UCL

Reinforcement learning for market making in algorithmic sports trading

Author
William De Vena

Student Number
22157928

Supervisor
Prof. Philip Treleaven

Second Supervisor
Dr. Laura Toni

Technical Supervisor
John Goodacre

2022-2023

Final Report - ELEC0054
MSc in Integrated Machine Learning Systems
Dept. of Electrical and Electronic Engineering

Abstract

This research investigates the application of Reinforcement Learning (RL) for market making in sports trading/betting and was conducted in collaboration with Quant Sports Trading Ltd. Market making plays a crucial role in financial markets by providing liquidity, enhancing market efficiency and narrowing the bid-ask spread. In particular, market makers ensure that there is always a ready market for buyers and sellers, by continuously quoting bid and ask prices, while profiting from the bid-ask spread. On the other hand, sports betting involves participants placing wagers on sports event outcomes. Originally dominated by bookmakers, sports betting has been revolutionized by online exchanges which allow participants to bet against each other using back and lay mechanisms, providing the opportunity to profit from both sides of a wager, similar to financial markets, where participants can profit from both buying and selling a security. The emergence of online exchanges has created an intersection with the realm of trading, opening up opportunities for market makers.

This research comprises three primary experiments:

- **Experiment 1: In-depth analysis of sports exchange data.** This experiment focuses on analysing the key features of sports exchange data like volumes, volatility, liquidity, and differences between pre-game and in-play data.
- **Experiment 2: Implementation and testing of baseline models.** It aims at evaluating baseline models to set a benchmark for market making in the sports market. A crucial part of this experiment includes the design of a novel framework for the simulation of a sports trading environment.
- **Experiment 3: Development, training and testing of a novel RL agent.** This last experiment has the objective of developing an agent, using state-of-the-art RL, with the aim of surpassing the performance of the baseline models.

With these investigations, this research presents the following original contributions to science:

- **Insights into sports exchange data:** identification of key characteristics and unique patterns specific to the sports trading market.
- **Novel framework for the simulation of a sports trading environment**: introduction of a novel simulation framework for the training and testing of market making models.
- **Benchmark market making models:** evaluation of baseline models' performance in the sports market, comparing their effectiveness and highlighting their strengths and weaknesses.
- **Novel market making RL agent:** design of a novel RL agent that outperforms the baseline models.

Overall, this research elucidates the promising potential of applying RL for market making in sports trading. The developed RL agents were successful in outperforming the established baseline models, especially in the crucial metric of Profit and Loss (PnL). Additionally, the RL agents were not only effective in learning profitable market making strategies but also demonstrated some flexibility across market conditions different from those they were trained in. Particularly, the agent trained with the Proximal Policy Optimization (PPO) algorithm emerged as the most effective across a range of performance and risk metrics. These findings collectively indicate the viability and effectiveness of RL-based market making strategies in the dynamic and complex environment of sports trading.

Impact statement

In addition to the contribution to science and the academic field, this research holds significant implications and potential impact on the industry of sports trading. In particular, the findings and outcomes of this research have the potential to directly influence and benefit various stakeholders within this industry.

In particular, traders and betting professionals can utilize the developed models and techniques to refine their strategies. By incorporating the insights and methodologies proposed in this research, they can enhance their decision-making processes, optimize risk management strategies, and ultimately improve profitability. The application of these models and techniques can provide valuable guidance and support for traders in navigating the complex and dynamic sports trading environment.

Furthermore, the introduction of sophisticated market making models derived from this research can contribute to the enhancement of the overall efficiency of the sports market. These models can improve the alignment between market prices and true probabilities, resulting in fairer odds for participants. The increased market efficiency not only benefits traders and betting professionals but also provides a more transparent and equitable marketplace for all participants.

Overall, the impact of this research extends beyond academic contributions and has direct implications for industry practitioners. By empowering traders with refined strategies and enhancing market efficiency, this research aims to drive positive transformations within the sports trading industry, leading to improved outcomes for all stakeholders involved.

Acknowledgements

I would like to express my profound gratitude to those who have been instrumental in the successful completion of this thesis.

First, I extend my sincerest appreciation to my first supervisor, Prof. Philip Treleaven, for his unwavering guidance, invaluable insights, and constant encouragement throughout this research journey. Your expertise and mentorship have been pivotal in shaping this work. I am also thankful to my second supervisor Dr. Laura Toni for her guidance and the insights she provided at various stages of this research.

Special thanks to my technical supervisor, John Goodacre from Quant Sports Trading Ltd, for his invaluable technical guidance and expertise. Collaborating with you and drawing from your wealth of knowledge has been a rewarding experience. I would also like to extend my gratitude to the entire team at Quant Sports Trading Ltd, for the support and for providing me with the resources and data necessary for the project.

Continuing on, I would like to dedicate a special word of gratitude to my parents. Their support, immense sacrifices, and constant belief in me have been the bedrock upon which I have built my life journey. It is their teachings, values, and the countless sacrifices they have made that have shaped me and given me the strength and perseverance to pursue this challenging journey. To my parents, who have always placed my dreams and aspirations above their own, thank you. Your love and dedication have made all the difference, and I owe this achievement as much to you as I do to my own efforts.

Last but certainly not least, I owe a debt of gratitude to Beatrice. You have been my rock through the ups and downs, always there to provide support and encouragement. You have not only believed in me but also continuously pushed me to become a better person. Thank you for always standing by me and for all the challenging moments we have overcome together. Your love means the world to me and I could not have been more lucky to have you by my side.

Contents

List of Figures

List of Tables

Chapter 1

This first chapter has the scope of introducing the reader to the context of the research. In particular, the chapter is going to briefly introduce the four main topics, which are algorithmic trading, market making, sports betting and reinforcement learning, together with the intersection of these four which is the main scope of this research. Additionally, the chapter discusses the underlying motivations, objectives, experiments, contributions to science and impact on the sports trading industry. Finally, the thesis structure is going to be outlined.

1 Introduction

In today's technologically driven financial markets, trading has undergone significant transformations, largely attributable to the emergence of algorithmic trading, also known as quantitative trading. As stipulated by the UK's Financial Conduct Authority (FCA) [2], algorithmic trading makes use of algorithms to automate trading actions and parameters of an order, including whether to initiate the order, the timing, price, or quantity of the order, with minimal to no human intervention. The fundamental tools of this trading approach encompass computational power, mathematical models, algorithms, and Machine Learning (ML), designed to amplify speed, efficiency, and accuracy while reducing human errors and biases. In current financial markets, algorithmic trading has become an indispensable instrument, offering benefits such as improved efficiency, liquidity, superior risk management, reduced transaction costs, data-informed decision-making, and broadened market accessibility. These advantages have led to a revolution in trading practices, fostering increased market competitiveness and accessibility [3].

Furthermore, market making is a vital function in financial markets where participants, called market makers or liquidity providers, continuously provide bid and ask prices for securities, ensuring liquidity and facilitating efficient trading. In particular, market makers stand ready to buy or sell at certain prices, allowing other market participants to easily transact and always have the possibility to buy or sell a security, thus providing liquidity, and profiting from the difference between the price at which the buy (bid) and the one at which they sell (ask), called bid-ask spread. Algorithmic market making automates this process, enhancing efficiency, liquidity, risk management, and cost-effectiveness through the use of advanced algorithms, mathematical models and ML techniques [4, 5, 6, 7].

Moreover, sports betting has a long history but in the last years, with the advent of online betting platforms and betting exchanges, it has witnessed substantial evolution and growth [8, 9]. Traditional sports betting involved individuals placing wagers with bookmakers who provided odds on different outcomes. However, the advent of online exchanges has transformed the landscape. In particular, online exchanges, such as Betfair, act as intermediaries, enabling direct peer-to-peer betting among individuals. Unlike traditional bookmakers, exchanges allow users to both back and lay bets, providing opportunities to act as both bettors and bookmakers. In particular, while backing represents betting on an outcome, laying is betting against the outcome and they can be thought of as the equivalent of buying and selling in financial markets (more details on the mechanisms in Chapter 2). Furthermore,

exchanges provide transparency by giving access to the order book, enabling users to assess supply and demand for specific outcomes. Additionally, participants can trade bets during events (called in-play betting), allowing for dynamic adjustments based on unfolding circumstances. Finally, the exchange model, by facilitating competitive odds, offers increased market efficiency.

The intersection of sports betting and algorithmic trading techniques has opened up new possibilities, where computational algorithms, ML models, and data analysis are applied to the sports betting market [10]. In particular, algorithmic sports traders, by leveraging historical and real-time data, similar to algorithmic traders in financial markets, seek to identify patterns, exploit pricing inefficiencies, and optimize trading strategies. Furthermore, in sports trading, with the advent of online exchanges and algorithmic trading techniques, market making possibilities have emerged. More specifically, algorithmic market making in sports trading offers advantages such as enhanced liquidity, fairer pricing (by reducing back-lay spreads), and efficient execution of bets, as well as profit opportunities.

Moreover, regarding RL, with the advent of Deep Learning, it has shown great potential in solving challenging tasks in several domains, such as board games [11, 12], biology [13], autonomous vehicles [14], and robotics [15]. Additionally, recent works [7, 16, 17, 18, 19] have also demonstrated that both RL and Deep RL have great potential in market making. In particular, RL algorithms, with their ability to learn from interactions with an environment and optimize decision-making, offer a promising approach for tackling the challenges of market making. Hence, this research aims to explore the potential of RL to develop market making strategies in sports trading.

This study was conducted in collaboration with Quant Sports Trading Ltd under the supervision of John Goodacre.

1.1 Research Motivations

This work is driven by four key observations. Firstly, the field of algorithmic sports trading is relatively new and underexplored compared to other domains such as equities, derivatives, commodities, and cryptocurrencies. This presents an opportunity to make valuable contributions and advance the understanding of algorithmic trading in the context of sports betting.

Secondly, the market making task, while not new, remains a highly challenging and unresolved problem. Existing market making approaches often rely on strong assumptions that do not align with the complexities and dynamics of real trading environments. Consequently, there is a need for innovative and adaptable market making strategies that can effectively navigate the unique characteristics of sports trading markets.

Thirdly, there is a noticeable scarcity of research specifically targeting market making in the sports betting market. This research gap highlights the potential for novel insights and advancements in this domain, which can significantly contribute to the development of effective market making strategies tailored to sports trading.

Lastly, RL has demonstrated immense potential in market making within financial markets. By leveraging RL techniques, this research project aims to explore the application of RL algorithms in the specific context of sports trading, capitalizing on the successes and advancements observed in financial markets. Collectively, these motivations drive the research project towards filling the research gap, developing innovative market making strategies, and leveraging the potential of RL to enhance trading performance.

1.2 Research Objectives

To address the previously stated motivations and contribute to the field of algorithmic sports trading this research will focus on three main objectives. The first objective is to obtain a deep understanding of the key characteristics and unique dynamics of the sports trading market, with a specific focus on analyzing the key features of exchange data. This objective entails comprehensive research and analysis of data related to prices (odds), volumes, volatility, liquidity, and differences between pre-game and in-play trading. By gaining a thorough understanding of these factors, valuable insights can be obtained regarding the intricacies of sports trading.

The second objective is to establish a benchmark for the development of novel models and strategies by evaluating the performance of existing baseline approaches. By evaluating and benchmarking these models, their effectiveness, strengths, and weaknesses can be identified, providing valuable insights for the development of novel strategies.

Finally, the third objective is to explore the application of ML and RL techniques in developing innovative market making strategies tailored to sports trading. This objective aims to leverage the power of ML and RL algorithms to design and train agents that can adapt and optimize their strategies based on real-time market data.

1.3 Research Experiments

To achieve the mentioned objectives, this research project comprises three primary experiments. The first experiment, titled **"In-depth analysis of sports exchange data"** (Chapter 3), is centred on the collection, preprocessing, and analysis of high-frequency exchange data. The focus of this first experiment is to thoroughly explore the primary features of the data. By comparing the pre-game and in-play phases, and discerning potential correlations between the various features, this experiment aims at providing a comprehensive understanding of the underlying dynamics and patterns present in the data.

Subsequently, the second experiment, **"Implementation and Testing of baseline models"** (Chapter 4), focuses on setting a benchmark for the third experiment and for future developments. One of its crucial stages is the formulation of a framework to simulate the sports trading environment. This framework serves as a simulation environment for both testing the models, but also for training the RL agents, which would not be possible through only historical data.

Finally, the third experiment, **"Development, Training and Testing of a novel RL agent"** (Chapter 5), centres on harnessing the power of ML and RL techniques. The primary goal is to design and train an RL agent capable of undertaking the market making task, with a view to outperforming the baseline models.

All the code written for these experiments is stored in two GitHub repositories, one for the first experiment (link) and one for the second and third experiment (link).

1.4 Scientific Contributions

With the experiments just described, this research significantly contributes to the advancement of knowledge in the field of sports trading. The following are the key scientific contributions:

- Valuable insights into the key characteristics and unique patterns specific to sports trading, uncovering important aspects that can enhance the understanding of this market.
- Introduction of a novel framework for the simulation of a sports trading environment, that enables the training and testing of market making models without using historical data.
- Setting of a benchmark for the market making task in the sports trading market, by evaluating the performance of existing baseline models.
- Introduction of a novel market making RL agent, showcasing its ability to optimize trading strategies and achieve superior performance.

These scientific contributions provide significant value to the academic community, practitioners, and the broader sports trading industry.

1.5 Thesis Structure

The remaining part of this thesis is organized in the following structure:

- **Chapter 2 - Background and Literature Review**. This chapter provides an in-depth exploration of the research topic by presenting the necessary background information and reviewing the relevant literature. It starts by discussing the application domain with its three main areas: algorithmic trading, sports betting and market making. It then dives into the theory and methodology of ML and RL utilized in the work.
- **Chapter 3 - In-depth analysis of sports exchange data**. It focuses on the first experiment of this work. In particular, it provides details on the data collected through the Betfair exchange and on its analysis and exploration, delving into the findings of its main features, like volumes, volatility, liquidity, and differences between pre-game and in-play data.

- **Chapter 4 - Implementation and Testing of baseline models**. This chapter describes the second experiment of this work. In particular, it describes the theory behind the models used, their implementation and the evaluation process, concluding with the analysis of the results obtained, providing valuable metrics that are going to be compared in the third experiment with those obtained with the developed RL agent.
- **Chapter 5 - Development, Training and Testing of a novel RL agent**. This fifth chapter focuses on the third and last experiment of this work. It outlines the development process of the RL agent, including the different RL algorithms tested, how the state space is represented, the agent's architectures and the reward function. The chapter also discusses the training procedure used to optimize the agent's performance and the testing phase, where the agent is evaluated and performance metrics are analyzed. Finally, the chapter concludes with a comprehensive assessment of the RL agent's capabilities, highlighting its strengths and potential areas for improvement.
- **Chapter 6 - Conclusions and Future Work**. This final chapter draws the final remarks and conclusions of the project. It begins by revisiting the objectives and summarizing the key discoveries. The chapter then underscores the research's innovative contributions to the domain of sports trading and market making. Furthermore, it discusses the main challenges and constraints encountered during the project. Finally, it concludes by suggesting potential avenues for future research.

Chapter 2

This chapter provides an in-depth exploration of the research topic by presenting the necessary background information and reviewing the relevant literature. This chapter aims to establish a solid foundation of knowledge and understanding in the field, highlighting the key concepts, theories, and previous studies that inform the research. By examining the existing literature, this chapter identifies the research gaps, justifies the significance of the study, and sets the stage for the subsequent chapters that contribute to the existing body of knowledge.

2 Background and Literature Review

2.1 Algorithmic Trading

Algorithmic trading, as previously discussed, harnesses computational power, mathematical models, algorithms, and machine learning to refine and automate trading procedures, enhancing speed, efficiency, and precision. A standard algorithmic trading system (Fig. 1), that decides which securities to buy or sell, the optimal timing, and their quantities, encompasses five crucial phases [1]:

- Data access/cleaning: encompasses the acquisition, refining, and structuring of the necessary data.
- Pre-trade analysis: entails analyzing the data to discover potential trading opportunities.
- Trading signal generation: is about determining which assets to buy or sell, based on the output of the previous stage.
- Trade execution: executing orders to build the portfolio that was constructed in the trading signal generation stage.
- Post-trade evaluation: reviewing the outcomes of the trading operations, utilizing metrics such as P&L and Sharpe-ratio, among others.

Conventionally, a system executing these five phases consists of five fundamental components. The alpha, risk, and transaction fee models, contribute to the pre-trade analysis and their output feeds into the portfolio construction model, which in turn is responsible for the trading signal generation. Lastly, the output of the portfolio construction model feeds into the execution model, which finally executes the trades [20]. The following is a brief explanation of the single components:

- The alpha model, responsible for the identification of trading opportunities, utilises various factors, such as historical price patterns, fundamental data, or statistical analysis. This model has a pivotal role in determining which securities to buy and sell.
- The risk model evaluates potential risks linked to each trading position, examining elements like volatility, correlations between markets, and exposure to distinct sectors or asset groups. It oversees the strategy's risk exposure. Occasionally, the risk requirements are directly embedded into the alpha model.

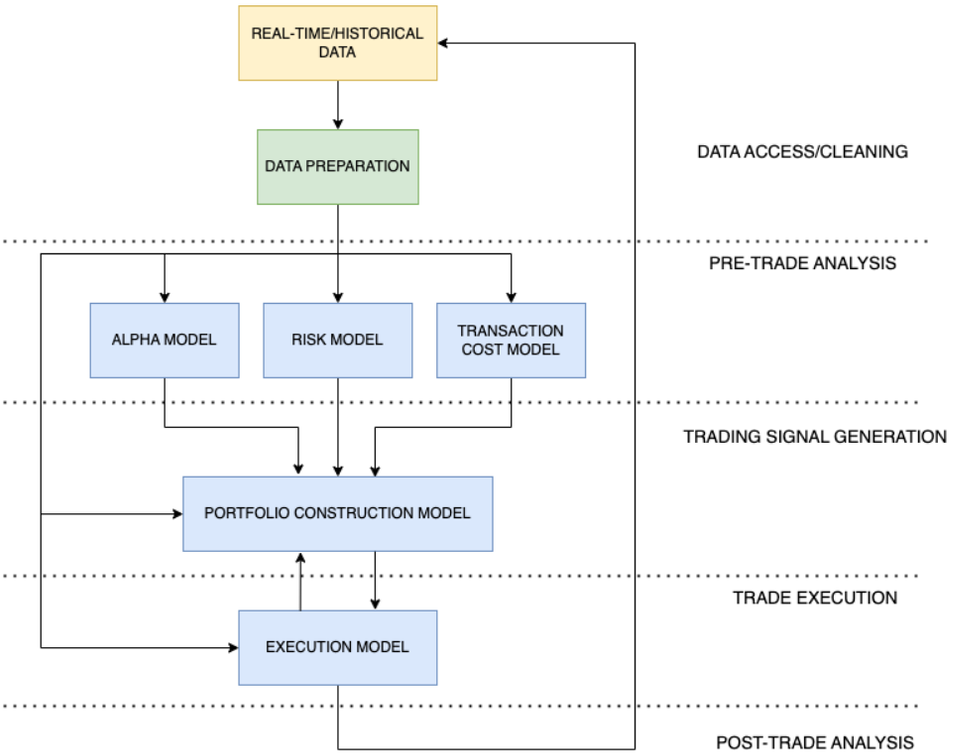


Figure 1: Typical structure of an algorithmic trading system.

- The transaction fee model is responsible for predicting the costs of the necessary trading operations. It takes as input factors such as bid-ask spreads and the order size. By integrating transactional fee considerations, the trading strategy aims not only at maximizing profits but also at minimizing costs.
- The portfolio construction model utilizes the outputs of the first three models to compose the new portfolio. It considers factors like the desired risk-return profile, diversification objectives, and other requirements to build an optimal portfolio.
- The execution model is tasked with executing the trades necessary to build the new portfolio composed by the portfolio construction model. It utilizes factors like market liquidity and aims at minimizing market impact and transaction costs.

It is worth noting that the described framework serves as a general guide and is not universally applicable to all trading systems. Variations in the components and their interconnections can occur based on specific needs. For example, some systems might exclude certain elements, consolidate multiple modules into a single unit, or feature recursive links between modules. Such flexibility enables traders and developers to customize the system to meet their particular needs and align it with their trading strategies. Moreover, as machine learning becomes increasingly integrated into algorithmic trading, contemporary systems often include two additional stages that enable dynamic adaptability to market changes. These are the data optimization and model optimization stages. These stages allow the system to dynamically adjust its features and models in response to shifts in market behaviour (Fig. 2) [21, 3].

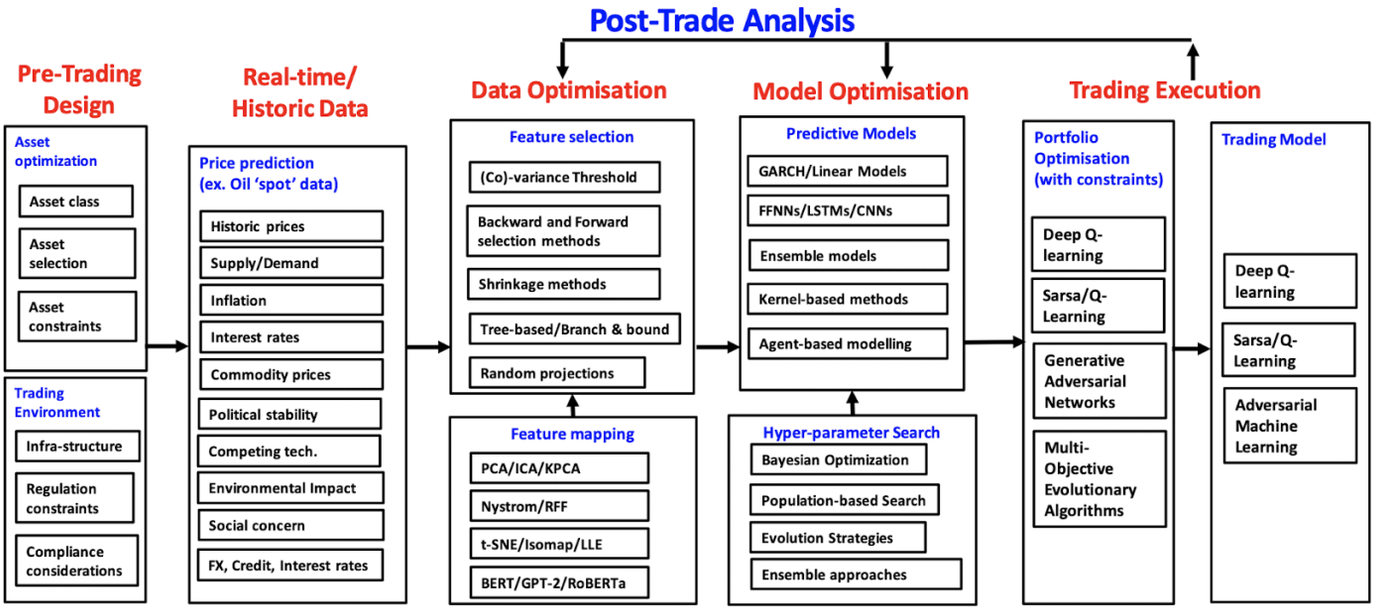


Figure 2: Typical structure of a modern self-evolving algorithmic trading system [1].

2.2 Sports Betting

Sports betting is a form of gambling that involves predicting the outcomes of sporting events and placing wagers based on those predictions. Participants, known as bettors, wager money on various sports, including football, basketball, tennis, and horse racing, among others. Sports betting odds represent the probability of an event occurring and determine the potential payout for a winning bet. There are different types of odds used in sports betting:

- Decimal Odds (e.g., 2.50): Expressed as a decimal number, these odds represent the total payout for every unit wagered, including the initial stake.
- Fractional Odds (e.g., 3/2): Represented as a fraction, these odds indicate the potential profit relative to the stake. For example, fractional odds of 3/2 mean that a successful bettor will win three units for every two units wagered.
- Moneyline Odds (e.g., +150 or -200): Commonly used in the United States, moneyline odds can be positive or negative. Positive odds indicate the potential profit from a $100 wager, while negative odds represent the amount needed to wager to win $100.

2.2.1 Bookmakers and Exchanges

In sports betting, bookmakers and exchanges serve as the primary platforms for placing bets, each operating with distinct mechanisms for profit generation. Bookmakers are integral to sports betting operations as they establish odds and accept wagers from bettors. Their revenue model centres around the concept of the overround. The overround ensures that bookmakers maintain a profit margin on the total amount wagered, regardless of the actual outcome of an event. The overround can be calculated by summing the reciprocal

of the odds for all possible outcomes. For instance, consider a simple example involving a tennis match between Player A and Player B:

- Player A: decimal odds of 2.0
- Player B: decimal odds of 3.0

To calculate the overround, we first convert the decimal odds to implied probabilities using the formula:

$$\text{Implied probability} = \frac{1}{\text{Decimal odds}} \tag{1}$$

So the implied probabilities for Player A and Player B are respectively 0.5 and 0.33: The overround can be calculated by summing the implied probabilities and subtracting it from 1:

$$\begin{aligned}\text{Overround} = (1 - (&\text{Implied probability of Player A} + \\ &\text{Implied probability of Player B})) = 0.12\end{aligned} \tag{2}$$

The bookmaker sets the odds in such a way that the overround ensures they retain a margin on the total amount wagered, usually around 5% to 10%. In particular, they set the odds lower than the ones that would fairly represent the probabilities of the outcomes. This margin compensates for their operating costs and serves as their profit. On the other hand, exchanges, such as Betfair, operate as peer-to-peer marketplaces, offering a distinct approach to sports betting. Unlike bookmakers, exchanges do not assume direct risk or set odds. Instead, they serve as intermediaries, facilitating bets between individual bettors. Exchanges generate revenue through commission fees imposed on the net returns of winning bets. In contrast to bookmakers, exchanges do not have a vested interest in the outcome of bets placed on their platforms. Instead, they earn income through the commission charged on the net winnings of successful bets, typically ranging from 2% to 5%. In addition to the revenue models discussed above, there is another key distinction between bookmakers and exchanges that significantly impacts the dynamics of sports betting. While bookmakers primarily offer the opportunity to back bets (betting for an outcome to occur), exchanges introduce the unique concept of laying bets (betting against an outcome to occur). In particular, the back and lay mechanisms are fundamental features of betting exchanges, allowing participants to both support and oppose outcomes. Backing means betting on an event to happen, similar to buying a security (in this case a bet). Laying a selection involves betting against an outcome, analogous to selling a security. Calculating the net returns from a back bet involves the following equation:

$$r_b = \begin{cases} \mu \times (\rho - 1) & \text{if wins the bet} \\ -\mu & \text{otherwise} \end{cases} \tag{3}$$

where μ, ρ are respectively the amount risked on the bet and the decimal odds. On the other hand, the net returns of laying a bet are calculated as:

$$r_l = \begin{cases} \mu & \text{if wins the bet} \\ -\mu \times (\rho - 1) & \text{otherwise} \end{cases} \tag{4}$$

Moreover, backing high and laying low on the same outcome can create a profitable opportunity, analogous to buying low and selling high in trading. In particular, this strategy allows participants to lock in a profit, called also cashing out, regardless of the outcome, by exploiting variations in odds [21]. As an example of the cash-out mechanism, suppose a bettor backs one player in a tennis match with a stake of 10 at odds of 2.0. As the event unfolds and the player's performance improves, the odds drop to 1.5. At this point, if the agent places a lay bet with a calculated stake of 12, he locks in a profit of 2: if the player wins, he gains 20 (10*2.0) from the back bet and loses 18 (12*1.5) from the lay bet, while if the player loses he gains 12 from the lay bet and loses 10 from back bet. The stake of the counter bet can be calculated as follows:

$$\mu_2 = \frac{(\rho_1 + 1) * \mu_1}{\rho_2 + 1} \tag{5}$$

where μ_1 and ρ_1 are the stakes and odds of the first bet, while μ_2 and ρ_2 are the stakes and odds of the counter bet that locks in a profit.

2.3 Market Making

As already mentioned, by providing liquidity, market makers are a crucial component of financial markets. A market maker is a firm or individual who stands ready to buy and sell a particular security on a regular and continuous basis at publicly quoted prices, called bid and ask prices. The difference between the bid and ask prices, known as the bid-ask spread, is how market makers primarily generate profit. Market makers assume a certain level of risk by holding a particular security in their inventory, with the anticipation that they can sell it, making a profit. However, they are exposed to the inventory risk, that derives from the price fluctuations and can lead to potential losses. In particular, the risk comes from the fact that the value of these securities can fluctuate and when a market maker has a non-flat inventory, meaning he has an imbalance between the amount he bought and sold, he is exposed to these fluctuations. For example, consider a market maker who buys 100 shares and sells 60, resulting in an inventory of +40 shares. In this case, if the price of that stock rises the market maker makes a profit but if it drops he suffers a loss, hence he is exposed to market movements. Therefore, because in general market makers want to have market-neutral strategies (not exposed to the market movements), effective risk management is crucial to protect their capital and ensure their long-term viability.

As already mentioned, the main advantage that market makers bring to the market is that they provide liquidity. Additional advantages are that they improve price stability and facilitate price discovery, hence improving market efficiency. In particular, market makers help to maintain price stability by absorbing supply and demand shocks and their continuous presence in the market helps to prevent drastic price fluctuations. Furthermore, by facilitating efficient price discovery, market makers contribute to the overall efficiency of the market, because their continuous trading activities help to ensure that security prices accurately reflect their intrinsic value.

In the existing literature, market making has been extensively studied in the context of tra-

ditional financial markets, such as equities, options, and currencies. Numerous studies have focused on developing models and strategies to improve market making performance and mitigate risks. In particular, [22] is considered the first formal analysis of market microstructure and optimal market making conditions, focusing on modelling temporary imbalances between buy and sell orders and addressing the problem of market makers' inventory imbalance. It is widely acknowledged as a pioneering study in market making and has influenced subsequent research in market microstructure and market making.

Furthermore, [23] presented a rigorous model specifically focused on an individual market maker operating in a single stock market. Subsequently, [24] expanded it by incorporating additional elements of uncertainty, introducing a multi-period strategy for the market maker, and considering the demand side of the market. Their research enhanced the understanding of market making dynamics by addressing these key factors and providing a more comprehensive framework for analyzing market behaviour.

Moreover, [4] enhanced the framework to a quantitative market making strategy, combining the utility framework of [24] with the micro-structure of actual limit order books. In particular, the key concepts introduced by the authors are the following: the mid-price used as the "true" price, the explicit utility function and the orders' arrival intensity. Regarding the orders' arrival intensity, to model the arrival rate of buy and sell orders reaching the agent, they incorporated findings from the econophysics field [25] and used an exponential arrival rate. More specifically, the mathematical framework defined by the authors of [4] is centred around the mid-price S of the asset in question, defined as:

$$dS_t = \sigma dW_t \tag{6}$$

where σ is the standard deviation (represents the volatility) and W_t a one-dimensional Brownian motion. Moreover, the bid and ask prices are set as:

$$\begin{aligned} S_t^b &= S_t - \delta_t^b \\ S_t^a &= S_t + \delta_t^a \end{aligned} \tag{7}$$

Hence, the end goal of the agent can be described as finding the optimal S_t^b and S_t^a. Furthermore, the authors define the wealth (cash) and inventory of the agent as dependent on the arrival of market orders. In particular, the wealth is defined as:

$$dX_t = S_t^a dN_t^a - S_t^b dN_t^b \tag{8}$$

where N_t^a and N_t^b are the amounts of the asset that the agent bought and sold and they are modelled as Poisson processes with intensities λ^a and λ^b. In particular, λ^a (λ^b) is dependent on the difference between the ask (bid) price and the mid-price, that is δ:

$$\begin{aligned} \lambda^a &= Ae^{-k\delta^a} \\ \lambda^b &= Ae^{-k\delta^b} \end{aligned} \tag{9}$$

where A and k are constants and measures of the liquidity of the market. In particular, this

definition of λ^a and λ^b follows the logic that the further S_t^a and S_t^b are from the mid-price the smaller the probability of getting the orders filled. Hence, the closer they are to the mid-price the bigger the bought and sold quantity N_t^a and N_t^b. On the other hand, the inventory is defined as:

$$dq_t = dN_t^b - dN_t^a \tag{10}$$

Finally, the agent's objective function is the constant absolute risk aversion (CARA) utility function, defined as follows:

$$\max_{\delta^b,\delta^a} E[-e^{-\gamma(X_T+q_T S_T)}] \tag{11}$$

where γ is called risk aversion parameter, T is the time horizon and $X_T + q_T S_T$ represents the final wealth at time T. In particular, X_T is the wealth accumulated until time T and $q_T S_T$ is the wealth that the agent can accumulate by liquidating the inventory q_T at price S_T. Regarding γ, it represents the risk aversion of the agent. In particular, a γ of 0 represents a risk-neutral agent, while a bigger value represents a more risk-averse one. To find the optimal solution, the authors follow the analysis in [24] and use the dynamic programming principle, showing that the objective function solves a Hamilton-Jacobi-Bellman equation (HJB) and approximating it with a more tractable PDE. Subsequently, they solve the problem in two steps: (1) they solve the PDE in order to find what they call the reservation price and (2) use the reservation price to derive the optimal S_t^b and S_t^a.

In particular, the reservation price represents an adjustment of the mid-price that accounts for the inventory. More specifically, the reservation price includes the desire of the agent to maintain a flat inventory. Hence, if the agent has a positive inventory, the reservation price is going to be smaller than the mid-price, to have a higher probability of selling more, while with a negative inventory, the agent would want to buy more, hence the reservation price will be higher. Specifically, the reservation price is defined with the following equation:

$$r_t = s - q\gamma\sigma^2(T-t) \tag{12}$$

Having $(T-t)$ in the equation follows the logic that the closer to the final time-step T the closer the reservation price is going to be to the mid-price because when closer to T the agent would want to sell and buy more to balance the inventory and possibly arrive to T with a flat one. Finally, the optimal bid and ask prices are defined as follows:

$$\begin{aligned} S_t^b &= r_t - \frac{1}{2}\gamma\sigma^2(T-t) - \frac{1}{\gamma}\ln(1+\frac{\gamma}{k}) \\ S_t^a &= r_t + \frac{1}{2}\gamma\sigma^2(T-t) + \frac{1}{\gamma}\ln(1+\frac{\gamma}{k}) \end{aligned} \tag{13}$$

This framework set a solid base for the market making problem, developing a tractable solution and showing evidence that it can generate positive returns. However, it was also based

on several assumptions that do not completely match the realities of real markets and lack some aspects that are part of them. Hence, subsequent works enhanced it by adding some of these features that make the resulting framework more useful in real applications.

In particular, the authors of [26] added an inventory constraint, making the framework more realistic, and showed that the HJB equation is transformable into a more tractable system of linear ordinary differential equations. Moreover, in [6] the authors incorporated price impact and adverse selection into the framework making it even more realistic. Finally, in [27] the authors generalize the framework to the multi-asset case. In particular, the authors highlight how in practice market makers are typically responsible for multiple securities and how applying an independent market making strategy to each asset is suboptimal in terms of risk management. In fact, one of the tools that market makers often use to mitigate their risk is hedging, which consists in reducing the risk associated with an asset by taking an offsetting position in a related one. Hence, by applying independent market making strategies the agent does not exploit the power of hedging, resulting in a suboptimal solution [21].

2.4 Reinforcement Learning

RL is a subfield of machine learning that deals with the problem of decision-making in dynamic and uncertain environments. Unlike other machine learning approaches, RL learns through trial-and-error interactions with an environment. It is particularly well-suited for scenarios where an agent needs to learn how to make sequential decisions to maximize a reward [28]. RL has gained significant attention in recent years due to its successes in various domains. It has demonstrated remarkable achievements in several domains, such as playing complex games, like AlphaGo [11] and AlphaZero [12], robotics [15], autonomous vehicles [14], and more recently it has been a crucial component in the training of ChatGPT with a technique called Reinforcement Learning from Human Feedback.

2.4.1 Key Components of Reinforcement Learning

RL consists of several fundamental components that work together to enable an agent to learn and make optimal decisions [29]. These components include the agent, the environment, the state, the action, the reward, the model and the policy. In particular, the agent represents the decision-making entity. It interacts with the environment, observes its state, takes actions, and learns from the received rewards. The environment encapsulates the external world with which the agent interacts. It is characterized by a set of states and the agent's actions within the environment lead to transitions between these states. In particular, a state s represents the current condition of the environment and provides the necessary information for the agent to make decisions. More specifically, the states are typically represented as Markov states, meaning that every state is independent of the past states (history) and provides sufficient information to the agent. More formally, a state is defined as a Markov State if

$$P[s_{t+1}|s_t] = P[s_{t+1}|s_t, s_{t-1}, ..., s_1] \tag{14}$$

Moreover, an action a represents the decision or behaviour that the agent can take in a given

state. On the other hand, the reward r is the feedback or evaluation signal that the agent receives from the environment and represents the desirability or quality of the agent's actions in a given state. Furthermore, a model is the agent's representation of the environment and predicts what the environment will do next. Specifically, the model predicts both the next state and the next immediate reward. The next state prediction is defined as:

$$P^a_{ss'} = P[s_{t+1} = s' | s_t = s, a_t = a] \tag{15}$$

while the prediction of the next reward is defined as:

$$R^a_s = E[r_{t+1} | s_t = s, a_t = a] \tag{16}$$

Finally, the policy π is the strategy or rule that the agent follows to select actions in different states. It maps states to actions, specifying the agent's behaviour. The policy can be deterministic, where it directly determines the action to take for a given state, or stochastic, where it defines a probability distribution over actions for each state. In particular, a deterministic policy is defined as:

$$a = \pi(s) \tag{17}$$

while a stochastic one is defined as:

$$\pi(a|s) = P[a_t = a | s_t = s] \tag{18}$$

In particular, the cumulative reward from a given state S_t is often discounted by a factor $\gamma \in [0,1)$ per time step, which encourages the agent to prefer immediate rewards over distant ones. The expected cumulative reward, also known as the value function $V(s)$, is defined as:

$$V(s) = E[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s_t = s] \tag{19}$$

If the agent follows an optimal policy π^*, it achieves the maximum possible value from any state s, which is

$$V^*(s) = \max V(s) \tag{20}$$

Similarly, the action-value function $Q(s,a)$ under a policy π is defined as the expected cumulative reward received after taking an action a in state s and represents an alternative method to estimate the goodness of a state. With the optimal policy π^*, the function $Q^*(s,a)$ gives the maximum expected cumulative reward when action a (the output of π^*) is taken in state s. The end goal of RL algorithms is to find the optimal policy π^* that maximizes the expected cumulative reward [21].

2.4.2 Types of RL and Deep RL

RL algorithms can be distinguished based on what they try to learn and approximate. Different types of algorithms approach the learning and decision-making process in distinct ways. Value-based algorithms aim to estimate value functions, such as the state-value function $V(s)$ or the action-value function $Q(s,a)$. One of the most popular algorithms in this category is Q-learning. On the other hand, policy-based algorithms directly optimize the agent's policy π to maximize the cumulative reward. In particular, rather than estimating value functions, these algorithms learn directly a policy that maps states to actions. Furthermore, model-based algorithms take a different approach by learning a model of the environment dynamics. More specifically, they aim to capture the transition probabilities $P^a_{ss'}$ and reward expected values R^a_s to simulate and plan ahead. By utilizing the learned model, these algorithms can make informed decisions. In fact, once the agent has an accurate model of the environment, he can plan his future actions to maximize the reward. On the other hand, model-free algorithms combine elements of value-based and policy-based methods but do not have a model of the environment. Finally, another type is actor-critic algorithms, which employ a dual-model architecture where an actor model learns the policy while a critic model estimates the value function. Additionally, hybrid approaches that combine elements from multiple algorithm types have been explored for enhanced performance and adaptability. In conclusion, each type of algorithm has its own strengths and advantages, making them suitable for different scenarios and environments.

On the other hand, in the last years, the advent of Deep Learning has revolutionized the field, enabling RL algorithms to scale to more complex problems that were previously intractable [30]. In particular, Deep RL is a subset of RL that leverages deep neural networks to approximate value or policy functions. By utilizing deep neural networks, Deep RL algorithms are capable of handling high-dimensional state spaces and complex decision-making tasks. Notable examples of Deep RL algorithms include DQN (Deep Q-Network) and DDPG (Deep Deterministic Policy Gradient) [21].

2.4.3 RL in Market Making

Market making is between the many domains in which RL showed great potential. In particular, by leveraging its ability to learn from past experiences and iteratively refine its actions, an agent trained through RL can dynamically adapt to changing market conditions, effectively navigating the complexities of financial markets and making informed decisions to maximize profits. In market making, both value-based (like Q-learning and SARSA [28] [7]) and policy-based algorithms (like the deep policy gradient method [31]) have been used. Furthermore, the variables that represent the state normally include bid and ask prices, inventory, order-flow imbalance, liquidity, volatility and other market indicators, while the possible actions for the agent are to set its bid and ask prices and the volumes to buy and sell. Furthermore, in most cases, the reward function used is either the PnL or some variation of it, which normally includes a penalization term for the inventory.

The first work to explore RL for market making is [32]. In particular, the authors explored the application of three RL algorithms, namely a Monte Carlo method, SARSA, and an actor-critic method, using the inventory, order imbalance and market quality measures as state

variables. Furthermore, as the reward function, the authors used a linear combination of PnL, inventory and market quality measures. The results showed evidence that the actor-critic algorithm was able to generate policies that effectively adjusted bid and ask prices, successfully balancing the trade-off between profit and quoted spread. Additionally, the authors showed that stochastic policies outperform the deterministic ones.

Moreover, in [33] the authors propose "spread-based" strategies that capitalize on the mean-reverting nature of the mid-price and exploit opportunities when the mid-price deviates from the previous period's price. Additionally, an online algorithm is employed to select the minimum quoted spread in each time period. In this work, the agent's states are defined by the current inventory and price data. Furthermore, [7] builds upon the work of [32]. In particular, they use an action space that contains ten actions, where the first nine actions represent a pair of orders with a specific spread, while the final action allows the agent to clear their inventory using a market order. The agent state includes the inventory and the active quoting distances, while the market state includes market spread, mid-price movement, book/queue imbalance, signed volume, volatility, and relative strength index.

On the other hand, the authors of [18] employed an actor-critic method to perform market making in cryptocurrencies. In particular, they use the top 15 levels of the limit order book, the Trade Flow Imbalance and the Order Flow Imbalance with a lookback window as state variables and to represent the agent's state they use risk and position indicators. Finally, some works focused on making the RL agent robust to adversarial conditions. In particular, [19] employed a second adversarial agent to make the first agent more robust to perturbations and uncertainty [21].

Chapter 3

This chapter focuses on the first experiment of the research project. In particular, its primary goal is to conduct a comprehensive exploration and analysis of sports exchange data. By examining key features such as volumes, volatility, liquidity, and differences between pre-game and in-play data, this experiment aims to gain valuable insights into the unique characteristics and patterns specific to sports trading. The chapter begins by introducing the dataset used, then delves into the details of the exploration and analysis and finally illustrates the results.

3 In-depth analysis of sports exchange data

3.1 Data

This first experiment relies on a comprehensive dataset obtained through the Betfair exchange historical data service. Access to this data was made possible by utilizing a PRO account, which ensures access to high-quality and reliable historical data. The dataset specifically focuses on tennis matches that took place in January 2023, for a total of more than 4000 events. This timeframe provides a substantial amount of data to examine and analyze, offering valuable insights into the dynamics of sports trading. The dataset includes information on a wide range of tennis matches, encompassing both very popular and lesser-known events, providing a diverse and representative sample of the market. In terms of size, the collected dataset amounts to approximately 340 MB.

It is important to note that the dataset exclusively focuses on match odds, meaning the data refers to bets placed on one of the two players winning the match. In particular, data on other types of bets, such as set betting or handicap betting, has been intentionally omitted from this analysis to maintain a specific focus on match odds and related market dynamics.

The files provided by the Betfair Historical Data service, which constitute the dataset, are provided in a zip format. This compression format allows for efficient storage and retrieval of data, facilitating data management and analysis. The dataset is stored in JSON format within the compressed files and each file represents a single event (in this case a tennis match). The structured nature of the data in JSON format enables easier parsing and extraction of relevant information for analysis. Each file contains a collection of JSON objects, each representing an update on the order book. In particular, by parsing the file, the order book can be reconstructed at each time step and then it can be used to extract meaningful features necessary for the data analysis and exploration stage.

3.2 Data Analysis and Exploration

3.2.1 Data collection and parsing

The first critical task undertaken was the development of a robust code infrastructure to efficiently collect, parse, and store the collected data. This essential process presented notable

challenges, particularly in parsing the data. The raw data obtained from the Betfair Historical Data service required thorough processing and transformation to be in a format suitable for analysis. To address this, substantial effort was invested in implementing all necessary functionalities to parse the data and output it in an easily visualizable and processable format. By building a reliable data infrastructure, this research project ensured the availability of well-organized data for further analysis. This stage's successful completion also eliminated the risk of significant delays or impediments in subsequent project stages that may require additional data.

3.2.2 Handling missing data

Following the completion of the data parsing stage, the exploration and analysis commenced. Initially, the presence of missing data was meticulously analyzed to assess its impact on the dataset's integrity and the potential need for data imputation strategies. Addressing missing data is crucial to maintain the accuracy and reliability of subsequent stages. During this stage, it was observed that certain events in the dataset contained a significant amount of missing data. While some missing data can be handled through imputation techniques, data found with an excessive amount of missing values was classified as unsuitable for analysis and excluded from further stages.

At this stage of the project, the remaining data, which contained none or acceptable amounts of missing data, was not immediately processed to remove or fill the missing entries. Instead, the decision was made to defer data cleaning to further stages of the project. The main reason for deferring this process was the need to carefully evaluate if and in what amount the historical data would have been used in conjunction with simulated data in later stages of the research, such as training and testing of the RL agents, in which case a more complete data cleaning will be needed. Nevertheless, the presence of missing data was carefully considered during the data analysis phase to ensure the integrity and reliability of the findings.

3.2.3 Feature extraction and Exploratory data analysis (EDA)

The next step in the data exploration involved feature extraction. Several key features relevant to sports trading and market making were identified and extracted from each event in the dataset. These features included volume matched (in pounds) at each order book update; total volume matched, which is the total amount of volume traded until that moment (cumulative sum of volume matched at each order book update); back-lay spread, the difference between the best back and lay prices; last traded price; mid-price, which is the average between the best back and lay prices; order book imbalance, which represents the imbalance between the back and lay sides and is calculated as follows:

$$OBI = \frac{V_b - V_l}{V_b + V_l} \tag{21}$$

where V_b and V_l are the total volume of the back and lay side, and it takes values between $+1$ and -1, where these two values represent respectively a complete imbalance towards the back and lay side; available volume on the back and lay side; and finally the time difference (in seconds) between each order book update. Fig. 3 shows examples of some of the features extracted. In particular, it shows features extracted from the data of the Australia

Open 2023 final played by Tsitsipas and Djokovic on the 29th of January.

The extraction of these specific features was driven by the research focus on market making. In particular, as a market maker, understanding the liquidity and volume dynamics is crucial for making informed decisions on back and lay prices. Hence, the features related to trading volumes were extracted to gain insights into the liquidity patterns before and during the sports events. Additionally, a market maker's primary objective is to capture spreads, which are the price differences between back and lay orders. Therefore, features like the back-lay spread were included in the analysis to understand the spread dynamics and potential opportunities for market making strategies. Furthermore, order book imbalance is an important factor that impacts market makers' decisions, providing information on the demand and supply balance at specific price levels, indicating potential shifts in price direction.

Moreover, the extracted features play a crucial role in the subsequent stages of the project. In particular, in the upcoming experiments, these features will be considered when designing the observation space of the RL agents.

After extracting the relevant features, an exploratory data analysis (EDA) approach was employed to delve deeper into the dataset's characteristics. This involved using summary statistics of the extracted features, as well as visualizing and plotting them, to identify specific patterns and gain valuable insights. The visualization of the data in graphical form facilitated the examination of trends, relationships, and potential anomalies.

3.2.4 Correlation analysis

Correlation analysis was conducted to investigate the relationships between the various features extracted from the data. Correlation analysis is a valuable statistical tool used to quantify the strength and direction of associations between different variables. By exploring correlations, insights into how these features interact and potentially influence the market dynamics are obtained.

To perform the correlation analysis, the Pearson correlation coefficient was utilized, which measures the linear relationship between two continuous variables. The formula for the Pearson correlation coefficient r is as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{22}$$

where x_i and y_i represent the individual data points for the two features, $\bar{x}$ and $\bar{y}$ are their respective means and n is the total number of data points. The correlation coefficient r ranges from -1 to $+1$, with the first indicating a perfect negative correlation, the second indicating a perfect positive correlation, and 0 indicating no linear correlation between the variables.

The importance of correlation analysis lies in its ability to reveal potential interdependencies between features. By identifying correlated features, a deeper understanding of how changes in one variable might affect another can be obtained. This knowledge is essential

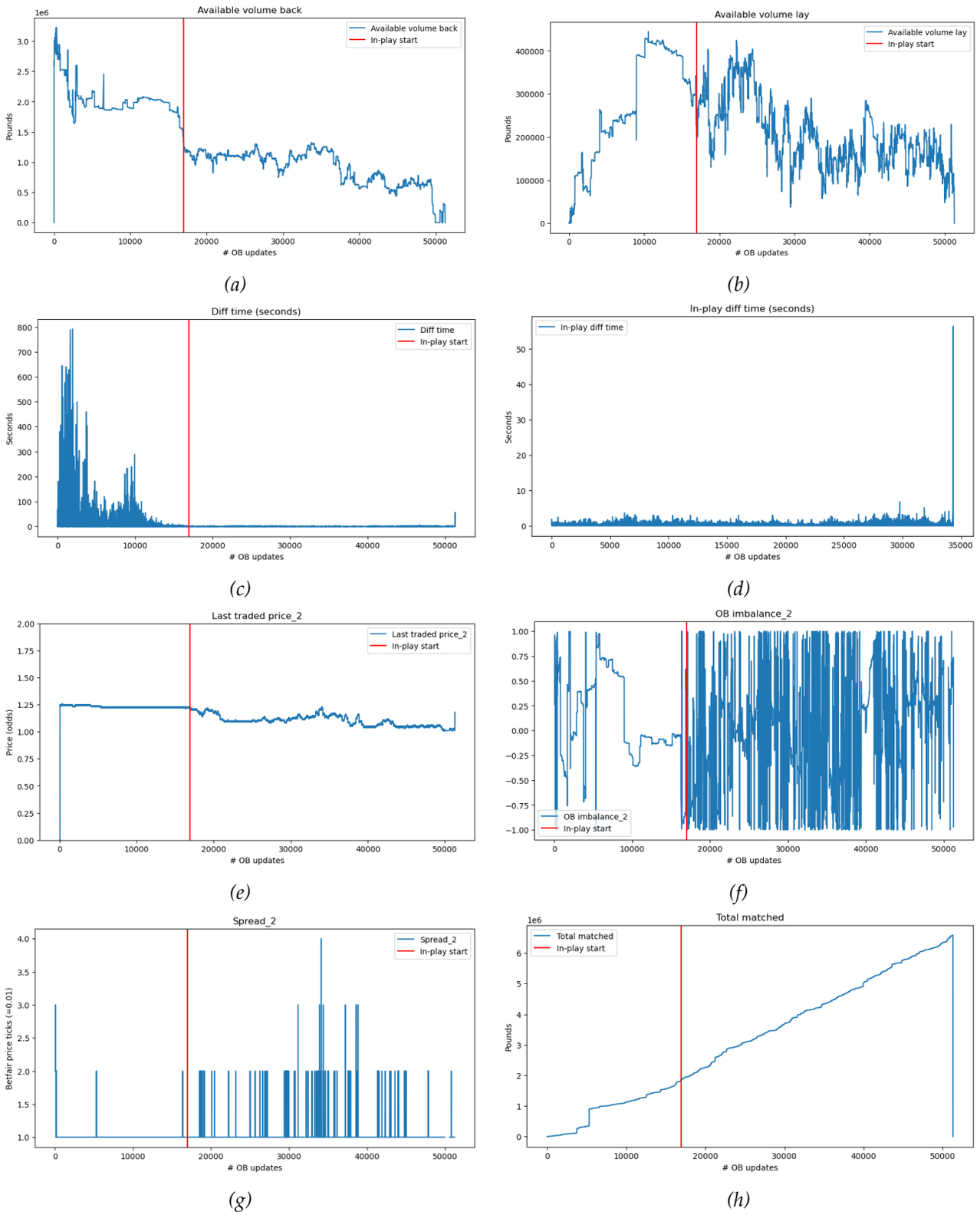
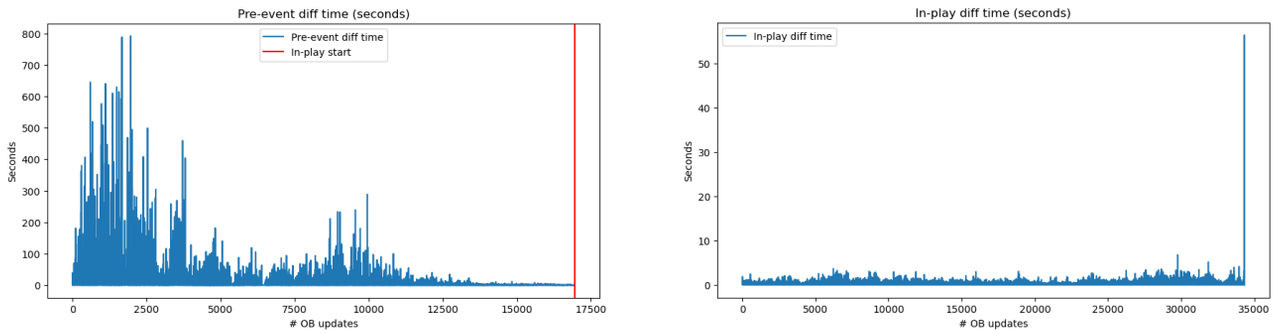


Figure 3: Examples of some of the extracted features from the data of the match between Tsitsipas and Djokovic played on the 29th of January 2023 for the Australia Open final. In particular, in this case, the features regard only the data on Djokovic (runner 2). The features are: available back volume (a), available back volume (b), difference in seconds between each OB update on all the match (pre event and in-play) (c), difference in seconds between each OB update during in-play (d), last traded price (e), OB imbalance (f), back-lay spread (g) and total volume matched (h).



(a) *(b)*

Figure 4: Plots of the time between each order book update in the pre-event (a) and in-play (b) periods of the Australia Open 2023 final (Djokovic vs Tsitsipas). As noticeable, in the pre-event period, the updates are much less frequent, arriving at a maximum of nearly 800 seconds (13.3 min) between an update and another.

in trading as it can help traders and market makers develop more robust and informed strategies. Moreover, in the context of ML, it helps us select relevant features, avoid multicollinearity issues, and improve model interpretability. For our RL agent development, understanding correlations guides us in fine-tuning input features, leading to a more accurate and adaptable model. More specifically, in ML and RL, utilizing correlated features can lead to unnecessary complexity without adding substantial benefits. This concept aligns with Occam's razor, which suggests that simpler models are preferred when they provide comparable performance.

Finally, it is crucial to note that the Pearson coefficient specifically measures linear relationships between variables. This is an important consideration when interpreting the results of the correlation analysis, as it may not capture more complex, non-linear associations between features.

3.3 Results

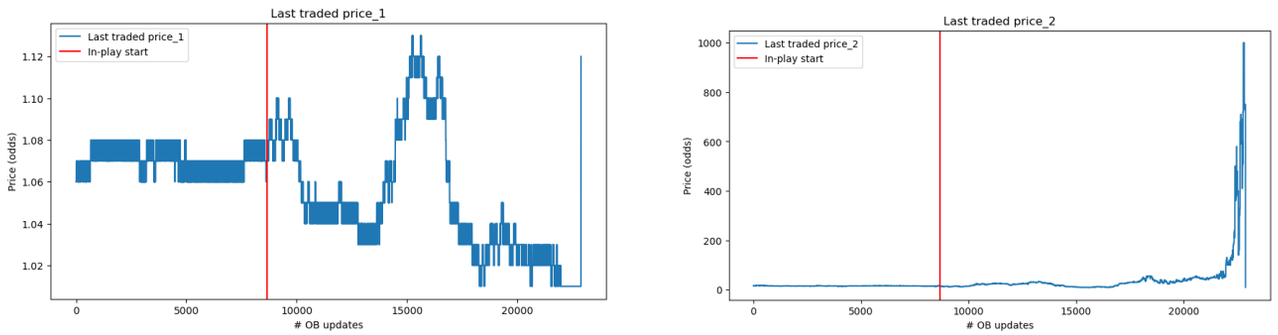
3.3.1 Exploratory Data Analysis

Starting from the feature which represents the time in seconds between each order book update (*Diff time*,), the analysis revealed interesting insights into the frequency of order book updates during the pre-event and in-play periods. The order book updates are noticeably more frequent during the in-play period compared to the pre-event period. The mean and standard deviation of the feature are significantly smaller during the in-play period, indicating that the order book is updated more rapidly when matches are in progress (Fig. 4 and Table 1). This increased frequency is attributed to orders arriving more quickly during the in-play period. The higher order frequency observed during the in-play period indicates a more favourable environment for market makers due to an increased likelihood of getting their orders matched. As the order flow becomes more active and frequent, the probability of market makers' orders being executed rises, making the in-play phase potentially more profitable.

Moreover, the analysis of the volatility of the prices during the in-play period revealed sig-

Table 1: Mean and standard deviation of the Diff time feature, which represents the time in seconds between each order book update, during the pre-event and in-play periods, in three different matches: the two semifinals (first and second column) and the final (third column) of the Australia Open 2023.

	Djokovic vs Paul	Tsitsipas vs Khacanov	Djokovic vs Tsitsipas
Pre-event			
Mean	18.9	25.9	9.6
Std	68.06	117.9	35.13
In-play			
Mean	**0.59**	**0.29**	**0.31**
Std	**1.13**	**0.46**	**0.50**



(a) *(b)*

Figure 5: Plots of the last traded price of the two players, Djokovic (a) and Paul (b), in the Australia Open 2023 semifinal played on the 27th of January, where Djokovic won. It is noticeable how the volatility during the in-play period is much higher.

nificantly higher levels compared to the pre-event phase (Fig. 5). The amplified volatility in the in-play period poses greater challenges and risks for market makers, mainly due to the inherent connection between volatility and inventory risk. As volatility escalates, the market prices fluctuate rapidly, increasing the likelihood of market makers' inventories being exposed to price fluctuations and leading to potential losses. Additionally, visualizing the plots of the prices throughout the different matches in the dataset, unveiled a pronounced surge in volatility towards the end of the match, especially of the price of the player that is losing. This intensified volatility in the closing stages might indicate a heightened level of uncertainty and market instability, which demands heightened vigilance from market makers to navigate the challenging market conditions effectively.

Furthermore, the matched volumes at each order book update are crucial in the context of market making, as they directly influence a market maker's chances of getting their back and lay orders matched. On the other hand, it is important to remember that while higher volumes offer greater potential for absolute returns on trades, they also come with increased risk, leading to higher potential losses. During both the pre-event and in-play periods, the mean volumes of matched orders are not significantly high, but the high standard deviations show considerable variability, suggesting the presence of occasional spikes in matched volumes (Table 2 shows three examples). This variation indicates that certain instances experience significantly higher trading activity, possibly indicating particular events

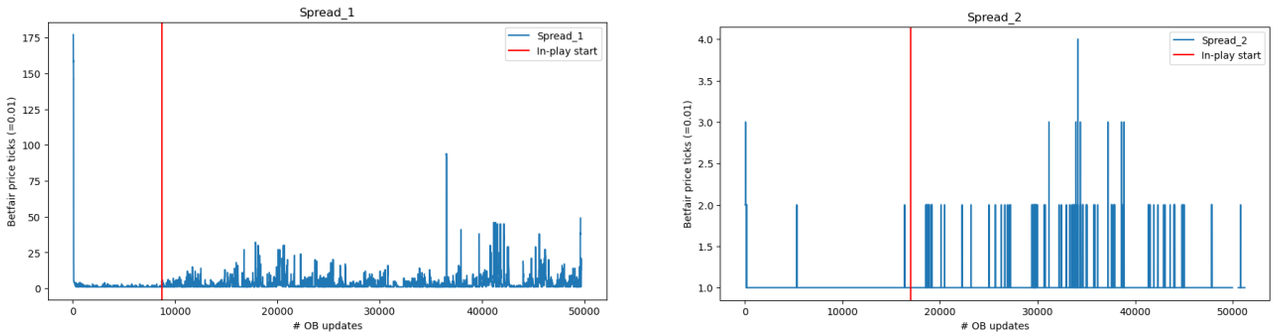
Table 2: Mean and standard deviation of the Matched volume feature, which represents the volume matched at each order book update, during the pre-event and in-play periods, in three different matches: the two semifinals (first and second column) and the final (third column) of the Australia Open 2023. As noticeable, the matched volumes are higher during the in-play period.

	Djokovic vs Paul	Tsitsipas vs Khacanov	Djokovic vs Tsitsipas
Pre-event			
Mean	136.86	47.01	109.98
Std	2197.22	487.95	4421.30
In-play			
Mean	**237.46**	**101.93**	**137.43**
Std	**3192.99**	**1010.61**	**1570.40**

in the match. However, a crucial observation arises when comparing the matched volumes between the pre-event and in-play periods. As previously established, the in-play period witnesses significantly more frequent order book updates. Consequently, normalizing the matched volume for each time range, shows that much higher volumes are matched during the in-play phase. This normalization underscores the attractiveness of the in-play period for market makers, as it provides more significant trading opportunities due to higher volumes matched over shorter time intervals.

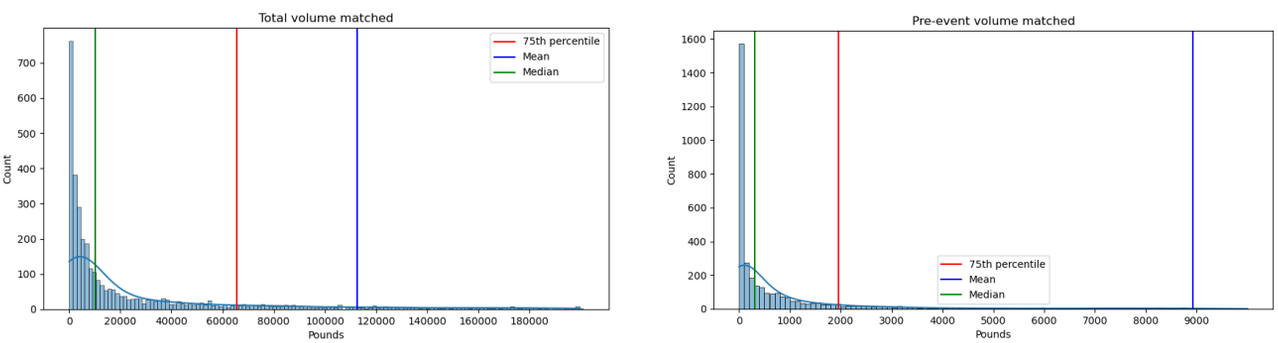
Another crucial feature in the market-making context is the back-lay spread. The analysis reveals that during the in-play phase, the back-lay spread tends to exhibit higher values compared to the pre-event period (Fig. 6 shows three examples). This finding suggests that there is a more significant price discrepancy between the best back and lay prices during in-play, indicating greater potential for market making opportunities. Moreover, during the in-play period, the spread experiences higher volatility with frequent peaks. This heightened volatility can be attributed to the increased price fluctuations occurring during in-play. As market participants react to changing events and outcomes during a sports match, the prices become more volatile, leading to wider spreads and presenting both opportunities and challenges for market makers. Overall, the higher values and increased volatility of the back-lay spread during the in-play period underscore its importance as a key factor influencing market making strategies. Understanding and managing the back-lay spread dynamics can enable market makers to make informed decisions and capitalize on favourable trading opportunities, while also navigating the potential risks associated with high volatility.

Moreover, by visualizing the distribution of the total volume matched (during all the event and not at each order book update) across all the matches in the dataset, it becomes evident that the distribution exhibits an exponential shape, with a large number of events characterized by relatively low volumes and a few events with significantly higher volumes. Fig. 7 illustrates the distributions of the total and pre-event volume matched, highlighting the exponential trend in both. Additionally, Table 3 provides summary statistics, demonstrating that, on average, the volume matched during the pre-event period is a small percentage of the total volume matched. This finding reinforces the idea that the in-play period offers more attractive trading opportunities for market makers due to the higher volumes.



(a) *(b)*

Figure 6: Plots of the back-lay spread for one of the players in two example matches: Tsitsipas vs Khachanov (a) and Tsitsipas vs Djokovic (b). Both matches are of the Australia Open 2023 tournament (the first is one of the semifinals and the second is the final).



(a) *(b)*

Figure 7: Plots of the distribution of the total (a) and pre-event (b) volume matched across all the matches in the dataset.

Table 3: Aggregate statistics of the distribution of the total volume matched across all the matches in the dataset.

Volume matched					
	Mean	Median	75th perc.	95th perc.	99th perc.
Total	112485.13	10110.37	65585.62	438582.06	1800592.25
Pre-event	8926.40	302.95	1960.35	37729.32	124794.99

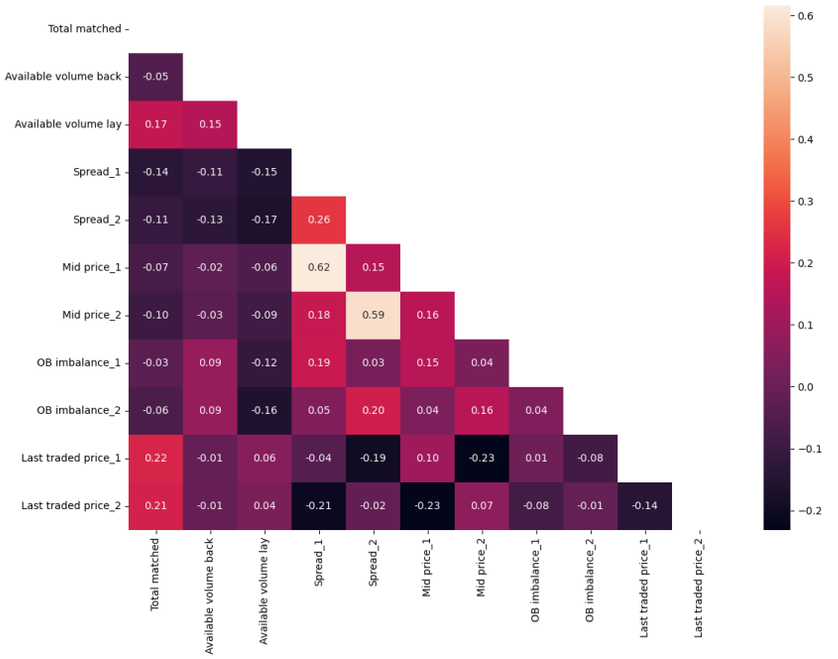


Figure 8: Mean correlation matrix of the features extracted from the dataset.

In summary, the in-play period stands out as more favourable for market makers, thanks to the higher volumes and trading activity, which increase their likelihood of getting orders matched and offer greater potential for profitable trades. However, the heightened volatility during this period also represents bigger risks and necessitates effective risk management strategies to safeguard against potential losses.

3.3.2 Correlations

By computing the mean correlation matrix of all the matches in the dataset, it was observed that there is only one correlation with a significant value: the correlation between the mid-price and back-lay spread, which has an average Pearson coefficient of 0.6 (Fig. 8). This correlation can be intuitively understood based on the nature of these two features. The mid-price represents the average between the best back and lay prices, while the spread measures the difference between these two prices. The high peaks of the spread during the in-play phase, mentioned in Section 3.3.1, cause momentary peaks in the mid-price as well. However, it is crucial to remember that the observed correlation between the mid-price and back-lay spread is a mean correlation calculated across all the matches in the dataset. This means that while there is a significant correlation on average, it may not be present in every individual match.

In conclusion, the lack of strong correlations between other features indicates that the dataset exhibits a certain degree of independence and diversity in its patterns. This finding suggests that various features capture distinct aspects of the market dynamics, which is valuable information for developing comprehensive market making strategies.

3.4 Discussion

This first experiment focused on providing an in-depth analysis of sports exchange data, emphasizing the unique characteristics of the sports trading market. The exploration of key features such as trading volumes, liquidity, and volatility revealed distinct patterns that are essential for understanding market dynamics. Specifically, the in-play trading period, compared to the pre-event phase, emerged as a crucial phase for traders and market makers. While offering better opportunities for profitable trading due to increased activity and liquidity, it also poses challenges related to higher volatility and associated risks. For these reasons, the focus of subsequent experiments was specifically placed on the in-play period.

Moreover, the correlation analysis undertaken provided valuable insights, even if no strong linear relationships were found. It is important to note that the Pearson coefficient measures only linear relationships, limiting our understanding of the possible non-linear interactions between the features. However, the lack of strong correlations can be viewed as a confirmation of the unique and independent contribution of each variable to market dynamics, affirming the need for comprehensive strategies in market making.

This analysis served as the foundational pillar for the subsequent experiments, enabling the formulation of strategies and models that are not just theoretically sound, but also practically viable in the sports trading market environment. Overall, this experiment's findings played an instrumental role in shaping the subsequent methodological choices in both the simulation of a sports trading environment and in the development and evaluation of market making agents in Chapters 4 and 5.

Chapter 4

This chapter delves into the second experiment of the thesis: simulating a sports betting trading environment and testing various market making baseline strategies. The chapter discusses various approaches to environment simulation and justifies the chosen method: combining a sport-specific Markov model for tennis with the Avellaneda-Stoikov framework, described in Section 2.3. The aim is to emulate critical aspects of a real trading environment within the constraints of available data and project scope. Finally, this experiment lays the groundwork for the third experiment, where RL agents will be implemented, trained and tested in this same environment.

4 Implementation and Testing of baseline models

As a reminder, the first experiment of the project, (Section 3), dived deep into the primary features and characteristics of sports betting markets, through the analysis of exchange data. The analysis provided valuable insights into the dynamics of these markets, highlighting the attractiveness of in-play betting for market makers due to higher trading volumes and increased trading opportunities.

With the foundation established in the first experiment, the project now moves into testing market making strategies. Before testing various market making strategies, the immediate task at hand is to build a simulated trading environment that accurately represents the dynamics of sports betting markets. The importance of this simulated environment cannot be overstated, as it serves as the experimental ground where all future strategies will be trained, tested and evaluated. Selecting the right approach to create this environment requires careful consideration of various factors, including the available data, complexity, and the specific nature of sports betting.

4.1 Simulating a trading environment

To simulate a trading environment in the sports betting market, several options were considered. In particular, the first option includes using historical data: this option would involve leveraging the data collected in the first experiment and additional one. The historical data could be used directly to train and test strategies as it includes directly the order book updates. However, the available historical data is limited and messy, which can lead to an inaccurate representation of the actual trading environment. Additionally, for training RL agents, the amount of historical data is insufficient as RL typically requires substantial data for effective learning.

The second option is to directly simulate the Limit Order Book (LOB) and its dynamics. This approach would arguably be the most comprehensive and realistic, capturing the intricacies of a real market. However, simulating the LOB accurately is complex and requires granular data on the individual orders submitted. Unfortunately, such data is not available and without proper data, not only it is challenging to accurately train the ML models or calibrate the non-ML models that simulate the LOB, but there is also the risk of ending up

with a generalised simulation. This generalized simulation would not accurately represent the specific dynamics of a sports betting market LOB, which is the focal point of this project. Thus, the absence of appropriate data would cause a deviation from the project's scope, making this option less viable.

Finally, the third option refers to using a sport-specific model to simulate the price (odds) time series of the bet, in conjunction with the Avellaneda-Stoikov (AS) framework (Section 2.3). The AS framework models several aspects of a real trading environment with a LOB, such as order arrivals, competitiveness, and the probability of getting orders filled. However, it is important to note that while the AS framework does model aspects of a real market with a LOB, it does not produce specific updates of the LOB at each timestep like a direct LOB simulation would. Instead, it employs simplified models and strong assumptions to simulate some of its aspects, providing a more abstract representation of a LOB.

After careful consideration, the third option was selected. The direct simulation of the LOB was deemed too complex and outside the scope of the project. Moreover, the lack of suitable data to calibrate the models would have resulted in a non-specific and potentially misleading representation of the LOB dynamics in the sports betting market. By leveraging a sports-specific price model in conjunction with the AS framework, the simulated environment provides a more accurate and practical basis for testing various market making strategies in the context of sports betting.

In the following section, the details of the model that simulates the price time series will be explained. The details of the AS framework have been already explained in Section 2.3 and Section 4.1.2 will dive into the adjustments made to adapt it to the sports betting market.

4.1.1 Tennis Markov model

Tennis, by its inherent nature, is a game of sequences. Each match is organized into points, games and sets, with each phase representing a distinct state. This sequential aspect of tennis can be effectively represented through a Markov model (Fig. 9), where each state in the model corresponds to a particular point score in the game (e.g., 0-15, 0-30, 15-30, etc.). In this model, the transition between states is determined by the probabilities associated with a player's performance, specifically their likelihood of winning a point when serving and when returning. Each player's probability of winning a point under these circumstances forms the basis for the transition probabilities between states in the Markov model. Thus, through the use of the Markov model, the point-by-point progression of a tennis match can be effectively simulated. The model outputs the time series of the probability of one player winning the match at every point. From these probabilities, the odds (or price) can be directly derived as the inverse of the probability $(1/p)$.

The following is an explanation of how the model works. It starts by defining p as the probability that the server wins a given point during the game. The probability that the returner wins the point is then $(1 - p)$. Since each point in tennis can be viewed as a Bernoulli trial with two possible outcomes (either the server or returner wins), the total number of points won in a game follows a Binomial distribution. The probability that the server wins

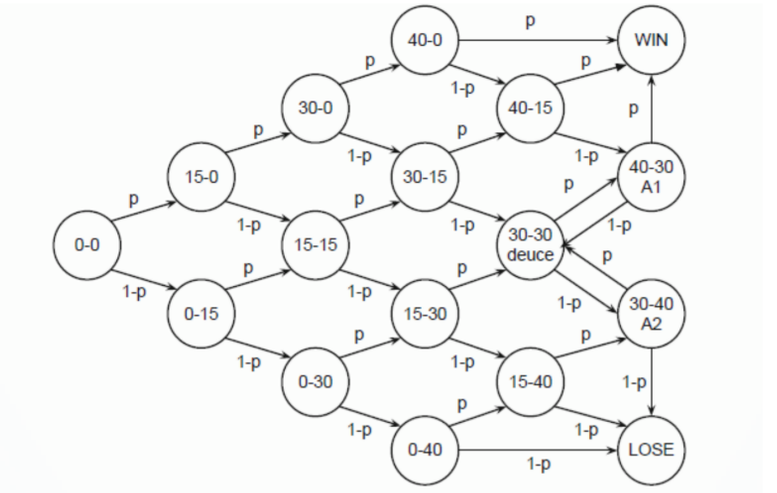


Figure 9: Markov model that represents the structure of a tennis game.

x out of n points is:

$$P(x;n,p) = \binom{n}{x} p^x (1-p)^{n-x} \tag{23}$$

The total probability $P(Win\ Game)$ of the server winning a game is the sum of the probabilities of winning to love (4-0), winning to 15 (4-1), winning to 30 (4-2), and winning after deuce (5-3 or more):

$$P(Win\ Game) = P_g(4,0) + P_g(4,1) + P_g(4,2) + P(Deuce) \tag{24}$$

where $P_g(4,0) = P(4;4,p)$, which corresponds to winning all of the four points played, $P_g(4,1) = P(4;5,p)$, $P_g(4,2) = P(4;6,p)$ and so on.

The calculation of $P(Deuce)$, the probability that the server enters deuce and wins, can be complex due to the condition of winning two consecutive points from deuce. In particular, the probability of winning from deuce is the product of $P_g(3,3)$ and an infinite sum of the possible states from which the player can win two consecutive points:

$$P(Deuce) = P_g(3,3) \sum_{n=0}^{\infty} P_g(n+2,n) \tag{25}$$

Luckily, the infinite sum is an infinite geometric series that can be reduced, resulting in the following equation:

$$P(Deuce) = P_g(3,3) \frac{p^2}{1-2p(1-p)} \tag{26}$$

Once the probability of winning a game is calculated, it is possible to calculate the probability of winning a set with a similar logic. In particular, the probability of a player winning a set can be computed as the sum of the probabilities of all possible game scores that would

lead to a victory in the set:

$$P(Set) = P_s(6,0) + P_s(6,1) + P_s(6,2) + P_s(6,3) \tag{27}$$
$$+P_s(6,4) + P_s(7,5) + P_s(6,6)P(Win\ Tiebreak) \tag{28}$$

Each of these probabilities is calculated similarly to the probability of winning a game. For example, the probability of winning a game score 6-1 would be the probability of the player winning 6 games out of the 7 played and be calculated with the Equation 23, substituting the p with the probability of winning a game. Note that in tennis, there are different formats of set scoring and rules, but for the sake of this project, the tiebreak set rules were adopted. Under these rules, if a set reaches a score of 6-6, a final tiebreak game is played to determine the winner of the set.

With the probability of winning a set calculated, it is now possible to compute the probability of winning a match. In particular, this is the sum of the probabilities of all possible set scores that result in a match victory:

$$P(Match) = P_m(2,0) + P_m(2,1) \tag{29}$$

Once again, for determining the overall winner of a tennis match, different rules can apply. In this project, the best-of-three type is used, where the possible outcomes to win a match are winning 2-0 or 2-1.

In using the Markov model for simulating the progression of a tennis match, it is essential to acknowledge certain assumptions it employs which, while simplifying the modelling process, might limit its accuracy. One of the primary assumptions is the Markov property itself, which states that a state depends only on itself and not on the sequence of states that preceded it. However, in tennis, the manner in which a player arrives at a certain point can significantly influence the outcome. For instance, if a player has consecutively won or lost the last points, made a comeback, or experienced other notable game dynamics, their mental and physical state might be affected, which can subsequently impact their performance. Moreover, the model is based on the two key probabilities that a player wins a point while serving or returning. These probabilities are typically estimated using a frequentist approach, which entails counting the number of points won out of the total points played. In real-life scenarios, however, these probabilities are influenced by a multitude of factors such as the type of court, the period of the year, player injuries, and so on. Consequently, the estimated probabilities might not be completely precise. Additionally, the model assumes these two probabilities to remain constant throughout the match. This overlooks real-world factors such as fatigue, psychological shifts, and other game dynamics, which can significantly impact a player's performance and hence, the probability of winning a point.

Furthermore, it is important to note that while the Markov model does a good job in simulating the overall progression of the match, it does not account for price fluctuations that occur during the points or in between them. In the real-world betting environment, traders continue to trade and move the price around during these periods. These intrapoint and interpoint price movements, driven by trader behavior and other external factors, are not captured in the Markov model.

In conclusion, despite these simplifications, the Markov model remains a valuable tool in this context. It may not perfectly mirror every aspect of a real-life tennis match and of a trading environment, but it retains key properties that are essential for our purpose of simulating the price time series in the betting environment. It represents a balance between realism and computational tractability and provides a reasonably accurate and practical basis for testing various market making strategies.

4.1.2 Adjustments to the Avellaneda-Stoikov framework

The AS framework, in its original conception, was tailored to suit the mechanisms of financial markets with a specific emphasis on equities. Hence, in order to maintain the fidelity of the used trading environment to the idiosyncrasies of the sports betting market, it becomes necessary to institute specific adjustments to the AS framework. Not making these necessary modifications might lead to a representation that does not accurately capture the essence of sports betting, failing to pursue the primary objective of the simulated environment of offering an authentic representation of the sports betting market.

The first adjustment regards the mid-price model. In the original AS framework, the mid-price was simulated using a Brownian motion. This stochastic process is commonly used in finance to represent random movements. However, in this project, as already mentioned in the previous section, the Tennis Markov model is adopted to simulate the price time series.

Moreover, in the original AS framework, the inventory was represented as a single numerical value, which encapsulated the net amount of the asset being held. However, the intricacies of sports betting demand a different approach to representing the inventory. More specifically, the inventory represents the cumulative betting position that the agent or model retains as a consequence of all the bets it has placed. Therefore, it is represented as a single back bet, characterized by specific stakes and odds. The crucial point of this representation hinges on the ability to combine several bets, irrespective of them being on the back or lay side, into a singular back bet. This is accomplished by first converting all lay bets into back bets but with a negative stake and once this transformation yields an array of back bets, these can be combined into a single representation. The stakes of the final bet are the sum of the stakes, while the odds are calculated as a weighted average. An essential aspect to note here is that both stake and odds can take negative values within this framework. Specifically, a negative stake is indicative of a lay position. On the other hand, negative odds signal that the agent has secured a loss, when the stake is positive (back position), or a profit when the stake is negative (lay position).

Furthermore, the difference between the two main operations in financial markets (buying and selling) and in sports trading (backing and laying), introduces distinctions in the way the value of a position (or inventory) is represented. In particular, the original AS framework represents the value of an inventory by associating it with the potential cash that an agent or model could secure if it decided to liquidate its entire inventory at that current timestep. Hence, the value was conceived as the product of the inventory and the current mid-price. However, in this context, because of the backing and laying mechanisms, the value of an inventory cannot be understood as the potential cash one could secure from liquidating bets.

Instead, it should be seen as the potential profit that can be ensured by placing a counter-bet. This act of counter-betting, often called "cashing out", involves placing a bet with a calculated volume (stake) on the opposite side of the order book, ensuring a profit regardless of the event's outcome (an example is shown in Section 2.2). It is crucial to note that this profit could either be positive, if the odds moved in favour of the position (decreased for a back position or increased for a lay one), or negative if they moved in the opposite direction.

The following is an explanation of how the PnL is derived at any given timestep t. If the inventory of the model at time t is represented as (S_t, O_t) where S_t is the stake and O_t is the odds, and the current mid-price (or the current odds) is denoted by p_t, the PnL is derived in the following way. Firstly, determine the stake or volume S_l that the model would need to bet on the lay side at the current mid-price. This can be calculated using:

$$S_l = \frac{(O_t + 1) * S_t}{p_t} \tag{30}$$

Using S_l, the PnL can be calculated as the difference in the profit when both positions (S_t, O_t) and $(-S_l, p_t)$ are taken:

$$PnL = S_t * O_t - S_l * p_t = S_l - S_t \tag{31}$$

where $S_t * O_t - S_l * p_t$ and $S_l - S_t$ represent the profit of the two possible outcomes of the bet (which in this case are equal).

Additionally, in the traditional AS framework, the cash process had an important role, being included in the calculation of the PnL (Profit and Loss). In particular, in the financial markets context, buying can be seen as an investment of cash into an asset and selling as an extraction of cash from that investment. Hence, in this dynamic, cash and inventory act as counterbalances: a rise in cash corresponds to a decrease in inventory and vice-versa. However, the sports betting environment does not align with this rationale. Backing and laying cannot be seen as investing or liquidating cash from a bet. Therefore, including cash in the PnL does not resonate with the sports betting framework, hence it was removed from its representation.

In conclusion, the sports betting market's distinct mechanisms required to diverge from the traditional AS framework. The Brownian motion that simulated the mid-price was substituted with the Tennis Markov model, the inventory representation was modified to adhere to the betting mechanisms and consequently, its value representation was changed, shifting its value from the obtainable liquidation cash to the profit achievable through the cash-out mechanism. The inclusion of cash in the PnL, standard in financial settings, was omitted due to its misfit in the sports betting environment. These adjustments ensure that the simulation closely mirrors the real sports betting market, providing a valid platform for evaluating baseline models and RL agents.

4.2 Baseline Models

In order to evaluate the effectiveness of the developed RL agents (Chapter 5), it is crucial to compare them against baseline models. In this study, baseline models are evaluated from

two primary categories: the fixed-offset model and the random-offset model. In particular, within the fixed-offset model, three distinct offsets (0.2, 0.5, and 0.8) are tested, each forming its own individual model, resulting in a total of four baseline models.

The fixed-spread model, as the name suggests, quotes a constant spread irrespective of the state of the market. Let the constant spread be denoted by s. If S_t is the mid-price (simulated with the Markov Model) at time t, the bid and ask prices, B_t and A_t respectively, are given by:

$$B_t = S_t - \frac{s}{2} \tag{32}$$

$$A_t = S_t + \frac{s}{2} \tag{33}$$

Unlike the fixed-spread model, the random-spread model introduces a degree of variability in the quoted spread. In this model, the spread s_t at time t is randomly chosen from a uniform distribution within a predetermined interval $[s_{min}, s_{max}]$. The bid and ask prices, B_t and A_t respectively, are then determined as:

$$B_t = S_t - \frac{s_t}{2} \tag{34}$$

$$A_t = S_t + \frac{s_t}{2} \tag{35}$$

Another potential baseline model worth mentioning is the AS optimal quotes model, as introduced in [4] and explored in Section 2.3. This model could have served as a baseline for this study, however, due to the modifications made to the original AS framework, as discussed in the preceding subsection, the equations derived for the optimal quotes no longer hold relevance or logical coherence. This deviation from the original framework made the AS optimal quotes model unsuitable as a valid point of comparison.

4.3 Testing and Results

To evaluate the efficacy and performance of the baseline models, they were subjected to a rigorous testing environment. The primary testing ground was constructed using the tennis Markov model, which simulates the price time series, and the AS framework, which encapsulates the complexities and nuances of real trading settings by accounting for aspects such as competitiveness on order, order arrivals and others. The Markov model, as previously discussed, provides a realistic and dynamic representation of price movements based on the probability transitions inherent in a tennis match. Utilizing the tennis Markov model allows for simulating real-world price fluctuations and uncertainties commonly found in trading scenarios. Fig 10 shows an example of simulated price time series. On the other hand, the AS framework offers a comprehensive representation of the trading world, and by integrating it into our tests, it is ensured that the baseline models are not only tested against price movements but also against the intricacies of trading dynamics.

4.3.1 Testing methodology

To evaluate the performance of each baseline model, a detailed testing process was conducted. Each model was subjected to multiple simulations, derived from various combina-

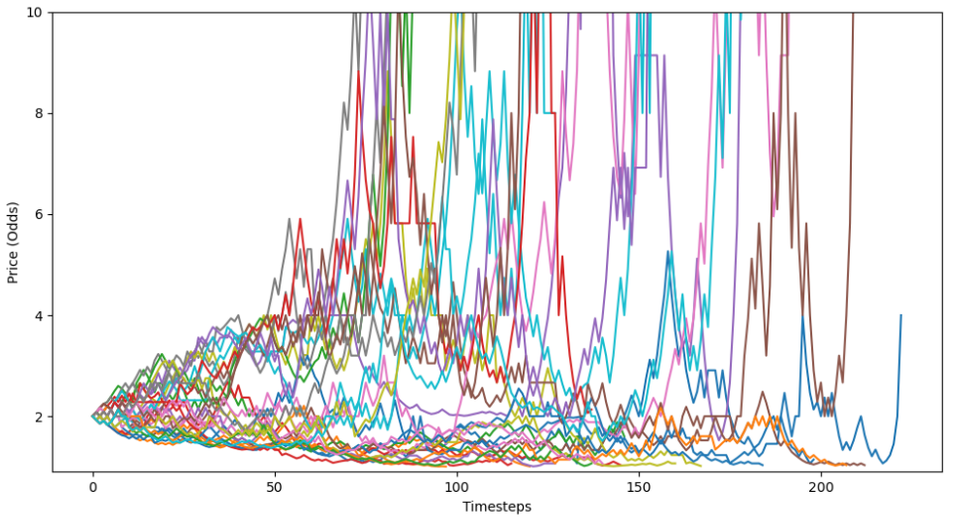


Figure 10: Example of price time series simulated with the Markov model.

tions of the environment's parameters. The goal was to measure how the models reacted under different scenarios and ensure the robustness and generalizability of the results. The parameters that were varied for these simulations include:

- Markov Model Probabilities: the two probabilities denote the chances of each opponent winning on their serve. Literature suggests a plausible range of 60-70% [34]. Hence, for this analysis, combinations starting from 0.60 up to 0.70 were considered, incremented by 0.02, resulting in 36 possible combinations.
- Parameter k of the AS Framework: a crucial element for the Avellaneda-Stoikov framework, this parameter represents market liquidity and directly influences the probability of orders getting filled. Values for this parameter were explored ranging from 3 to 12, incremented by 1, leading to a total of 10 distinct values. The choice of this range was based on its perceived plausibility. As previously highlighted, there was a lack of proper data to calibrate the parameters of the simulated environment. Consequently, a plausible range was selected to ensure meaningful simulation results.

Given these parameter selections, the models were tested using a total of 360 combinations. For every combination of parameters, 100 unique iterations were performed. Each iteration generated a distinct price time series, emulating the dynamics of a tennis match. Hence, each model was tested on 36000 iterations in total. This rigorous testing approach offered insights into the behaviour of each baseline model, ensuring the robustness of the results.

4.3.2 Performance and Risk metrics

To critically assess and compare the models' efficacy, it is fundamental to introduce a comprehensive set of evaluation metrics. The metrics used are classified into two primary categories: performance and risk. The reason for employing both types of metrics stems from the inherent trade-offs between the two. While high returns are always desired, they often come with increased risk. A model that yields high profits but experiences significant

volatility might not be sustainable in the long run. On the other hand, a model that showcases moderate profits but with minimal volatility might be more reliable. By assessing both the reward (performance) and the potential downside (risk) of each model, we ensure a balanced and complete comparison. This dual evaluation allows us to identify models that strike an optimal balance between maximizing returns while managing and mitigating risks.

The following are the performance metrics used:

- Final PnL (Profit and Loss): represents the net monetary value gained or lost by the model at the end of the trading period (one tennis match), offering a straightforward measure of profitability. Note from Section 4.1.2 that the PnL is calculated as the potential profit the agent can lock with the current position, using the cash-out mechanism.

- Maximum PnL: stands for the highest point on the PnL curve over the trading period. It represents the maximum profit achieved by the trading model during that timeframe. Maximum PnL offers a glimpse into the potential upside of the model, portraying the best-case scenario in terms of financial gains. This metric helps traders understand the profitability of their strategy during favourable market conditions. It is instrumental in assessing the effectiveness of a trading model and in strategizing for future trades.

- Mean Return: evaluates the average return earned by the model over the trading period. Unlike the final PnL, which provides the total profit or loss over the entire period, the Mean Return measures the consistent performance of the model at regular intervals, giving insights into its stability and regularity. While the final PnL offers a snapshot of the model's overall success or failure, the Mean Return allows for an understanding of bet-to-bet profitability. It is crucial to include both to get a comprehensive view of the model's effectiveness and consistency. The Mean return is calculated as follows:
$$\text{Mean return} = \sum_{t=0}^{T} \frac{PnL_{t+1} - PnL_t}{PnL_t} \tag{36}$$
where T is the total number of timesteps in the trading period.

- Sharpe Ratio: evaluates the returns of the model relative to its risks, offering insights into the risk-adjusted performance of a model. The Sharpe Ratio is calculated as follows:
$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p} \tag{37}$$
where R_p is the expected return, calculated as the mean of the returns, R_f is the risk-free rate, which in this case is considered as 0 and σ_p is the returns' standard deviation.

- Sortino Ratio: concentrating solely on negative volatility, it is a nuanced metric that considers only downward risk, useful for identifying potential substantial drops. Unlike the Sharpe Ratio, which considers both upward and downward volatility, the Sortino Ratio concentrates solely on negative volatility. By focusing only on the downside, it offers a more precise measure of a model's risk relative to its potential for neg-

ative returns. The Sortino Ratio is calculated as follows:

$$\text{Sortino Ratio} = \frac{R_p - R_f}{\sigma_d} \tag{38}$$

where σ_d is the standard deviation of the negative returns.

On the other hand, the risk metrics used include:

- Volatility of returns: calculated as the standard deviation of the returns, it measures the potential variability or risk associated with the returns of a model.
- Mean Inventory Stake: indicates the average position held by the model, offering insights into the model's typical exposure and potential vulnerabilities. Note that the inventory stake was chosen as a risk metric because its magnitude directly correlates with the associated risks; the larger its absolute value, the greater the liabilities. Conversely, inventory odds were not selected as risk metrics because its value, in itself, does not signify risk. As previously mentioned, we operate under the assumption that bets are never held until the match's conclusion. As such, the risks stemming from the odds are purely relative to the current mid-price (or mid-odds). For instance, even if a model has high inventory odds, if the current mid-price (mid-odds) are also high, the associated risk is not necessarily substantial. However, a high inventory stake always indicates a heightened risk level.
- Minimum PnL or Maximum Loss: refers to the minimum point of the PnL curve over the trading period. This metric provides insight into the downside risk of the model, essentially showing the worst-case scenario in terms of financial loss. By understanding the Minimum PnL, traders can make informed decisions regarding risk management, setting stop losses, and allocating capital. It is an essential metric to gauge the vulnerability of a model during adverse market conditions.

4.3.3 Results

For clarity in this section, a simplified naming convention will be adopted for the models under discussion. The models with fixed offsets will be referred to by their respective numbers (e.g., the model with a fixed offset of 0.2 will be called "Fixed_0.2"), and the random offset model will be termed "Random".

Upon analyzing the various metrics (Table 4 and Fig. 11), it becomes evident that the Fixed_02 model stands out with superior performance metrics values (the very high value of the Sortino ratio of the Random model is given by a very skewed distribution, hence the Sortino ratio of the Fixed_02 has, in reality, a better value). However, the better performances come at higher risks. In fact, the risk metrics of the Fixed_02 model have the worst values (apart from the volatility, but again the very high value is caused by a distribution with very high skewness).

In conclusion, as anticipated, all four models display limited performances. However, despite their inherent simplicity and the expected outcomes, surprisingly the mean final PnL for all four models yields positive values. This is likely thanks to the simulations with a low

Model	Final PnL	Mean Return	Volat.	Min PnL	Max PnL	Sharpe ratio	Sortino ratio	Mean Inv. Stake
Fixed_0.2	**0.91**	-0.06	10.12	-1.39	**2.63**	**0.05**	1.17	0.02
Fixed_0.5	0.65	**0.005**	3.38	-0.55	1.43	0.03	0.43	**-0.001**
Fixed_0.8	0.31	6e-4	**1.63**	**-0.25**	0.68	0.01	0.23	0.002
Random	0.66	-6e-8	1e+10	-1.37	2.26	0.02	**1e+9**	-0.005

Table 4: The table shows the mean value for all the metrics used to evaluate the baseline models.

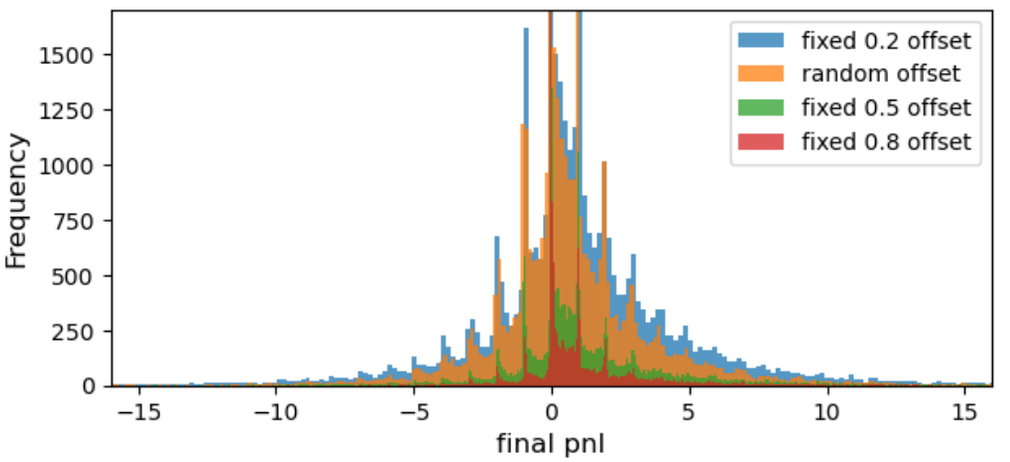


Figure 11: The plot shows the distributions of the Final PnL metric for the four baseline models.

value of the parameter k (indicative of high market liquidity) counterbalancing the simulations with a high value of k, where the models recorded negative PnL.

4.4 Discussion

In this experiment, the focus was on the implementation and evaluation of four baseline models to establish a benchmark for market making in sports trading. These models, while simplistic in their design, served as an essential starting point for understanding how traditional approaches perform in the unique environment of sports trading. Additionally, a crucial aspect of this experiment was the simulation of the trading environment, which was foundational for both this and subsequent experiments.

The trading environment simulation served dual purposes. First, it provided a controlled setting in which the baseline models could be rigorously tested and evaluated. Second, it established a flexible framework that will be integral for the training and testing of RL agents in Chapter 5. This simulation framework allows for accurate replication of pivotal real-world market aspects while offering computational efficiency and the versatility to adjust various parameters, thereby providing a robust platform for future research. However, several limitations were encountered in the design process. In particular, the lack of granular data on order arrivals restricted the ability to calibrate or train models that directly simulate the LOB. This constraint made it difficult to create a fully realistic trading environment, which in turn could impact the generalizability of our findings to real-world scenarios.

Finally, the performance and risk metrics revealed that these baseline models had limited effectiveness, reinforcing the need for more sophisticated approaches to navigate the complexities of sports trading markets.

Chapter 5

This chapter dives into the culmination of the research journey: the development, training, and evaluation of an RL agent tailored for market making in sports betting markets. Building upon the simulated environment and baseline models discussed in the previous chapters, this segment explains the rationale behind employing RL and describes the design and structure of the agent. The chapter will then detail the training process and highlight the results obtained, providing insights into how RL can be applied to sports betting market making.

5 Development, Training and Testing of a novel RL agent

In this final experiment, the primary objective was to design, train, and test an RL agent tailored for market making in the sports betting market. Building on the foundational knowledge from the previous chapters and leveraging the simulated environment from Chapter 4, this experiment sought to answer whether an RL agent can outperform the baseline models.

5.1 Implementation tools

For the execution of this and the previous experiments, the Python programming language was employed due to its versatility and a wide array of libraries and tools in the field of machine learning and data analysis.

The primary tool for implementing the trading environment was Gymnasium. This open-source library offers a platform to develop custom RL environments and evaluate different algorithms. On the other hand, to implement the RL algorithms and architectures, Stable-Baselines3 was chosen. An enhancement over the original Stable Baselines library, it provides a collection of reliable implementations of RL algorithms. One of the key advantages of employing Stable-Baselines3 was its capability to allow the focus to remain on the design and conduct of the experiments, eliminating implementation overheads. In addition, Tensorboard was utilized for efficient logging of results, visualizations, and hyperparameters throughout the training processes.

By harnessing these tools, an efficient workflow was achieved, enabling greater concentration on the core objectives of the experiment rather than technical impediments.

5.2 Environment representation

Designing a suitable trading environment for RL necessitates the careful selection and representation of observation space, action space, and reward structure.

5.2.1 Observation space

The observation space is a pivotal component in the realm of RL, acting as the lens through which the agent perceives the environment. A robust and rich representation is paramount for the agent's success, as it empowers him with a comprehensive view of the environment

and provides him with the essential tools to make well-informed decisions. Additionally, with the capabilities of Deep Learning models, agents are well-equipped to process and draw meaningful insights from elaborate observation spaces.

As mentioned in [35], defining the observation space is one biggest challenges in tackling a trading task with RL. In particular, the observation space is a pertinent distinction between applications where RL had huge successes, like games or toy problems, and finance problems. In games, the state space, albeit complex, is naturally and unequivocally defined by the problem itself. For instance, in board games like chess, the arrangement of the pieces dictates the state, while in video games, the pixels might define it. However, in financial problems, the definition of the state space becomes more ambiguous and can be viewed as a product of modelling choices. Taking the example of a limit order book, while its state (often reducible to a few limits) and perhaps its history are innate components, the state space can be broadened to incorporate a plethora of signals. These might range from market trends, historical or implied volatilities and market volumes, among others. The vastness and variability of the state space present a challenge, as it becomes difficult to ascertain the relevant variables from the redundant ones. This underpins the significance of carefully crafting the observation space in trading problems.

In the initial stages of the experiment, the observation space was characterized by a straightforward setup. It encapsulated details about the inventory, such as the stake and odds, alongside the current timestep. Subsequently, as expected, the need to enhance the agent's observation emerged. In particular, this need stemmed from a critical distinction between the nature of market making in sports betting and in other financial contexts, like equities. In particular, while in equities the market making task is often modelled such that the agent is driven to maintain a mean-reverting inventory, thereby leading to a risk-neutral strategy with hopefully an increasing cash process, this strategy is not directly translatable to sports betting. Within the framework of sports betting and the trading environment modelling employed in this study, aiming for a mean-reverting inventory does not correlate with profits. Consequently, while in equities, a representation that focuses solely on the inventory and the current timestep might suffice, in sports betting, such a representation proves inadequate. Hence, several richer representations were tested.

In particular, in addition to the inventory and timestep, these were added:

- Current Price: Unlike other markets where the absolute value of the price might hold relevance only in relation to its past values, in sports betting, the absolute price value in itself offers crucial information because it represents the underlying odds.
- Momentum Indicator: To give the agent a clearer understanding of the current market trends, a momentum indicator was added to the observation space. Momentum indicators are crucial in trading as they highlight the direction and strength of a market trend. They can be pivotal in making buying or selling decisions as they signal the continuity or change in the market direction. The momentum indicator M_t was calculated as follows:

  $$M_t = p_t - p_{t-k} \tag{39}$$

  where p_t is the price a timestep t and k a constant set to 15.

- Volatility Indicator: By incorporating the volatility indicator, the agent can better gauge the potential risks and rewards of the market. It enables the agent to understand market fluctuations better and adjust its strategy accordingly. The volatility indicator V_t is calculated as the standard deviation of the price values from $t - k$ to t.

5.2.2 Action space

The action space delineates the set of possible moves or decisions an agent can make at any given timestep in response to the environment's current state. In the sports betting context, while the challenge in defining the observation space lies in its intricacy, defining the action space is relatively more straightforward. In the established research, the majority of existing works opt for discrete actions, typically specifying the fixed offsets that the agent can quote. This approach offers a simple and concise representation, making it both intuitive for the agent and computationally efficient.

Hence, for this study, the action space was conceptualized as a discrete set, representing the back and lay offsets (from the mid-price) the agent can decide to quote. Given the dynamic nature of sports betting markets and the results of the analysis of the first experiment (Chapter 3) on spread features, a range from 0 to 1.0 (excluded) with increments of 0.1 was chosen for these offsets. This configuration yields 10 possible offsets for both the back and lay sides, resulting in a total of 100 potential action combinations. Furthermore, by incorporating 0 as an offset, it effectively introduces a market order as a viable action for the agent. The rationale behind including a market order action is to provide the agent with the flexibility to ensure the execution of an order on a particular side, regardless of achieving an optimal price. In specific market conditions, such as high volatility, the agent might prioritize execution certainty over obtaining a favourable price. This option provides the agent with the adaptability to navigate such scenarios effectively.

5.2.3 Reward function

The reward function is a crucial aspect of RL as it provides feedback to the agent on the quality of its decisions and drives its learning process. A well-defined reward function can significantly influence the effectiveness and efficiency of the agent's learning, guiding it towards desired behaviours and optimal strategies.

In this study, the reward function used is based on the PnL. This measure provides immediate feedback based on the agent's performance in monetary terms. However, while this approach offers a direct evaluation of the agent's trading decisions, it also presents certain challenges when applied to the market making scenario.

The primary challenge of designing a reward function for market making scenarios is to encourage the agent towards liquidity provision without speculating on the direction of price movements. Simply rewarding based on raw PnL can inadvertently promote speculative behaviours, as the agent is rewarded for profits made due to favourable price shifts. To address this, literature offers solutions like introducing an inventory penalization. This approach aims to ensure that the agent aspires to maintain a minimal inventory, thereby negating the opportunity to profit from price fluctuations. However, in the sports betting

domain, this approach is not directly applicable. As previously noted, striving for a flat inventory in sports betting does not equate to profitability. This discrepancy underscores the need to adapt traditional market making reward strategies to the unique characteristics of the sports betting market, ensuring the agent is driven towards strategies that are genuinely profitable in this specific context.

Due to time constraints, the scope of this research was limited to testing the reward function based on the PnL. Future research could explore the implementation of more sophisticated reward functions that consider the nuances of sports betting, to develop agents that are better aligned with the optimal and desirable behaviour in this specific domain.

5.3 RL agents and architectures

The selection of an appropriate RL algorithm is pivotal to the success of the learning agent, especially in complex trading environments like sports betting. The landscape of RL offers approaches ranging from traditional algorithms to state-of-the-art Deep Learning-based techniques. For the scope of this project, the focus was primarily on Deep RL approaches. The rationale behind this choice lies in the empirical success Deep RL methods have demonstrated in the literature, especially in the context of trading and market making. Such financial domains often deal with high-dimensional input signals, characterized by noise and non-stationarity. Deep RL algorithms, with their intrinsic capability to process and extract meaningful patterns from complex data, have showcased an edge in handling these challenges more adeptly than their non-deep counterparts.

In the quest to find the most effective algorithm for this task, three different RL algorithms were explored, encompassing both value-based and policy-based approaches: Deep Q-Networks (DQN), Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO). As highlighted by [36], there is not a definitive consensus in the literature about the superiority of one approach over the other in market making scenarios. Both value-based and policy-based algorithms have been successfully employed in various market making scenarios, suggesting that their effectiveness might be context-dependent.

Value-based methods typically estimate the value of states or state-action pairs and derive a policy implicitly from these value estimates. On the other hand, policy-based methods aim to directly learn a policy without necessarily estimating values. Given the intricacies and nuances of the sports betting market, it was deemed essential to investigate the performance of both paradigms to ascertain which would align more cohesively with the project's objectives.

5.3.1 Deep Q-Networks (DQN)

Deep Q-Networks (DQN), introduced in [37], is a pioneering algorithm in the domain of Deep RL. Originating from the traditional Q-learning algorithm, its central principle is that it employs deep neural networks to approximate the Q-values, which represent the expected future rewards for each action taken in a given state. The utilization of neural networks in DQN enables it to manage larger and more complex state representations, which traditional

Q-learning struggles to handle due to its tabular nature.

One of the major innovations in DQN is the use of an experience replay buffer. This is essentially a memory bank where the agent stores its past experiences or transitions—each being a tuple consisting of the current state, action taken, reward received, next state, and whether the episode ended. During training, instead of learning from the most recent transition, the agent randomly samples a mini-batch of transitions from this replay buffer. This practice has two primary advantages: it breaks the correlation between consecutive samples, making the training process more stable and data-efficient; and it allows the algorithm to revisit and learn from previous experiences multiple times, which is particularly valuable in environments where collecting new experiences can be costly or time-consuming.

Additionally, DQN introduces a key architectural innovation: the use of two separate neural networks, the Q-network and the target network. The Q-network is the primary model that is updated continually and is used to select actions based on the current policy. On the other hand, the target network is a duplicate of the Q-network but is updated less frequently. It is used to compute the target Q-values during the learning phase, thereby providing more stable and consistent targets for updates. This separation mitigates the risk of the updates to the Q-values becoming oscillatory or divergent, an issue that can arise when the same network is used to both select actions and evaluate those actions.

5.3.2 Advantage Actor-Critic (A2C)

Advantage Actor-Critic (A2C), the synchronous version of Asynchronous Advantage Actor-Critic (A3C) [38], stands as a pivotal algorithm in the RL sphere, combining the strengths of both value-based and policy-based methods. Its name encapsulates its dual nature: "Actor" refers to the policy part of the model which outputs a distribution over actions, and "Critic" estimates the value function, guiding the Actor in refining its policy.

The core idea behind A2C, in addition to the dual aspect of the actor and the critic, is to reduce the variance in the policy gradient estimation, making the training process more stable. This is achieved by introducing the advantage function, which evaluates the relative value of taking an action compared to the average action value in a given state. In essence, the advantage function measures how much better or worse an action is compared to the average action in that state.

More specifically, in policy-based methods, the aim is to find an optimal policy $\pi(a|s)$ that maximizes the expected reward. In Deep RL methods, where the policy π is represented by a neural network, the optimal policy is found by updating iteratively the network's parameters θ with the policy gradient $\nabla_\theta J(\theta)$, where $J(\theta)$ represents the expected reward when following policy π_θ:

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta) \tag{40}$$

where α is the learning rate. In the general actor-critic algorithm, the policy gradient is calculated as follows:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^{\pi_\theta}(s,a)] \tag{41}$$

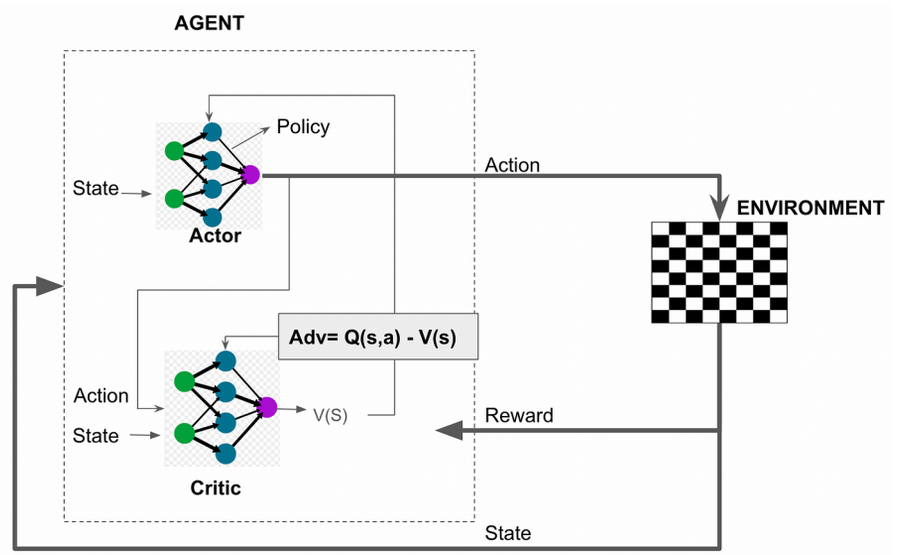


Figure 12: Illustration of the A2C algorithm.

However, directly using the action-value function $Q^{\pi_\theta}(s,a)$ can lead to high variance, making the training process unstable. A high variance means that the updates to the policy can be large, causing oscillations and slow convergence. To reduce variance, A2C uses the critic network to estimate the state-value function $V^{\pi}(s)$ that is then used to calculate the advantage function:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s) \tag{42}$$

The advantage function describes how much better or worse an action a is at state s compared to the average value of that state. By using the advantage function in the calculation of the policy gradient, the expression becomes:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) A^{\pi}(s,a)] \tag{43}$$

With this formulation, the updates are in the direction of actions that are better than the average, reducing the variance of the gradient estimator. Fig 12 shows an illustration of the mechanism just explained.

5.3.3 Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO), introduced in [39], has gained widespread recognition in the RL community for its effectiveness and efficiency. It builds upon traditional policy gradient methods but introduces modifications to improve stability and reduce the potential for large, destructive policy updates.

The essence of PPO lies in its objective function $L(\theta)$. Unlike traditional policy gradient methods which seek to maximize expected returns, PPO aims to keep the updated policy close to the old policy. It achieves this by adding a constraint to the objective function, ensuring that the new policy does not deviate too far from the old one:

$$L(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{44}$$

where

$$r_t(\theta) = \frac{\pi_{\theta_t}(a_t|s_t)}{\pi_{\theta_{t-1}}(a_t|s_t)} \tag{45}$$

represents the likelihood ratio between the new and old policy, and $\hat{A}_t$ is the estimated advantage function at time t. The clip function ensures that the ratio remains within a specified range, determined by the hyperparameter ϵ (set by the authors at 0.2). Through these modifications, PPO strikes a balance between allowing sufficient policy updates for learning and preventing excessively large updates that could harm the learning process.

5.3.4 Function approximators

For all the RL algorithms used in this study, one of the key components is the function approximator. In this project, for all the algorithms tested, shallow Multi-layer Perceptrons (MLPs) were used. The reason for this decision, as also highlighted in [36], is the type of input provided to these networks. In many cases, deep NNs handle raw data, hence the need for a deep architecture, to have the capacity to perform the feature extraction stage. In this case, because the networks receive handcrafted features, this capacity is not necessary.

Specifically, the chosen MLP designs have 2 layers with Tanh activation function, for A2C and PPO (both actor and critic networks), and 3 layers with ReLU activation function, for DQN.

5.4 Training

5.4.1 Training data and challenges

As previously elaborated in Chapter 4, the utilization of real historical data for training and testing both baselines and RL agents was confronted with challenges. A significant obstacle was the insufficiency of data, further exacerbated after a data cleaning stage that in case of utilizing the data for training and testing would have been needed. Consequently, simulated data emerged as the viable choice for the training and testing processes.

Drawing from insights in [35], it is imperative to highlight an intriguing paradox that surfaces in this approach. Even if the RL approaches employed are model-free, the constraints tied to the unavailability of real historical data necessitate the adoption of a model that simulates the trading environment. Hence, even if the RL algorithms utilized do not necessitate a model of how the trading environment evolves, simulating the data that these algorithms necessitate to train, requires a model of the environment. In this project's scenario, this model was constructed using a variation of the AS framework complemented by a tennis Markov model. This strategy, while providing a foundation, does come with its inherent limitations. The challenge lies in the fact that the RL agent is trained and tested on data derived from a simplified representation of the actual trading environment. This model, by its very design, is bound to exclude numerous intricacies and nuances integral to a real-world trading ecosystem.

Moreover, due to the trading environment model's omission of the LOB, further limitations

are encountered. Several features, frequently acknowledged in literature such as Order Flow Imbalance, Queue Imbalance, and Spread, become inaccessible as state-space variables. This means that a rich source of information, often instrumental in crafting more robust RL models, is left unused.

5.4.2 Hyperparameters and Reproducibility

Reproducibility remains a pressing concern in many RL studies, often leading to scepticism regarding the validity of certain research findings. Recognizing this issue, as highlighted by the authors in [40], in this project, a significant emphasis was placed on ensuring reproducibility across all experiments.

To maintain a consistent environment, it was ensured that random seeds of all libraries utilizing random generators were set and kept invariant throughout the duration of the project. These seeds are meticulously recorded in a configuration file within the repository to ensure that anyone attempting to reproduce the results can recreate the exact conditions of the experiments. Furthermore, to eliminate ambiguity and potential deviations, all hyperparameters associated with the RL agents are explicitly documented and made available.

By taking these measures, this project strives to maintain transparency, reliability, and adherence to best practices, thus enhancing the integrity and validity of its findings.

5.4.3 Training procedure

The training of the RL agents starts with the initialization of the weights of the agent's function approximators. This was achieved using the default method available in PyTorch. Specifically, the weights of the linear layers of the MLPs are initialized through the Kaiming method [41]. The Kaiming method, in this case, employs a uniform distribution, with the bounds $(-\frac{1}{\sqrt{X_{in}}}; +\frac{1}{\sqrt{X_{in}}})$, where X_{in} is the number of input features. The choice of the Kaiming initialization method was motivated by its proven advantages: it fosters faster convergence, stabilizes training and addresses common challenges in DL like the vanishing and exploding gradient problems.

For evaluation and monitoring purposes, the primary metric employed was the mean final reward over 100 episodes. It is important to highlight that other metrics frequently used in RL, such as the mean episode length, were not pertinent in this particular context. In this scenario, the mean episode length is not indicative of the agent's performance but rather is a random variable determined by the length of the trading period, which in turn is decided by the length of the price time series simulated by the tennis Markov model.

One of the crucial decisions to take for the training of the RL agents was the configuration of the environment. Specifically, the possible options were two:

- Fixed environment configuration: In this approach, the agent is trained in a static environment with set parameters. Such a method provides a stable training environment, but it may restrict the agent's exposure to only a subset of conditions. This could

potentially make the agent less adaptive when tested with diverse environment configurations and in real-world scenarios.

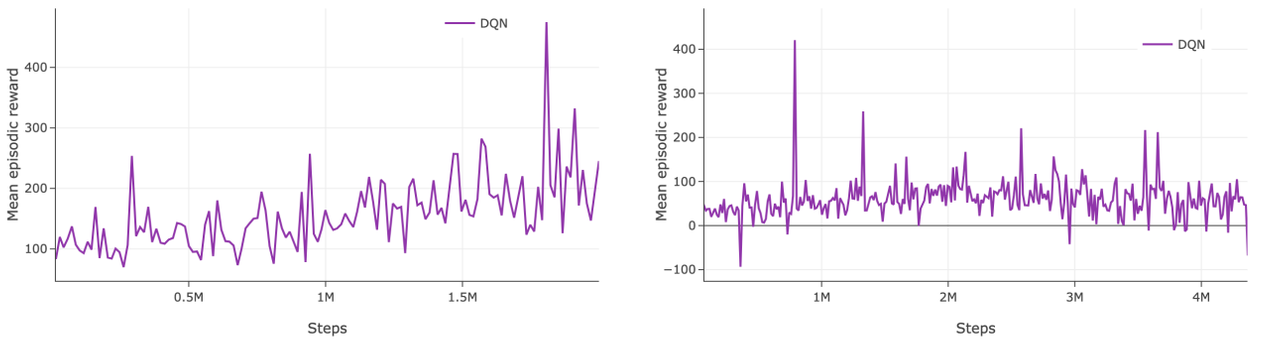
- Randomized environment configuration: Conversely, this strategy advocates for altering the environment parameters randomly throughout the training. By randomizing these parameters, the agent gets trained on a broader spectrum of conditions, which might improve its overall adaptability.

Specifically, the mentioned parameters are the same that were used in Chapter 4 in the testing procedure of the baseline models: the two Markov Model probabilities and the parameter k of the AS framework. Notably, the k parameter is pivotal as it represents the market liquidity. Consequently, varying values of this parameter depict very different market conditions.

Regarding the two options, while the second is without a doubt the more complete option, it also has the risk of introducing instability into the training process. Changing the environment parameters consistently might throw the agent off its learning trajectory, making it challenging to maintain a steady learning curve, hence impeding the agent's successful training. On the other hand, while with the fixed environment, one might guarantee stable training, it would also sacrifice the adaptability and flexibility of the agent when faced with different conditions that were not part of the training environment. Hence, to determine the best approach to use for the rest of the experiment, both options were tested on one of the three algorithms, which is DQN. In the fixed environment configuration, the specific parameters were set as follows:

- The parameter k was assigned a value of 4, indicative of a high-liquidity market. The rationale behind this specific choice stems from the inherent nature of the parameter itself. As mentioned, the parameter k represents market liquidity. A value of 4 was chosen because it represents a market with fairly high liquidity. High liquidity is vital for the training of the RL agent because it allows the agent to receive more immediate feedback on its actions. The rewards become more closely tied to the agent's actions, rather than being predominantly influenced by price fluctuations. This direct correlation and immediate feedback facilitate more informed learning and decision-making. In scenarios with low liquidity markets (where k had a higher value), it was observed that the agent could place only about 1-3 orders during a match. This limitation meant that for the majority of its actions (prices quoted), the agent did not get to witness the direct impact of its decisions. Such an environment could hinder the learning process and the agent's ability to adapt effectively.

- The two Markov Model probabilities were both set to 0.65. This was chosen to simulate a balanced match where each player has an equal chance of winning a point. Such a setup guarantees that the price time series generated by the Markov model will be evenly split between those converging to 1 (indicating a win by the player) and those that eventually will rise (indicating a loss by the player), ensuring the lack of bias towards any specific outcome in the training stage.



(a) *(b)*

Figure 13: The two plots show the mean episodic reward of the training of DQN using the two environment configurations: fixed parameters (a) and varying random parameters (b). As noticeable, in (a) the mean episodic reward shows a positive trend, while in (b) the mean episodic reward shows no substantial growth even after 5 million steps.

5.4.4 Training results

Starting from the test of the two training configurations (just mentioned in the previous section), the results (Fig. 13) highlighted that, as expected, the varying environment resulted in an unstable training regime, with the mean episodic reward showing no growth after 5 million training steps. Conversely, with the fixed environment, in 2 million steps the plot shows evidence of a positive trend, suggesting some learning progress. Hence, the fixed configuration was chosen for the subsequent training of the other algorithms. All three algorithms were trained for a total of 4 million steps.

While all the plots show positive growth, it is evident that the mean episodic reward does not exhibit outstanding growth throughout the training (Fig 14), as it happens in other RL contexts. This observation aligns with the initial expectations given the intrinsic complexity of the task and of the environment. While the AS framework serves as a simplification, it effectively mimics the core challenges of a real trading environment that incorporates a LOB. One of the primary challenges comes from its competitive nature. The AS framework clearly emulates this competition, making it challenging for the agent to gain a significant and stable profit across the various simulations. Further accentuating this challenge is the agent's limited view of essential features of the environment. As highlighted in Section 5.4.1, the agent is deprived of key LOB features. These features, which are central to numerous studies, play a pivotal role in predicting near-future price fluctuations and can offer insights that could enhance the agent's predictive capabilities.

5.5 Testing and Results

The trained RL agents underwent testing using the identical procedure detailed in Section 4.3.1, which was employed for the baselines' evaluation. Adhering to this methodology ensures not only the accuracy and validity of the results but also their comparability. Additionally, the trained RL agents were also tested in the fixed configuration environment they were trained on (with k=4 and the Markov model's probabilities set to 0.65). The pri-

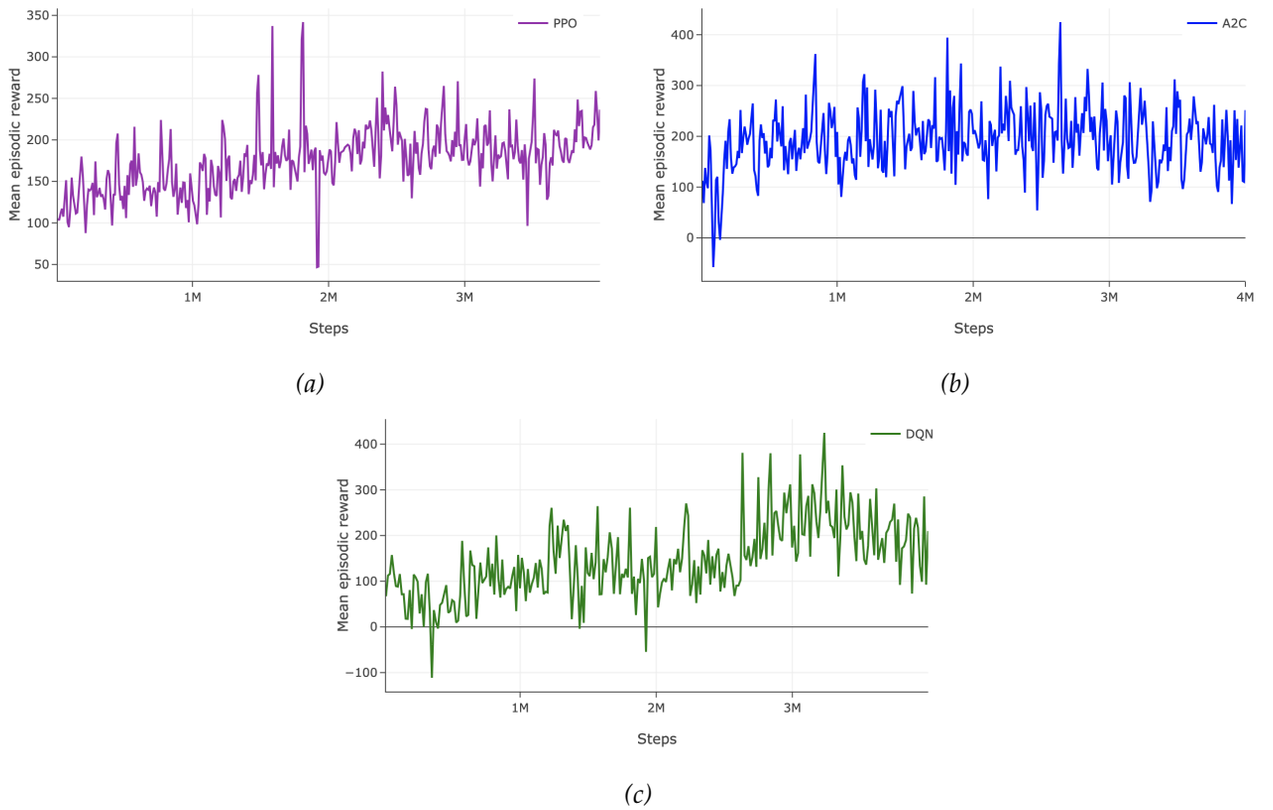


Figure 14: The three plots show the mean episodic reward of the training of the three algorithms: PPO (a), A2C (b) and DQN (c).

mary motivation behind this was to provide insights into how well the agents performed in the environment they were specifically trained for. For clarity in this and the next sections, a naming convention will be adopted for the two test procedures. The test procedure that used all the possible combinations of the environment's parameters will be referred to as "All_comb", while the procedure with the fixed environment will be referred to as "Fixed_env".

Examining the results of "All_comb" (Table 5 and Fig. 15), it is evident how the RL agents outperform the baseline models on two critical metrics: Final PnL and Max PnL (PPO and DQN on the Final PnL and all three on the Max PnL). In particular, the two metrics, especially the Final PnL, which holds significant weight in assessing the effectiveness of a trading model, show that the RL agents have successfully managed to build positive profits. However, it is worth noting that the results show also that the RL agents failed to maintain low risk and volatility, compared to the baseline models.

On the other hand, the results of "Fixed_env" (Table 6 and Fig. 16) show that in the specific environment they have been trained in, the RL agents outperform the baseline models on all metrics, with the exception of the Min PnL and Mean Inventory Stake. However, noteworthy to mention that the PPO agent's Min PnL is marginally close to the best score (-0.55 against -0.54). It is also crucial to remember that within the context of this study and of sports trading, in contrast to other market making contexts, possessing a flat inventory is

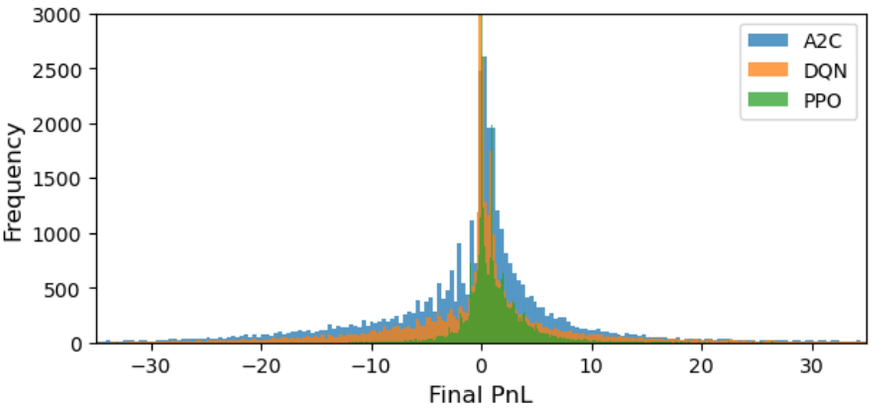


Figure 15: The plot shows the distributions of the Final PnL metric for the three RL agents in the "All_comb" test.

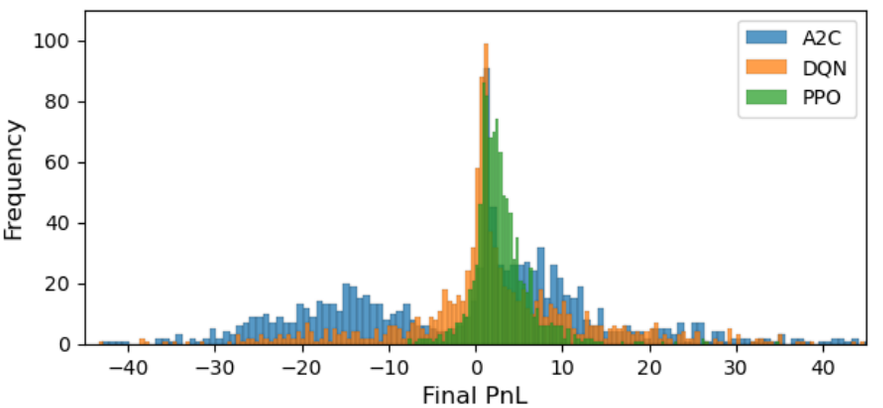


Figure 16: The plot shows the distributions of the Final PnL metric for the three RL agents in the "Fixed_env" test.

not desirable, because it cannot lead to a profit. Hence, while it represents a good indicator of the risk undertaken during the trading period, it is not desirable to aim for a zero Mean Inventory Stake. In fact, the negative values of the PPO agent (-1.31 in "All_comb" and -2.79 in "Fixed_env") show evidence that the agent successfully learned to aim for an inventory with both stake and odds negative, which represents the state where the agent locked in a profit regardless of the price fluctuations or the event outcome.

Moreover, between the RL algorithms, the one trained using PPO stands out as the best performer. In fact, both in "All_comb" and "Fixed_env", PPO has the highest Final PnL and Sharpe Ratio, as well as the lowest Volatility.

5.5.1 Correlations of actions and state variables

To better understand the behaviours of the agents, a correlation analysis was performed between their actions (quoted back and lay offsets, and spread) and state variables (volatility indicator, momentum indicator, price, inventory stake and inventory odds). To calculate the correlation, the Pearson coefficient was used. Fig 17 shows the results.

Interestingly, the correlation matrix of A2C shows the same values for all the state variables. Further analysis revealed that this is caused by the fact that the agent learned to quote a fixed spread of 0.6 with a back offset of 0.1 and a lay offset of 0.5 during the entire trading period. This approach worked well in the "Fixed_env" environment, achieving top scores in metrics like the Mean Return, Sortino ratio, and Max PnL. However, in the broader "All_comb" test, A2C's performance was scarce on different metrics, resulting in being the only model with a negative mean Final PnL among all the tested models, including the baselines.

The DQN agent, on the other hand, revealed some weak positive correlations. Specifically, between the spread and the inventory odds, lay offset and inventory odds, and the lay offset and volatility indicator. Additionally, it also presents a negative correlation between the price and the back offset, suggesting that when the price drops, the agent aims to increase the probability of getting the back order filled (and vice-versa), which is in line with the logic that by holding a back position, a decreasing price represents a profit.

Moreover, PPO exhibited the same correlations of DQN but with additional ones. Correlations were found between the momentum indicator and both the lay offset (positive correlation) and the back offset (negative correlation). This correlation follows the same logic as the correlation between price and back offset (found both in PPO and DQN). However, it is important to highlight how, while the DQN agent has a correlation only between price and the back offset, the PPO agent presents, in addition to that correlation, the correlation between both the back and lay offset with the momentum indicator. This is evidence that the PPO agent better understood the connection between increasing or decreasing prices and profit mechanisms of backing and laying. Additionally, another interesting correlation in the PPO agent is the one between volatility and spread, suggesting that the agent might be increasing the spread to mitigate potential risks during volatile periods.

Finally, it is important to highlight that while these correlations provide valuable insights into the decision-making of the agents, they are relatively weak, indicating that these state variables influence the actions but the overall logic that the agents follow is more complex and cannot be explained only by the correlation between two variables. In fact, while the Pearson coefficient measures linear relationships, the relationships between the state variables and actions, especially given the non-linear nature of the agents' architectures, are probably more complex and non-linear.

5.6 Discussion

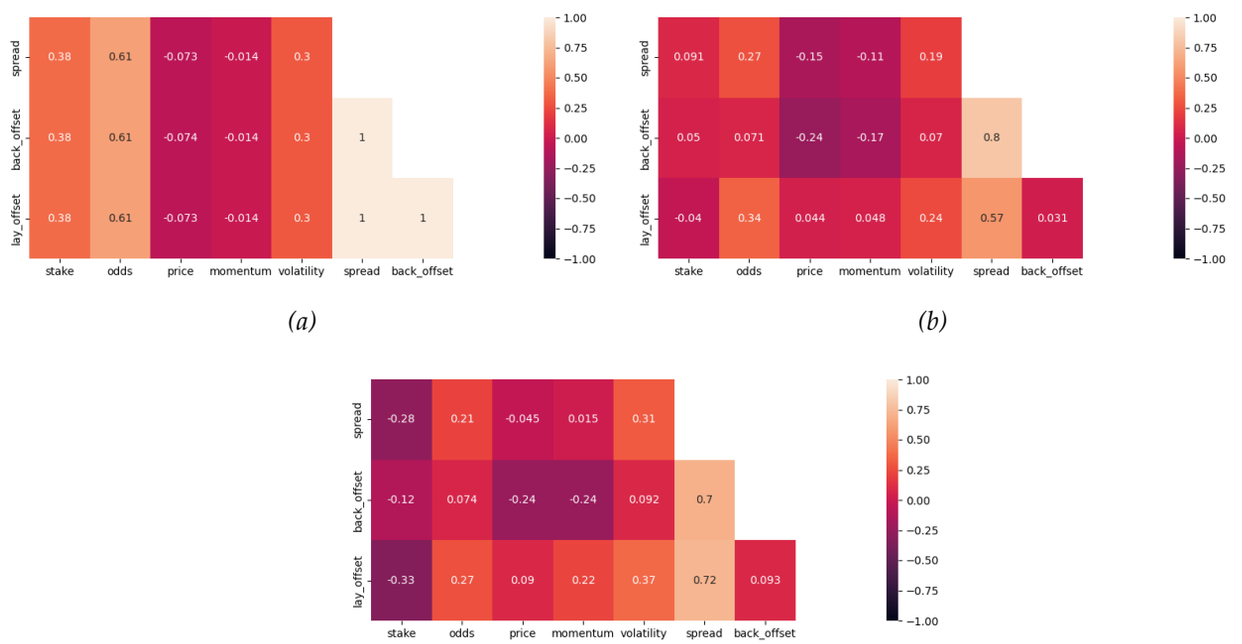
In this final experiment, the focus was on the development, training, and testing of a novel RL agent. One of the primary objectives was to surpass the performance of the baseline models established in the previous chapter. Overall, the experiment can be considered successful, as the RL agents managed to outperform the Final PnL of the baseline models in both "Fixed_env" and "All_comb" test environments. It is critical to highlight that while other performance metrics are very important, the Final PnL is of paramount importance as it served as the reward function during the training phase. Hence, outperforming the baselines in this metric is indicative of a successful training process.

Model	Final PnL	Mean Return	Volat.	Min PnL	Max PnL	Sharpe ratio	Sortino ratio	Mean Inv. Stake
Fixed_0.2	0.91	-0.06	10.12	-1.39	2.63	0.05	1.17	0.02
Fixed_0.5	0.65	0.005	3.38	-0.55	1.43	0.03	0.43	-0.001
Fixed_0.8	0.31	6e-4	1.63	-0.25	0.68	0.01	0.23	0.002
Random	0.66	-6e-8	1e+10	-1.37	2.26	0.02	1e+9	-0.005
DQN	**1.13**	-1e+8	2e+9	-3.72	**6.92**	0.03	6e+7	0.86
A2C	-0.002	-0.07	18.63	-5.02	**6.07**	0.03	0.67	4.92
PPO	**1.28**	-0.25	10.43	-0.78	**2.69**	0.04	0.75	-1.31

Table 5: The table shows the mean value for all the metrics calculated on the "All_comb" test.

Model	Final PnL	Mean Return	Volat.	Min PnL	Max PnL	Sharpe ratio	Sortino ratio	Mean Inv. Stake
Fixed_0.2	2.47	0.04	4.23	-1.68	4.81	0.07	0.48	0.04
Fixed_0.5	1.83	0.10	3.84	-0.97	3.21	0.08	0.61	-0.01
Fixed_0.8	0.80	-0.03	3.76	-0.54	1.61	0.05	0.44	0.002
Random	1.41	-4.70	60.1	-1.58	3.38	0.04	0.55	-0.04
DQN	**2.48**	-0.08	6.39	-4.46	**8.43**	0.05	0.41	-1.49
A2C	0.19	**0.14**	8.38	-9.68	**10.18**	0.03	**1.10**	9.63
PPO	**3.06**	0.03	**2.97**	**-0.55**	4.39	**0.09**	0.55	-2.79

Table 6: The table shows the mean value for all the metrics calculated on the "Fixed_env" test (k=4 and Markov model's probabilities set to 0.65).



(a) *(b)*

(c)

Figure 17: The four images show the correlation matrices between state variables (on the x-axis) and the actions (on the y-axis) of the agents trained using three RL algorithms: A2C (a), DQN (b) and PPO (c).

Moreover, the RL agents' ability to outperform the Final PnL in the "All_comb" testing environment suggests that they have gained a level of adaptability. It shows that the strategies and behaviours learned in the fixed environment were to a certain degree transferable and applicable to a wider set of conditions. However, while the agents were successful with the Final PnL metric in both environments, they failed to outperform the baselines on most of the metrics in the "All_comb" environment. This highlights a significant challenge: the need to train agents that are more robust and agnostic to varying environmental parameters, particularly market liquidity. This remains a key area for future research and development.

The correlation analysis between actions and state variables provided valuable insights into the agents' behaviour, despite not revealing any strong correlations. While it was interesting to see how the A2C agent adopted a fixed spread strategy, the DQN and, more significantly, the PPO agent demonstrated to some degree an understanding of the underlying mechanisms and logic required for profitable trading through backing and laying, as well as the connection with price fluctuations and volatility. This understanding represents an encouraging basis for further refinement and development of these agents.

In conclusion, aside from the previously mentioned limitations of the simulated environment, which does not fully capture the intricacies of a real LOB, there are two other key aspects that differentiate the simulated environment from a real sports trading environment. Firstly, unlike financial markets where fees are imposed on transactions, sports trading exchanges take a commission on positive profits. This would naturally reduce the profits generated by the agents. However, it is crucial to understand that while commissions on trades in financial markets can convert a profitable trading period into a losing one, commissions in sports trading are only on positive profits. Thus, they may scale down the profits but they can not turn them into losses. Still, it is important to include all aspects of real trading environments, hence, future works could include the commissions, by incorporating them directly into the reward function. Secondly, sports trading exchanges introduce an artificial time delay on the orders going into the LOB. The reason is to prevent mechanisms like "court-side" betting, where bettors who are physically close to the event, have a time advantage. The time delay poses an additional challenge for the market making agents because, in practice, this means the agent cannot immediately act on observed state variations but instead must anticipate the state of the market in the next seconds. Hence, in future works, to improve the applicability to real trading scenarios, it is important to incorporate these delays into the simulated environment.

Chapter 6

In this final chapter, the study culminates by summarizing the objectives, key findings, insights, and contributions made throughout the research, as well as the main challenges encountered. Furthermore, the chapter delineates potential avenues for future research, emphasizing the exploration of novel methodologies and advanced modelling techniques to enhance the robustness and applicability of the trained agents in real-world trading environments.

6 Conclusions and Future Work

6.1 Summary

This research was anchored around three core objectives in the realm of algorithmic sports trading and market making:

- Obtain a deep understanding of the key characteristics and unique dynamics of the sports trading market, with a specific focus on analyzing the key features of exchange data.
- Setting a benchmark in market making in the sports trading market by evaluating baseline models used in other works.
- Designing, training and testing cutting-edge market making agents using ML and RL methodologies.

To achieve the outlined objectives, three pivotal experiments were conducted, each providing its own set of insights:

- In-depth analysis of sports exchange data (Chapter 3). In this experiment, special attention was given to crucial features like trading volumes, liquidity, and market volatility, especially during the in-play trading period. The aim was to understand the unique characteristics that differentiate the sports trading market from traditional financial markets.
- Implementation and Testing of baseline models (Chapter 4). The purpose of this experiment was to establish a performance benchmark for market-making in the sports trading domain. In this context, a simulated trading environment was created, which would later serve as the training and testing ground for the RL agents.
- Development, Training and Testing of a novel RL agent (Chapter 5). This final experiment aimed at the development, training, and testing of a novel RL-based market making agent. This experiment sought to employ state-of-the-art methodologies in RL and ML to design an agent capable of outperforming the established baseline models in the simulated sports trading environment.

Together, these experiments provided a comprehensive approach to address the research objectives.

6.2 Conclusions

In the first experiment, the in-play trading period was highlighted as a particularly notable phase for traders. During this phase, heightened trading activity and volumes create an environment that is generally favourable compared to the pre-event phase. These conditions increase the likelihood of orders getting matched and offer greater potential for profitable trades. However, it is not without its challenges. The pronounced volatility in this period means there is a greater risk, necessitating better risk management strategies to navigate potential pitfalls. Additionally, the analysis of the dataset revealed a lack of strong correlations between its various features. This independence in the data indicates that each feature captures a unique aspect of the market dynamics, an insight that is important for formulating comprehensive market making strategies.

Moving on to the second experiment, four baseline models were evaluated using several performance and risk metrics. The results, as anticipated, showed scarce performances from all four models. This was largely attributed to the simplicity of their design and the lack of ability to consider or utilize any state variables or understand the environment they operate in.

In the third and final experiment, the focus was on the development of cutting-edge market making agents using RL methodologies. Three RL algorithms (DQN, A2C, and PPO) were used to train the agents and test them against established baseline models. The results turned out to be promising, in fact, two of three RL agents (DQN and PPO) outperformed the baseline models in terms of Final PnL in both the testing procedure that included all possible combinations of the environment's parameters ("All_comb") and the one with the specific environment's parameters they were tested in ("Fixed_env"). Particularly, the PPO model emerged as the best performing agent in both testing scenarios. Moreover, the RL agents demonstrated, to a certain degree, adaptability and flexibility, as evidenced by their performance in the "All_comb" configuration. This adaptability was most apparent in their ability to perform well in environments different from the one they were trained in. Additionally, a correlation analysis of the actions and state variables revealed that the RL agents, particularly DQN and PPO, had some understanding of the intricacies of sports trading and market making, such as the connection between price fluctuations, volatility and profitable trading actions.

Within the realm of this research, several aspects underscore its novelty and distinct approach. To begin with, based on the author's best knowledge, there is an absence of existing literature regarding the use of RL for market making in the sports betting arena. This research, therefore, ventures into this domain, potentially laying the groundwork for future investigations. Further accentuating the novelty of this study is the innovative adaptation of the Avellaneda-Stoikov (AS) framework (Section 4.1.2). By replacing the conventional Brownian motion model with a tennis Markov model and modifying the PnL process, a modified version of the framework was established, introducing a novel framework for simulating a sports trading environment with a LOB.

In terms of limitations encountered during the project, the predominant one was the lack of granular data on order arrivals. The absence of this data restricted the adoption of more

sophisticated LOB simulation models. Consequently, this limitation imposed constraints on achieving higher accuracy in simulating the trading environment. Moreover, it denied the RL agents from accessing valuable LOB features, which could have potentially enhanced its performance and predictive capabilities.

6.3 Future works

A primary consideration for further research is the acquisition of more diverse and granular data. Possessing a comprehensive dataset, enriched with high-granular information on order arrivals, can facilitate the calibration and training of more sophisticated models that simulate directly the LOB. Such data would not only refine the modelling and simulation processes but might also present an opportunity to train RL agents directly on real historical data.

In general, aiming at a more accurate simulation of the market and its LOB should be central in future works. A more accurate and realistic simulation is fundamental to obtaining results that are transferable to real-world market scenarios. Moreover, with a LOB model, extracting essential features becomes feasible and incorporating them into the training process can provide agents with valuable insights, guiding their decision-making processes. In alternative, future works could continue to work using the introduced variation of the AS framework, without simulating directly the LOB, but utilizing a more complex and accurate model to simulate the mid-price. In particular, while the Markov model has served as the bedrock for simulating the mid-price, as discussed in Section 4.1.1, it has its inherent limitations. Advanced models could provide a more granular simulation of the mid-price, offering insights into market efficiency and the nuances of price movements. Also, as mentioned in Section 5.6, additional aspects of a real sports trading environment, like commission and imposed delays, are not included in the simulation. Hence, future works could work towards incorporating these aspects, to further improve the applicability to real trading scenarios.

Furthermore, this study has predominantly operated within a discrete setting. In particular, the assumption was that the mid-price updated only at points. This approach, while providing a structured framework, does not entirely encapsulate the fluid nature of real-world trading where price fluctuations occur continuously, both in intra-point and inter-point phases. Delving into a more continuous setting could offer a more realistic environment.

Additionally, a notable enhancement in the RL agent's performance could be obtained by integrating a predictive component. By employing a sport-specific model that forecasts in real-time the outcome of the match, hence forecasting the price movements, the agent could be better positioned to strategize its market making decisions. Such a predictive edge could potentially enable the agent to navigate the market with increased precision, maximizing its profitability while minimizing risks.

Moreover, due to time constraints, this study focused on utilizing a reward function solely based on PnL. While this approach provided a good representation of the final goal of making a profit, it is essential to consider that more nuanced reward functions could yield better

results. Future research could explore the development of more sophisticated reward functions that account for additional desirable aspects of a market making strategy, like liquidity provision and risk management.

On the methodological front, a future research direction could explore the potential of Distributional RL [42]. The intrinsic risk-based nature of this type of RL makes it a compelling approach for market making and, in general, trading scenarios. Given that financial markets are often influenced by both expected outcomes and the variability of these outcomes, Distributional RL's focus on the entire distribution could offer a more holistic perspective. In particular, this perspective can provide agents with a richer set of information to make informed decisions and manage risks effectively.

Moreover, the exploration of Representation Learning approaches represents a promising direction. Most contemporary studies in market making have traditionally been anchored to the use of handcrafted features, which, while effective, come with inherent limitations. Even if meticulously designed, they often encompass biases of the designer and might miss out on intricate patterns present in the data. Representation Learning, on the other hand, provides an automated method for feature extraction, allowing models to discern and capture pivotal information from the raw data autonomously.

Finally, expanding upon the scope of this research, attention could be directed towards other sports beyond tennis. Investigating the dynamics and intricacies of various sports would not only enrich the research perspective but also pave the way for the development of sport-agnostic agents. Such agents would be equipped with the capability to operate across a multitude of sports, demonstrating versatility and robustness in their trading strategies.

References

[1] A. Koshiyama, N. Firoozye, and P. Treleaven, "Algorithms in future capital markets," *Available at SSRN 3527511*, 2020.

[2] "Algorithmic trading," https://www.handbook.fca.org.uk/handbook/glossary/G3552a.html, accessed: 2023-05-17.

[3] W. De Vena, "Machine learning in algorithmic trading - emerging topics in integrated machine learning systems," 2023.

[4] M. Avellaneda and S. Stoikov, "High-frequency trading in a limit order book," *Quantitative Finance*, vol. 8, no. 3, pp. 217–224, 2008.

[5] O. Guéant, C.-A. Lehalle, and J. F. Tapia, "Dealing with the inventory risk," 2011.

[6] Á. Cartea, S. Jaimungal, and J. Ricci, "Buy low, sell high: A high frequency trading perspective," *SIAM Journal on Financial Mathematics*, vol. 5, no. 1, pp. 415–444, 2014.

[7] T. Beysolow II and T. Beysolow II, "Market making via reinforcement learning," *Applied Reinforcement Learning with Python: With OpenAI Gym, Tensorflow, and Keras*, pp. 77–94, 2019.

[8] R. Casadesus-Masanell and N. Campbell, "Platform competition: Betfair and the uk market for sports betting," *Journal of Economics & Management Strategy*, vol. 28, no. 1, pp. 29–40, 2019.

[9] K. Croxson, J. J. Reade *et al.*, *Exchange vs. dealers: a high-frequency analysis of in-play betting prices*. Department of Economics, University of Birmingham Birmingham, 2011.

[10] O. Hubáček, G. Šourek, and F. Železnỳ, "Exploiting sports-betting market using machine learning," *International Journal of Forecasting*, vol. 35, no. 2, pp. 783–796, 2019.

[11] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[12] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.

[13] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[14] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[15] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[16] J. Fernandez-Tapia, "High-frequency trading meets reinforcement learning: Exploiting the iterative nature of trading algorithms," *Available at SSRN 2594477*, 2015.

[17] Y. Patel, "Optimizing market making using multi-agent reinforcement learning," *arXiv preprint arXiv:1812.10252*, 2018.

[18] J. Sadighian, "Deep reinforcement learning in cryptocurrency market making," *arXiv preprint arXiv:1911.08647*, 2019.

[19] B. Gašperov and Z. Kostanjčar, "Market making with signals through deep reinforcement learning," *IEEE Access*, vol. 9, pp. 61 611–61 622, 2021.

[20] R. K. Narang, *Inside the black box: A simple guide to quantitative and high frequency trading*. John Wiley & Sons, 2013, vol. 846.

[21] W. De Vena, "Reinforcement learning for market making in algorithmic sports trading - interim report," 2023.

[22] M. B. Garman, "Market microstructure," *Journal of financial Economics*, vol. 3, no. 3, pp. 257–275, 1976.

[23] H. R. Stoll, "The supply of dealer services in securities markets," *The Journal of Finance*, vol. 33, no. 4, pp. 1133–1151, 1978.

[24] T. Ho and H. R. Stoll, "Optimal dealer pricing under transactions and return uncertainty," *Journal of Financial economics*, vol. 9, no. 1, pp. 47–73, 1981.

[25] M. Potters and J.-P. Bouchaud, "More statistical properties of order books and price impact," *Physica A: Statistical Mechanics and its Applications*, vol. 324, no. 1-2, pp. 133–140, 2003.

[26] O. Guéant, C.-A. Lehalle, and J. Fernandez-Tapia, "Dealing with the inventory risk: a solution to the market making problem," *Mathematics and financial economics*, vol. 7, pp. 477–507, 2013.

[27] O. Guéant, "Optimal market making," *Applied Mathematical Finance*, vol. 24, no. 2, pp. 112–154, 2017.

[28] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[29] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[30] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[31] M. Zhao and V. Linetsky, "High frequency automated market making algorithms with adverse selection risk control via reinforcement learning," in *Proceedings of the Second ACM International Conference on AI in Finance*, 2021, pp. 1–9.

[32] N. T. Chan and C. Shelton, "An electronic market-maker," 2001.

[33] J. Abernethy and S. Kale, "Adaptive market making via online learning," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[34] C. Gray, "Game, Set and Stats," *Significance*, vol. 12, no. 1, pp. 28–31, 02 2015. [Online]. Available: https://doi.org/10.1111/j.1740-9713.2015.00799.x

[35] A. Capponi and C.-A. Lehalle, *Machine Learning and Data Sciences for Financial Markets: A Guide to Contemporary Practices*. Cambridge University Press, 2023.

[36] B. Gašperov, S. Begušić, P. Posedel Šimović, and Z. Kostanjčar, "Reinforcement learning approaches to optimal market making," *Mathematics*, vol. 9, no. 21, p. 2689, 2021.

[37] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[38] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[40] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[42] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International conference on machine learning*. PMLR, 2017, pp. 449–458.