093/rfs/10.4.969

Battalio, R. H. (1997), 'Third Market Broker-Dealers: Cost Competitors or Cream Skimmers?', *Journal of Finance* **52**(1), 341–352. doi:10.1111/j.1540-6261.1997.tb03819.x

Battalio, R. H. and Loughran, T. (2008), 'Does payment for order flow to your broker help or hurt you?', *Journal of Business Ethics* **80**(1), 37–44. doi:10.1007/s10551-007-9445-x

Battalio, R., Jennings, R. and Selway, J. (2001), 'The Relationship Among Market-Making Revenue, Payment for Order Flow, and Trading Costs for Market Orders', *Journal of Financial Services Research* **19**(1), 39–56. doi:10.1023/A:1011181317099

Bessembinder, H. and Kaufman, H. M. (1997), 'A cross-exchange comparison of execution costs and information flow for NYSE-listed stocks', *Journal of Financial Economics* **46**(3), 293–319. doi:10.1016/S0304-405X(97)00032-9

Bianchi, D., Babiak, M. and Dickerson, A. (2022), Trading volume and liquidity provision in cryptocurrency markets, Technical report. doi:10.1016/j.jbankfin.2022.106547

Boehmer, E., Jones, C. M., Zhang, X. and Zhang, X. (2021), 'Tracking Retail Investor Activity', *Journal of Finance* **76**(5), 2249–2305. doi:10.1111/jofi.13033

Bollen, N. P., Smith, T. and Whaley, R. E. (2004), 'Modeling the bid/ask spread: measuring the inventory-holding premium', *Journal of Financial Economics* **72**(1), 97–141. doi:10.1016/S0304-405X(03)00169-7

Brauneis, A., Mestel, R., Riordan, R. and Theissen, E. (2021), 'How to measure the liquidity of cryptocurrency markets?', *Journal of Banking & Finance* **124**, 106041. doi:10.1016/j.jbankfin.2020.106041

Brauneis, A., Mestel, R. and Theissen, E. (2021), 'What drives the liquidity of cryptocurrencies? A long-term analysis', *Finance Research Letters* **39**, 101537. doi:10.1016/j.frl.2020.101537

Brolley, M. and Cimon, D. A. (2020), 'Order-flow segmentation, liquidity, and price discovery: The role of latency delays', *Journal of Financial and Quantitative Analysis* **55**(8), 2555–2587. doi:10.1017/S002210901900067X

Brownlee, J. (2016), *Deep learning with Python: Develop deep learning models on Theano and TensorFlow using Keras*, Machine Learning Mastery. Available from https://machinelearningmastery.com/deep-learning-with-python/.

Chakravarty, S. (2001), 'Stealth-trading: Which traders' trades move stock prices?', *Journal of Financial Economics* **61**(2), 289–307. doi:10.1016/S0304-405X(01)00063-0

Chakravarty, S. and Sarkar, A. (2002), 'A model of broker's trading, with applications to order flow internalization', *Review of Financial Economics* **11**(1), 19–36. doi:10.1016/S1058-3300(01)00031-3

Challet, D., Chicheportiche, R., Lallouache, M. and Kassibrakis, S. (2018), 'Statistically validated lead-lag networks and inventory prediction in the foreign exchange market', *Advances in Complex Systems* **21**(08), 1850019. doi:10.1142/S0219525918500194

Chollet, F. et al. (2015), 'Keras', www.keras.io. Package available from https://github.com/fchollet/keras.

Comerton-Forde, C., Malinova, K. and Park, A. (2018), 'Regulating dark trading: Order flow segmentation and market quality', *Journal of Financial Economics* **130**(2), 347–366. doi:10.1016/j.jfineco.2018.07.002

Csáji, B. C. et al. (2001), 'Approximation with artificial neural networks', *Faculty of Sciences, Etvs Lornd University, Hungary* **24**(48), 7. M.Sc. Thesis. Available from https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=35272edb07ee4f0c9957eb0c631d2a74c24fe446.

Degryse, H., Van Achter, M. and Wuyts, G. (2022), 'Plumbing of securities markets: The impact of post-trade fees on trading and welfare', *Management Science* **68**(1), 635–653. doi:10.1287/mnsc.2020.3880

Diebold, F. X. and Mariano, R. S. (1995), 'Comparing Predictive Accuracy', *Journal of Business & Economic Statistics* **13**(3), 253–263. doi:10.1080/07350015.1995.10524599

Easley, D., Kiefer, N. M. and O'Hara, M. (1996), 'Cream-Skimming or Profit-Sharing? The Curious Role of Purchased Order Flow', *Journal of Finance* **51**(3), 811–833. doi:10.2307/2329223

Eldan, R. and Shamir, O. (2016), The power of depth for feedforward neural networks, *in* V. Feldman, A. Rakhlin and O. Shamir, eds, '29th Annual Conference on Learning Theory', Vol. 49 of *Proceedings of Machine Learning Research*, PMLR, Columbia University, New York, New York, USA, pp. 907–940. Available from https://proceedings.mlr.press/v49/eldan16.html.

Felder, C. and Seemüller, J. (2022), Intelligent Inventory Management for Cryptocurrency Brokers, *in* 'Proceedings of the Third ACM International Conference on AI in Finance', ICAIF '22, Association for Computing Machinery, New York, NY, USA, pp. 1–8. doi:10.1145/3533271.3561661

Feng, W., Wang, Y. and Zhang, Z. (2018), 'Informed trading in the Bitcoin market', *Finance Research Letters* **26**, 63–70. doi:10.1016/j.frl.2017.11.009

Fleming, M. and Nguyen, G. (2013), Order flow segmentation and the role of dark trading in the price discovery of u.s. treasury securities, Staff Report 624, Federal Reserve Bank of New York, New York, NY. Available from http://hdl.handle.net/10419/93629.

Fox, M. B., Glosten, L. and Rauterberg, G. (2019), *The New Stock Market: Law, Economics, and Policy*, Columbia University Press, New York Chichester, West Sussex. doi:10.7312/fox-18196

Francioni, R. and Schwartz, R. (2008), Dialog with Reto Francioni, *in* R. A. Schwartz, A. Colaninno and J. A. Byrne, eds, 'Competition in a Consolidating Environment', Springer US, Boston, MA, pp. 1–5. doi:10.1007/978-0-387-75943-2_1

Garriott, C. and Walton, A. (2018), 'Retail order flow segmentation', *Journal of Trading* **13**(3), 13–23. doi:10.3905/jot.2018.13.3.013

Géron, A. (2018), *Neural networks and deep learning*, O'Reilly Media. Available from https://www.oreilly.com/library/view/neural-networks-and/9781492037354/.

Grammig, J. and Theissen, E. (2012), 'Is Best Really BETTER? Internalization of Orders in an Open Limit Order Book', *Schmalenbach Business Review (sbr)* **64**(2), 82–100. doi:10.1007/BF03396891

Grundy, B. D. and McNichols, M. (2015), 'Trade and the Revelation of Information through Prices and Direct Disclosure', *Review of Financial Studies* **2**(4), 495–526. doi:10.1093/rfs/2.4.495

Gu, S., Kelly, B. T. and Xiu, D. (2020), 'Empirical Asset Pricing via Machine Learning', *Review of Financial Studies* **33**(5), 2223–2273. doi:10.1093/rfs/hhaa009

Hagerty, K. and McDonald, R. L. (1996), Brokerage, Market Fragmentation and Securities Market Regulation, *in* A. W. Lo, ed., 'The Industrial Organization and Regulation of the Securities Industry', National Bureau of Economic Research Conference Report series, University of Chicago Press, pp. 35–56. Available from https://www.nber.org/system/files/chapters/c8101/c8101.pdf.

Hansch, O., Naik, N. Y. and Viswanathan, S. (1999), 'Preferencing, Internalization, Best Execution, and Dealer Profits', *Journal of Finance* **54**(5), 1799–1828. doi:10.1111/0022-1082.00167

Hoerl, A. E. and Kennard, R. W. (1970), 'Ridge regression: Biased estimation for nonorthogonal problems', *Technometrics* **12**(1), 55–67. doi:10.2307/1271436

Ji, S., Kim, J. and Im, H. (2019), 'A Comparative Study of Bitcoin Price Prediction Using Deep Learning', *Mathematics* **7**(10). doi:10.3390/math7100898

Jiang, J., Kelly, B. T. and Xiu, D. (2022), '(Re-) Imag (in) ing Price Trends', *Journal of Finance* **forthcoming**. Available at SSRN 3756587. doi:10.2139/ssrn.3756587

Kelley, E. K. and Tetlock, P. C. (2013), 'How Wise Are Crowds? Insights from Retail Orders and Stock Returns', *Journal of Finance* **68**(3), 1229–1265. doi:10.1111/jofi.12028

Kim, O. and Verrecchia, R. E. (1991), 'Market reaction to anticipated announcements', *Journal of Financial Economics* **30**(2), 273–309. doi:10.1016/0304-405X(91)90033-G

Kingma, D. P. and Ba, J. (2015), Adam: A method for stochastic optimization, *in* '3rd International Conference on Learning Representations, ICLR May 7-9, 2015', ICLR, San Diego, CA, USA. doi:10.48550/arxiv.1412.6980

Klein, T., Pham Thu, H. and Walther, T. (2018), 'Bitcoin is not the New Gold – A comparison of volatility, correlation, and portfolio performance', *International Review of Financial Analysis* **59**, 105–116. doi:10.1016/j.irfa.2018.07.010

Kumpan, C. (2006), 'Carrot and Stick – The EU's Response to New Securities Trading Systems', *European Company and Financial Law Review* **3**(4), 383–407. doi:10.1515/ECFR.2006.017

Kwan, A., Masulis, R. and McInish, T. H. (2015), 'Trading rules, competition for order flow and market fragmentation', *Journal of Financial Economics* **115**(2), 330–348. doi:10.1016/j.jfineco.2014.09.010

Larrymore, N. and Murphy, A. (2009), 'Internalization and market quality: An empirical investigation', *Journal of Financial Research* **32**(3), 337–363. doi:10.1111/j.1475-6803.2009.01253.x

LeCun, Y. A., Bottou, L., Orr, G. B. and Müller, K.-R. (2012), Efficient BackProp, *in* 'Neural networks: Tricks of the trade', Springer, pp. 9–48. doi:10.1007/978-3-642-35289-8_3

Linnainmaa, J. (2010), 'Do limit orders alter inferences about investor performance and behavior?', *Journal of Finance* **65**(4), 1473–1506. doi:10.1111/j.1540-6261.2010.01576.x

Lucas, I. (2020), *Quantifying Systematic Internalisers' Activity: Their Share in the Equity Market Structure and Role in the Price Discovery Process*, Autorité des Marchés Financiers. Available from https://www.amf-france.org/sites/institutionnel/files/2020-06/202005_etude_internalisateurs_integrale_va.pdf.

Macey, J. R. and O'Hara, M. (1997), 'The Law and Economics of Best Execution', *Journal of Financial Intermediation* **6**(3), 188–223. doi:10.1006/jfin.1997.0219

Makarov, I. and Schoar, A. (2020), 'Trading and arbitrage in cryptocurrency markets', *Journal of Financial Economics* **135**(2), 293–319. doi:10.1016/j.jfineco.2019.07.001

Malinova, K. (2012), 'Why allow internalization?', *Available at SSRN* **2120136**. Working paper, unpublished. doi:10.2139/ssrn.2120136

Martens, M. and van Dijk, D. (2007), 'Measuring volatility with the realized range', *Journal of Econometrics* **138**(1), 181–207. 50th Anniversary Econometric Institute. doi:10.1016/j.jeconom.2006.05.019

Masters, T. (1993), Designing Feedforward Network Architectures, *in* T. Masters, ed., 'Practical Neural Network Recipies in C++', Morgan Kaufmann, San Francisco, pp. 173–185. doi:10.1016/B978-0-08-051433-8.50015-X

Nair, V. and Hinton, G. E. (2010), Rectified Linear Units Improve Restricted Boltzmann Machines, *in* 'Proceedings of the 27th International Conference on International Conference on Machine Learning', ICML'10, Omnipress, Madison, WI, USA, pp. 807–814. Available from https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf.

Nielsen, M. A. (2015), *Neural networks and deep learning*, Vol. 25, Determination Press. Available from http://neuralnetworksanddeeplearning.com/.

Paquin, D. (2020), 'Coinbase Pro API'. Available from https://github.com/danpaquin/coinbasepro-python.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830. Available from https://dl.acm.org/doi/10.5555/1953048.2078195.

Peterson, M. A. and Sirri, E. R. (2003), 'Order Preferencing and Market Quality on U.S. Equity Exchanges', *Review of Financial Studies* **16**(2), 385–415. doi:10.1093/rfs/hhg009

Preece, R. (2012), *Dark Pools, Internalization, and Equity Market Quality*, CFA Institute. Available from https://www.cfainstitute.org/en/advocacy/policy-positions/dark-pools-internalization-and-equity-market-quality.

Preece, R. and Rosov, S. (2014), 'Dark Trading and Equity Market Quality', *Financial Analysts Journal* **70**(6), 33–48. doi:10.2469/faj.v70.n6.2

Scaillet, O., Treccani, A. and Trevisan, C. (2018), 'High-Frequency Jump Analysis of the Bitcoin Market', *Journal of Financial Econometrics* **18**(2), 209–232. doi:10.1093/jjfinec/nby013

Schwartz, R., Cormack, M., Domowitz, I., Federspiel, F., Francioni, R., Mogavero, D. and Ryan, M. (2005), How Best to Integrate the Order Flow, *in* R. A. Schwartz, J. A. Byrne and A. Colaninno, eds, 'Coping with Institutional Order Flow', Springer US, Boston, MA, pp. 59–83. doi:10.1007/0-387-25881-7_3

SEC (U.S. Securities and Exchange Commission) (2013), Trade Execution: What Every Investor Should Know, *in* 'Investor Publications'. Available from https://www.sec.gov/reportspubs/investor-publications/investorpubstradexec.

Shen, D., Li, X. and Zhang, W. (2017), 'Baidu news coverage and its impacts on order imbalance and large-size trade of chinese stocks', *Finance Research Letters* **23**, 210–216. doi:10.1016/j.frl.2017.06.008

Silantyev, E. (2019), 'Order flow analysis of cryptocurrency markets', *Digital Finance* **1**(1), 191–218. doi:10.1007/s42521-019-00007-w

Smith, S. L., Elsen, E. and De, S. (2020), On the Generalization Benefit of Noise in Stochastic Gradient Descent, *in* 'Proceedings of the 37th International Conference on Machine Learning', ICML'20, PMLR, pp. 9058–9067. Available from https://dl.acm.org/doi/10.5555/3524938.3525778.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting', *Journal of Machine Learning Research* **15**(56), 1929–1958. Available from https://dl.acm.org/doi/10.5555/2627435.2670313.

Thomadakis, A. (2022), 'Shortening the settlement cycle: Why Europe should not wait too long to introduce T+1', *ECMI Commentary no 77 | January 2022* . Available from https://www.ecmi.eu/publications/commentaries/shortening-settlement-cycle-why-europe-should-not-wait-too-long-introduce.

Tibshirani, R. (1996), 'Regression Shrinkage and Selection via the Lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* **58**(1), 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x

Wang, J.-N., Liu, H.-C., Zhang, S. and Hsu, Y.-T. (2021), 'How does the informed trading impact bitcoin returns and volatility?', *Applied Economics* **53**(28), 3223–3233. doi:10.1080/00036846.2020.1814944

Zou, H. and Hastie, T. (2005), 'Regularization and variable selection via the elastic net', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **67**(2), 301–320. doi:10.1111/j.1467-9868.2005.00503.x

Zou, Q., Xie, S., Lin, Z., Wu, M. and Ju, Y. (2016), 'Finding the best classification threshold in imbalanced classification', *Big Data Research* **5**, 2–8. Big data analytics and applications. doi:10.1016/j.bdr.2015.12.001

Appendix

3.A Distribution of profits and losses on internal order matches

This section analyzes whether different order prices of internally matched orders would impact our revenue differently than price changes between confirming and executing an order at an exchange. To this end, we compare the distribution of profits per order when we internally match all contrary orders as described in Section 3.5.1 with the distribution of profits per order when we route all orders to the exchange. As approximation to the broker's execution price at the exchange for an order received in t, we consider the market price in $t + 1$.

We calculate the profit on internal order matches as follows: For each buy order in the order history, we weight the order price by the order volume internally matched and subtract from it the volume-weighted order price of the sell order(s) against which we internally match that order. For instance, if we receive a buy order about 1 BTC with an order price of €40, 000 and internally match 90% of the total order amount against a sell order about 0.75 BTC with an order price of €40,050 and 30% of a sell order about 0.50 BTC with an order price of €40,025, the nominal profit on the internally matched part of the buy order is $90\% \cdot €40,000 - (100\% \cdot 0.75 \cdot €40,050 + 30\% \cdot 0.50 \cdot €40,025) = €36,000 - €36,041.25 = -€41.25$. Finally, we scale the nominal profit by the initial volume-weighted order price of the buy order, i.e., $-€41.25 \div (90\% \cdot €40,000) = -0.11\%$. Hence, a positive profit indicates that we received more money from the buyer than we spent to the seller(s), whereas a negative profit indicates that we spent more for buying than we received from selling the asset.

Correspondingly, we define the profit on an order routed to the exchange by the percentage difference between the volume-weighted order price and the volume-weighted execution

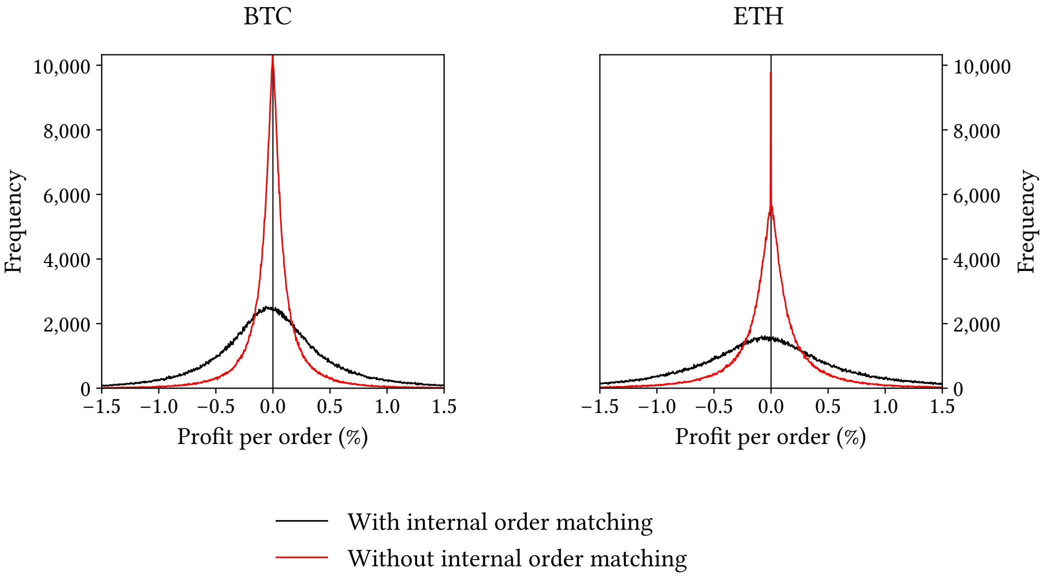

Figure 3.15: Frequency distribution of profits per order when internally matching all contrary orders (black) and when routing all orders to the exchange (red) in the range $[-1.5\%, 1.5\%]$ for BTC and ETH. We calculate profits according to Section 3.A.

price at the exchange. Continuing the example from above, if we execute the other 10% of the buy order at the exchange at a price of €40, 005, we generated a percentage profit of (10% · €40, 000 − 10% · €40, 005) ÷ €40, 000 = −0.01%. Hence, the number indicates the impact of price changes between confirming and executing orders on our revenue. This calculation also considers the close-out of open positions when internal order matching was not possible within $t + 2$ despite we routed the order to inventory. Note that we have to calculate profits on internal matches only for buy orders, as this already covers internally matched sell orders.

Figure 3.15 shows the frequency distribution of profits per order for the two scenarios. We find that profits fluctuate more when matching orders internally than when routing all orders to the exchange, although both scenarios show an equal distribution of profits and losses. This means, for example, that in each of the two scenarios a profit of 0.5% occurs about as frequently as a profit of −0.5%, with the absolute frequency being higher in the internal matching scenario than in the scenario without internal matching. When we pass all orders to the exchange, the price differences are smaller compared to the internal matching scenario, resulting in fewer large losses and profits, and we have more frequent very small profits and losses, respectively. We conclude that the impact of price differences in internally matched orders on the broker's profit is on average zero and thus similar to the impact of differences between order prices and execution prices on the exchange. Analogously, the graph also conveys that close-outs of the position in $t + 2$ can also be equally both profitable and loss-making. On this basis, we can disregard these price effects in a cross-sectional analysis.

3.B Daily order volume

Figure 3.16 illustrates the daily average order volume across the observation period. The chart emphasizes that the first half of 2021 accounts for the largest part of order volume in the data set, with BTC reporting the largest order volume. When transferring Figure 3.16 to Figure 3.17 in Appendix 3.C, we find that bid-ask spreads are large (small) on days with high (low) trade volume, which explains why volume-weighted spreads are larger than equal-weighted spreads (see Section 3.6.5). However, we still use volume-weighted spreads in order to demonstrate total internalization-driven cost savings measured by the order volume.

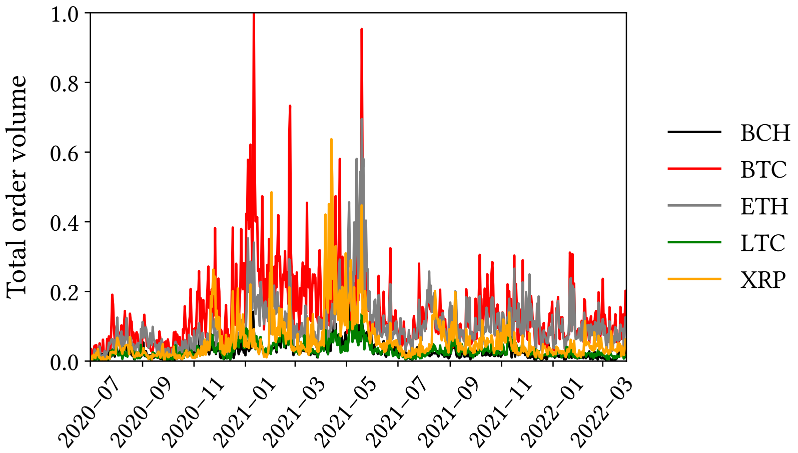

Figure 3.16: Daily total order volume by currency pair across the entire observation period. We normalize order volume in the range [0, 1] for data confidentiality purposes and aggregate buy and sell orders.

3.C Historical bid-ask spreads

We estimate bid-ask spreads according to the Abdi and Ranaldo (2017) estimator presented in equation (3.1). Figure 3.17 illustrates the average volume-weighted daily estimated bid-ask spreads by exchange. Bid-ask spreads are similar across all three exchanges and the lowest for the BTC-EUR and ETH-EUR pairs. Overall, the estimates are smaller compared to Brauneis, Mestel and Theissen (2021), potentially indicating that spreads narrowed throughout the past

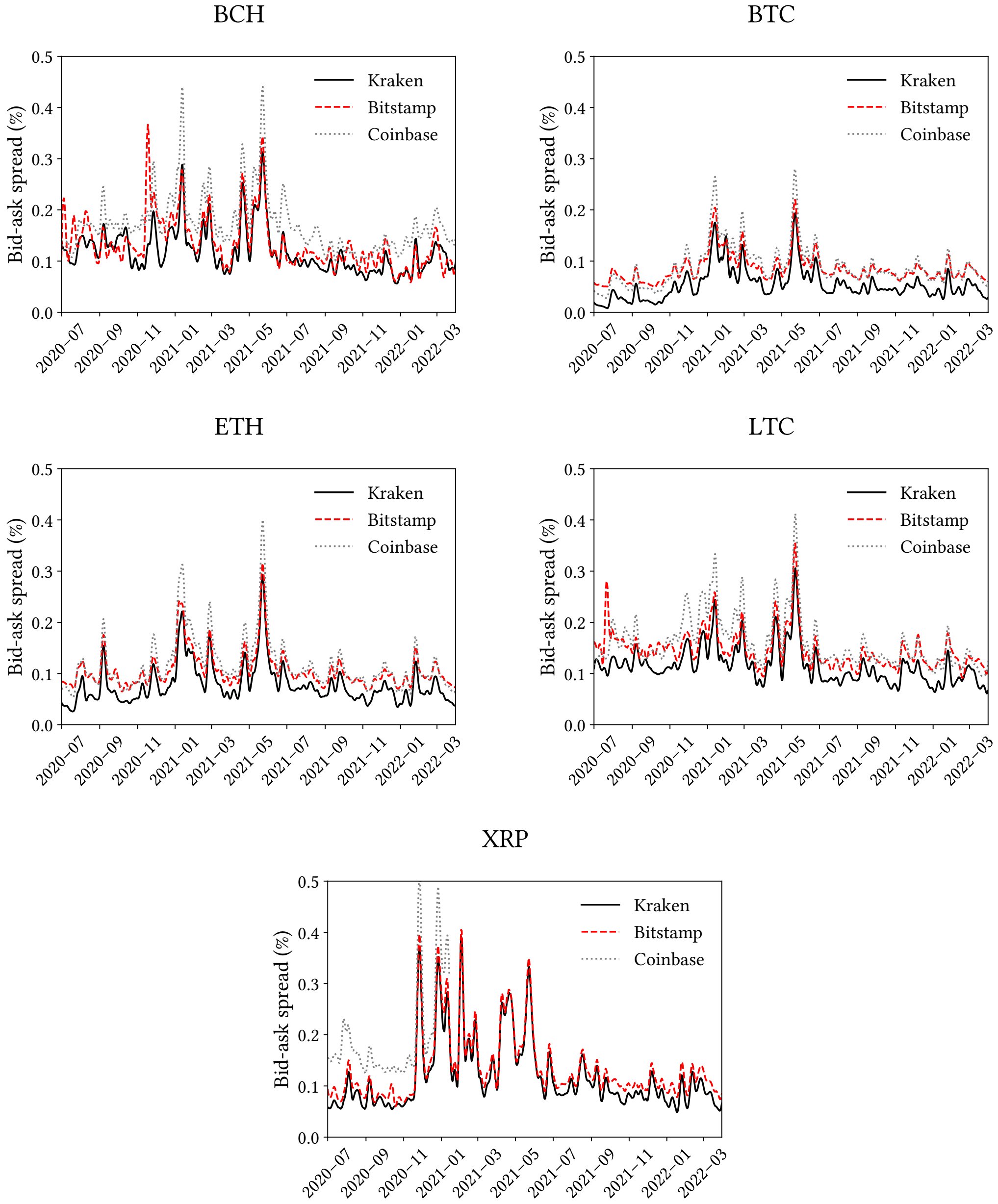

Figure 3.17: Average daily volume-weighted bid-ask spreads between July 01, 2020 and March 30, 2022. Reported bid-ask spreads base on the Abdi and Ranaldo (2017) estimator (equation (3.1)). We could not retrieve OHLCV data for XRP from Coinbase starting from 2021.

two years. In Section 3.4, we weight bid-ask spreads based on the share of daily trade volume between the three exchanges: For instance, if Kraken has a trade volume of €10,000 and an average daily spread of 10 basis points, whereas Coinbase and Bitstamp each have a trade volume of €5,000 and an average daily spread of 5 basis points, the volume-weighted average daily bid-ask spread across all three exchanges is 7.5 basis points.

3.D Market and order flow dynamics before and after order arrival (continued)

Figure 3.18 continues the analysis of Figure 3.7 and reports the hourly average total trade volume (top left), standard deviation (top right), and bid-ask spread (bottom) for internally matched orders and orders routed to the exchange. We calculate the total trade volume as the sum of the nominal trade volume in Euro across the exchanges of Bitstamp, Coinbase and Kraken. Standard deviation is the one-hour standard deviation of one-minute log returns, where log returns are the trade volume-weighted log returns of mid prices at Bitstamp, Coinbase and Kraken. Lastly, the bid-ask spread is the trade volume-weighted bid-ask spread based on the Abdi and Ranaldo (2017) estimator across the three exchanges (equation (3.1)).

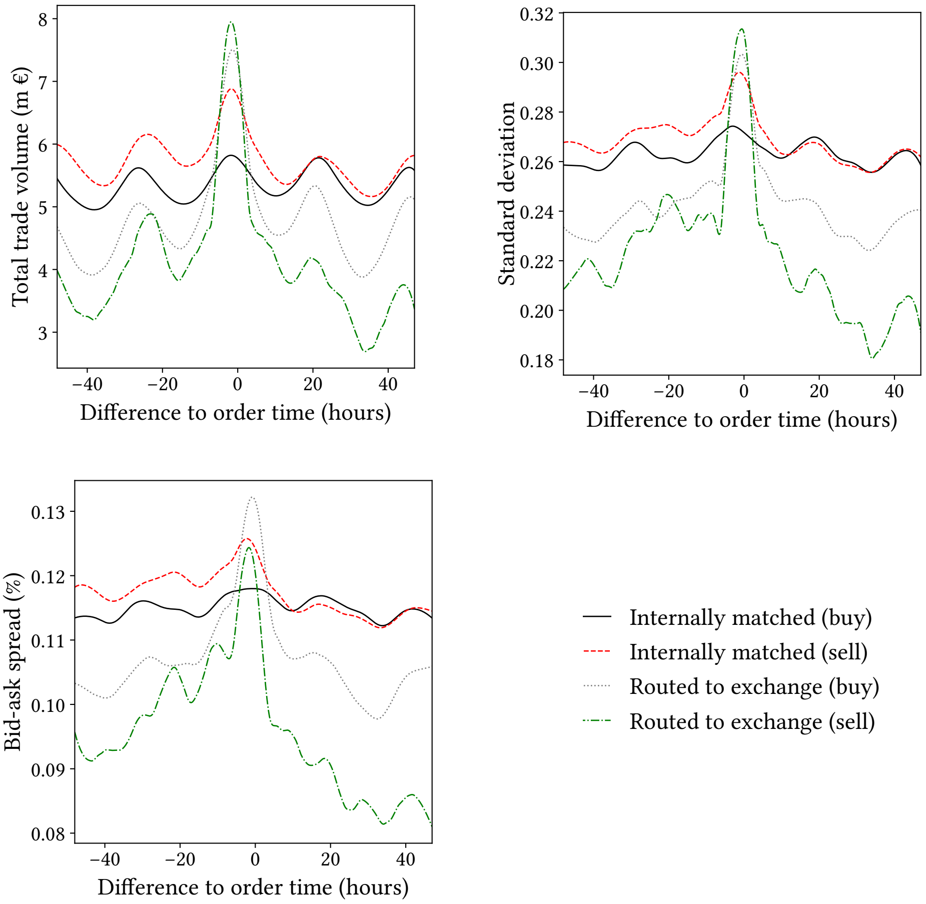

Figure 3.18: Market and order flow dynamics before and after order arrival (continued). Internal order matches base on the optimal internal matching rate determined in Section 3.5.1. Reported values are equal-weighted averages within each currency pair and volume-weighted averages across pairs.

3.E Sensitivity of internal matching to inventory constraints

As we assume a discretionary inventory limit of €100,000 in Section 3.3.2, we also should analyze whether a different choice would impact our order execution performance. To this end, Figure 3.19 illustrates the internal matching rate for inventory limits between €100,000 and €1,000,000. Internal matching generally is positively related with the inventory limit, as a larger inventory limit allows us to store more orders in inventory and thus match more orders internally. A shift from €100,000 to €200,000 leads to a 1 percentage point increase in internal matching rate in the case of size-based segmentation and a 0.2 percentage point increase in the case of ANN prediction-based segmentation. Consequently, the inventory limit may limit internal matching to some extent, but the impact on trade execution decisions is small compared to other factors such as the eligible order size or the segmentation methodology.

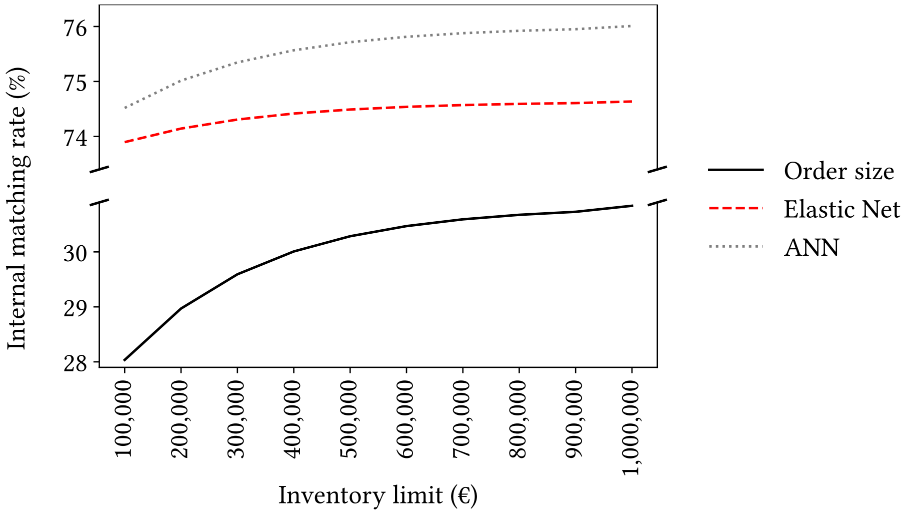

Figure 3.19: Sensitivity of internal matching to the inventory constraint in the range €100,000 to €1,000,000. Reported numbers are volume-weighted by trade volume. An inventory limit of €300,000 allows us to hold open positions of up to €300,000.

4 Prediction-based Limit Order Trading

Christopher Felder

Abstract

Managing the trade-off between volume and margin is among the most fundamental challenges for dealers in a securities market. We attempt to overcome this trade-off by incorporating predictions for buyer- and seller-initiated trades when submitting limit orders. Using the Avellaneda–Stoikov model as an example, we show how dealers can adjust quotes to predictions and thereby capture larger spreads at constant volume. Simulations on historical limit order book data illustrate that our model allows dealers to both increase market making revenues through trade flow-optimized positioning in the order book and reduce adverse selection cost through preempted adverse price movements.

Keywords: Sequence learning, Limit order trading, Avellaneda–Stoikov model, Adverse price selection, Cryptocurrency.

JEL classification: C45, C53, C55, C63, D49, G17.

4.1 Introduction

Intuitively, dealers aim to repeatedly earn the 'spread' between buy and sell prices by simultaneously buying and selling an asset without accumulating a large net position. As a result, dealers face a trade-off between trade volume and margin: Dealers with tight spreads trade frequently but at unfavorable prices, while wide spreads lead to high margins but less volume (Guéant, 2017). Additionally, following Guilbaud and Pham (2013), dealers face inventory risk arising from the stochastic behavior of their net position (Avellaneda and Stoikov, 2008), execution risk arising from limit orders not being executed or only partially executed (Handa and Schwartz, 1996; Kühn and Stroh, 2010), and adverse selection risk arising from trading with informed traders (Frey and Grammig, 2008). Thus, much of the research literature has been concerned with the question of whether there is an optimal strategy for posting limit orders (Ahuja et al., 2017).

Avellaneda and Stoikov (2008) propose a stochastic optimal control model in which the dealer's reservation price is a function of the net position: When the dealer is long (short), the reservation price is below (above) the mid price to incentivize execution of sell (buy) orders and return to a neutral inventory, thereby allowing the dealer to manage inventory risk. Numerous studies discuss extensions of the model, see Ahuja et al. (2017), Aydoğan et al. (2022), Bayraktar and Ludkovski (2014), Cartea and Jaimungal (2013), Cartea and Jaimungal (2015), Cartea et al. (2014), Fodra and Labadie (2012), Guéant (2017), Guéant et al. (2013) or Guilbaud and Pham (2013)—to name a few. Like Avellaneda and Stoikov (2008), most of these studies assume that market orders follow a stochastic process. In contrast, Cao et al. (2009), Cont et al. (2014), Dixon (2018*a*), Kercheval and Zhang (2015), Tsantekidis et al. (2017), Tsantekidis et al. (2020), Zhang et al. (2019) and Zheng et al. (2013) present evidence of mid price predictability from the limit order book (LOB). Moreover, Hirschey (2021) finds that high-frequency traders anticipate specific order flow trends, while Fodra and Labadie (2012) shows that high-frequency traders can improve their Profit & Loss (P&L) with directional bets.

Motivated by these discussions, we introduce an approach to integrate trade flow forecasting into limit order trading. We model buyer- and seller-initiated trades using a Recurrent Neural Network (RNN)—a powerful sequence learner well suited for modeling the high-frequency dynamics of an LOB (Dixon, 2018*b*)—that processes tick-level LOB and trade data to predict average 24-hour percentile ranks of buyer- and seller-initiated trade sizes during the next five seconds, respectively. For example, a 90% prediction for buyer-initiated trades means that the average size of buyer-initiated trades over the next five seconds is greater than 90% of all buyer-initiated trades over the last 24 hours.

Using the Avellaneda–Stoikov model as an example, we adjust limit order prices to these predictions in two ways: First, if our model predicts large (small) trades, we increase (decrease) the depth of orders in the LOB, i.e., post orders deeper (closer) in the LOB. For example, when predicting large buy orders and small sell orders, we increase both the ask and bid price to maximize the potentially captured spread on the ask side and increase the chance of being consumed on the bid side. Second, we shift the reservation price to protect against the anticipated price movement, which we derive from the imbalance between the predictions for buy and sell trade sizes. Hence, when anticipating a price increase (decline), we shift the reservation price up (down) to protect from being adversely selected by informed buyers (sellers) and participate in the price increase (decline) through long (short) exposure, regardless of our current net position. In order to control by how much we adjust prices in response to predictions, we scale price adjustments using sensitivity parameters, where high (low) sensitivity indicates relatively larger (smaller) adjustments to Avellaneda–Stoikov prices.

We test our model on historical LOBs of the BTC-USD and ETH-USD pairs and arrive at the following findings: The RNN improves prediction quality by about 40% over a standard feed-forward neural network and performs best when previous trade flow is balanced. Moreover, there is an empirical optimum of sensitivity to predictions, which states that we should be twice as sensitive to predictions when adjusting order depth to trade sizes than when adjusting reservation price to adverse price movements. Our limit order model significantly improves P&L over a wide range of test periods compared to the Avellaneda–Stoikov model, e.g., by up to 7% for one-hour test periods, but this improvement also comes with higher risk and depends on market dynamics. As two main contributors to the P&L surplus, we identify higher market making revenues due to larger spreads earned especially when the order book is less liquid, and reduced adverse selection costs by preempting adverse price movements, with the cost reduction being proportional to the magnitude of the adverse price movement. Accordingly, our model provides an effective approach to both overcoming the trade-off between volume and margin and reducing adverse selection cost by taking trade flow forecasts into account when deciding whether we can afford a wider spread to increase profitability.

The remainder of the article is structured as follows: Section 4.2 introduces to the basics of LOBs and high-frequency trading. Section 4.3 presents our data set and Section 4.4 derives a sequence prediction model for high-frequency trade flow. In Section 4.5, we illustrate how dealers can adjust order prices to predictions for limit order trading. Finally, Section 4.6 presents the results from the trading simulation and Section 4.7 concludes.

4.2 High-frequency trading in a limit order book

4.2.1 Limit order book (LOB)

A limit order[22] is an order to buy or sell a specified amount of an asset at a particular price. Upon submission, the trader must specify the price limit, amount, and side of the trade. If execution is not possible at the time of arrival, the order queues in the LOB at the respective price level. The LOB is an aggregation of all open limit orders awaiting execution by incoming market orders (Haider et al., 2022) and can be viewed as a continuous double auction where traders compete by submitting bid and ask orders (Lim and Gorse, 2018). The LOB represents market demand and supply at any point in time, "a cross-section of orders to execute at various price levels away from the market price" (Dixon, 2018*a*). LOBs typically follow a price-time priority, with orders queued according to their price level and arrival time. For more information on LOBs, we refer to Gould et al. (2013) and Abergel et al. (2016).

The counterpart of limit orders are market orders, which serve impatient traders by instantaneous order execution. A market order has no price limit and executes immediately against open orders in the LOB, starting with the best price level and, if necessary, continuing with subsequent price levels until the exchange's matching engine has been able to match the total order volume. This process drives price discovery and carries a price risk for the impatient trader, as the final execution price could be worse than the best market price currently displayed. Buy (sell) market orders execute against the best ask (bid) price, i.e., the 'spread' is a cost for immediate execution of the trade.

[22] Section 4.2.1 describes limit order books following Gould et al. (2013), Abergel et al. (2016), Lim and Gorse (2018), Spooner et al. (2018), Dixon (2018*a*) and Haider et al. (2022).

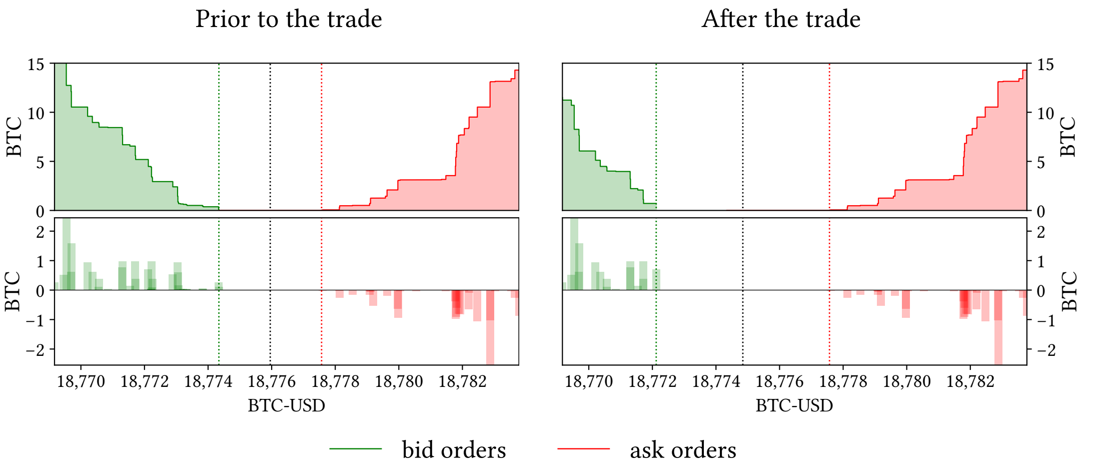

Figure 4.1: Impact of a large market sell order on the LOB. We take LOB snapshots of the BTC-USD pair from Coinbase at `2022-09-26 07:08:00.0516 UTC` (left) and `2022-09-26 07:08:00.0563 UTC` (right) and represent the LOB in aggregate (top) and plain (bottom) format. Between the two snapshots, a sell market order about 4.68 BTC arrived and matched with open buy orders. Dashed lines represent the best bid (green) and best ask price (red).

The mid price marks the equal-weighted midpoint between the best bid and best ask price. Changes in the mid price indicate that either traders cancel their open limit orders or market orders arrive and execute against open limit orders at either the bid or the ask side. Figure 4.1 illustrates the impact of a large seller-initiated trade on the LOB using the Coinbase BTC-USD pair as an example. The top (bottom) charts show the LOB in aggregate (plain) format prior to the arrival of the market order (left) and after the order has arrived and fully executed against open limit orders (right). The dashed lines represent the respective price levels of the best bid (green) and best ask price (red). As the arriving market order is larger than open limit orders at the top of the bid side, the order consumes multiple price levels between $18,774 and $18,772, causing the mid price to fall from $18,776 to less than $18,775, indicated by the $2 decrease of the best bid and the widened spread.

4.2.2 High-frequency trading

Traditionally, the role of dealers was taken by 'official' market makers who had signed a contract with an exchange or listed company (Guéant, 2017). Today, most modern exchanges are organized as order-driven markets (Guilbaud and Pham, 2013), where any participant can become a market maker. Menkveld (2013) and Guéant (2017) refer to this category as 'new' market makers, which can be any type of high-frequency trader. With a share of up to 80% (Hagströmer and Nordén, 2013), market making activities dominate high-frequency limit order trading over other high-frequency strategies such as statistical arbitrage.

The objective of dealers is to provide liquidity to the market by continuously offering buy and sell prices through the LOB without accumulating large net positions (Menkveld, 2013) in order to minimize market risk. The prices offered therefore should take into account a dealer's net position: Symmetrically placing orders around the mid price, e.g., at the top of the order book, risks adverse selection, as downward or upward trends could lead to a disproportionate number of filled buy or sell orders, respectively, resulting in a large net position that the dealer would have to offset at unfavorable prices. Thus, market making initially is non-directional and keeps inventory close to zero (Menkveld, 2013).

A common approach in the literature[23] for determining a dealer's optimal buy and sell price is to treat market making as a stochastic optimal control problem introduced by Ho and Stoll (1981), who analyzes the optimal price of a single stock for a monopolistic dealer. Avellaneda and Stoikov (2008) extend this approach and formulate the problem mathematically for a dealer who optimizes the expected utility of the terminal profit. Consistent with empirical indications presented in Gabaix et al. (2006), Gopikrishnan et al. (2000), Maslov and Mills (2001), Potters and Bouchaud (2003), and Weber and Rosenow (2005) (Ahuja et al., 2017), the authors assume that market order flow follows a Poisson process with rate $Ae^{-\kappa\delta}$, where δ is the distance between order price and mid price, and A and κ represent order book liquidity. The authors solve the stochastic control problem using a Hamilton–Jacobi–Bellman equation with an asymptotic expansion for a small inventory position (Fushimi et al., 2018) and derive the indifference or reservation price r_{AS} and the optimal bid-ask spread $\delta^a + \delta^b$ as

$$r_{AS} = s - q\gamma\sigma^2 (T - t) \tag{4.1}$$

$$\delta^a + \delta^b = \gamma\sigma^2 (T - t) + \frac{2}{\gamma}\ln\left(1 + \frac{\gamma}{\kappa}\right) \tag{4.2}$$

where s denotes the mid price, q depicts the dealer's inventory net position, σ is the standard deviation of the mid price, γ is an inventory risk aversion parameter, and 'AS' indicates the Avellaneda–Stoikov model. Accordingly, the dealer's optimal ask price is $r^a_{AS} = r_{AS} + \frac{1}{2}\left(\delta^a + \delta^b\right)$ and the optimal bid price is $r^b_{AS} = r_{AS} - \frac{1}{2}\left(\delta^a + \delta^b\right)$. Following equation (4.2), the optimal bid-ask spread is a decreasing function of time: Close to terminal ($t \longrightarrow T$), $\delta^a + \delta^b$ converges towards $\frac{2}{\gamma}\ln\left(1 + \frac{\gamma}{\kappa}\right)$ and thus becomes symmetrical to the mid price, increasing the dealer's chance of finishing the trading day with a zero net position. The ask and bid order's depth in the order book is defined by $\delta^a = r^a_{AS} - s$ and $\delta^b = s - r^b_{AS}$, respectively.

A large literature discusses extensions of the Avellaneda–Stoikov model: Guéant et al. (2013) include inventory constraints and apply the model to longer time scales, Cartea and Jaimungal (2015) regard the market impact on the mid price, and Aydoğan et al. (2022) take stochastic volatility, drift effects, and market influences into account. Studies focusing on remodeling the market order flow process include Bayraktar and Ludkovski (2014), who consider the liquidation problem and model order arrivals as a function of the liquidation price, Cartea et al. (2014), who integrate stronger market order dynamics, and Ahuja et al. (2017) and Fodra and Labadie (2012), both of which replace the Brownian process with a mean-reverting process such as the Ornstein-Uhlenbeck process. Correspondingly, Chakraborty and Kearns (2011) demonstrate the general profitability of market making algorithms on mean-reverting time series. Ahuja et al. (2017) find that, far from terminal, the optimal bid and ask prices are constant. Inspired from Kühn and Stroh (2010), Guilbaud and Pham (2013) consider market orders and limit orders together with stochastic spreads modeled by a continuous-time Markov chain, which is, according to Guéant (2017), "one of the only market making models really well suited to stocks" as it does not assume discrete nature of prices and takes time-priority

[23] This paragraph describes the Avellaneda–Stoikov model according to Guéant (2016), Fushimi et al. (2018), Ahuja et al. (2017), Hummingbot (2021), Hummingbot (2023) and Avellaneda and Stoikov (2008).

into account. Cartea and Jaimungal (2013) embed a discrete-time Markov chain and try minimizing the dealer's adverse selection risk. Bringing many of the aforementioned studies together, Guéant (2017) generalizes the proposed extensions by equipping the Avellaneda–Stoikov model with general intensity functions and extending it to the multi-asset case. More recently, reinforcement learning has led to improvements over stochastic optimal control formulations—see Spooner et al. (2018), Lim and Gorse (2018), Ganesh et al. (2019), Guéant and Manziuk (2019), Gašperov and Kostanjčar (2021), or Schnaubelt (2022).

On the question of whether market order flow *is* stochastic, numerous studies examine the relationship between LOBs and price discovery. Yet, Dixon (2018*a*) acknowledges that there is "no consensus on the extent to which limit order books convey predictive information". Studies arguing that limit orders are not informative include Glosten (1994) and Seppi (1997). In contrast, Cao et al. (2009), Zheng et al. (2013), Cont et al. (2014), Kercheval and Zhang (2015), Palguna and Pollak (2016), Tsantekidis et al. (2017), Dixon (2018*a*), Zhang et al. (2019), or Tsantekidis et al. (2020) argue that LOBs may contain predictive information. For more studies, we refer to the survey of Zaznov et al. (2022). In terms of high-frequency trading, Dixon (2018*a*) outlines that price predictions may serve to reduce the likelihood of adverse price selection through preempted adverse price movements, Li et al. (2014) design a limit order trading algorithm based on sentiment predictions from LOB and market news, and Haider et al. (2022) implement price predictions into a reinforcement learning pricing strategy.

4.3 Data

Schnaubelt (2022) outlines that cryptocurrency exchanges "represent a well suited test environment" for limit order trading models, as—in contrast to global stock exchanges—data are easily and free to access and "major cryptocurrency exchanges operate like established limit order exchanges and exhibit very similar stylized facts" (Makarov and Schoar, 2020; Schnaubelt et al., 2019). As one of the most liquid centralized cryptocurrency exchanges with more than 100 million traders as of October 2022 (Coinbase, 2022), we therefore consider the Coinbase exchange as a suitable example to test our limit order model. Accordingly, we focus on the two most liquid pairs, 'BTC-USD' and 'ETH-USD'.

Since tick-level data from Coinbase is not downloadable, we record LOB and trade messages between `2022-08-16 05:37:27 UTC` and `2022-10-05 16:07:40 UTC` via WebSocket using `cbpro` (Paquin, 2020), the Python client for the Coinbase Pro API. We receive batches of level-2 messages every 50 milliseconds through the `level2_batch` channel and price updates on each match through the `ticker` channel. After recording, we replay the messages to reconstruct LOBs and trade flows, aggregate both data sets into one-second intervals and cut the LOBs behind the top 50 price levels on each side of the book. To obtain comparable results between the two pairs, we index all prices at 100 with the first recorded price.

For the split into training, validation and test periods, we follow the reasoning of Section 3.4 and use n-fold cross validation that divides the entire period into n equal-sized sub-intervals, for each of which we train independent models. We choose an overlapping split of five days in length, of which the first three days are for training (60%), the fourth day is for validation (20%), and the last day is for test (20%). As illustrated in Figure 4.2, the overlap is four days, i.e., we shift the sub-intervals by one day, yielding a total of 45 sub-intervals, allowing us to test each day once with an independent model. As the shift length equals the length of the test period, each trade occurs exactly once across all test periods. Considering that we use our

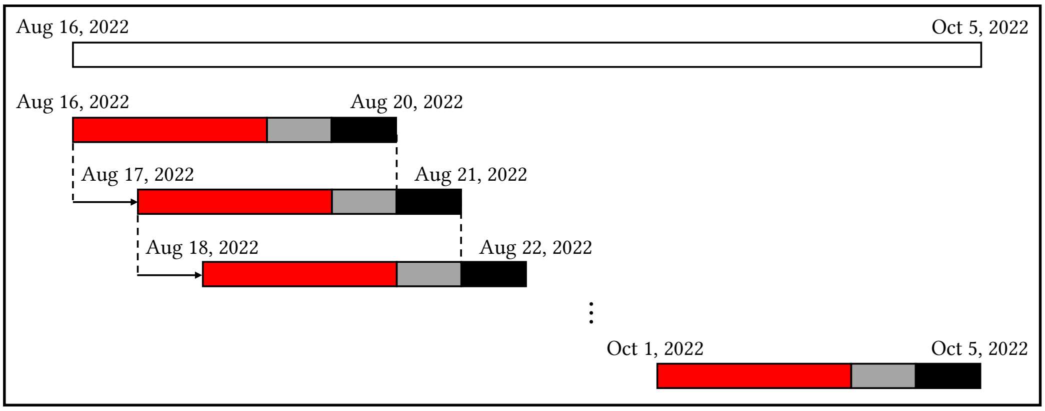

Figure 4.2: Division of the observation period for n-fold cross validation into overlapping five-days sub-intervals, consisting of three days (60%) training (red), one day (20%) validation (gray) and one day (20%) test (black). The overlap is four days.

model for simulations of continuously trading cryptocurrency markets, it is probably most realistic to evaluate performance for one gap-less period rather than for individual days.

4.4 High-frequency trade flow prediction

4.4.1 24-hour trade size percentile ranks

We model buyer- and seller-initiated trade flows using the percentile ranks of trade sizes in the frequency distribution of all trade sizes over the past 24 hours. Analogous to Roscoe (1975), we calculate the 24-hour percentile rank R of a trade size as the share of trade sizes that are less than that trade size, indicating whether the trade tends to be large or small relative to other trades from the previous 24 hours.

We define the 24-hour percentile rank R of a particular trade size by

$$R = \frac{M + 0.5 \cdot F}{N} \tag{4.3}$$

where M is the cumulative frequency of all trade sizes less than or equal that trade size, F is the frequency of that trade size in the distribution of all trade sizes, and N is the number of all trades occurred within the past 24 hours. Thus, R is defined in the range $[0, 1]$, with $R > 0.5$ ($R < 0.5$) indicating a relatively large (small) trade. Considering the differences between bid and ask market order flow, we calculate R separately for buyer-initiated trades (R^b) and seller-initiated trades (R^a). For example, if a buyer-initiated trade about 1.5 BTC has a percentile rank of $R^b = 0.90$, this means that 90% of all buyer-initiated trades over the past 24 hours are smaller than 1.5 BTC. To obtain comparable results across all sub-intervals, we calculate percentile ranks prior to separating training, validation and test periods. Figure 4.9 illustrates that, although we calculate percentiles based on sequences along the time axis, the percentile ranks are evenly distributed across test periods.

Together, R^b and R^a convey two pieces of information: First, R^b and R^a indicate whether trades are more likely to consume one or multiple price levels. Figure 4.18 in Appendix 4.A illustrates that, empirically, trades with a low (high) trade size percentile rank consume one (multiple) price level(s), e.g., trade sizes with $R = 90\%$ consume five price levels on average. Second, the difference between R^b and R^a indicates the imbalance between the buyer- and seller-initiated trade flow. For example, a small R^b combined with a large R^a may represent a sell surplus. In contrast to the widely used definition (Albers et al., 2021; Silantyev, 2019), we follow Sadighian (2019) and define the trade flow imbalance TFI in the range $[-1, 1]$ by

$$TFI = \frac{R^b - R^a}{R^b + R^a} \tag{4.4}$$

where $TFI < 0$ ($TFI > 0$) indicates a sell (buy) surplus. Consistent with Chan (2017) and Silantyev (2019), we consider TFI as an approximation of the price movement, with $TFI > 0$ ($TFI < 0$) indicating a price increase (decline).

4.4.2 Sequence learning for limit order trading

We present a limit order trading model that adjusts buy and sell prices according to the predicted 24-hour percentile rank of future seller- and buyer-initiated trade sizes. The prediction task is a regression task with R^b and R^a as target variables. As Silantyev (2019) shows that trade flow imbalance is auto-correlated and Dixon (2018*a*) states that high-frequency price "increments are neither independent nor stationary and depend on the state of the order book", we assume that the observations of R^b and R^a, respectively, are auto-correlated and not i.i.d. Hence, we define our prediction task as a sequence prediction model that uses lagged values of R^b and R^a sequences to predict subsequent values of that sequences. A sequence learner proven successful in modeling high-frequency dynamics (Dixon, 2018*b*) is the RNN. While some researchers (Tsantekidis et al., 2020; Zhang et al., 2019) prefer Long Short-Term Memory (LSTM) over RNN, Dixon (2018*b*) outlines that for high-frequency LOB modeling, "short-term memory is adequate", i.e., the auto-correlation lowers for larger lags and the vanishing gradient problem of RNNs (Bengio et al., 1994; Hochreiter, 1998; Hochreiter et al., 2001) therefore is not an issue, and hypothesizes that "there is little to no benefit in using an LSTM" (see also Section 2.2). Given these findings as well as the higher computation time of LSTMs, which may affect the monetary benefit in a high-frequency environment, we follow Dixon (2018*b*) and model the market order flow with RNN networks.

Introduced by Rumelhart et al. (1986*a*,*b*), the RNN network[24] is an artificial neural network architecture capable to extract temporally encoded information from time-series data by using feedback connections with neurons (units) of the same or previous layers. A hidden state h transfers information contained in cell outputs across time steps and facilitates stepwise processing of sequences to connect information between time steps. Hidden state loops

[24] We describe RNNs following Aggarwal et al. (2018), Olah (2015), Géron (2018), and Brownlee (2017).

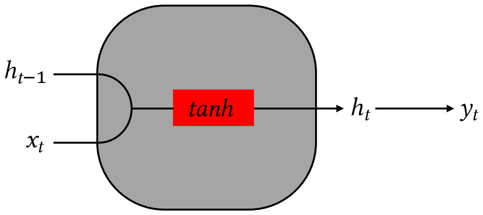

Figure 4.3: Structure of an RNN cell following Olah (2015) with input x, hidden state h, cell output y and a hyperbolic tangent activation function at time t.

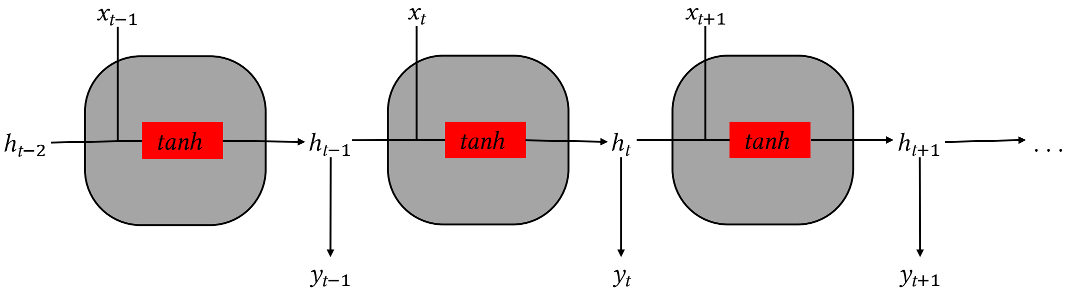

Figure 4.4: Unfolded RNN cell with a direct feedback connection and a hyperbolic tangent activation function between $t-1$ and $t+1$ following Olah (2015) and LeCun et al. (2015).

within the network have a memory function and allow past information to persist. Figure 4.3 illustrates the structure of an RNN cell where the vector of hidden states h_t is the output of a cell at time t and serves as the cell memory for the next time step. The cell passes the current hidden state to the next time step of the sequence to process them again together with the current input.

At each time step $t = 1, ..., T$, the RNN cell processes both h_{t-1} and the vector of current elements of the input sequence, x_t, to generate the new hidden state $h_t = tanh(w_x x_t + w_h h_{t-1} + b_h)$ through a $tanh$ activation function, where b_h is the bias vector and w_x (w_h) is a weight matrix determining which elements of x_t (h_{t-1}) to keep and which to remove. This procedure ensures that the network takes into account the important knowledge acquired in previous time steps when processing any new information. The model output at time t is $y_t = w_y h_t + b_y$. In our case, y_t is a 2×1 vector containing R_t^b and R_t^b. The ultimate model output of the final hidden state is $y_T = w_y \cdot h_T + b_y$. Figure 4.4 illustrates an unfolded RNN cell with a direct feedback connection, revealing a chain structure where each chain block represents one time step. In that sense, the RNN is the basis of the LSTM (Section 2.2), which adds a second type of state and replaces the $tanh$ layer with three different gates with unique tasks.

4.4.3 Predictors of trade flow

Since we use aggregated one-second data (see Section 4.3), we can process predictors and target variables as time series where one time step represents one second. Besides the sequences of R_t^b and R_t^a, we additionally include sequences of predictors proposed by the price prediction literature: First, we include the price level distance, which is the distance of each of the 10 best price levels on both sides of the order book from the mid price, expressed as a percentage of the mid price (Tsantekidis et al., 2020). Second, we include the cumulative depth size, which reports the cumulative sum of open limit order sizes at each of the 10 best price levels (Tsantekidis et al., 2020). Both variables yield a vector of 20 unique values for each time step. Third, we include the order flow imbalance OFI (Cont et al., 2014; Palguna and Pollak, 2013), which we calculate as the difference between the cumulative bid and ask size at the tenth price levels on each side, divided by their sum (Cartea et al., 2015). For example, if the cumulative size at the 10th price level on the bid side is 5 ETH and on the ask side 4 ETH, it is $OFI = (5-4)/(5+4) = 0.11$. Finally, we include the percentage change in the mid price (Tsantekidis et al., 2020; Zaznov et al., 2022) and the distance between the highest and lowest trade price of all matches within each one-second interval (Zaznov et al., 2022).

In total, given that we process n observations per batch, each time step includes five predictors with shape $n \times 1$ and two predictors with shape $n \times 20$, i.e., we process 45 predictors per time step. A key parameter in sequence learning is the sequence length or lookback period, which determines the number of lags we provide to our model in order to deliver a prediction. Motivated by Dixon (2018*a*), who chooses a sequence length of 10 for tick-level data, we use a sequence length of 10 seconds, i.e., we process 45 predictors over 10 time steps to deliver a prediction for R^b and R^a. The final vector of input variables thus has shape $n \times 45 \times 10$. Prior to training, we standardize the input data by subtracting the mean and dividing by the standard deviation across all sub-intervals (LeCun et al., 2012).

4.4.4 Network configuration

We use a three-layer network architecture with two visible layers and one hidden layer. The number of units in the first and last layer equals the dimension of the vector of input and output variables, respectively. The activation function for the two output units is the sigmoid function, which squashes predictions into the range $[0, 1]$, corresponding to the range for which R^b and R^a are defined, thereby facilitating learning. Determining the number of units in the hidden layer is not as straightforward. Out of architectures tested with 16, 32, 64, 128, and 256 hidden units, we choose 64 hidden units, resulting in the lowest mean squared prediction error for the validation sets. As activation function we select $tanh$, scaling values between $[-1, 1]$ and thereby avoiding dropping information by scaling towards 0. The depth of the network corresponds to the chosen sequence length: Each time step unfolds the RNN module once and increases the depth by one. We do not consider deep RNNs (Pascanu et al., 2014) that extend the hidden state loop operation across multiple layers, as Dixon (2018*b*) presents promising results when using a standard RNN for modeling limit order book dynamics.

Each training step, we minimize the mean squared prediction error, a loss function suited for regression problems in modeling high-frequency dynamics (Dixon et al., 2019), across batches of 8 random samples using the stochastic gradient descent (Graves, 2013; Rojas, 2013) with an initial learning rate of 0.01. As we have a large amount of possibly irrelevant or correlated data, a major task in training our network is regularization. We regularize training in multiple ways: First, stochastic gradient descent introduces randomness to the optimization routine and improves generalization. Second, we consider a stateless RNN, i.e., memory does not transfer between sequences of different batches and persist for one batch. Third, we allow the network to shuffle samples within batches. Fourth, we apply 20% dropout (Srivastava et al., 2014) to the input layer and, as proposed by Graves (2013), to the non-recurrent connections in the hidden layer. Finally, we use early stopping and halt training when there has been no improvement for the validation set for two epochs. Corresponding to Sections 2.2.2 and 3.5.4, we train RNNs with `keras` (Chollet et al., 2015) and Google's `TensorFlow` (Abadi et al., 2015).

4.5 Prediction-based limit order trading

4.5.1 Prediction-based order depth

We propose order depth as a function of the predicted trade size percentile ranks, $\widehat{R}^b$ and $\widehat{R}^a$. Assumed that we make predictions one period ahead and use these predictions to adjust r^a_{AS} and r^b_{AS}, respectively, the function should fulfill two requirements: First, if $\widehat{R}^b < \frac{1}{2}$ ($\widehat{R}^a < \frac{1}{2}$), i.e., predicting small buy (sell) orders, we should post sell (buy) orders closer to the top of the book to increase the hit probability, whereas if $R^b > \frac{1}{2}$ ($R^a > \frac{1}{2}$), we should post sell (buy) orders deeper in the book to maximize the captured spread. Second, the spread adjustment should be proportional to the percentile rank, i.e., the adjustment for $\widehat{R}^b = 0.9$ should be larger than that for $\widehat{R}^b = 0.6$. Based on the Avellaneda–Stoikov model, we define the spread-adjusted ask price r^a_{SP} and the spread-adjusted bid price r^b_{SP} by

$$r^a_{SP} = r_{AS} + \frac{1}{2}\left(1 + \frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\right)\left(\delta^a + \delta^b\right) \tag{4.5}$$

$$r^b_{SP} = r_{AS} - \frac{1}{2}\left(1 + \frac{1}{\alpha_1}\left(2\widehat{R}^a - 1\right)\right)\left(\delta^a + \delta^b\right) \tag{4.6}$$

where $\alpha_1 \geq 0$ is a scaling parameter that determines how sensitive order depth is to predicted percentile ranks, i.e., by how much we widen or tighten the spread in response to a prediction. As equation (4.5) can be rewritten by $r^a_{SP} = r^a_{AS} + \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)$, we simply add a weighted version of $\delta^a + \delta^b$ to r^a_{AS}. The terms $\frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)$ and $\frac{1}{\alpha_1}\left(2\widehat{R}^a - 1\right)$ are positive for $\widehat{R}^b > \frac{1}{2}$ and $\widehat{R}^a > \frac{1}{2}$, respectively, as we wish to post orders deeper in the LOB, zero for $\widehat{R}^b = \frac{1}{2}$ and $\widehat{R}^a = \frac{1}{2}$ as predicted orders are neither large nor small and we are indifferent between widening or tightening spreads, and negative for $\widehat{R}^b < \frac{1}{2}$ and $\widehat{R}^a < \frac{1}{2}$ as we wish to narrow the spread. For $\alpha_1 = 1$, the terms are defined in the range $[-1, 1]$, implying that the maximum spread adjustment ($\widehat{R}^b, \widehat{R}^a = 1$) is $r^a_{SP} - r^a_{AS} = \frac{1}{2}\left(\delta^a + \delta^b\right)$ and $r^b_{AS} - r^b_{SP} = \frac{1}{2}\left(\delta^a + \delta^b\right)$, i.e., the order's depth in the order book is twice the depth given by the Avellaneda–Stoikov model, and the minimum spread adjustment ($\widehat{R}^b, \widehat{R}^a = 0$) is $r^a_{SP} - r^a_{AS} = -\frac{1}{2}\left(\delta^a + \delta^b\right)$ and $r^b_{AS} - r^b_{SP} = -\frac{1}{2}\left(\delta^a + \delta^b\right)$, i.e., $r^a_{SP}, r^b_{SP} = r_{AS}$. Smaller (larger) values for α_1 are associated with more (less) impact of predictions on the depth.

Figure 4.5 illustrates $r^a_{SP} - r^a_{AS}$ (left) and $r^b_{SP} - r^b_{AS}$ (right) as functions of α_1 and $\widehat{R}^b$ and $\widehat{R}^a$, respectively. Both charts emphasize that small (large) values for α_1 are associated with strong (little) changes to the quotes as order depth reacts more (less) strongly to predictions. If

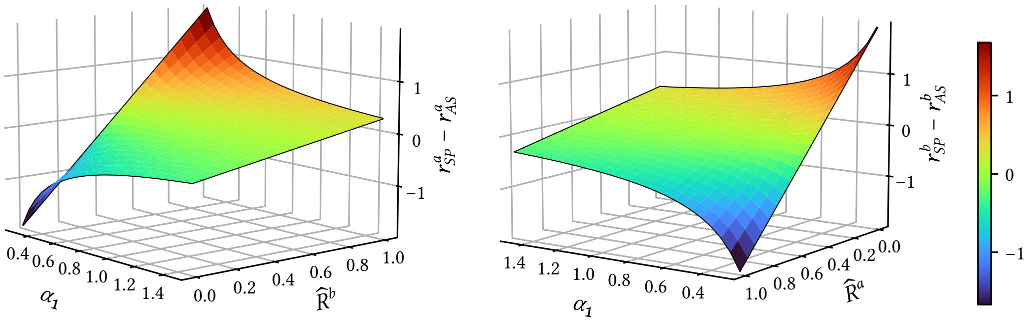

Figure 4.5: Prediction-based spread adjustments in the range $\alpha_1 = [0.25, 1.5]$ and $\widehat{R}^b, \widehat{R}^a = [0, 1]$. We measure spread adjustments by the difference between r^a_{SP} and r^a_{AS} (left) and difference between r^b_{SP} and r^b_{AS} (right plot). For illustration purposes, we use the following parameters: $\gamma = 0.25$, $\sigma = 0.3$, $T = 1$, $t = 0.3$, $\kappa = 2$, $q = 0$. Note that the axis origin in the left (right) chart is on the left (right).

$\widehat{R}^b > \frac{1}{2}$, we increase the ask price to exploit a larger spread from arriving large buy orders, and lower the ask price if $\widehat{R}^b < \frac{1}{2}$ to increase the chance of being consumed by small buy orders. Correspondingly, we decrease the bid price if $\widehat{R}^a > \frac{1}{2}$ and increase it if $\widehat{R}^a < \frac{1}{2}$. One may interpret our extension as a way to reduce adverse selection cost: Informed trades typically are larger than uninformed trades (Chakravarty, 2001; Linnainmaa, 2010). By increasing spreads when predicting large orders, we potentially protect against adverse selection by informed traders placing large orders (Sandås, 2001).

4.5.2 Prediction-based reservation price

Adverse price selection occurs when better informed traders pick off our limit orders on one side of the order book (Cartea and Jaimungal, 2013), resulting in an adverse price movement for our limit orders on the other side, which now queue away from the inside market (Dixon, 2018*a*). Following Dixon (2018*a*), one approach "potentially reducing the likelihood of adverse price movement" between the fill of a sell and the fill of a buy order is predicting price movements, thereby preempting unfavorable execution of orders. For example, Cartea and Jaimungal (2013) simulates an informed dealer using a hidden Markov model and reports reduced adverse selection costs. If we expect a mid price increase (decline), we may increase the reservation price in order to avoid disproportionate execution of sell (buy) orders and hold (close) an open long position or close (hold) an open short position. Thus, incorporating directional forecasts can help controlling both inventory risk and adverse selection risk.

We propose the reservation price as a function of the predicted trade flow imbalance, $\widehat{TFI}$, given by $\widehat{TFI} = \left(\widehat{R}^b - \widehat{R}^a\right)\left(\widehat{R}^b + \widehat{R}^a\right)^{-1}$, analogous to equation (4.4). This function should fulfill two requirements: First, if $\widehat{TFI} > 0$, we should increase the reservation price in order to avoid adverse filling of sell orders and incentivize long exposure for participating in the price increase, whereas if $\widehat{TFI} < 0$, we should decrease the reservation price for avoiding adverse filling of buy orders and incentivize short exposure. Second, considering our net position, these directional adjustments should be allowed to contradict with the Avellaneda–Stoikov model, i.e., if we predict a price increase (decline), we should be able to use a reservation price greater (less) than r_{AS} even if we are long (short). Based on the Avellaneda–Stoikov model (equation (4.1)), we define the directional forecast-adjusted reservation price r_{DF} by

$$r_{DF} = s + \left(sgn\left(q\right) \frac{1}{\alpha_2} \widehat{TFI} - 1 \right) q\gamma\sigma^2 \left(T - t\right) \tag{4.7}$$

where $\alpha_2 \geq 0$ measures how sensitive the reservation price is to a prediction and $sgn(q)$ is the signum function of q, taking -1 (resp. 1) for $q < 0$ (resp. $q > 0$) and 0 for $q = 0$. Rewriting equation (4.7) as $r_{DF} = r_{AS} + sgn\left(q\right) \frac{1}{\alpha_2} \widehat{TFI}\ q\gamma\sigma^2 \left(T - t\right)$ illustrates that we simply add or subtract a weighted version of the Avellaneda–Stoikov correction term $q\gamma\sigma^2(T-t)$. For $\alpha_2 = 1$ and $q > 0$, the maximum adjustment ($\widehat{TFI} = 1$) is $r_{DF} - s = 0$, i.e., we neutralize the Avellaneda–Stoikov correction, and the minimum adjustment ($\widehat{TFI} = -1$) is $r_{DF} - s = 2\left(r_{AS} - s\right)$, i.e., we double the distance to the mid price. Smaller (larger) values for α_2 imply a higher (lower) price sensitivity to $\widehat{TFI}$. For instance, if $\widehat{TFI} = 0.25$, the price adjustment is $2q\gamma\sigma^2(T - t)$ for $\alpha_2 = \frac{1}{2}$ and $\frac{1}{2}q\gamma\sigma^2(T - t)$ for $\alpha_2 = 2$.

Figure 4.6 illustrates the difference between r_{DF} and r_{AS} for values in the range $q = [-2, 2]$ and $\widehat{TFI} = [-1, 1]$. If predicting a sell surplus, it is $r_{DF} < r_{AS}$, either contradicting ($q < 0$) or

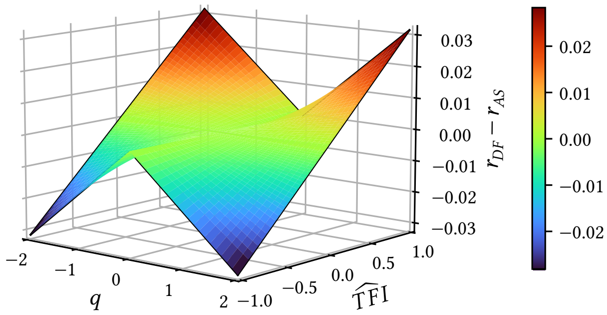

Figure 4.6: Prediction-based reservation price adjustments in the range $q = [-2, 2]$ and $\widehat{TFI} = [-1, 1]$. We measure reservation price adjustments by the difference between r_{DF} and r_{AS} based on the following parameters: $\alpha_2 = 1$, $\gamma = 0.25$, $\sigma = 0.3$, $T = 1$, $t = 0.3$.

reinforcing ($q > 0$) the Avellaneda–Stoikov correction. Analogous, if predicting a buy surplus, it is $r_{DF} > r_{AS}$, either contradicting ($q > 0$) or reinforcing ($q < 0$) the Avellaneda–Stoikov correction. The directional forecast-adjusted ask and bid quote is $r^a_{DF} = r_{DF} + \frac{1}{2}\left(\delta^a + \delta^b\right)$ and $r^b_{DF} = r_{DF} - \frac{1}{2}\left(\delta^a + \delta^b\right)$, respectively.

4.5.3 Consolidated prediction-based limit order model

Finally, we combine both *DF* and *SP* models to one consolidated limit order model that considers r_{DF} as reservation price and determines bid and ask quotes based on equations (4.5) and (4.6) with replacing r_{AS} by r_{DF}. As a result, the consolidated ask price is $r^a_{SPDF} = r_{DF} + \frac{1}{2}\left(1 + \frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\right)\left(\delta^a + \delta^b\right)$ and, correspondingly, the consolidated bid price is $r^b_{SPDF} = r_{DF} - \frac{1}{2}\left(1 + \frac{1}{\alpha_1}\left(2\widehat{R}^a - 1\right)\right)\left(\delta^a + \delta^b\right)$.

One might wonder how *DF* and *SP* relate to each other and whether *DF* and *SP* adjustments to *AS* quotes would offset each other in the consolidated model. To understand their relationship, we compare the absolute difference between *SPDF* quotes and *AS* quotes (e.g., $|r^a_{SPDF} - r^a_{AS}|$) with the absolute difference between *SP* or *DF* quotes and *AS* quotes (e.g., $|r^a_{DF} - r^a_{AS}|$ or $|r^a_{SP} - r^a_{AS}|$). By using absolute differences, we take into account that *SP* or *DF* corrections occur in both directions, either increasing or decreasing *AS* quotes, which allows us to summarize the comparison in four cases represented by inequalities (4.8)–(4.11). For instance, if inequality (4.8) holds, this means that *SPDF* does not offset or reinforces the *DF* adjustment, whereas if it does not hold, this means that the *SPDF* adjustment to *AS* is less than the *DF* adjustment, i.e., *SPDF* contradicts *DF* and neutralizes (parts of) the *DF* adjustment.

$$|r^a_{SPDF} - r^a_{AS}| \geq |r^a_{DF} - r^a_{AS}| \tag{4.8}$$

$$|r^b_{SPDF} - r^b_{AS}| \geq |r^b_{DF} - r^b_{AS}| \tag{4.9}$$

$$|r^a_{SPDF} - r^a_{AS}| \geq |r^a_{SP} - r^a_{AS}| \tag{4.10}$$

$$|r^b_{SPDF} - r^b_{AS}| \geq |r^b_{SP} - r^b_{AS}| \tag{4.11}$$

With regard to *DF* strategy, we find in Appendices 4.B.ii and 4.B.iii that inequality (4.8) holds if $\widehat{R}^b \geq \widehat{R}^a$ and $\widehat{R}^b \geq \frac{1}{2}$ or if $\widehat{R}^b < \widehat{R}^a$ and $\widehat{R}^b \leq \frac{1}{2}$, and inequality (4.9) holds if $\widehat{R}^b \geq \widehat{R}^a$ and $\widehat{R}^a \leq \frac{1}{2}$ or if $\widehat{R}^b < \widehat{R}^a$ and $\widehat{R}^a \geq \frac{1}{2}$. For instance, when *DF* shifts the reservation price upward ($\widehat{R}^b > \widehat{R}^a$), *SP* does so as well if buy orders are relatively large ($\widehat{R}^b > \frac{1}{2}$) or sell orders are relatively small ($\widehat{R}^a < \frac{1}{2}$). Thus, if predicting a buy surplus, we increase the reservation price to protect from being adversely selected by informed buyers, thereby implicitly increasing (decreasing) the depth of buy (sell) orders in the LOB. This does not hold if the predicted buy surplus is based on a relatively small buy volume, as in that case, we choose a closer spread to increase the chance of being consumed by small orders.

Appendices 4.B.ii and 4.B.iii outline that inequalities (4.8) and (4.9) also hold in the opposite cases, e.g., $\widehat{R}^b \geq \widehat{R}^a$ with $\widehat{R}^b < \frac{1}{2}$ for inequality (4.8), if the *DF* adjustment to the reservation price is *smaller* than the *SP* adjustment to the order depth. For instance, given $\widehat{R}^b \geq \widehat{R}^a$ with $\widehat{R}^b < \frac{1}{2}$, inequality (4.8) holds if $\frac{1}{2\alpha_1}\left(1 - 2\widehat{R}^b\right)\left(\delta^a + \delta^b\right) > 2\left(r_{DF} - r_{AS}\right)$. Figure 4.19 in Appendix 4.C numerically illustrates that for small q and market standard values for γ, σ and κ, this condition is given in most possible combinations of $\widehat{R}^b$ and $\widehat{R}^a$. Hence, as our trading strategies are designed to keep inventory close to zero, we infer that *DF* largely shifts prices in the same direction as *SP*.

With regard to *SP* strategy, we find that, analogous to *DF* strategy, inequality (4.10) holds if $\widehat{R}^b \geq \widehat{R}^a$ and $\widehat{R}^b \geq \frac{1}{2}$ or if $\widehat{R}^b \leq \widehat{R}^a$ and $\widehat{R}^b < \frac{1}{2}$, and inequality (4.11) holds if $\widehat{R}^b \leq \widehat{R}^a$ and $\widehat{R}^a \geq \frac{1}{2}$ or if $\widehat{R}^b \geq \widehat{R}^a$ and $\widehat{R}^a < \frac{1}{2}$. If predicting large sell orders ($\widehat{R}^a > \frac{1}{2}$) and a sell surplus ($\widehat{R}^b < \widehat{R}^a$), we post buy orders deeper in the book to maximize the captured spread, thereby implicitly lowering the midpoint between ask and price, which is consistent with lowering the reservation in response to the sell surplus. If predicting a buy surplus instead, the *DF* adjustment would contradict the *SP* adjustment in response to $\widehat{R}^a \geq \frac{1}{2}$. Corresponding to *DF* strategy, we show in Appendices 4.B.iv and 4.B.v that inequalities (4.8) and (4.9) can also hold in the opposite scenarios, such as $\widehat{R}^b < \widehat{R}^a$ with $\widehat{R}^b \geq \frac{1}{2}$ for inequality (4.10), when the *DF* adjustment to the reservation price is *larger* than the *SP* adjustment to the order depth. However, Figure 4.20 in Appendix 4.C numerically illustrates that for small q, this condition does mostly not hold. Regarding the fact that the trading strategies aim to minimize q, we infer that, in these opposite cases, *SP* largely would offset the *DF* price adjustment.

4.5.4 Trading model

We propose a limit order trading model in which we predict the five-second averages of R^a and R^b, use these predictions to determine our bid and ask price, and then simultaneously place a corresponding buy order and a corresponding sell order in the limit order book. After five seconds, the orders have either been filled by incoming market orders or we cancel them and submit new orders. Even if both orders get filled in less than five seconds, we still wait five seconds before submitting new orders. We choose a five-second interval to present a robust model that does not depend on perfect market conditions, latency, and other microscale features, and instead can be easily embedded in any practical setup. As target variables we consider the equal-weighted averages of all five values in the five-second intervals of R^b and R^a, respectively, while keeping the predictors in one-second granularity (see Section 4.3). Thus, our prediction conveys information about how large market orders in the next five seconds will be on average relative to the orders in the last 24 hours.

We follow the market making literature and base our trading model on several assumptions: First, the money market pays no interest and we can submit and cancel orders at no cost (Avellaneda and Stoikov, 2008). Second, there is no latency (Guilbaud and Pham, 2013), which is a rather minor condition since we consider five-second intervals to be largely independent of latency. Third, our limit orders have no impact on the market (Avellaneda and Stoikov, 2008; Cartea et al., 2014; Fodra and Labadie, 2012; Spooner et al., 2018), which is a simplifying assumption as, in practice, limit orders *do* affect other traders and price discovery. However, the alternative measure would be a live trading algorithm, which is difficult to evaluate: If one runs multiple live trading strategies in parallel, they would affect each other and thus bias trading results, whereas if run sequentially or simultaneously in different markets, the results might not be comparable. As a result, market making research must make this simplifying assumption to obtain empirical indications of trading performance. The assumption is most realistic when modeling highly liquid assets (Zhang et al., 2019) and order sizes that are small compared to the total quantity traded in the market (Spooner et al., 2018). We meet these two requirements by focusing on the two most liquid cryptocurrency pairs in the market (see Section 4.3) and modeling order sizes of 0.1 BTC and 0.1 ETH, respectively.

Next, we parameterize the Avellaneda–Stoikov model. First, we assume a constant risk aversion of $\gamma = 0.5$, where large (small) values of γ indicate high (low) risk aversion. Second, we assume an order book density of $\kappa = 2$, where large (small) values of κ indicate regimes where trading is fast (slow) and volatility is low (high), leading to small (large) bid-ask spreads and orders placed far from (close to) the mid price, forcing us to narrow (widen) our own spread (see equation (4.2)) (Cartea and Jaimungal, 2013). For further reading on the calibration of κ, we refer to Laruelle (2013). Third, corresponding to R^b and R^a, we compute σ as the 24-hour standard deviation of the mid price, i.e., each σ_t represents the volatility of the mid price between t and 24 hours ago.

We set the maximum net position at ±0.5 BTC and ±0.5 ETH, respectively. Once reached the inventory limit, we stop posting orders on the respective side until the net position is less than 0.5. We trade BTC and ETH simultaneously, with hedging of positions between BTC and ETH being not allowed, and calculate P&L following Avellaneda and Stoikov (2008) separately for each currency pair in three steps: First, our net position at the end of interval t is the net position from $t - 1$ plus bid and less ask orders executed in t. Second, our cash position at the end of t is the cash position in $t - 1$, less the money spent for bid orders and plus the money received from ask orders executed in t. Finally, our P&L at the end of t equals the new cash position plus the new inventory position scaled by the mid price (Spooner et al., 2018). As we use indexed prices (Section 4.3), P&L numbers reported do not represent currency units. In the following sections, we determine P&L as equal-weighted average of both currency pairs.

Due to the LOB's aggregate form, we have no access to the time queue at a particular price level. While we can separate new from waiting orders based on trade history, if the size at a price level decreases without a trade at that time, we cannot identify which orders were canceled, and we do not know whether our order was ahead or behind that order (Spooner et al., 2018). We follow Spooner et al. (2018) and "assume that cancellations are distributed uniformly throughout the queue", i.e., the "probability that the canceled order is ahead [...] is proportional to the amount of volume ahead [...] compared to the amount of volume behind" our order. At the beginning of a five-second interval, we submit orders at the end of the queue of the respective price level. Each one-second interval, we update the LOB and match it with the market order flow. Accordingly, any increase (resp. decrease) in volume at a particular price level that is not related to a trade will queue behind our order (resp. be evenly distributed among the orders before and after our order).

4.5.5 Empirical calibration of α_1 and α_2

As we have historical data, instead of analytically approximating the likelihood of order consumption by α_1 and α_2, we can empirically determine which values of α_1 and α_2 would have been P&L-optimal. For this calibration, we assume to have full knowledge of the future and determine quotes according to the realized R^b and R^a, which represents the scenario of trading with a perfect prediction model, resulting in the empirically maximum achievable P&L for each α_1, α_2.

Figure 4.7 illustrates the average one-hour P&L, i.e., the average P&L at the end of each hour, and the execution rate ExR, i.e., the share of executions in all submissions, achieved by *SP* as a function of α_1 and achieved by *DF* as a function of α_2 based on the validation sets. We find that P&L$_{SP}$ and P&L$_{DF}$ have their optima for $\alpha_1 < 1$ and $\alpha_2 < 1$, respectively, indicating relatively high sensitivity to predictions. For very small α_1 and α_2, *SP* and *DF* are less profitable as orders get filled rarely. For $\alpha_1, \alpha_2 < 1$, *SP* and *DF* outperform *AS* despite less order executions, suggesting larger spreads earned or reduced adverse selection cost. The optimal calibration for *SP* is $\alpha_1 = 0.5$, resulting in a 35% improvement in P&L over *AS*, and for *DF* it is $\alpha_2 = 0.9$, resulting in a 10% P&L improvement.

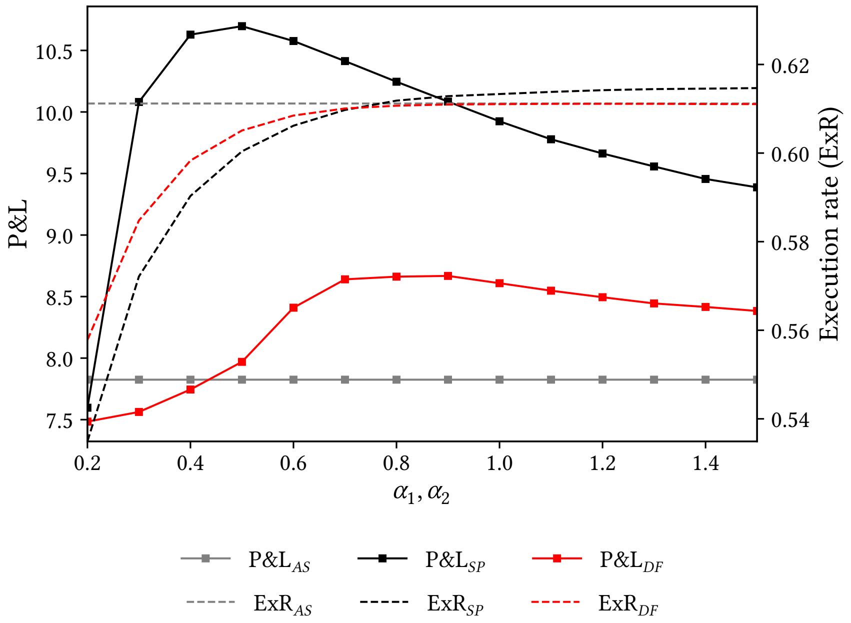

Figure 4.7: Average one-hour P&L (solid lines with markers) and order execution rate (dashed lines) of *SP* strategy (black) and *DF* strategy (red) as functions of α_1 and α_2, respectively, based on realized values R^b and R^a. The gray lines represent the *AS* benchmark, which is independent of α_1 and α_2.

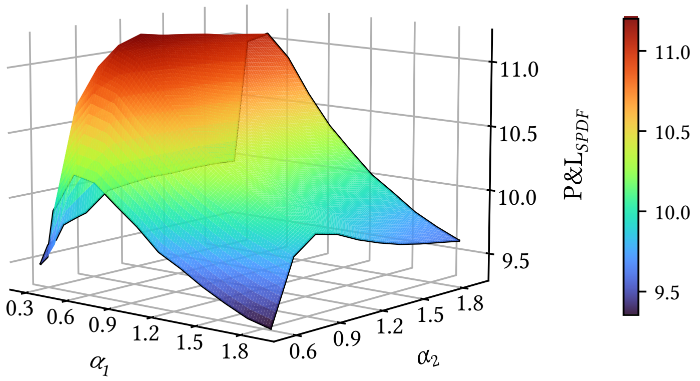

Figure 4.8: Average *SPDF* one-hour P&L as a function of α_1 and α_2 based on realized values R^b and R^a (validation sets).

As under particular circumstances (see Appendix 4.B), *SP* and *DF* adjustments to order prices could offset each other, we need to calibrate α_1 and α_2 separately for the consolidated model. Figure 4.8 illustrates the average one-hour P&L achieved by *SPDF* as a function of α_1 and α_2. Similar to Figure 4.7, P&L is the lowest for large and very small values of α_1 and α_2, and has its maximum for the combination of $\alpha_1 = 0.55$ and $\alpha_2 = 1.00$, improving P&L by more than 40%. This calibration is similar to the stand-alone strategies *SP* and *DF*, requiring us to be about twice as sensitive to predictions when adjusting order depth to trade flow than when adjusting the reservation price to adverse price movements.

4.6 Trading simulation with order predictions

4.6.1 Prediction quality

We analyze prediction quality benchmarked against a single-layer feedforward neural network (SFN), which directly maps the inputs to R^b and R^a based on sigmoid activation functions. We train SFN with the training and validation periods mentioned in Section 4.3 and test it with the same test periods we use for RNN. Figure 4.9 illustrates the frequency distribution of $\widehat{R}^b$ and $\widehat{R}^a$ over all test periods. Regarding the distribution of R^b and R^a, RNN apparently captures the true case better than SFN by achieving a stretched and rather flat predictions distribution. Both models predict very large and small values poorly, with substantially less (RNN) or hardly any (SFN) predictions in the upper and lower 10%. The SFN distribution is slightly right-skewed with more predictions in the upper 20% than in the lower 20%.

Table 4.1 reports average prediction errors aggregated by side and prediction error sign. Throughout all groups, RNN shows smaller prediction errors than SFN. For instance, if overpredicting percentile ranks of buyer-initiated trade sizes ($\widehat{R}^b - R^b > 0$), SFN ranks them on average 23 percentile ranks too high, whereas RNN ranks them on average 14 ranks too high, suggesting an improvement over SFN by 40%. The mean squared prediction error of RNN is less than half of the SFN model, which corresponds with the flatter frequency distribution in Figure 4.9. Moreover, Figure 4.21 in Appendix 4.D illustrates that mean squared prediction errors depend on the predicted percentile rank, with RNN showing the best (worst) prediction quality when predicting average (extreme) values. Accordingly, RNN predictions are most

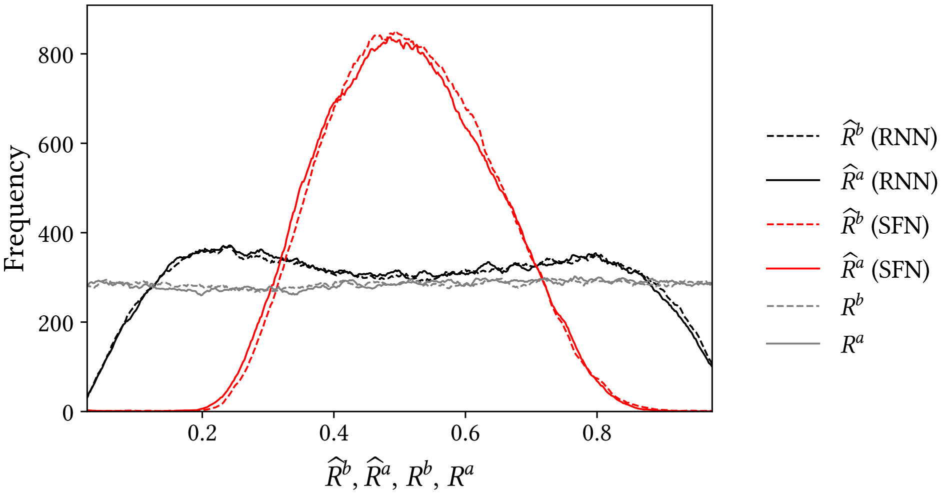

Figure 4.9: Frequency distribution of realized and predicted percentile ranks. Black (red) lines focus on RNN (SFN) predictions $\widehat{R}^b$ and $\widehat{R}^a$, whereas gray lines focus on the realized values R^b and R^a. Solid (dashed) lines represent seller- (buyer-) initiated trades.

reliable for predictions inside the range $\widehat{R}^b, \widehat{R}^a = [0.2, 0.8]$, whereas SFN exhibit the smallest errors when predicting large percentile ranks.

Next, we investigate whether prediction quality depends on market sentiment. Analogous to the sequence length, we group the last ten seconds before a prediction into two equal-sized intervals: The first (second) interval TFI_{1-5} (TFI_{6-10}) represents the aggregated TFI of the first (last) five seconds, where we calculate TFI according to Section 4.4.1 based on the 24-hour percentile ranks of aggregated five-second trade sizes. Figure 4.10 shows the mean squared error as a function of TFI_{1-5} and TFI_{6-10} for RNN and SFN predictions. Prediction models perform well when the preceding trade flow is relatively balanced between buyer- and seller-initiated trade volume, and poorly when there is an excess of either buyer- or seller-initiated trades. This behavior is less pronounced for RNN than for SFN predictions, indicating that RNN outperforms SFN when making predictions based on imbalanced trade flows. For practical purposes, a trader may conclude that predictions are most reliable when trade volume is balanced, and most prone to error when it is biased during the last 10 seconds.

Side	$\widehat{R}^b$			$\widehat{R}^a$		
$\widehat{R} - R$	< 0	> 0	All	< 0	> 0	All
Average prediction error						
RNN	−0.14	0.14	−0.00	−0.14	0.14	−0.00
SFN	−0.22	0.23	0.01	−0.22	0.22	0.00
Mean squared prediction error (%)						
RNN	3.78	3.78	3.78	3.83	3.80	3.82
SFN	6.96	7.21	7.09	7.14	7.04	7.09

Table 4.1: Average and mean squared prediction errors aggregated by prediction error sign. We calculate prediction errors by $\widehat{R}^b - R^b$ and $\widehat{R}^a - R^a$, respectively. Prediction error sign group $\widehat{R} - R < 0$ ($\widehat{R} - R > 0$) includes all predictions smaller (larger) than the realized values. Each observation and each test period has the same weight.

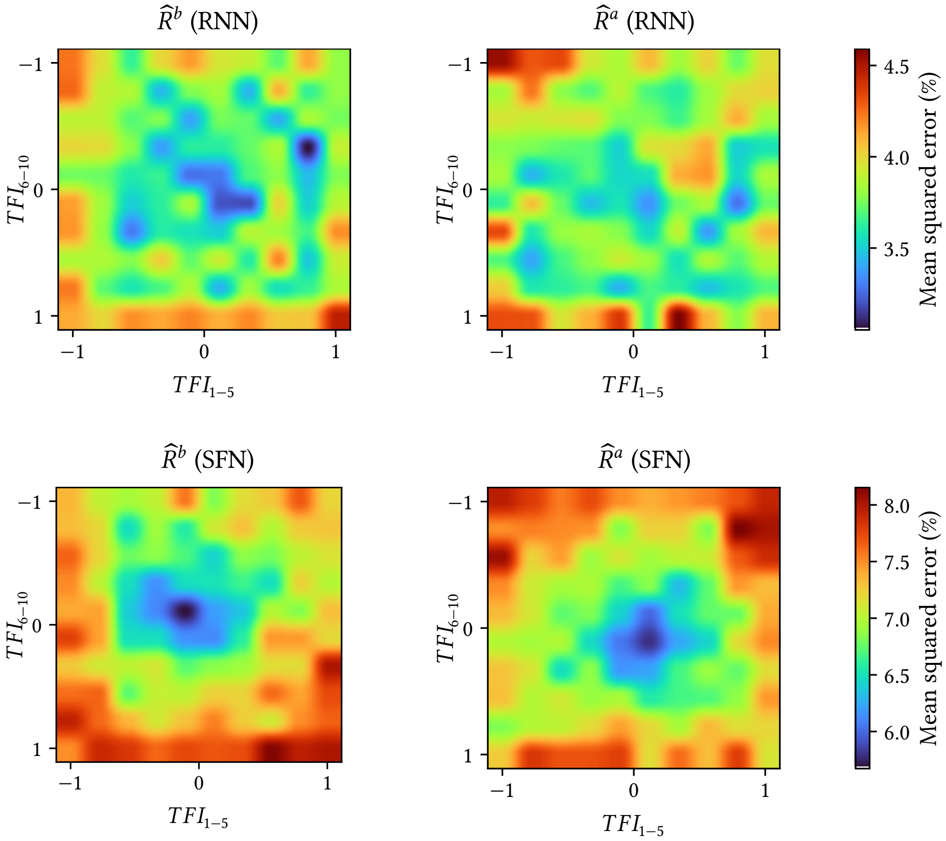

Figure 4.10: Mean squared prediction error by market sentiment. We measure market sentiment by the combinations of the two last five-second *TFI* before a prediction, TFI_{1-5} and TFI_{6-10}. The two top (bottom) plots are based on RNN (SFN) predictions, while the two right (left) plots refer to buy (sell) predictions. Note that color bars belong to the prediction model of the respective row.

An alternative method to evaluate prediction quality is transforming our prediction task into a classification task, where the positive class represents a trade size percentile rank greater than 0.5, and the negative class represents a percentile rank less than or equal 0.5. Table 4.3 in Appendix 4.D shows that SFN reaches a classification accuracy of 64% and RNN reaches an accuracy of 82%, i.e., in 64% (resp. 82%) of all observations, SFN (resp. RNN) predicts correctly whether buy and sell market order flow in the next five seconds will be less or greater than the 24-hour median. Thus, RNN improves SFN by about 30%.

4.6.2 Trading performance

We conduct trading simulations separately for *SP*, *DF*, and *SPDF* strategy, once based on RNN and once based on SFN predictions. Table 4.2 presents the equal-weighted average of cumulative P&Ls achieved within non-overlapping periods of 1 minute to 12 hours. In the majority of periods, prediction-based strategies lead to an improvement over *AS*, e.g., we improve one-hour P&L by up to 7%. We also perform t-tests between the P&L distributions of prediction-based strategies and that of *AS* strategy and find that increases in P&L are statistically significant in most cases. RNN-based strategies produce, on average, a higher P&L than

Strategy		1 min	5 min	1 h	12 h
AS		0.13 (0.25)	0.66 (0.79)	7.89 (5.71)	81.37 (32.95)
SP	RNN	0.14^{***} (0.28)	0.70^{***} (0.95)	8.06^{**} (7.48)	101.93^{***} (44.71)
	SFN	0.14^{***} (0.27)	0.68^{*} (0.92)	8.01^{*} (7.29)	91.86^{**} (42.27)
DF	RNN	0.13^{**} (0.25)	0.69^{**} (0.80)	8.34^{***} (5.74)	88.10^{**} (32.56)
	SFN	0.13 (0.25)	0.66 (0.80)	7.88 (5.76)	81.00 (32.91)
SPDF	RNN	0.14^{***} (0.28)	0.75^{***} (0.93)	8.45^{***} (7.29)	104.13^{***} (43.61)
	SFN	0.14^{**} (0.26)	0.67 (0.90)	7.92 (7.08)	92.01^{**} (41.06)

Table 4.2: Equal-weighted average of cumulative P&L after periods between 1 minute (min) and 12 hours (h). Numbers in brackets report the P&Ls' standard deviation. Numbers followed by ***, ** and * indicate statistical difference from *AS* at the 1%, 5% and 10% level (t-test).

SFN-based strategies. Moreover, the standard deviation of P&Ls illustrate that higher profit chances come at the cost of higher risk. Only *DF* actually achieves a meaningful increase in profits while keeping standard deviation constant.

In order to quantify this P&L improvement, we define the P&L surplus as the difference

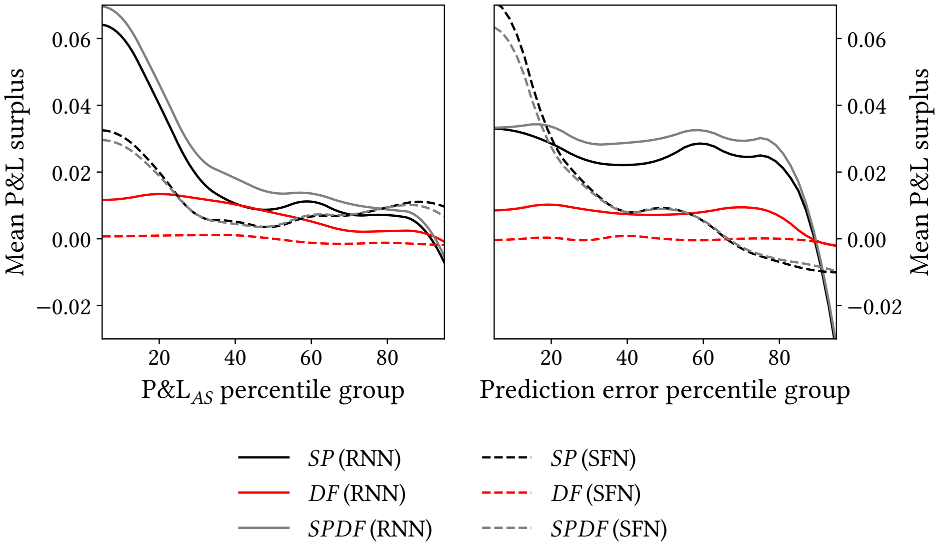

Figure 4.11: Average one-minute P&L surplus as a function of one-minute P&L$_{AS}$ percentile groups (left) and prediction error percentile groups (right). Percentile group 1 (100) contains the worst (best) 1% observations of a variable. Prediction errors are the sum of absolute prediction errors recorded for buy and sell predictions over one minute. Solid (dashed) lines refer to RNN (SFN) predictions.

between the P&L of a prediction-based strategy and the P&L of *AS* (P&L$_{AS}$). The left chart in Figure 4.11 illustrates the average one-minute P&L surplus as a function of one-minute P&L$_{AS}$ percentile groups, where percentile group 1 (100) contains the worst (best) 1% of all one-minute P&L$_{AS}$. In both RNN- and SFN-based strategies, we benefit from prediction-based strategies when *AS* performs poorly. The better *AS* performs, the less beneficial or more detrimental prediction-based strategies are.

The right chart plots the P&L surplus over one-minute prediction error percentile groups, where percentile group 1 (100) contains the 1% smallest (largest) one-minute prediction errors. Since we have prediction errors for buy and sell forecasts separately, while the P&L aggregates executed buy and sell orders, we sum the absolute values of prediction errors recorded for buy and sell forecasts within one minute. The graph illustrates that P&L surplus is large (small) when prediction errors are small (large), suggesting that prediction quality is an indicator of P&L performance. Besides, the charts show that *DF* is less dependent on prediction quality and P&L$_{AS}$ than *SP* or *SPDF* strategy.

Next, we analyze the relationship between P&L and the dynamics of order flow and trade flow. To this end, Figure 4.12 illustrates each strategy's average one-minute P&L as a function of one-minute *TFI* and one-minute *OFI*. We calculate one-minute *TFI* based on minute-by-

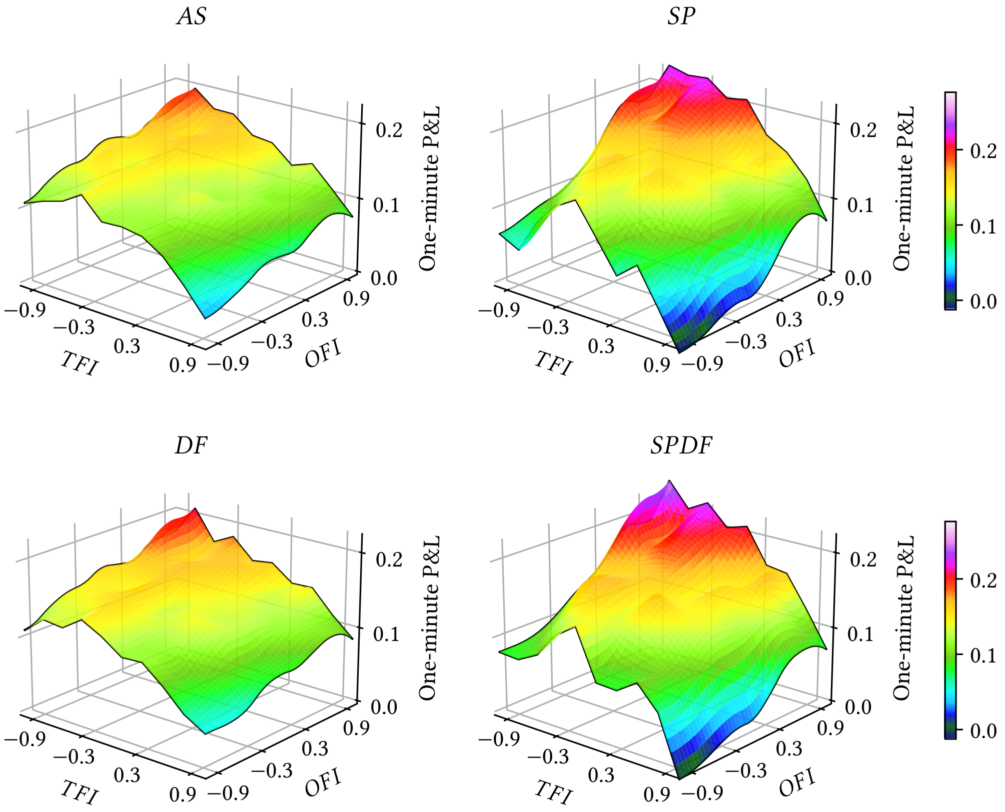

Figure 4.12: Average one-minute P&L achieved by *AS* strategy (top left) and prediction-based strategies using RNN predictions as a function of one-minute *TFI* and *OFI*. We calculate *OFI* as an equal-weighted average of all one-second order book states. *TFI* < 0 (*OFI* < 0) indicates a sell surplus in matched orders (open limit orders). Each chart refers to the same color bar.

minute aggregated buy and sell trades, whereas for *OFI*, we consider the equal-weighted averages of all one-second LOB records within a minute, since the order book is a snapshot at a given point in time and aggregation is not possible. All four charts show that trading is least profitable during a seller overhang in the LOB ($OFI < -0.5$) in combination with a buyer surplus in trade flow ($TFI > 0.5$), potentially representing high competition among waiting sellers such that we can catch only a small share of impatient buyers. P&L is the highest in the opposite scenario, $OFI > 0.5$ with $TFI < -0.5$, indicating that we are more successful in competing with buyers than with sellers (see Section 4.6.3).

Overall, *AS* and *DF* (*SP* and *SPDF*) are less (more) dependent on market dynamics and therefore provide a less (more) volatile P&L profile. Apparently, *SP* and *SPDF* are superior to *AS* and *DF* only if $OFI > 0$ and $TFI < 0$, whereas in all other situations, *AS* and *DF* are superior to *SP* and *SPDF*. Hence, imbalances in trade flow and order flow may indicate whether a strategy is more profitable than another one. In practice, firms could use this information and switch between trading strategies in response to changes in trade flow and order flow, e.g., switch to *AS* or *DF* (*SP* or *SPDF*) when the buyer share (seller share) in trade flow is large while the buyer share (seller share) in the LOB is small.

Since P&L includes both market making revenues and unrealized directional changes of the net position (Spooner et al., 2018), we next aim to understand each component's contribution to P&L. We calculate P&L generated from directional movements, P&L^{DR}, as the change in value of the net position caused by the mid price change (Anolli and Petrella, 2007), i.e., if the mid price increases by 0.1 while we are long 2 assets, it is P&L^{DR} = 0.2. The left plot in Figure 4.13 reports RNN-based P&L^{DR} surplus as a function of one-minute mid price change decile groups, i.e., decile group 1 (10) contains the 10% minutes with the most negative (positive) mid price changes. In case of *DF*, P&L^{DR} surplus is the highest (smallest) when the mid price sharply (slightly) increases or decreases. As P&L^{DR} can be viewed as an approximation of adverse selection cost, this result suggests that *DF*'s reduction in adverse selection cost is proportional to the magnitude of price movement: The larger the positive or negative price movement, the larger the reduction in adverse selection cost.

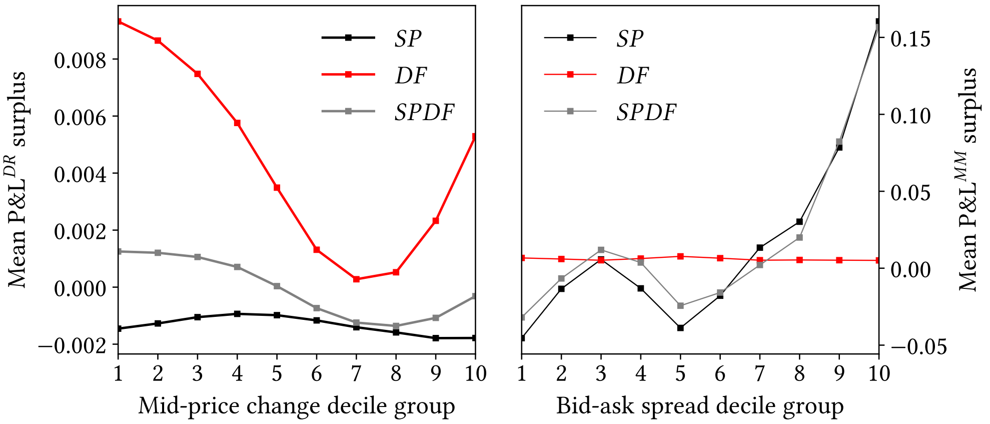

Figure 4.13: Average one-minute P&L^{DR} surplus as a function of mid price change decile groups (left) and average one-minute P&L^{MM} surplus as a function of bid-ask spread decile groups (right). All strategies base on RNN predictions. Bid-ask spreads are the average spreads of all one-second snapshots per minute.

As second P&L component, we calculate P&L from market making activities, P&L^{MM}, by subtracting P&L^{DR} from P&L (Spooner et al., 2018). Thus, we implicitly subsume realized directional revenues (Anolli and Petrella, 2007) under market making revenues. The right plot in Figure 4.13 shows average P&L^{MM} surplus as a function of one-minute bid-ask spread decile groups, where decile group 1 (10) contains the 10% minutes with the smallest (largest) average bid-ask spread across all one-second snapshots per minute. Both *SP* and *SPDF* achieve the largest increase in market making revenues when bid-ask spreads are large, suggesting that the strategies are most successful at adjusting their LOB position to trade flow when the order book is less liquid. From both charts in Figure 4.13, we infer that P&L surplus of *DF* is mainly driven by reduced adverse selection cost, while P&L surplus of *SP* and *SPDF* is mainly due to increased market making revenues.

4.6.3 Limit order book dynamics

In Section 4.6.2, we gained an understanding of how the trading strategies relate to each other from a P&L perspective. Yet, we have no information how they affect our positioning in the LOB. To this end, Figure 4.14 illustrates the average depth of ask orders, δ^a, and the average depth of bid orders in the order book, δ^b, as functions of bid-ask spread percentile groups (top left), the net position (top right), the predicted trade flow imbalance (below left), and one-second P&L percentile groups (below right). For consistency purposes with previous sections, we continue to analyze BTC-USD and ETH-USD pairs together, although the LOB characteristics may differ between the two pairs. Figure 4.23 in Appendix 4.G shows that the results for the aggregated order books also hold for the stand-alone order books.

The top left graph shows that *SP* and *SPDF* place orders deeper in the book than *AS* and *DF*, suggesting that *SP* and *SPDF* seek increasing the margin through wider spreads rather than increasing trade volume through tighter spreads. *AS* and *DF* quotes are less sensitive to bid-ask spreads, i.e., when bid-ask spreads increase, *AS* and *DF* follow this trend slower than *SP* and *SPDF*. It is $\delta^a > \delta^b$ in most cases, indicating that our average net position is short (see Figure 4.22 in the Appendix). Thus, despite prediction-based corrections, the dominant parameter determining δ^a and δ^b remains q. Correspondingly, the top right graph shows that δ^a is large (small) when we are short (long) and δ^b is small (large) when we are short (long) to return to a neutral net position.

The chart below left illustrates *DF*'s and *SPDF*'s protection against adverse price selection: δ^a increases for $\widehat{TFI} > 0$ to protect against informed buyers, and decreases for $\widehat{TFI} < 0$ to incentivize short exposure for positively participating in the price decline. Accordingly, buy prices are lower, represented by increasing δ^b, when predicting a sell surplus, although the effect is less pronounced than for sell orders, which could be due to the fact that we are short on average and thus generally place buy orders deeper in the book. The chart below right illustrates that submitting sell orders deep in the book is risky: The highest and lowest P&L are associated with the deepest sell orders, while sell orders close to the top of the order book are associated with average P&L. Again, this relationship is less clear for buy orders.

Next, we compare the price level priority of orders between prediction-based strategies and the *AS* strategy. We start counting price levels at the top of the book as position 1, continuing with the second-best price level as position 2, and so on. Thus, a position difference of 1 (−1) means that the prediction-based strategy is one price level behind (ahead of) *AS*, i.e., it has a one price level lower (higher) priority. Figure 4.15 shows the average price level difference for RNN-based strategies as a function of $\widehat{TFI}$ (left) and *OFI* (right). The left chart shows that

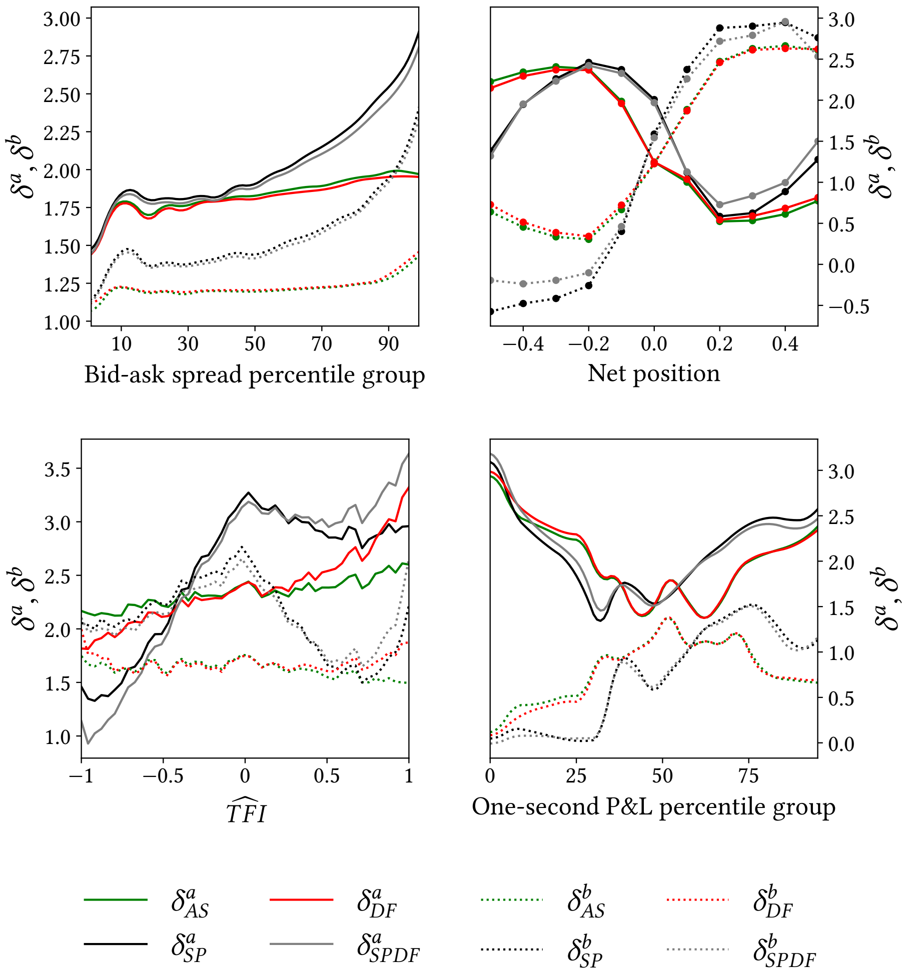

Figure 4.14: Average depth of ask orders δ^a (solid lines) and bid orders δ^b (dotted lines) in the order book for both BTC-USD and ETH-USD pairs. *SP*, *DF*, and *SPDF* base on RNN predictions. δ^a and δ^b denote in basis points. `Net position' represents each strategy's individual net position. `Bid-ask spread' is the one-second bid-ask spread at the end of a one-second interval.

all three strategies post sell (buy) orders deeper in the book when predicting a price increase (decline) to protect against informed buyers (sellers). This effect is most pronounced for *DF* and *SPDF*, since *SP* initially does not consider directional forecasts. The right chart illustrates that *SP* and *SPDF* orders have a lower (equal) priority when there is either (neither) a large buy or (nor) a large sell overhang in the order book. In contrast, *DF* posts sell orders closer to the top of (deeper in) the book when $OFI > 0$ ($OFI < 0$), thus acts contrary to $\widehat{TFI}$.

To understand how our positioning in the LOB contributes to P&L, the bar chart in Figure 4.16 illustrates the execution rate by price level. An execution rate of 30% at price level 3 means that 30% of our orders queuing at price level 3 are filled within five seconds. The circle markers indicate the average margin earned, measured by $\delta^a + \delta^b$. A margin of 5 basis points at price level 2 means that we earn 5 basis points on average if one buy and one sell order, each waiting at the second best price level, get simultaneously filled by market orders. As relevant price levels, we consider the price level at the time of execution. If we submit an

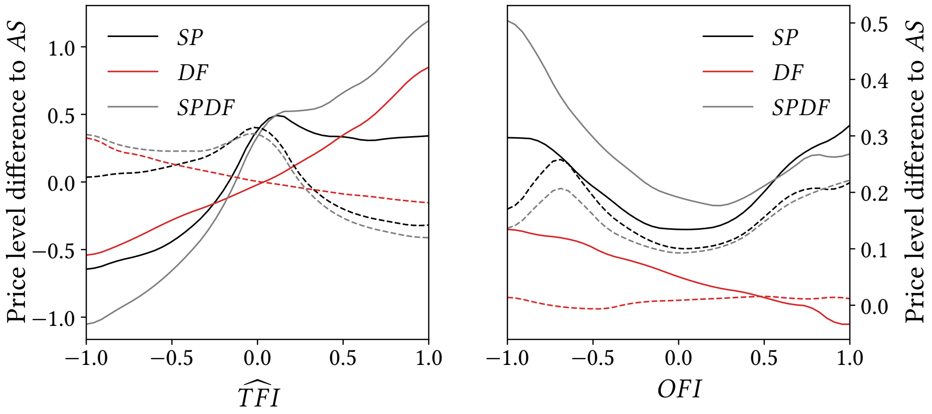

Figure 4.15: Average difference in price levels between orders submitted by RNN-based trading strategies and orders submitted by *AS* as functions of $\widehat{TFI}$ and *OFI*. Results represent averages of limit orders in both the BTC-USD and ETH-USD order book. Solid (dashed) lines belong to ask (bid) orders.

order at price level 2, and one second later the volume at the best price level gets filled, the order slips to position 1. Thus, execution on a price level greater than 1 always requires a market order greater than the total size on position 1.

While at the top of the order book all strategies achieve a similar execution rate, *SP* and *SPDF* exceed *AS* and *DF* by five to ten percentage points for higher price levels. Thus, when *SP* and *SPDF* submit orders at, e.g., the third-best price level, they historically have a five percentage point higher chance of being consumed. Similarly, the margins earned at the top of the order book are comparable for all strategies, but fall much larger from price level 2 for *SP* and *SPDF* than for *AS* and *DF*. This suggests that *SP* and *SPDF* can execute deeper orders

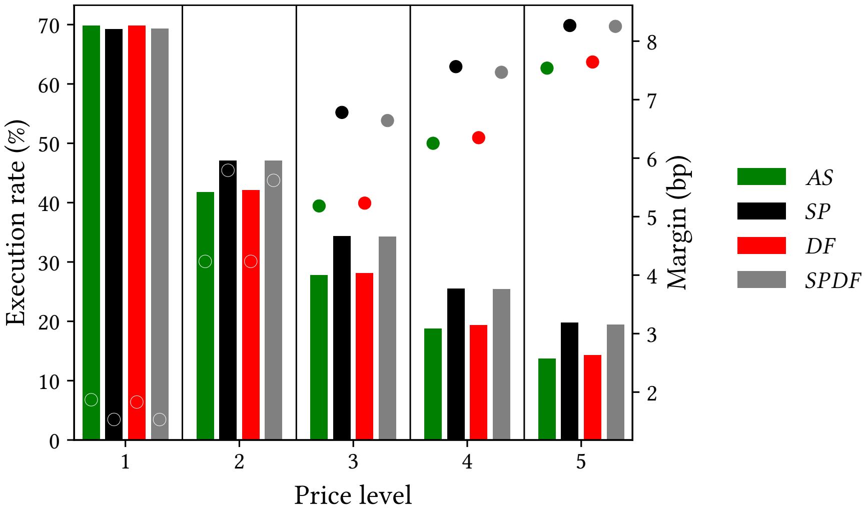

Figure 4.16: Execution rate (bar chart) and average margin earned (circle marker plot) by price level. Execution rate measures the share of executed orders in all orders waiting at a given price level. Earned margin measures the average $\delta^a + \delta^b$ of simultaneously executed bid and ask orders at corresponding price level positions in basis points.

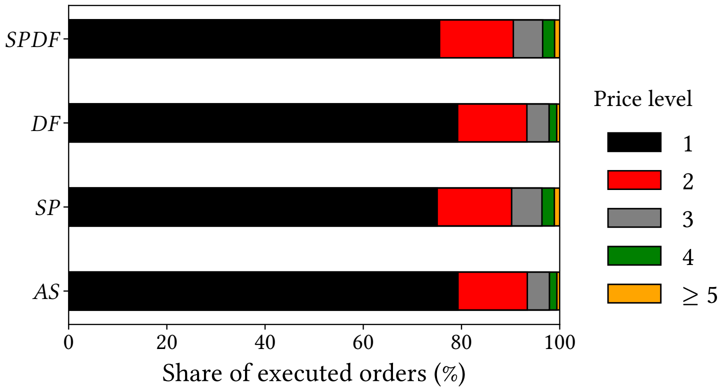

Figure 4.17: Share of order executions by price level. The figure illustrates the respective share of order executions by price level in all order executions under a particular trading strategy. Price level indicates the price level at the time of execution.

even when the bid-ask spread is large, whereas *AS* and *DF* execute on these levels only when the bid-ask spread is small anyway and, accordingly, earn less margin. Thus, *SP* and *SPDF* maximize the margin earned by increasing the depth of orders while keeping the execution share at a high level. In Figure 4.17, we analyze the share of executed orders by price level in all order executions. For instance, 79% (14%) of all orders executed by *AS* have been waiting at the first (second) price level at the time of execution, whereas this share is 75% (15%) for *SP* strategy. Thus, *SP* and *SPDF* execute about 5 percentage points more orders at price levels 2 to 5 than *AS* and *DF*. We infer that the higher market making revenue of *SP* and *SPDF* largely comes from a higher execution share of orders with more profitable spreads.

4.7 Conclusion

In this paper, we present a data-driven attempt to overcome dealers' trade-off between volume and margin. To this end, we develop a sequence prediction model for buyer- and seller-initiated trades that predicts how the trade sizes of the next five seconds relate to the trade sizes of the previous 24 hours. Using the Avellaneda–Stoikov model as an example, we show how to adjust reservation price and order depth in response to predictions based on empirically calibrated sensitivity parameters. In a trading simulation, we find both higher market making revenues through trade flow-optimized positioning in the order book and lower adverse selection costs through anticipated price movements.

For future work, we see two approaches to improving our method. Given the recent development of reinforcement learning approaches in high-frequency trading, one approach would be to incorporate our predictions into the training of an agent, thus providing predictive information to the agent. Second, instead of approximating price changes using trade flow imbalance, predicting price changes directly could improve our protection against adverse selection as negative (positive) trade flow imbalance need not always be associated with a price decrease (increase).

Overall, our model can help market makers to anticipate market movements when posting limit orders. As we process publicly available data in five second frequency, technical implementation for cryptocurrency trading might not be limited to professional traders.

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 https://www.tensorflow.org/.

Abergel, F., Anane, M., Chakraborti, A., Jedidi, A. and Toke, I. M. (2016), *Limit order books*, Cambridge University Press. doi:10.1017/CBO9781316683040

Aggarwal, C. C. et al. (2018), 'Neural networks and deep learning', *Springer* **10**(978), 3. doi:10.1007/978-3-319-94463-0

Ahuja, S., Papanicolaou, G., Ren, W. and Yang, T.-W. (2017), 'Limit order trading with a mean reverting reference price', *Risk and Decision Analysis* **6**(2), 121–136. doi:10.3233/RDA-160118

Albers, J., Cucuringu, M., Howison, S. and Shestopaloff, A. Y. (2021), 'Fragmentation, price formation and cross-impact in bitcoin markets', *Applied Mathematical Finance* **28**(5), 395–448. doi:10.1080/1350486X.2022.2080083

Anolli, M. and Petrella, G. (2007), 'Internalization in European Equity Markets Following the Adoption of the EU MiFID Directive', *Journal of Trading* **2**(2), 77–88. doi:10.3905/jot.2007.682142

Avellaneda, M. and Stoikov, S. (2008), 'High-frequency trading in a limit order book', *Quantitative Finance* **8**(3), 217–224. doi:10.1080/14697680701381228

Aydoğan, B., Uğur, Ö. and Aksoy, Ü. (2022), 'Optimal limit order book trading strategies with stochastic volatility in the underlying asset', *Computational Economics* pp. 1–36. doi:10.1007/s10614-022-10272-4

Bayraktar, E. and Ludkovski, M. (2014), 'Liquidation in limit order books with controlled intensity', *Mathematical Finance* **24**(4), 627–650. doi:10.1111/j.1467-9965.2012.00529.x

Bengio, Y., Simard, P. and Frasconi, P. (1994), 'Learning long-term dependencies with gradient descent is difficult', *IEEE Transactions on Neural Networks* **5**(2), 157–166. doi:10.1109/72.279181

Brownlee, J. (2017), *Long short-term memory networks with python: Develop sequence prediction models with deep learning*, Machine Learning Mastery. Available from https://machinelearningmastery.com/lstms-with-python/.

Cao, C., Hansch, O. and Wang, X. (2009), 'The information content of an open limit-order book', *Journal of Futures Markets: Futures, Options, and Other Derivative Products* **29**(1), 16–41. doi:10.1002/fut.20334

Cartea, Á. and Jaimungal, S. (2013), 'Modelling asset prices for algorithmic and high-frequency trading', *Applied Mathematical Finance* **20**(6), 512–547. doi:10.1080/1350486X.2013.771515

Cartea, Á. and Jaimungal, S. (2015), 'Risk metrics and fine tuning of high-frequency trading strategies', *Mathematical Finance* **25**(3), 576–611. doi:10.1111/mafi.12023

Cartea, Á., Jaimungal, S. and Penalva, J. (2015), *Algorithmic and high-frequency trading*, Cambridge University Press. Available from https://www.cambridge.org/de/academic/subjects/mathematics/mathematical-finance/algorithmic-and-high-frequency-trading?format=HB&isbn=9781107091146.

Cartea, Á., Jaimungal, S. and Ricci, J. (2014), 'Buy low, sell high: A high frequency trading perspective', *SIAM Journal on Financial Mathematics* **5**(1), 415–444. doi:10.1137/130911196

Chakraborty, T. and Kearns, M. (2011), Market making and mean reversion, *in* 'Proceedings of the 12th ACM Conference on Electronic Commerce', EC '11, Association for Computing Machinery, New York, NY, USA, pp. 307–314. doi:10.1145/1993574.1993622

Chakravarty, S. (2001), 'Stealth-trading: Which traders' trades move stock prices?', *Journal of Financial Economics* **61**(2), 289–307. doi:10.1016/S0304-405X(01)00063-0

Chan, E. (2017), *Machine Trading: Deploying Computer Algorithms to Conquer the Markets*, Wiley trading series, Wiley. doi:10.1002/9781119244066

Chollet, F. et al. (2015), 'Keras', www.keras.io. Package available from https://github.com/fchollet/keras.

Coinbase (2022), 'Coinbase Homepage: About'. Retrieved October 2022 from https://www.coinbase.com/about.

Cont, R., Kukanov, A. and Stoikov, S. (2014), 'The price impact of order book events', *Journal of Financial Econometrics* **12**(1), 47–88. doi:10.1093/jjfinec/nbt003

Dixon, M. (2018*a*), 'A high-frequency trade execution model for supervised learning', *High Frequency* **1**(1), 32–52. doi:10.1002/hf2.10016

Dixon, M. (2018*b*), 'Sequence classification of the limit order book using recurrent neural networks', *Journal of Computational Science* **24**, 277–286. doi:10.1016/j.jocs.2017.08.018

Dixon, M. F., Polson, N. G. and Sokolov, V. O. (2019), 'Deep learning for spatio-temporal modeling: dynamic traffic flows and high frequency trading', *Applied Stochastic Models in Business and Industry* **35**(3), 788–807. doi:10.1002/asmb.2399

Fodra, P. and Labadie, M. (2012), 'High-frequency market-making with inventory constraints and directional bets'. arXiv preprint, unpublished. doi:10.48550/arxiv.1206.4810

Frey, S. and Grammig, J. (2008), Liquidity supply and adverse selection in a pure limit order book market, *in* 'High Frequency Financial Econometrics', Springer, pp. 83–109. doi:10.1007/978-3-7908-1992-2_5

Fushimi, T., González, C. and Herman, M. (2018), 'Optimal high-frequency market making'. Stanford Education. Available from https://stanford.edu/class/msande448/2018/Final/Reports/gr5.pdf.

Gabaix, X., Gopikrishnan, P., Plerou, V. and Stanley, H. E. (2006), 'Institutional investors and stock market volatility', *The Quarterly Journal of Economics* **121**(2), 461–504. doi:10.1162/qjec.2006.121.2.461

Ganesh, S., Vadori, N., Xu, M., Zheng, H., Reddy, P. and Veloso, M. (2019), 'Reinforcement learning for market making in a multi-agent dealer market'. arXiv preprint, unpublished. doi:10.48550/arxiv.1911.05892

Gašperov, B. and Kostanjčar, Z. (2021), 'Market making with signals through deep reinforcement learning', *IEEE Access* **9**, 61611–61622. doi:10.1109/ACCESS.2021.3074782

Géron, A. (2018), *Neural networks and deep learning*, O'Reilly Media. Available from https://www.oreilly.com/library/view/neural-networks-and/9781492037354/.

Glosten, L. R. (1994), 'Is the electronic open limit order book inevitable?', *Journal of Finance* **49**(4), 1127–1161. doi:10.2307/2329182

Gopikrishnan, P., Plerou, V., Gabaix, X. and Stanley, H. E. (2000), 'Statistical properties of share volume traded in financial markets', *Physical Review E* **62**(4), R4493–R4496. doi:10.1103/PhysRevE.62.R4493

Gould, M. D., Porter, M. A., Williams, S., McDonald, M., Fenn, D. J. and Howison, S. D. (2013), ‘Limit order books’, *Quantitative Finance* **13**(11), 1709–1742. doi:10.1080/14697688.2013.803148

Graves, A. (2013), ‘Generating sequences with recurrent neural networks’. arXiv preprint, unpublished. doi:10.48550/arxiv.1308.0850

Guéant, O. (2016), *The Financial Mathematics of Market Liquidity: From Optimal Execution to Market Making (1st ed.)*, Chapman and Hall/CRC, New York. doi:10.1201/b21350

Guéant, O. (2017), ‘Optimal market making’, *Applied Mathematical Finance* **24**(2), 112–154. doi:10.1080/1350486X.2017.1342552

Guéant, O., Lehalle, C.-A. and Fernandez-Tapia, J. (2013), ‘Dealing with the inventory risk: a solution to the market making problem’, *Mathematics and financial economics* **7**(4), 477–507. doi:10.1007/s11579-012-0087-0

Guéant, O. and Manziuk, I. (2019), ‘Deep reinforcement learning for market making in corporate bonds: beating the curse of dimensionality’, *Applied Mathematical Finance* **26**(5), 387–452. doi:10.1080/1350486X.2020.1714455

Guilbaud, F. and Pham, H. (2013), ‘Optimal high-frequency trading with limit and market orders’, *Quantitative Finance* **13**(1), 79–94. doi:10.1080/14697688.2012.708779

Hagströmer, B. and Nordén, L. (2013), ‘The diversity of high-frequency traders’, *Journal of Financial Markets* **16**(4), 741–770. doi:10.1016/j.finmar.2013.05.009

Haider, A., Wang, H., Scotney, B. and Hawe, G. (2022), ‘Predictive Market Making via Machine Learning’, *Operations Research Forum* **3**(1), 1–21. doi:10.1007/s43069-022-00124-0

Handa, P. and Schwartz, R. A. (1996), ‘Limit order trading’, *Journal of Finance* **51**(5), 1835–1861. doi:10.1111/j.1540-6261.1996.tb05228.x

Hirschey, N. (2021), ‘Do high-frequency traders anticipate buying and selling pressure?’, *Management Science* **67**(6), 3321–3345. doi:10.1287/mnsc.2020.3608

Ho, T. and Stoll, H. R. (1981), ‘Optimal dealer pricing under transactions and return uncertainty’, *Journal of Financial Economics* **9**(1), 47–73. doi:10.1016/0304-405X(81)90020-9

Hochreiter, S. (1998), ‘The vanishing gradient problem during learning recurrent neural nets and problem solutions’, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **6**(02), 107–116. doi:10.1142/S0218488598000094

Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J. (2001), Gradient flow in recurrent nets: The difficulty of learning longterm dependencies, *in* ‘A Field Guide to Dynamical Recurrent Networks’, IEEE Press, pp. 237–243. doi:10.1109/9780470544037

Hummingbot (2021), A comprehensive guide to Avellaneda and Stoikov’s market-making strategy, *in* ‘Hummingbot Academy’, Hummingbot Foundation. Available from https://blog.hummingbot.org/2021-04-avellaneda-stoikov-market-making-strategy/.

Hummingbot (2023), Documentation of `avellaneda_market_making` package, *in* ‘Hummingbot Docs’, Hummingbot Foundation. Available from https://docs.hummingbot.org/strategies/avellaneda-market-making/.

Kercheval, A. N. and Zhang, Y. (2015), ‘Modelling high-frequency limit order book dynamics with support vector machines’, *Quantitative Finance* **15**(8), 1315–1329. doi:10.1080/14697688.2015.1032546

Kühn, C. and Stroh, M. (2010), ‘Optimal portfolios of a small investor in a limit order market: a shadow price approach’, *Mathematics and Financial Economics* **3**(2), 45–72. doi:10.1007/s11579-010-0027-9

Laruelle, S. (2013), 'Faisabilité de l'apprentissage des paramètres d'un algorithme de trading sur des données réelles', *Hors-Série Microstructure des Marchés* **1**, 1–25. Available from https://events.chairefdd.org/faisabilite-de-l%E2%80%99apprentissage-des-parametres-d%E2%80%99un-algorithme-de-trading-sur-des-donnees-reelles/.

LeCun, Y. A., Bottou, L., Orr, G. B. and Müller, K.-R. (2012), Efficient BackProp, *in* 'Neural networks: Tricks of the trade', Springer, pp. 9–48. doi:10.1007/978-3-642-35289-8_3

LeCun, Y., Bengio, Y. and Hinton, G. (2015), 'Deep learning', *Nature* **521**(7553), 436–444. doi:10.1038/nature14539

Li, X., Deng, X., Zhu, S., Wang, F. and Xie, H. (2014), 'An intelligent market making strategy in algorithmic trading', *Frontiers of Computer Science* **8**(4), 596–608. doi:10.1007/s11704-014-3312-6

Lim, Y.-S. and Gorse, D. (2018), Reinforcement learning for high-frequency market making, *in* 'ESANN 2018-Proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning', ESANN, pp. 521–526. Available from https://www.esann.org/sites/default/files/proceedings/legacy/es2018-50.pdf.

Linnainmaa, J. (2010), 'Do limit orders alter inferences about investor performance and behavior?', *Journal of Finance* **65**(4), 1473–1506. doi:10.1111/j.1540-6261.2010.01576.x

Makarov, I. and Schoar, A. (2020), 'Trading and arbitrage in cryptocurrency markets', *Journal of Financial Economics* **135**(2), 293–319. doi:10.1016/j.jfineco.2019.07.001

Maslov, S. and Mills, M. (2001), 'Price fluctuations from the order book perspective—empirical facts and a simple model', *Physica A: Statistical Mechanics and its Applications* **299**(1-2), 234–246. doi:10.1016/S0378-4371(01)00301-6

Menkveld, A. J. (2013), 'High frequency trading and the new market makers', *Journal of Financial Markets* **16**(4), 712–740. doi:10.1016/j.finmar.2013.06.006

Olah, C. (2015), 'Understanding LSTM networks'. Available from https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Palguna, D. and Pollak, I. (2013), Non-parametric prediction in a limit order book, *in* '2013 IEEE Global Conference on Signal and Information Processing', IEEE, pp. 1139–1139. doi:10.1109/GlobalSIP.2013.6737102

Palguna, D. and Pollak, I. (2016), 'Mid-price prediction in a limit order book', *IEEE Journal of Selected Topics in Signal Processing* **10**(6), 1083–1092. doi:10.1109/JSTSP.2016.2593060

Paquin, D. (2020), 'Coinbase Pro API'. Available from https://github.com/danpaquin/coinbasepro-python.

Pascanu, R., Gulcehre, C., Cho, K. and Bengio, Y. (2014), How to construct deep recurrent neural networks, *in* 'Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014)'. doi:10.48550/arxiv.1312.6026

Potters, M. and Bouchaud, J.-P. (2003), 'More statistical properties of order books and price impact', *Physica A: Statistical Mechanics and its Applications* **324**(1-2), 133–140. doi:10.1016/S0378-4371(02)01896-4

Rojas, R. (2013), *Neural networks: a systematic introduction*, Springer Science & Business Media. doi:10.1007/978-3-642-61068-4

Roscoe, J. (1975), *Fundamental Research Statistics for the Behavioral Sciences*, Editors' Series in Marketing, 2 edn, Holt, Rinehart and Winston

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986*a*), Learning Internal Representations by Error Propagation, *in* J. McClelland and D. Rumelhart, eds, 'Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations', MIT Press, Cambridge, MA, USA, pp. 318–362. Available from https://apps.dtic.mil/dtic/tr/fulltext/u2/a164453.pdf.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986*b*), 'Learning representations by back-propagating errors', *Nature* **323**(6088), 533–536. doi:10.1038/323533a0

Sadighian, J. (2019), 'Deep reinforcement learning in cryptocurrency market making'. arXiv preprint, unpublished. doi:10.48550/arxiv.1911.08647

Sandås, P. (2001), 'Adverse Selection and Competitive Market Making: Empirical Evidence from a Limit Order Market', *Review of Financial Studies* **14**(3), 705–734. doi:10.1093/rfs/14.3.705

Schnaubelt, M. (2022), 'Deep reinforcement learning for the optimal placement of cryptocurrency limit orders', *European Journal of Operational Research* **296**(3), 993–1006. doi:10.1016/j.ejor.2021.04.050

Schnaubelt, M., Rende, J. and Krauss, C. (2019), 'Testing stylized facts of bitcoin limit order books', *Journal of Risk and Financial Management* **12**(1), 25. doi:10.3390/jrfm12010025

Seppi, D. J. (1997), 'Liquidity Provision with Limit Orders and a Strategic Specialist', *Review of Financial Studies* **10**(1), 103–150. doi:10.1093/rfs/10.1.103

Silantyev, E. (2019), 'Order flow analysis of cryptocurrency markets', *Digital Finance* **1**(1), 191–218. doi:10.1007/s42521-019-00007-w

Spooner, T., Fearnley, J., Savani, R. and Koukorinis, A. (2018), Market Making via Reinforcement Learning, *in* 'Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems', AAMAS '18, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 434–442. Available from https://dl.acm.org/doi/10.5555/3237383.3237450.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014), 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting', *Journal of Machine Learning Research* **15**(56), 1929–1958. Available from https://dl.acm.org/doi/10.5555/2627435.2670313.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M. and Iosifidis, A. (2017), Forecasting stock prices from the limit order book using convolutional neural networks, *in* '2017 IEEE 19th conference on business informatics (CBI)', Vol. 1, IEEE, pp. 7–12. doi:10.1109/CBI.2017.23

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M. and Iosifidis, A. (2020), 'Using deep learning for price prediction by exploiting stationary limit order book features', *Applied Soft Computing* **93**, 106401. doi:10.1016/j.asoc.2020.106401

Weber, P. and Rosenow, B. (2005), 'Order book approach to price impact', *Quantitative Finance* **5**(4), 357–364. doi:10.1080/14697680500244411

Zaznov, I., Kunkel, J., Dufour, A. and Badii, A. (2022), 'Predicting stock price changes based on the limit order book: A survey', *Mathematics* **10**(8). doi:10.3390/math10081234

Zhang, Z., Zohren, S. and Roberts, S. (2019), 'DeepLOB: Deep convolutional neural networks for limit order books', *IEEE Transactions on Signal Processing* **67**(11), 3001–3012. doi:10.1109/TSP.2019.2907260

Zheng, B., Moulines, E. and Abergel, F. (2013), 'Price jump prediction in limit order book', *Journal of Mathematical Finance* **3**(2), 242–255. doi:10.4236/jmf.2013.32024

Appendix

4.A Price level consumption by trade size percentile

Figure 4.18 presents the average number of price levels consumed by market orders as a function of R^b and R^a, respectively. We determine the number of price levels consumed based on the last price level a market order consumes. Hence, a price level consumption of 3 indicates that the market order matches with all orders of the first and second best price level, and all or some orders of the third best price level. As relevant price level, we consider price priority at the time of the order arrival. The minimum price level consumed is 1, i.e., the order has been fully matched with open limit orders from the top price level. Figure 4.18 suggests that larger trades are associated with larger margins earned by dealers, and R^b (R^a) thus may indicate the number of ask (bid) price levels consumed by the order as assumed in Section 4.4.1.

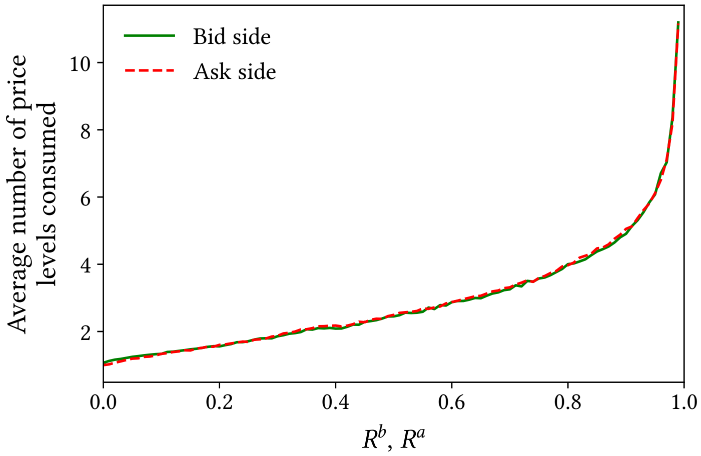

Figure 4.18: Average number of bid (green) and ask (red) price levels consumed as a function of R^a and R^b, respectively. We consider partially consumed price levels as fully consumed price levels.

4.B Conditions for inequalities between trading strategies

4.B.i Basics

This section examines what relationship must exist between $\widehat{R}^b$ and $\widehat{R}^a$ for inequalities (4.8) to (4.11) to be valid. In the derivation of necessary and sufficient conditions of the inequalities (4.8) to (4.11), only the following two types of problems occur ($x, y \in \mathbb{R}$):

- **Type 1:** $|x + y| \geq |x|$
- **Type 2:** $|x - y| \geq |x|$

We are looking for necessary and sufficient conditions for the validity of the respective inequality in the form of conditions on the real numbers x and y or on their relation to each other. We first focus on these conditions in Appendix 4.B.i, before focusing on the initial inequalities in Appendices 4.B.ii to 4.B.v.

Conditions for the validity of Type 1 inequality

The inequality $|x + y| \geq |x|$ holds under the following conditions for x and y:

1. **Case** $x \geq 0$: The inequality holds if
   a) $y \geq 0$ **or**
   b) $y < 0$ **and** $-y > 2 \cdot x$
2. **Case** $x < 0$: The inequality holds if
   a) $y \leq 0$ **or**
   b) $y > 0$ **and** $y > -2 \cdot x$

Conditions for the validity of Type 2 inequality

The inequality $|x - y| \geq |x|$ holds under the following conditions for x and y:

1. **Case** $x \geq 0$: The inequality holds if
   a) $y \leq 0$ **or**
   b) $y > 0$ **and** $y > 2 \cdot x$
2. **Case** $x < 0$: The inequality holds if
   a) $y \geq 0$ **or**
   b) $y < 0$ **and** $-y > -2 \cdot x$

4.B.ii Inequality of directional forecast-adjusted ask prices

Inequality (4.8) represents the inequality between directional forecast-adjusted ask prices, r^a_{DF}, and ask prices of the consolidated model, r^a_{SPDF}, in the form

$$|r^a_{SPDF} - r^a_{AS}| \geq |r^a_{DF} - r^a_{AS}|.$$

Transformations of the initial inequality

For $\alpha_1, \alpha_2 > 0$, it applies

$$r^a_{SPDF} - r^a_{AS} = r_{DF} - r_{AS} + \frac{1}{2}\frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)$$

and

$$r^a_{DF} - r^a_{AS} = r_{DF} - r_{AS}$$

and further

$$r_{DF} - r_{AS} = s + \left(sgn(q)\frac{1}{\alpha_2} \cdot \widehat{TFI} - 1\right) q\gamma\sigma^2(T-t) - \left[s - q\gamma\sigma^2(T-t)\right] =$$

$$sgn(q)\frac{1}{\alpha_2} \cdot \widehat{TFI} \cdot q\gamma\sigma^2(T-t) = \widehat{TFI} \cdot \frac{|q|\gamma\sigma^2}{\alpha_2}(T-t) = \frac{\widehat{R}^b - \widehat{R}^a}{\widehat{R}^b + \widehat{R}^a} \cdot \frac{|q|\gamma\sigma^2}{\alpha_2}(T-t). \tag{4.12}$$

It should be noted here that it applies

$$(r_{DF} - r_{AS}) \sim 0 \iff \left(\widehat{R}^b - \widehat{R}^a\right) \sim 0 \iff \widehat{R}^b \sim \widehat{R}^a \tag{4.13}$$

where $\sim$ denotes the possible equality and inequality signs $=, <, \leq, \geq, >$.

Conditions for the validity of the inequality

Now we can transform the initial inequality and get

$$|r^a_{SPDF} - r^a_{AS}| \geq |r^a_{DP} - r^a_{AS}|$$

$$\iff \left|(r_{DF} - r_{AS}) + \left(\frac{1}{2}\frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)\right)\right| \geq |r_{DF} - r_{AS}|$$

which, according to the case distinction in Section 4.B.i, with

- $x = r_{DF} - r_{AS}$
- $y = \frac{1}{2}\frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)$

holds for

1. **Case** $x \geq 0 \iff r_{DF} - r_{AS} \geq 0 \iff \widehat{R}^b \geq \widehat{R}^a$ **if**
   a) $y \geq 0 \iff \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right) \geq 0 \iff \widehat{R}^b \geq \frac{1}{2}$ **or**
   b) $y < 0 \iff \widehat{R}^b < \frac{1}{2}$ **and**
   $$-y > 2\cdot x \iff \frac{1}{2\alpha_1}\left(1 - 2\widehat{R}^b\right)\left(\delta^a + \delta^b\right) > 2\cdot\left(r_{DF} - r_{AS}\right) = 2\cdot\frac{\widehat{R}^b - \widehat{R}^a}{\widehat{R}^b + \widehat{R}^a}\cdot\frac{|q|\gamma\sigma^2}{\alpha_2}(T-t)$$
2. **Case** $x < 0 \iff r_{DF} - r_{AS} < 0 \iff \widehat{R}^b < \widehat{R}^a$ **if**
   a) $y \leq 0 \iff \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right) \leq 0 \iff \widehat{R}^b \leq \frac{1}{2}$ **or**
   b) $y > 0 \iff \widehat{R}^b > \frac{1}{2}$ **and**
   $$y > -2\cdot x \iff \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right) > 2\cdot\left(r_{AS} - r_{DF}\right) = 2\cdot\frac{\widehat{R}^a - \widehat{R}^b}{\widehat{R}^a + \widehat{R}^b}\cdot\frac{|q|\gamma\sigma^2}{\alpha_2}(T-t)$$

4.B.iii Inequality of directional forecast-adjusted bid prices

Inequality (4.9) represents the inequality between directional forecast-adjusted bid prices, r^b_{DF}, and bid prices of the consolidated model, r^b_{SPDF}, in the form

$$|r^b_{SPDF} - r^b_{AS}| \geq |r^b_{DF} - r^b_{AS}|.$$

Transformations of the initial inequality

For $\alpha_1, \alpha_2 > 0$, it applies

$$r^b_{SPDF} - r^b_{AS} = r_{DF} - r_{AS} - \frac{1}{2}\frac{1}{\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)$$

and

$$r^b_{DF} - r^b_{AS} = r_{DF} - r_{AS}$$

Further, the relationships demonstrated by equations (4.12) and (4.13) apply.

Conditions for the validity of the inequality

Now we can transform the initial inequality and get

$$|r^b_{SPDF} - r^b_{AS}| \geq |r^b_{DF} - r^b_{AS}|$$

$$\Longleftrightarrow \left|(r_{DF} - r_{AS}) - \left(\frac{1}{2}\frac{1}{\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)\right)\right| \geq |r_{DF} - r_{AS}|$$

which, according to the case distinction in Section 4.B.i, with

- $x = r_{DF} - r_{AS}$
- $y = \frac{1}{2}\frac{1}{\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)$

holds for

1. **Case** $x \geq 0 \Longleftrightarrow r_{DF} - r_{AS} \geq 0 \Longleftrightarrow \widehat{R}^b \geq \widehat{R}^a$ **if**
   a) $y \leq 0 \Longleftrightarrow \frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right) \leq 0 \Longleftrightarrow \widehat{R}^a \leq \frac{1}{2}$ **or**

   b) $y > 0 \Longleftrightarrow \widehat{R}^a > \frac{1}{2}$ **and**

$$y > 2 \cdot x \Longleftrightarrow \frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right) > 2 \cdot \left(r_{DF} - r_{AS}\right) = 2 \cdot \frac{\widehat{R}^b - \widehat{R}^a}{\widehat{R}^b + \widehat{R}^a} \cdot \frac{|q|\gamma\sigma^2}{\alpha_2}(T - t)$$

2. **Case** $x < 0 \Longleftrightarrow r_{DF} - r_{AS} < 0 \Longleftrightarrow \widehat{R}^b < \widehat{R}^a$ **if**
   a) $y \geq 0 \Longleftrightarrow \frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right) \geq 0 \Longleftrightarrow \widehat{R}^a \geq \frac{1}{2}$ **or**

   b) $y < 0 \Longleftrightarrow \widehat{R}^a < \frac{1}{2}$ **and**

$$-y > -2 \cdot x \Longleftrightarrow \frac{1}{2\alpha_1}\left(1 - 2\widehat{R}^a\right)\left(\delta^a + \delta^b\right) > 2 \cdot \left(r_{AS} - r_{DF}\right) = 2 \cdot \frac{\widehat{R}^a - \widehat{R}^b}{\widehat{R}^a + \widehat{R}^b} \cdot \frac{|q|\gamma\sigma^2}{\alpha_2}(T - t)$$

4.B.iv Inequality of spread-adjusted ask prices

Inequality (4.10) represents the inequality between spread-adjusted ask prices, r^a_{SP}, and the ask prices of the consolidated model, r^a_{SPDF}, in the form

$$|r^a_{SPDF} - r^a_{AS}| \geq |r^a_{SP} - r^a_{AS}|.$$

Transformations of the initial inequality

For $\alpha_1, \alpha_2 > 0$, it applies

$$r^a_{SPDF} - r^a_{AS} = r_{DF} - r_{AS} + \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)$$

and

$$r^a_{SP} - r^a_{AS} = r_{AS} + \frac{1}{2}\left(1 + \frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\right)\left(\delta^a + \delta^b\right) - \left(r_{AS} + \frac{1}{2}\left(\delta^a + \delta^b\right)\right) = \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)$$

Further, the relationships demonstrated by equations (4.12) and (4.13) apply.

Conditions for the validity of the inequality

Now we can transform the initial inequality and get

$$|r^a_{SPDF} - r^a_{AS}| \geq |r^a_{SP} - r^a_{AS}|$$

$$\Longleftrightarrow \left|(r_{DF} - r_{AS}) + \left(\frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)\right)\right| \geq \left|\left(\frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)\right)\right|$$

$$\Longleftrightarrow \left|\left(\frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)\right) + (r_{DF} - r_{AS})\right| \geq \left|\left(\frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)\right)\right|$$

which, according to the case distinction in Section 4.B.i, with

- $x = \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)$
- $y = r_{DF} - r_{AS}$

holds for

1. **Case** $x \geq 0 \Longleftrightarrow \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right) \geq 0 \Longleftrightarrow \widehat{R}^b \geq \frac{1}{2}$ **if**
   a) $y \geq 0 \Longleftrightarrow r_{DF} - r_{AS} \geq 0 \Longleftrightarrow \widehat{R}^b \geq \widehat{R}^a$ **or**
   b) $y < 0 \Longleftrightarrow r_{DF} - r_{AS} < 0 \Longleftrightarrow \widehat{R}^b < \widehat{R}^a$ **and**

$$-y > 2 \cdot x \Longleftrightarrow r_{AS} - r_{DF} > \frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)$$

$$\Longleftrightarrow \frac{\widehat{R}^a - \widehat{R}^b}{\widehat{R}^a + \widehat{R}^b} \cdot \frac{|q|\gamma\sigma^2}{\alpha_2}(T - t) > \frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right)$$

2. **Case** $x < 0 \Longleftrightarrow \frac{1}{2\alpha_1}\left(2\widehat{R}^b - 1\right)\left(\delta^a + \delta^b\right) < 0 \Longleftrightarrow \widehat{R}^b < \frac{1}{2}$ **if**
   a) $y \leq 0 \Longleftrightarrow r_{DF} - r_{AS} \leq 0 \Longleftrightarrow \widehat{R}^b \leq \widehat{R}^a$ **or**
   b) $y > 0 \Longleftrightarrow r_{DF} - r_{AS} > 0 \Longleftrightarrow \widehat{R}^b > \widehat{R}^a$ **and**

$$y > -2 \cdot x \Longleftrightarrow r_{DF} - r_{AS} > \frac{1}{\alpha_1}\left(1 - 2\widehat{R}^b\right)\left(\delta^a + \delta^b\right)$$

$$\Longleftrightarrow \frac{\widehat{R}^b - \widehat{R}^a}{\widehat{R}^b + \widehat{R}^a} \cdot \frac{|q|\gamma\sigma^2}{\alpha_2}(T - t) > \frac{1}{\alpha_1}\left(1 - 2\widehat{R}^b\right)\left(\delta^a + \delta^b\right)$$

4.B.v Inequality of spread-adjusted bid prices

Inequality (4.11) represents the inequality between spread-adjusted bid prices, r^b_{SP}, and the bid prices of the consolidated model, r^b_{SPDF}, in the form

$$|r^b_{SPDF} - r^b_{AS}| \geq |r^b_{SP} - r^b_{AS}|.$$

Transformations of the initial inequality

For $\alpha_1, \alpha_2 > 0$, it applies

$$r^b_{SPDF} - r^b_{AS} = r_{DF} - r_{AS} - \frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)$$

and

$$r^b_{SP} - r^b_{AS} = r_{AS} - \frac{1}{2}\left(1 + \frac{1}{\alpha_1}\left(2\widehat{R}^b - 1\right)\right)\left(\delta^a + \delta^b\right) - \left(r_{AS} - \frac{1}{2}\left(\delta^a + \delta^b\right)\right) =$$
$$-\frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right).$$

Further, the relationships demonstrated by equations (4.12) and (4.13) apply.

Conditions for the validity of the inequality

Now, analogous to Section 4.B.iv, we can transform the initial inequality and get

$$|r^b_{SPDF} - r^b_{AS}| \geq |r^b_{SP} - r^b_{AS}|$$

$$\Longleftrightarrow \left|(r_{DF} - r_{AS}) - \left(\frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)\right)\right| \geq \left|\left(\frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)\right)\right|$$
$$\Longleftrightarrow \left|\left(\frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)\right) - (r_{DF} - r_{AS})\right| \geq \left|\left(\frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)\right)\right|$$

which, according to the case distinction in Section 4.B.i, with

- $x = \frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)$
- $y = r_{DF} - r_{AS}$

holds for

1. **Case** $x \geq 0 \Longleftrightarrow \frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right) \geq 0 \Longleftrightarrow \widehat{R}^a \geq \frac{1}{2}$ **if**
   a) $y \leq 0 \Longleftrightarrow r_{DF} - r_{AS} \leq 0 \Longleftrightarrow \widehat{R}^b \leq \widehat{R}^a$ **or**
   b) $y > 0 \Longleftrightarrow r_{DF} - r_{AS} > 0 \Longleftrightarrow \widehat{R}^b > \widehat{R}^a$ **and**

$$y > 2 \cdot x \Longleftrightarrow r_{DF} - r_{AS} > \frac{1}{\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)$$
$$\Longleftrightarrow \frac{\widehat{R}^b - \widehat{R}^a}{\widehat{R}^b + \widehat{R}^a} \cdot \frac{|q|\gamma\sigma^2}{\alpha_2}(T - t) > \frac{1}{\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right)$$

2. **Case** $x < 0 \iff \frac{1}{2\alpha_1}\left(2\widehat{R}^a - 1\right)\left(\delta^a + \delta^b\right) < 0 \iff \widehat{R}^a < \frac{1}{2}$ **if**

   a) $y \geq 0 \iff r_{DF} - r_{AS} \geq 0 \iff \widehat{R}^b \geq \widehat{R}^a$ **or**

   b) $y < 0 \iff r_{DF} - r_{AS} < 0 \iff \widehat{R}^b < \widehat{R}^a$ **and**

$$-y > -2 \cdot x \iff -(r_{DF} - r_{AS}) > \frac{1}{\alpha_1}\left(1 - 2\widehat{R}^a\right)\left(\delta^a + \delta^b\right)$$
$$\iff \frac{\widehat{R}^a - \widehat{R}^b}{\widehat{R}^a + \widehat{R}^b} \cdot \frac{|q|\gamma\sigma^2}{\alpha_2}(T - t) > \frac{1}{\alpha_1}\left(1 - 2\widehat{R}^a\right)\left(\delta^a + \delta^b\right)$$

4.C Numerical examples of inequalities between trading strategies

Figures 4.19 and 4.20 illustrate numerical examples for the inequalities (4.8) to (4.11) and explore under which combinations of $\widehat{R}^a$ and $\widehat{R}^b$ the inequalities hold given market standard

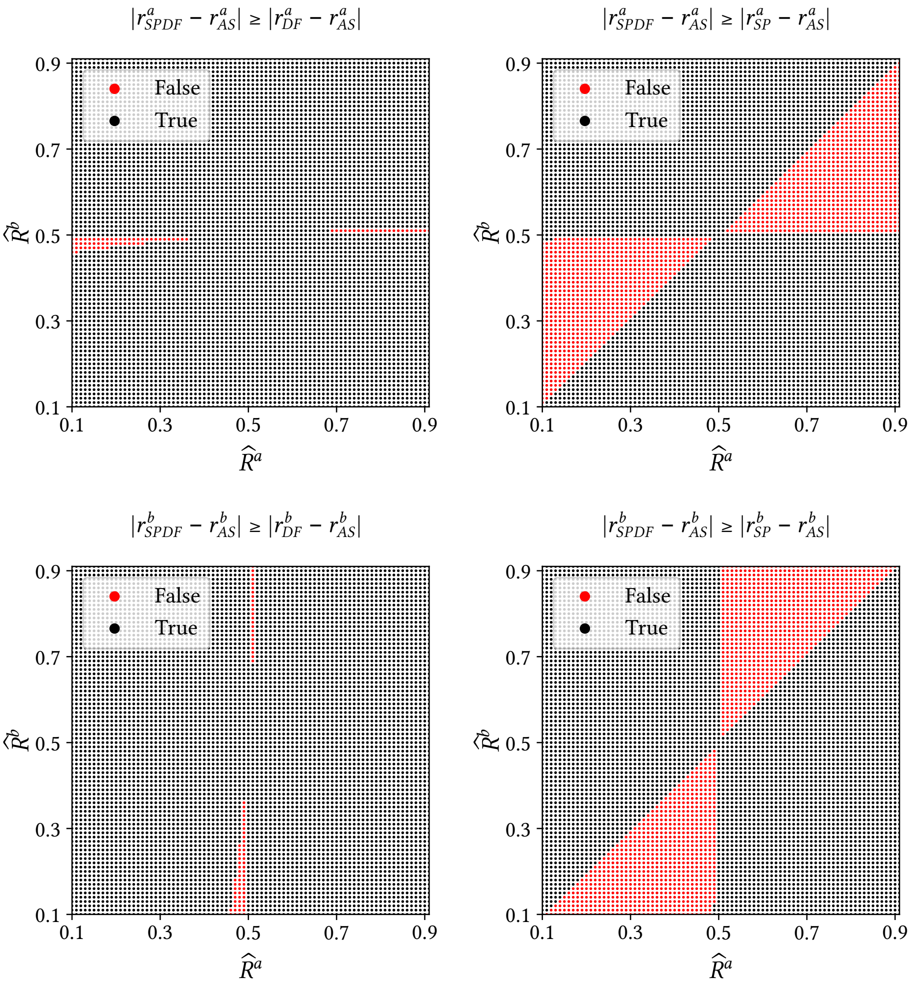

Figure 4.19: Inequalities between *SPDF*, *SP*, and *DF* for small inventory ($q = 1$). We test the inequalities (4.8) to (4.11) for all combinations of $\widehat{R}^a$ and $\widehat{R}^b$ with a granularity of 0.01 in the range $\widehat{R}^a, \widehat{R}^b = [0.1, 0.9]$. We assume the following parameters: $\alpha_1 = 1$, $\alpha_2 = 1$, $s = 100$, $\gamma = 0.5$, $\sigma = 0.3$, $T = 1$, $t = 0.3$, $\kappa = 2$.

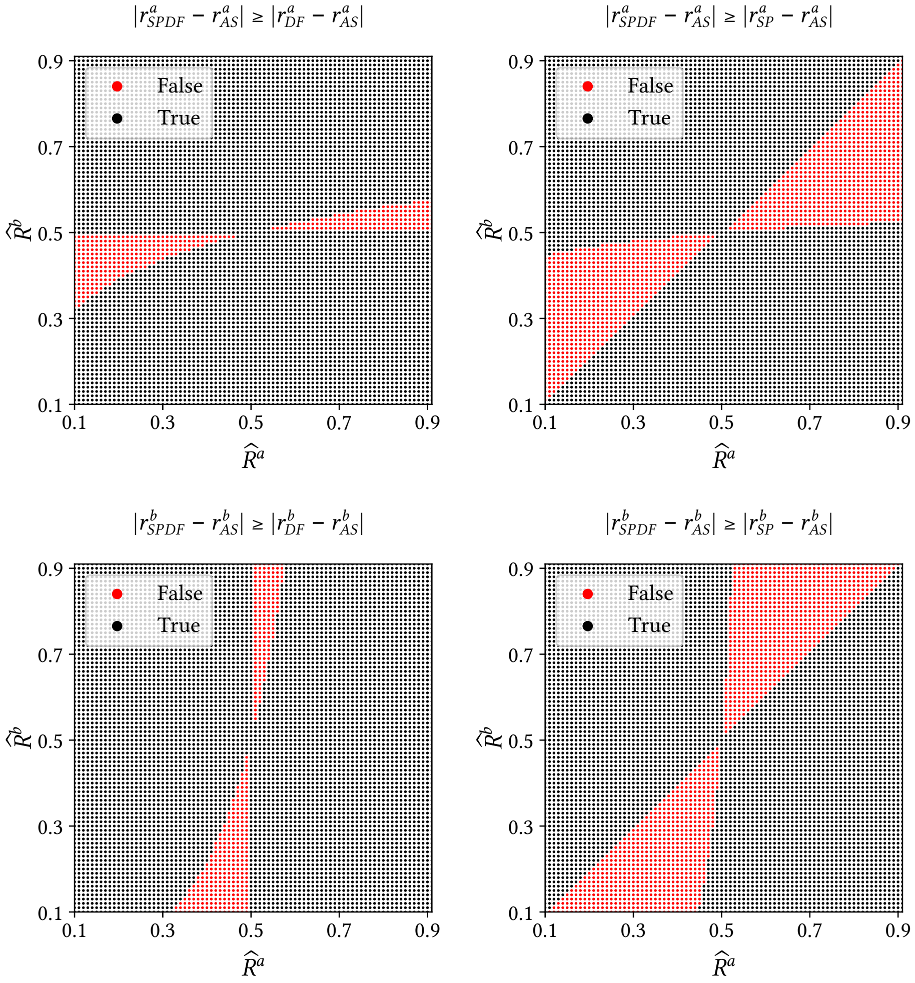

Figure 4.20: Inequalities between *SPDF*, *SP*, and *DF* for large inventory ($q = 5$). We test the inequalities (4.8) to (4.11) for all combinations of $\widehat{R}^a$ and $\widehat{R}^b$ with a granularity of 0.01 in the range $\widehat{R}^a, \widehat{R}^b = [0.1, 0.9]$. We assume the following parameters: $\alpha_1 = 1$, $\alpha_2 = 1$, $s = 100$, $\gamma = 0.5$, $\sigma = 0.3$, $T = 1$, $t = 0.3$, $\kappa = 2$.

values for γ, σ, and κ. To this end, we consider two relevant cases: First, Figure 4.19 assumes a small inventory ($q = 1$); second, Figure 4.20 assumes a large inventory ($q = 5$). 'True' (resp. 'False') indicates that the inequality holds (resp. does not hold).

In both Figures 4.19 and 4.20, the two top (bottom) charts focus on ask (bid) prices, whereas the two left (right) charts focus on *DF* (*SP*) strategy. Figure 4.19 emphasizes that for small q, *DF* adjusts prices in the same direction as *SP* for most combinations of $\widehat{R}^a$ and $\widehat{R}^b$. Only when predicted buy (sell) volume is neither small nor large with predicted sell (buy) volume being either very small or very large, *DF* offsets the *SP* adjustment to sell (buy) prices. This is due to the fact that the conditions given in Sections 4.B.ii and 4.B.iii are valid for small q, e.g., in case of inequality (4.8), if $\widehat{R}^b \geq \widehat{R}^a$ and $\widehat{R}^b < \frac{1}{2}$, small q fulfill the condition $\frac{1}{2\alpha_1}\left(1 - 2\widehat{R}^b\right)\left(\delta^a + \delta^b\right) > 2 \cdot \left(r_{DF} - r_{AS}\right)$ for most common values for κ, σ and γ.

On the other hand, Figure 4.20 shows that for increasing inventory, this relationship changes and *DF* more often offsets *SP* strategy, which may be due to the fact that the conditions for SP strategy to act in the same direction as DF are stronger: For instance, inequality 4.10 holds for $\widehat{R}^b < \frac{1}{2}$ and $\widehat{R}^b > \widehat{R}^a$ if $r_{DF} - r_{AS} > \frac{1}{\alpha_1}\left(1 - 2\widehat{R}^b\right)\left(\delta^a + \delta^b\right)$, which is opposite to the condition for inequality (4.8) and therefore holds only for large q. Hence, we infer that DF rarely offsets the SP strategy in the consolidated model, whereas SP largely offsets DF in the consolidated model, except for the a) cases defined in Sections 4.B.iv and 4.B.v.

4.D Distribution of prediction errors

To gain a better understanding of the conditions under which the prediction models perform either well or poorly, Figure 4.21 illustrates the mean squared prediction error as a function of the percentile groups of $\widehat{R}^b$ and $\widehat{R}^a$. Thus, the value plotted at the 20th percentile reflects the mean squared prediction error of all predictions that fall between the 19th and 20th percentiles of the respective prediction distribution. The plot illustrates that RNN predictions between the 20th and 80th percentiles are more accurate than small or large RNN predictions. For example, when RNN delivers a small prediction, the potential prediction error is larger on average than for an average prediction. In contrast, SFN achieves smaller prediction errors for extreme predictions. Nevertheless, RNN is superior to SFN in all percentile groups.

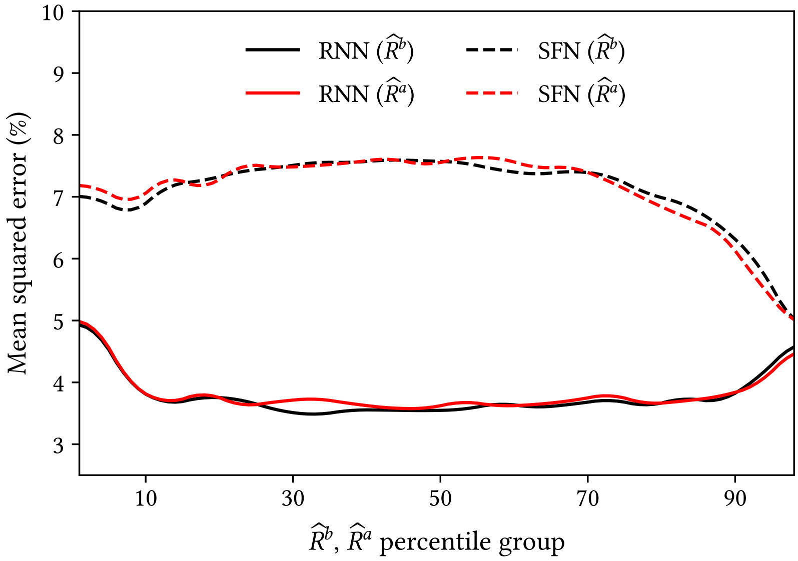

Figure 4.21: Mean squared prediction error as a function of $\widehat{R}^b$ and $\widehat{R}^a$. Black (red) lines represent buy (sell) predictions, where solid (dashed) lines represent RNN (SFN) predictions. According to Section 4.6.1, we calculate prediction errors by $\widehat{R}^b - R^b$ and $\widehat{R}^a - R^a$, respectively.

4.E Prediction accuracy of classification task

Table 4.3 reports prediction quality of the trade size classification task, where the positive class represents a trade size percentile rank greater than 0.5, and the negative class represents a percentile rank less than or equal 0.5. Accuracy ACC reports the number of true classifications in all observations (see equation (2.13)). True Positive (Negative) Rate reports the share of correct classifications in all positive (negative) observations. We calculate True Positive Rate

TPR analogous to equation (3.6) by $TPR = TP/(TP + FN)$ and True Negative Rate TNR analogous to equation (3.7) by $TNR = TN/(TN + FP)$.

	RNN			SFN		
	$\widehat{R}^b$	$\widehat{R}^a$	All	$\widehat{R}^b$	$\widehat{R}^a$	All
ACC	0.82	0.82	0.82	0.64	0.64	0.64
TPR	0.82	0.81	0.82	0.65	0.64	0.64
TNR	0.83	0.83	0.83	0.63	0.64	0.64

Table 4.3: Prediction accuracy of the trade size classification task

4.F Net position percentiles

Figure 4.22 shows the percentiles of net positions held as an equal-weighted average for both ETH-USD and BTC-USD pairs. For instance, the 20th percentile of net positions held under *SP* strategy is −0.3. Following Section 4.5.4, the inventory limit is ±0.5 BTC or ±0.5 ETH, respectively. The chart shows that in more than 50% (less than 20%) of all seconds, we are net short (net long), whereas in about 30% of all seconds the position is neutral. As a result, we post sell orders on average deeper than buy orders in the order book (see Figure 4.14).

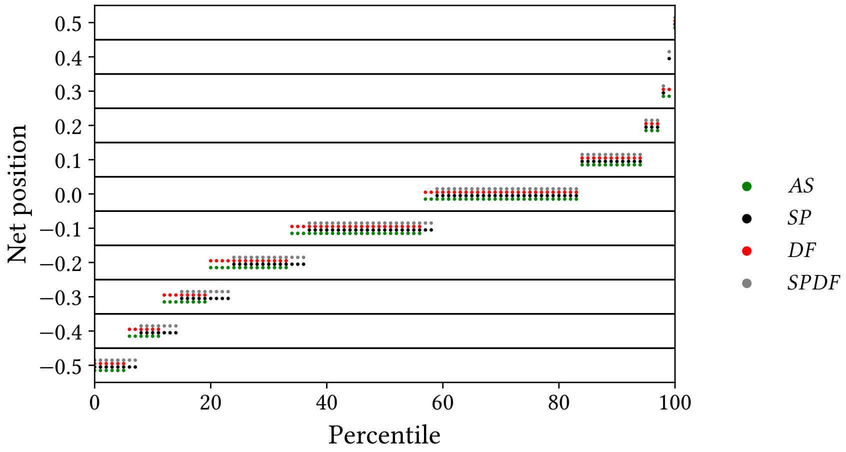

Figure 4.22: Percentiles of net positions held during the total trading period. We consider net positions in one-second intervals and calculate percentiles separately for each trading strategy. The net position can take any value out of $[-0.5, -0.4, -0.3, ..., 0.3, 0.4, 0.5]$.

4.G Depth of orders in the order book by currency pair

Since Figure 4.14 aggregates order books of BTC-USD and ETH-USD, we analyze in Figure 4.23 whether our findings for the aggregated order books also hold for stand-alone order books of the BTC-USD and ETH-USD pairs. Figure 4.23 illustrates that the dynamics of the stand-alone order books are largely similar to the dynamics demonstrated by Figure 4.14, which suggests that the relationships between trading strategies are similar for both BTC-USD and ETH-USD order books, allowing us to consider the order books of BTC-USD and ETH-USD together.

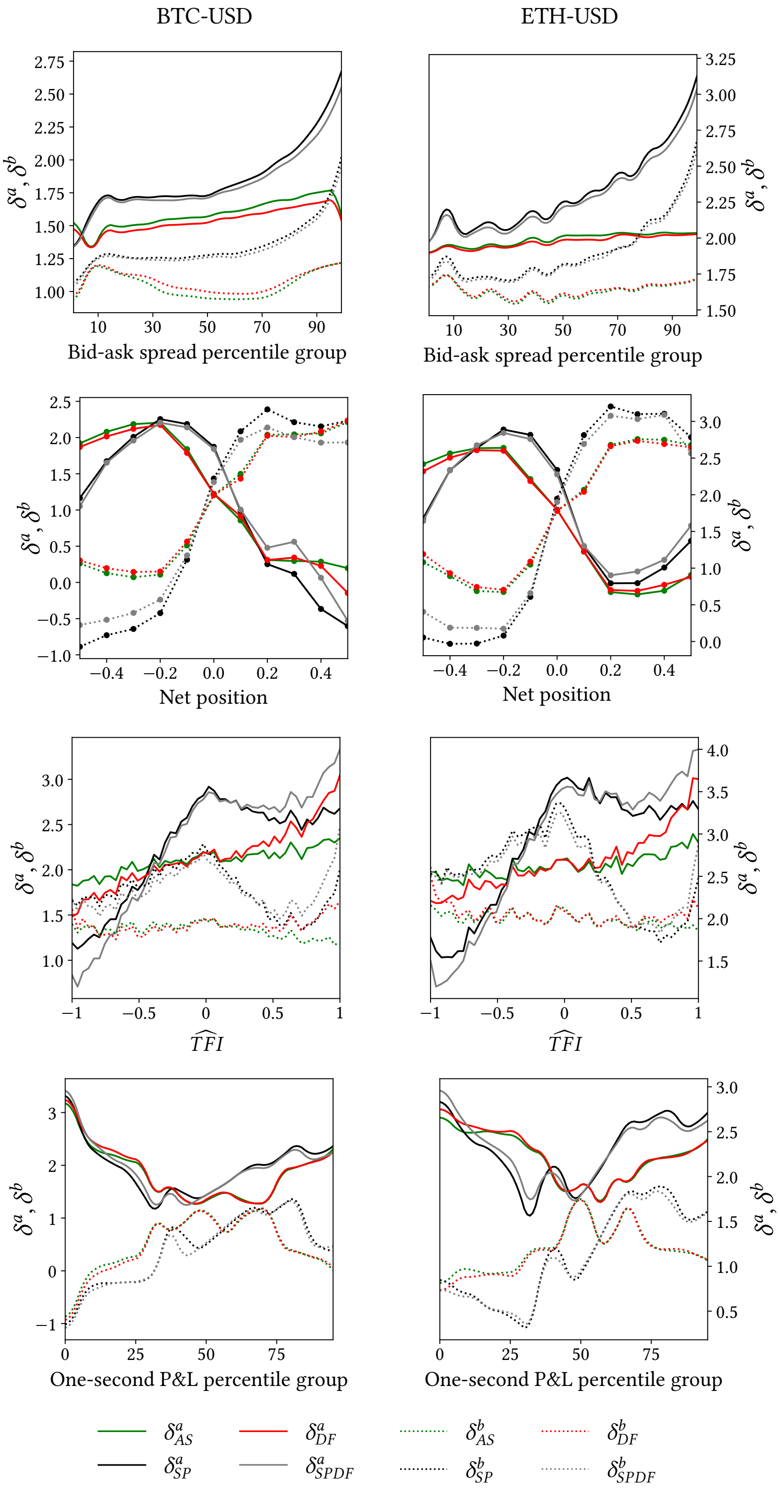

Figure 4.23: Average depth of ask orders δ^a (solid lines) and bid orders δ^b (dotted lines) in the order book for BTC-USD pair (left) and ETH-USD pair (right). SP, DF, and $SPDF$ base on RNN predictions. δ^a and δ^b denote in basis points. `Net position' represents each strategy's individual net position. `Bid-ask spread' is the one-second bid-ask spread at the end of a one-second interval.