QATAR UNIVERSITY

COLLEGE OF ENGINEERING

STRONG MARKET-MAKING USING DEEP REINFORCEMENT LEARNING: BITCOIN MARKET ANALYSIS

BY

ABDULHADY YOUNES FEZOONI

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Masters of Science in Computing

June 2024

.

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Abdulhady Fezooni defended on 01/05/2024.

Dr. Noora Fetais
Thesis/Dissertation Supervisor

Dr. Mohamed Alshraideh
Committee Member

Dr. Sumaya Al-Maadeed
Committee Member

Dr. Ridha Hamila
Committee Member

Approved:

Dr. Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

FEZOONI, ABDULHADY YOUNES, Masters:

June:2024, Masters of Science in Computing

Title: STRONG MARKET-MAKING USING DEEP REINFORCEMENT LEARNING: BITCOIN MARKET ANALYSIS

Supervisor of: Dr. Noora Fetais.

This study evaluates the use of deep reinforcement learning (DRL) in market-making, specifically in the Bitcoin market. DRL has shown promise in providing robust market-making capabilities, including enhanced market liquidity and risk management, which may lead to more efficient price discovery and lower volatility. The study also discusses the historical perspective of market-making techniques and explains how agents can use DL algorithms and RL principles to improve preset objectives in financial markets. It also reviews essential DRL algorithms like Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Q-Network (DQN) and their specialized applications and the possible effect of DRL-based market-making on market dynamics. This project uses data science and machine learning to study Bitcoin market data. It's important for financial market analysis, especially in volatile and speculative cryptocurrency markets. The models' effectiveness is evaluated with spread capture ratio, market impact, and profitability. The findings can help academics and financial institutions understand how DRL can improve market efficiency and stability.

DEDICATION

"To my family and friends, for their support and faith in me."

ACKNOWLEDGMENTS

" Thank you to Dr. Noora Fetais, who has been working with me diligently on my thesis, for their advice, knowledge, and patience. I've grown as a researcher and scholar thanks to your mentoring. "

" I would also like to thank the Graduate Programs Coordinator of the Computer Science and Engineering Department Prof. Khaled Shaban, Head of the Computer Science and Engineering Department Prof. Amr Mohamed, and Professor & Associate Dean for Research and Graduate Studies Prof. Ahmed Massoud, for their effort and support "

" I would like to express my appreciation to Dr. Rateb Jabbar, for his effort, time, and support given to me during the journey."

"Finally, I would like to thank Qatar University, for the support and for providing all the needs to achieve the requirements of this study"

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS:

AI	Artificial Intelligence
ML	Machine Learning
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
MM	Market-Making
HFT	High-Frequency Trading
DRW	Don Wilson
FinTech	Financial Technology
DQN	Deep Q-Network
PPO	Proximal Policy Optimization
TDQN	Trading Deep Q-Network
TD3	Twin Delayed Deep Deterministic Policy Gradient
D3QN	Dueling Double Deep Q-Network
DRQN	Deep Recurrent Q-Networks
DDPG	Deep Deterministic Policy Gradient
DRP	Dynamic Risk Pricing
A3C	Asynchronous Advantage Actor-Critic
LOB	Limit Order Book
PM	Portfolio Management
OE	Order Execution
LO	Limit Order
FIFO	First-In-First-Out
CNNs	Convolutional Neural Networks
RNNs	Recurrent Neural Networks

GAN	Generative Adversarial Networks
SRB	Simple Rule-Based
PNL	Profit and Loss
MASs	Multi-Agent Systems
PMM	Predictive Market-Making
CPE	Consolidated Price Equation
MADDPG	Multi-Agent Deep Deterministic Policy Gradient
LSTM	Long Short-Term Memory
SDAEs	Stacked Denoising Autoencoders
SARSA	State-Action-Reward-State-Action
MAP	Mean Absolute Positions
FFNN	Feed-Forward Neural Network
DRLMM	Deep Reinforcement Learning as Applied to Market-Making

CHAPTER 1: INTRODUCTION

Overview

Citadel Securities, one of the largest players in equity market making, announced a net trading revenue exceeding $6 billion in the year 2020 [1]. Similarly, Virtu Financial [2], another prominent market maker, declared an approximate trading income of nearly $3.5 billion. The overall yearly profits attributed to the high-frequency trading sector for US equity are believed to range from $5 billion to $10 billion. It's important to note the close relationship between market-making and high-frequency trading. An illustration of this connection is Flow Traders, a proprietary trading company that serves as a market maker, enhancing liquidity within the securities market through the application of high-frequency and quantitative trading technique. The financial results for Flow Traders in 2020 reported revenue totaling €933.4 million [3]. In the Chicago Mercantile Exchange landscape, DRW (Don Wilson) functions as a diversified firm involved in trading a variety of financial instruments such as fixed income, options, derivatives, energy, agriculture, and cryptocurrency. In the financial year 2021, the DRW Trading Group posted an annual revenue of €412 million [1].

One of the most significant scientific advancements of the twenty-first century, is the emergence and rapid evolution of artificial intelligence (AI), a transformative field that has revolutionized various domains across academia, industry, and society at large. By 2022, AI usage among businesses worldwide had increased by 4% points from the previous year, with 35% of businesses incorporating AI into their operations, forecasts predict that the global artificial intelligence market will soar to $1.59 trillion by 2030, notably, 91% of prominent companies consistently invest in AI, underlining the strategic importance they place on AI technologies,

according to a survey, 61% of employees believe that AI contributes positively to their work productivity, enhancing efficiency and outcomes. Surprisingly, 62% of consumers express their willingness to share data with AI systems to enhance their interactions and overall experiences with businesses, it was projected that around 15% of all customer service interactions globally in 2021 would be fully powered by AI, indicating the increasing integration of AI in customer service processes, currently, almost 1 in 4 sales teams rely on artificial intelligence in their day-to-day operations, embracing AI's potential to drive sales effectiveness, over half of the organizations, approximately 54%, have reported tangible cost savings and efficiencies as a direct outcome of implementing AI solutions, the number of AI-powered voice assistants is projected to experience a remarkable 146% surge from 3.25 billion in 2019 to 8 billion by 2023, and a significant majority of businesses, more than 3 in 4, consider trust in AI's analysis, results, and recommendations to be of utmost importance in their decision-making processes [4] [5]. The implications of AI are already being felt in many different areas, and it has the potential to revolutionize many different sectors. Businesses now need novel approaches to maintain a competitive edge, and AI has emerged as a powerful tool to help them do just that. It is possible to see the effects of AI in many other fields, such as business, teaching, medicine, retail, and transportation [6]. AI has the potential to boost output, cut expenses, improve precision, and personalize services for customers [7], [8]. It can also provide valuable insights into data that individuals may have trouble recognizing on their own. "One of the most significant things that humanity is working on is AI," remarked by Google CEO Sundar Pichai. Its significance is beyond that of fire, electricity, and all our other great technological advances. There are many benefits to it, but there are also some major negatives that you should be aware of [9]. AI has advanced significantly in the

past few years, particularly in a branch of machine learning called deep learning. Some people are worried about the social and ethical implications of these innovations as well as their potential applications. These concerns are shared by the general public, scientists, science policy analysts, and those who study artificial intelligence [10] [11].

To better evaluate, manage, invest, and secure financial resources, financial institutions are increasingly turning to artificial intelligence (AI) tools which aim to emulate human intelligence and decision-making [12]. Interest in the application of AI to the financial sector has been on the rise for several years [13]. Conventional AI in the finance sector has historically served functions like financial markets, trading, banking, insurance, risk management, regulation, and marketing. In contrast, the emergence of FinTech (financial technology) represents a newer generation that empowers activities such as digital currency management, lending, payments, asset and wealth management, risk and regulatory compliance, as well as accounting and auditing [14].

Technological advancements in the financial markets have traditionally been centered on improving trading procedures and overall market efficiency [15]. Market-making is one of the most essential functions in the global financial markets. This function requires providing consistent buy and sell quotes for a group of assets in order to keep the market liquid and to ensure accurate price discovery [16]. Classic market-making strategies often depend on heuristic principles and statistical models; however, recent advancements in AI and ML have sparked a large interest in the research of more complicated and adaptive ways to classic market-making tactics. One of these prospective avenues is the application of DRL methods in market-making. DRL is a subfield of ML that is created when the powerful class of

algorithms known as deep learning is paired with the paradigm of reinforcement learning, which is oriented on learning through interactions with one's environment in order to maximize the achievement of a preset goal. DRL enables market makers to potentially learn sophisticated patterns from historical market data, modify their strategies in real-time in reaction to shifting market conditions, and so on. This allows market makers to improve their capacity to supply continuous liquidity while efficiently managing risk. This capability can be strengthened by adopting DRL [17].

Market-Making (MM) tactics increase market activity, order stability, and liquidity in the ever-changing stock market. Buy-side high-frequency trading strategies capitalize on market spreads, which are the disparities between the best bid and ask prices. These spreads pose difficulties in devising a successful Market Making strategy [18][19]. Human professionals used their experience to design mechanical MM techniques. These rules were useful yet limited. First, they didn't understand the market's dynamics and strategy states. Second, they struggled to grasp these states' complex linkages and best trading behaviors [18]. Meanwhile, deep reinforcement learning allows agents to learn and adapt without rules [20]. Imagine an MM agent quoting bid and ask prices simultaneously, ready to exploit chances. This AI-driven agent masters decision-making using deep reinforcement learning. It can understand strategy states and make the most successful trades by learning from its experiences. Alpha Star shows incredible potential. This astonishing agent outperformed 99.8% of human gamers utilizing deep reinforcement learning, as the AI era dawns, its popularity soars (rise), and it already accounts for more than 70% of trade volumes in major countries like the United States and more than 40% in developing markets like China [21]. AI could also transform market-making. Deep reinforcement learning can make MM strategies efficient, adaptive, and profitable.

AI-powered MM agents could revolutionize market-making by embracing numbers and market conditions [9].

The potential of DRL to disrupt conventional market-making techniques has led to widespread interest in the field. DRL is a technique that combines DL and RL to help market makers learn complicated patterns from past market data, adjust strategies to new circumstances, and maintain good risk management all while providing continuous liquidity [17]. Due to its effectiveness in learning difficult games and other applications, DRL has garnered substantial adoption within the financial sector. DRL has emerged as a promising methodology for tackling intricate decision-making issues across various domains, with market-making being a prominent area that has attracted significant interest. Market-Making, the continuous buying and selling of financial instruments to provide liquidity in financial markets requires quick and adaptive decision-making to capitalize on market fluctuations and ensure efficient trading. Traditional market-making strategies rely on rule-based heuristics and mathematical models, which may struggle to capture the dynamic and intricate nature of financial markets. DRL, alternatively, offers a data-driven and adaptive approach that can learn optimal trading strategies directly from market data. The scientific community and the private sector have recently shown a great deal of interest in deep reinforcement learning. Alpha Star, which was taught to play the game using a deep reinforcement learning algorithm, now consistently achieves a 99.8%-win rate against human opponents [22]. To optimize difficult decision-making tasks from start to finish, a Deep Q-network (DQN) combines deep neural networks with RL to extract characteristics from data [23]. However, the majority of existing stock market-related deep reinforcement learning challenges are geared toward long- or short-term stock trading. Deng et al. employed a direct deep reinforcement learning

method to train an agent for trading financial assets, aiming to model real-time financial data.[24].

In recent years, DRL's use of neural networks and reinforcement learning algorithms to tackle the difficulties of market-making has shown considerable promise. DRL enables market-making agents to learn and adapt to changing market conditions by integrating deep neural networks, which are capable of managing complicated data patterns, with reinforcement learning techniques, which maximize decision-making based on rewards and punishments. Agents using dynamic risk pricing (DRP) in the market make judgments about when to place bids and offers, how much of a spread to use, and when to execute trades based on their analysis of market data such as order book information, historical price movements, and market depth [25]. These agents learn through trial and error how to maximize profits, decrease transaction costs, and control risk.

While DRL shows potential in market-making, it also comes with implementation issues that must be taken into account. There is a risk of overfitting historical data, which necessitates huge volumes of high-quality training data and careful construction of reward functions. When implementing DRL-based market-making methods, it is also important to take into account relevant market regulations and compliance prerequisites [26]. There are, however, complications associated with using DRL for market-making. It takes a lot of computing power, a lot of data, and a well-thought-out reward system to train a DRL agent. Additional difficulties that must be overcome include overfitting, model instability, and the requirement for ongoing learning in a rapidly changing market context. There are a number of potential benefits of implementing DRL into market-making. To begin, it can recognize complex, non-linear relationships in the market data, allowing agents to make smarter,

more flexible choices. Second, DRL agents can respond to the ever-evolving nature of the market by learning from past events and adjusting their approach accordingly. Third, by providing tighter spreads and deeper order books, market-making based on DRLs has the potential to increase liquidity provision and enhance market efficiency [27].

Principles of Market Design

It is necessary to go into the fundamental notions of market-making in order to lay the foundations for comprehending DRL-based market-making. The foundation for contemporary market-making tactics was established by earlier methodologies like the bid-ask spread and dealer-based models. Algorithmic trading and high-frequency trading methods emerged as financial markets developed, improving liquidity provision. This analysis of existing market-making techniques reveals the drawbacks of current approaches and paves the path for the implementation of more flexible DRL-based strategies.

Market Decisions Using DRL

In this thesis, we examine the many approaches that have been offered for using DRL in market-making. Modeling order books, managing stock, and automating trades are just some of the many facets of DRL-based market-making that have been investigated by academics. Case studies and empirical experiments are reviewed to show the potential gains in market-making efficiency and risk management that can be attained by employing DRL across a variety of market circumstances and asset classes.

The Limitations and Challenges

Despite the potential benefits, the widespread adoption of DRL-based market-making faces several obstacles [28]. One of the primary concerns highlighted by researchers pertains to the issue of substandard data quality, inadequately elucidated models, protracted training durations, and extensive processing demands. This thesis provides a comprehensive analysis of the challenges encountered in the field, along with potential strategies and advancements aimed at mitigating these obstacles. These include the utilization of sample-efficient algorithms and the implementation of transfer learning techniques. Below are the limitations and challenges faced during the implementation:

Limitations:

Model Overfitting: The disparity between training and testing performance, particularly in terms of profitability, could indicate overfitting. This means the models may have been too closely tailored to the training data, reducing their effectiveness on new data.

Market Dynamics: The cryptocurrency market is known for its high volatility and unpredictability. These characteristics can make it challenging for models to consistently predict market movements and apply profitable trading strategies.

Data Quality and Availability: The performance of machine learning models heavily relies on the quality and comprehensiveness of the data they are trained on. Limited or biased data can lead to underperforming models.

Challenges:

Adapting to Market Changes: Cryptocurrency markets can change rapidly. A significant challenge is developing a model that not only performs well on historical data but can also adapt and respond to new market conditions effectively.

Risk Management: While profitability is important, effectively managing risk is crucial in trading. The model must balance the pursuit of profit with the management of potential losses, especially in volatile markets.

Regulatory and Ethical Considerations: Cryptocurrency markets are subject to evolving regulatory landscapes. Ensuring compliance and ethical trading practices is both a challenge and a necessity.

The Motivation of Deep Reinforcement Learning for Market-Making

DRL has gained attention in the field of market-making due to its ability to handle complex and dynamic environments, optimize trading strategies, and adapt to changing market conditions. Market-Making entails the provision of liquidity to financial markets through the continuous quoting of bid and ask prices for a specific security. DRL can enhance market-making strategies by leveraging its ability to learn from data and adjust to evolving market circumstances. DRL has the capacity to discover complex patterns from past market data and optimize market-making tactics by combining deep learning, which can handle massive volumes of data, with reinforcement learning, which learns through interactions with an environment. Core DRL algorithms are explained at length to demonstrate their specialized applicability in financial markets. This includes DQNs, PPO, and A2C. Indeed, the use of DRL in market-making can offer several potential advantages. Here are some reasons why DRL is used for market making:

Complex Decision-Making: Market-Making involves making numerous trading decisions in a fast-paced and highly competitive environment. DRL models have the capability to acquire intricate patterns and associations from both historical and current market data [29]. They can process vast amounts of information and make

decisions based on multiple variables, including price movements, order flow, and market microstructure.

Adaptability: Financial markets are constantly evolving, and market conditions can change rapidly. DRL models excel at adaptability and can adjust their strategies to changing market dynamics [30]. They can learn from experience and update their decision-making process accordingly. This adaptability allows market makers to respond quickly to market shifts and optimize their trading strategies in real-time.

Optimization and Performance: DRL models can optimize trading strategies by learning from past experiences and exploring different actions to maximize rewards or minimize costs. These models can continually refine their decision-making process [30], improving their performance over time [31]. By leveraging DRL, market makers can strive for more efficient and profitable market-making operations.

Handling Uncertainty: Financial markets are inherently uncertain, and market makers need to manage risk effectively. DRL models can incorporate risk management techniques by considering uncertainty and potential downside risks. They can balance risk and reward trade-offs and adjust their trading strategies to minimize losses or exposure to adverse market conditions [32].

Nonlinear Relationships: Financial markets often exhibit nonlinear relationships [29], where the impact of one variable on market dynamics can be influenced by various factors. DRL models can capture these nonlinear relationships and make more accurate predictions and decisions compared to traditional linear models. This enables market makers to better understand and respond to market dynamics.

Continuous Learning: DRL models have the capability to learn continuously

from new data and adapt their strategies accordingly [33]. They can learn from both historical data and real-time market information, allowing them to incorporate the most up-to-date insights into their decision-making process. This continuous learning capability helps market makers stay competitive and adapt to changing market conditions over time.

Improved Efficiency: DRL-based market-making systems can automate various aspects of the trading process, leading to improved efficiency and reduced operational costs [33]. By automating the decision-making process, DRL can enable market-makers to quote prices more quickly and accurately, thereby improving liquidity provision and reducing bid-ask spreads.

Handling Complex Market Dynamics: Financial markets can exhibit complex dynamics, including non-linear relationships, high-frequency trading, and varying liquidity conditions [34]. DRL algorithms excel at handling such complexity by capturing patterns and exploiting market inefficiencies that might be difficult for traditional rule-based approaches. By processing substantial data sets and uncovering hidden patterns, these models enable market-makers to enhance their trading decisions with increased information and effectiveness.

Risk Management: DRL algorithms can incorporate risk management techniques and optimize trading strategies to minimize risk exposure. By considering factors such as market volatility, position limits, and risk tolerance [32], DRL-based market-making systems can better manage risk and avoid excessive losses. This can enhance the overall stability and profitability of market-making operations.

Scalability: DRL algorithms have the potential to scale effectively across multiple markets and instruments. Once trained, the algorithms can be deployed across various assets, allowing market-makers to operate in different markets

simultaneously. This scalability can enable market-makers to expand their trading activities and capture opportunities across a broader range of securities [33].

While DRL shows promise for market-making, it's important to note that its implementation requires expertise in AI, extensive computational resources, and careful consideration of risk management practices and regulatory requirements to ensure responsible and compliant trading activities.

Thesis Objective

The aim of this thesis is to explore the emerging topic of DRL in order to generate more precise market forecasts. We will discuss the theoretical and experimental foundations of market-making and DRL, analyze the challenges and potential benefits of applying DRL to market-making, and examine the most current results and activities at the cutting edge of this field. The goal is to develop an intelligent trading system using historical cryptocurrency price data that can make profitable trades while minimizing risk. This involves preprocessing and analyzing the data, creating a suitable reinforcement learning environment, and training different models such as PPO, A2C, and DQN to find the best-performing model. The main objective of the research:

- Develop a market-making framework based on Deep Reinforcement Learning: improve overall market-making profitability.
- Preprocess and analyze the given crypto_data.csv dataset containing Bitcoin price information.
- Train three different models (PPO, A2C, and DQN) with this environment and compare their performance.

- Analyze evaluation metrics including Spread Capture Ratio, Market Impact, and Profitability for both train and test datasets across all trained models.
- Identify the most effective model capable of making profitable trades consistently while considering risk management.

Thesis Overview

This chapter gives an overview of the research and presents the research objectives. In Chapter 2, a background to the main concepts utilized in this research is introduced. In Chapter 3, A deep description of using Deep Reinforcement Learning in Market-Making was presented. Chapter 4 is a review of the related work in using Deep Reinforcement Learning in Market-Making domain. Chapter 5 delves into a comprehensive mathematical explanation, clarifying the underlying principles and theoretical framework that substantiate the thesis's propositions and findings. In Chapter 6, the methodology and its practical applications take center stage, offering a detailed exploration of how the theoretical constructs explained in the previous chapters are implemented and tested in real-world scenarios. Finally, Chapter 7 provides a review of our findings and the work that will be done moving forward to incorporate deep reinforcement learning in market-making.

CHAPTER 2: BACKGROUND

Market-Making

Before going deep into the specifics of this thesis, we provide brief background information on MM in this part. Market makers are high-volume traders that actually "make a market" for assets, constantly prepared to buy or sell at any price to maintain market liquidity. Financial institutions, investment banks, and brokerages are some examples of market makers. Financial markets rely largely on liquidity, and by providing liquidity, market makers ensure that the music never stops. The enhancement of market-making has a huge influence on the whole financial industry. Over the past 20 years, we have steadily moved toward a more computerized financial system. As a result of this transition, computers, which use sophisticated algorithms and provide decisions in a matter of milliseconds, have replaced traditional market makers [35].

The market as we know it today was created with the appearance of market makers. Today, artificial intelligence has assumed the role of the market maker, which, with the aid of mathematical algorithms, enables a seamless flow of closed agreements and offers immediate liquidity. Undoubtedly, automated programs that can handle a million orders at once have revolutionized the trading industry. They have opened up new possibilities for using trading systems and, more importantly, have sparked the creation of new technologies that will improve market liquidity.

Market makers are unique market participants that always stand ready to enter into deals with other market makers, keeping the market dynamic. Market makers may also be described as traders who assume responsibility for maintaining pricing, demand, supply, and/or volume of transactions in financial instruments, foreign currencies, and/or items after such an agreement, one of which is the trade organizer.

There must be a second party participating in the transaction for each participant. Finding a buyer for your shares or currencies is the only thing you need to do in order to sell them. Similarly, if you wish to purchase assets, you must locate a seller. No matter what instrument is exchanged, a market maker is in charge of making sure there is always a buyer or a seller to make sure the transaction goes well. Market makers act as both brokers and dealers, a conflict of interest develops since, as brokers, they are obligated to give customers the greatest execution. As dealers, however, they take on the role of counterparties and engage in profitable trading.

Market makers can be split into two groups [35]:

- The biggest commercial banks are regarded as the first-level market makers. They are also referred to as institutional market makers. They collaborate with stock exchanges, reach agreements, and accept commitments in order to ensure asset turnover and supply and demand equilibrium. These suppliers include businesses that manipulate interest rates and foreign exchange rates in addition to commercial banks. Large banks, trading floors, brokerage firms, sizable funds, and wealthy people might all be among them.
- Intermediaries are market makers at the second level because they let smaller brokers and individual traders access the market. They run on their own liquidity, but if they need to, they can borrow money from the first level's liquidity suppliers. Market makers, as opposed to regular traders, focus on orders like Take Profit, Stop Loss, and pending orders while analyzing the market. Exchange participants fall under the category of speculative market makers when discussing the categories of market makers. When these market makers (such as small banks and individual

investors) deal, a price impulse is created due to their large stockpiles of assets.

Additionally, market takers need their own independent discussion. In contrast to market makers, who set or quote prices, market takers accept or take prices. Market makers, on the other hand, may only negotiate with market takers.

Market makers may be able to benefit smaller and private account investors. The disadvantages mostly manipulate advanced traders. The following are some advantages of market makers: Security Availability, Investor Confidence, Seamless Markets, Insider Trading, Conflict of Interest, and Impact Market Integrity [35].

A market maker supports transactions in a two-sided auction market, by holding both buy and sell offers. The market receives liquidity from an ever-present MM. Liquidity refers to having access to rapid trading opportunities at costs that accurately represent the state of the market. MMs earn money from the spread, or the variation between their buy and sell bids, as payment for providing liquidity. MM activity is often thought to stabilize prices and make it easier to find realistic pricing in the market. In different market institutions, market makers play a variety of roles [36]. Multiple MMs compete to quote prices in a pure dealer market, and market orders from investors are executed at the optimal price offered by market makers [37]. Hambly, B. et al, an entity, whether it's an individual trader or an organization, that generates profit through the placement of buy and sell limit orders for a specific financial product within the Limit Order Book (LOB) is identified as a market maker for that particular financial instrument [38]. Market makers are essential for providing liquidity and helping to sustain well-functioning, constant, and resilient financial markets across all the major exchange-traded and over-the-counter asset classes in the United States and globally [39].

The objective of market making differs from portfolio optimization and optimum execution in that it focuses on generating the bid-ask spread without accumulating undesirably big holdings, sometimes referred to as inventory [40]. Inventory risk, execution risk, and adverse selection risk are the three main forms of risk that a market maker must deal with [41]. Market makers face several key risks. Inventory risk refers to accumulating an undesirable large net position in a particular asset [42], execution risk is the chance that limit orders will not get filled within a preferred timeline [43], and adverse selection risk occurs when a market trend sweeps through a market maker's limit orders before the orders can be canceled, leading to losses. The following Figure 1 summarizes the overall advantages of algorithmic market-making [44]:

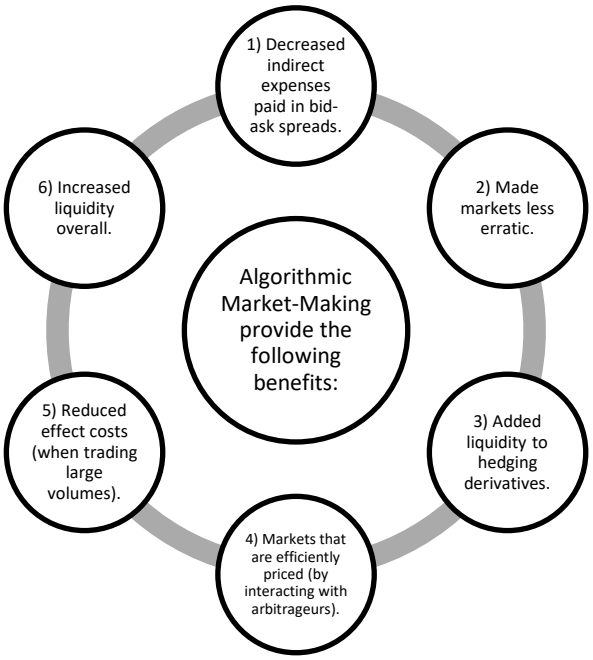

Figure 1: Algorithmic Market-Making

Portfolio Management:

Portfolio management (PM) refers to the process of managing a collection of investments, known as a portfolio, with the goal of achieving specific financial objectives. It involves making decisions about the allocation of resources, selecting appropriate investment options, and monitoring and adjusting the portfolio over time. The primary objective of portfolio management is to maximize the return on investment while managing the associated risks. This involves balancing the potential for higher returns with the need to diversify investments and minimize exposure to individual securities or asset classes. The overall aim of portfolio management is to optimize the risk-return tradeoff and help investors achieve their financial goals within their risk tolerance and investment timeframe [45].

Practically every work now carried out by humans is intended to be complemented by AI and even replaced by it. AI is applied in many different fields, driven by external pressures and technical advancements. Among them, financial AI applications have a bright future [46]. Using AI-based portfolio management tools can help investors make better decisions since they can offer data-driven insights [47]. A lot of data from many sources, including market trends, economic indicators, corporate reports, social media, and news, may be processed by AI. Additionally, it can evaluate and understand the data using machine learning and natural language processing and produce suggestions that may be put into practice. For instance, depending on the investor's objectives, preferences, and risk tolerance, an AI-based portfolio management tool might provide suggestions for which assets to purchase, sell, or hold. As part of their approach to portfolio management, portfolio managers participate in market-making activities.

Order Execution

Order execution is the procedure of receiving and fulfilling a purchase or sell order in the market on behalf of a client. Investors must execute a liquidation (or acquisition) order in order to buy (or sell) a specified number of shares in order to modify the new portfolio [45]. In essence, order execution has two goals: it must complete the entire order but also aims for a more economical execution with a focus on increasing profit (or lowering cost). As previously stated, the primary challenge in order execution lies in striking a balance between mitigating adverse market impacts arising from large trades executed rapidly and managing price risk. This, in turn, may lead to missed trading opportunities due to slower execution [48]. We begin by outlining several fundamental OE concepts:

Market Order

A market order is a command to purchase or sell a security right away. Although the execution of the order is guaranteed with this kind of order, the execution fee is not. A market order often executes at or close to the current bid (for a sell order) or ask (for a purchase order) price. Investors must keep in mind, nevertheless, that a market order may not be filled at the same price as the last traded price [48]. Market orders are carried out with certainty at the published prices in the market [49]. A market order is an instruction to purchase or sell a financial asset at the prevailing market price, signaling the intent to execute the transaction at the most favorable price at the present moment [45]. Market orders are frequently utilized when the investor is less concerned with the precise execution price and the immediacy of execution is a priority. They may be appropriate for highly liquid equities or for initiating or leaving positions fast. Market orders, however, may result

in considerable price effects or increased costs for bigger trades or in less liquid markets because there is no control over the execution price. Example: When the best offer price is $3.00 per share, an investor issues a market order to buy 1000 shares of the YX company. The investor's market order can be filled at a higher cost if other orders are filled before it. Additionally, a fast-moving market might also result in a huge market order having distinct portions executed at various prices. Let's use the above example where an investor issues a market order to buy 1000 shares of the YX company for $3.00 each. In a fast-moving market, the 500 shares order may execute at $3.00 per share and the remaining 500 shares execute at a higher price.

A market order will purchase or dispose of the shares at the market's best price at the moment the order is received. With a market order, you can be confident that you'll purchase or sell, but you have no control over the price at which you'll transact. Let’s see how a market order works [50]:

- A market buy will be made at the lowest bid price. If it purchases all of the available shares at the lowest ask, the ask above becomes the new lowest ask, and from there, more shares are purchased.
- The highest bid price will be used for a market sale. Similarly, if it sells all of the shares at the highest bid, the bid immediately below that will then become the highest bid, and that is where more shares will be sold.

Limit Order

Is a command to buy or sell a security at a certain price or higher. Only at the limit price or lower can a buy limit order be fulfilled, and only at the limit price or more can a sell limit order be fulfilled. For instance, an investor wishes to spend no more than $10 on shares of X stock. This amount might be specified in a limit order,

which would only be carried out if the price of ABC stock was $10 or less [51]. Thomas S et al, an offer to purchase or sell a certain quantity of an asset at a predetermined price (or better) is known as a limit order (LO) [52]. Each limit order specifies a price, a volume (the amount to be exchanged), and a direction (buy/sell or, equivalently, bid/ask). Additionally, when (1) the spread is wide, (2) the order size is large, and (3) they anticipate strong short-term price volatility, where traders put more limit orders compared to market orders [49].

In this form of order, you select the highest/lowest price at which you will buy/sell. If the trade is performed, a limit order guarantees the price at which you will purchase or sell, but it does not ensure that you will really transact at that price [50]. Let’s see how a limit order works:

- Buy limit order: If an investor wants to buy a security but is only willing to pay a certain price or lower, they can place a buy limit order. The investor specifies the maximum price they are willing to pay. If the market price of the security reaches or goes below the specified price, the buy limit order is triggered, and the broker or trading platform will execute the order at the specified price or better.
- Sell limit order: If an investor wants to sell a security but is only willing to sell at a certain price or higher, they can place a sell limit order. The investor specifies the minimum price they are willing to accept. If the market price of the security reaches or goes above the specified price, the sell limit order is triggered, and the broker or trading platform will execute the order at the specified price or better.

It's important to note that the execution of a limit order is not guaranteed. If the specified price is not reached, the limit order may remain unfilled until the market

reaches the specified price or better. Limit orders are typically used by investors who want to control the price at which they buy or sell a security and are willing to wait for the market to reach their desired price.

Limit orders can be useful in volatile markets or when an investor wants to be more specific about the price, they are willing to transact at, rather than relying on the prevailing market price at the time of the order.

Limit Order Book

The limit order book is the set of orders (prices at which you can deal) for a specific security. These orders might be on a single exchange or combined over several exchanges, depending on the security [50]. An electronic database maintained by an exchange all the buy and sell limit orders that are received for a particular instrument are kept in the limit order book (LOB). LOBs are organized according to price, time priority (FIFO order), and order direction (buy or sell). A limit order book is a record or database that displays all outstanding limit orders to buy or sell a particular security in a financial market. It provides transparency into the supply and demand dynamics for that security at different price levels. The limit order book is organized into two sides [52]: the buy side and the sell side, look at Figure 2. The buy side contains limit orders to buy the security, sorted by price from highest to lowest, while the sell side contains limit orders to sell the security, sorted by price from lowest to highest. The depth of the market, or the quantity of shares or contracts available at each price level, is also shown. As market orders or new limit orders arrive, they are matched against existing orders in the book based on price and time priority. Filled orders are removed, and the book is updated in real-time to reflect changes in order quantities and prices. The limit order book helps market participants

analyze market conditions, identify support and resistance levels, and make informed trading decisions based on the visible supply and demand for a security.

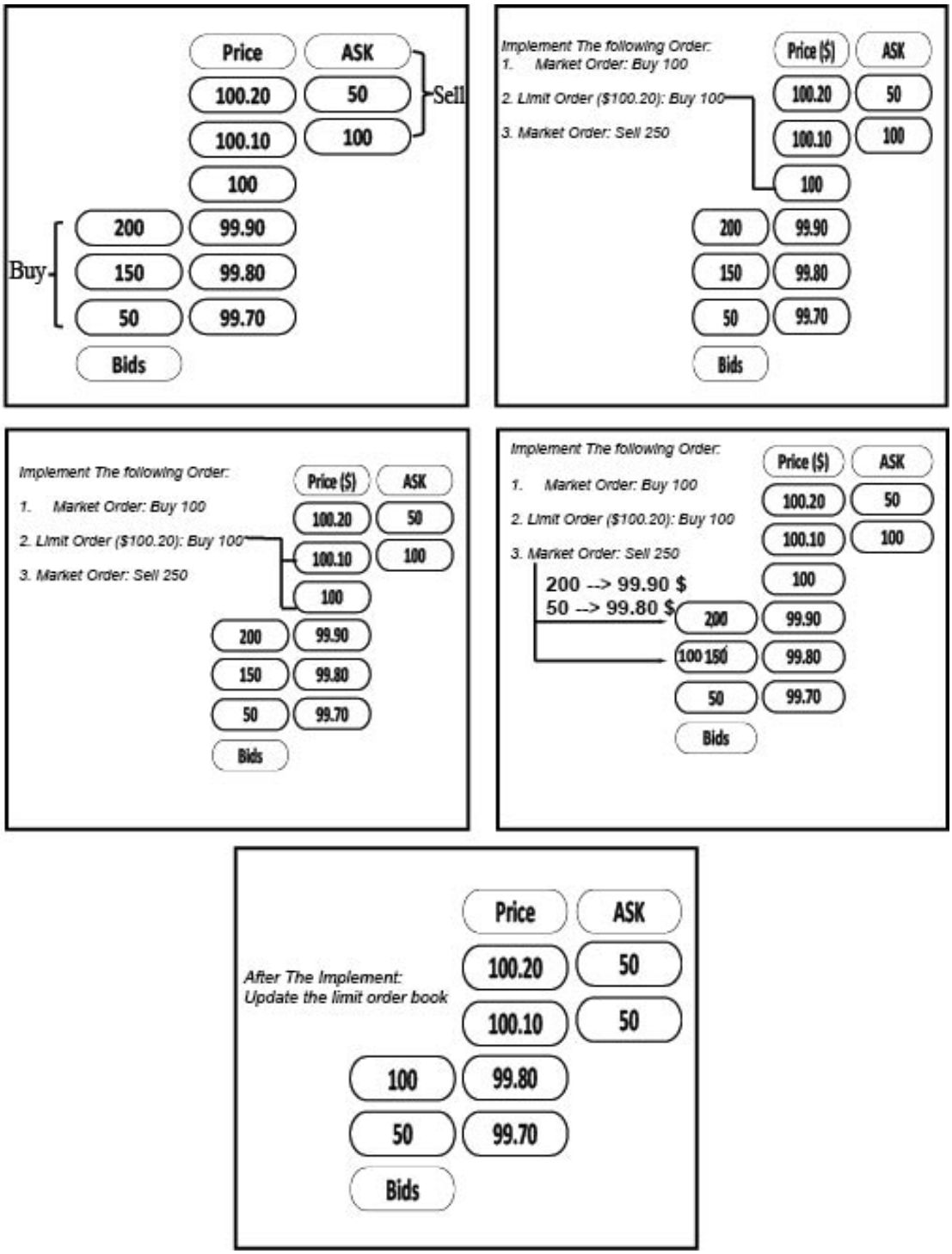

Figure 2: Limit Order Book Execution

Reinforcement Learning / Deep Reinforcement Learning (RL/DRL)

DRL is a branch of machine learning and artificial intelligence where intelligent computers may learn from their actions similar to how people learn from experience. An agent is automatically rewarded or punished according to their activities in this kind of machine learning. They are rewarded (reinforced) for doing actions that lead to the desired result. This technique is appropriate for dynamic situations that constantly change since a computer learns through trial and error. Even while reinforcement learning has been around for a long time, it was only lately that it was paired with deep learning, which produced amazing results. "Deep" in reinforcement learning refers to an artificial neural network with several (deep) layers that closely resemble the structure of the human brain. Deep learning demands a lot of training data and considerable computing power[53]. The proliferation of deep learning applications has been made possible by the expansion in data quantities over the past few years combined with sharp declines in the cost of computer power. In the financial Pit.AI, which stands for "solving intelligence for investment management, [54]" aims to use artificial intelligence, especially deep reinforcement learning, to outperform humans in managing investments and analyzing trading methods. With the improvements from Deep Learning, reinforcement learning has come a long way. DRL systems, algorithms, and agents that have already accomplished some unbelievable actions have been developed as a result of recent research attempts to combine Deep Learning with Reinforcement Learning. Such systems have not only exceeded the performance of the majority of classical and non-deep learning-based Reinforcement Learning agents, but they have also begun to outperform the best of human intelligence at tasks that were previously thought to require extremely high

levels of human intelligence, creativity, and planning skills [55].

DRL helps agents acquire the best decision-making rules through interactions with their environment [56]. A subfield of artificial intelligence that combines deep learning with reinforcement learning approaches. Due to its capacity to handle difficult issues involving sequential decision-making in dynamic situations, DRL has attracted a lot of interest recently.

Through interaction with the environment and feedback in the form of rewards or penalties, an agent learns to make decisions using the reinforcement learning paradigm [57]. On the other hand, deep learning makes use of artificial neural networks to analyze complicated data and derive useful representations. Deep neural networks are used as function approximators to manage high-dimensional input data and learn from unprocessed sensory inputs in DRL, which combines these two methods [52]. To maximize cumulative rewards over time, an agent in DRL interacts with the environment and makes decisions based on observations. Reward or penalty feedback is given to the agent, and iterative learning is utilized to update the agent's policy and value functions. The agent can learn complicated decision-making techniques by using deep neural networks to approximate its policy or value functions, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs) [58].

It's crucial to understand the idea of reinforcement learning. The agent may observe the situation and respond appropriately to help a network accomplish its objective. An input layer, an output layer, and several hidden layers make up this network architecture; the input layer is where the environment's state is kept. The model is based on several attempts to predict the future reward associated with each action taken in a certain state of the situation, Figure 3.

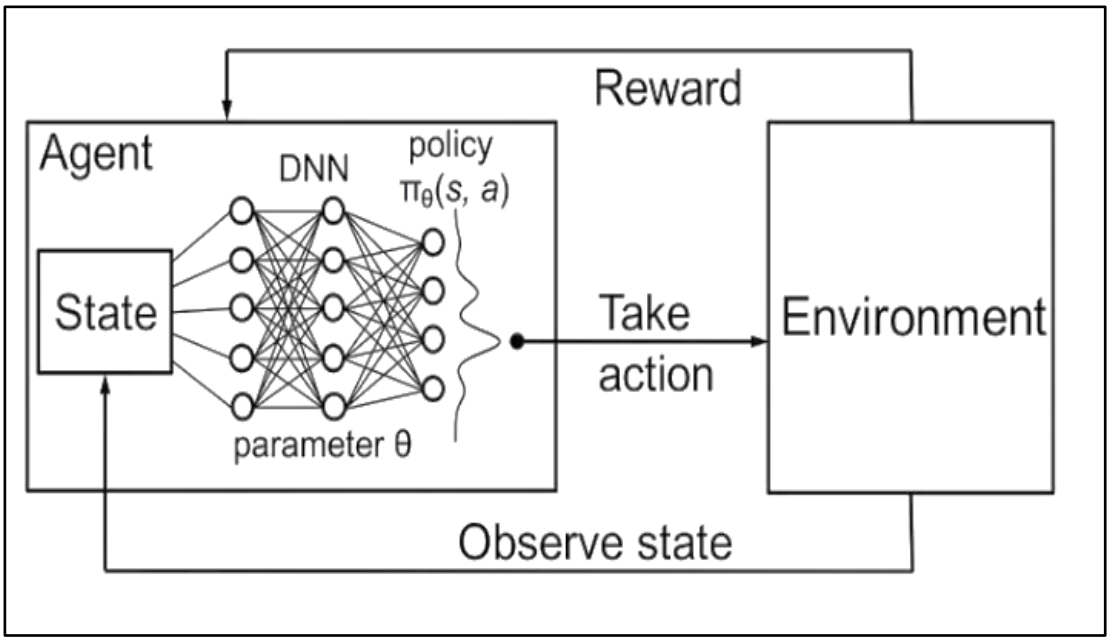

Figure 3: Schematic Structure of Deep Reinforcement Learning [59]

Before continuing, let's look at the schematic structure of DRL in Figure 3, and define them which will encounter when learning about RL and DRL [59][60] [9].

Agent: Agent (A) does actions that have an impact on the environment.

Action: It is the collection of every action or activity that the agent is capable of. The agent chooses one discrete action from a list of possible actions (a).

Reward (R): The environment provides feedback, which we use to judge whether the agent's behaviors in each state were appropriate. In the Reinforcement Learning scenario, where we want the machine to learn entirely on its own and the only criticism that would aid in learning is the feedback/reward it receives, it is

critical.

State: A state (S) is the specific circumstance in which the agent is now located.

Environment: Every action the reinforcement learning agent takes has an immediate impact on the environment. The reward is returned to the agent with a new state after the environment uses the agent's current state and action as information.

Policy (π): It determines what action to pick in a specific state in order to maximize the reward.

Value (V): It measures whether a certain state is ideal. It is the anticipated discounted rewards that the agent receives in accordance with the particular policy.

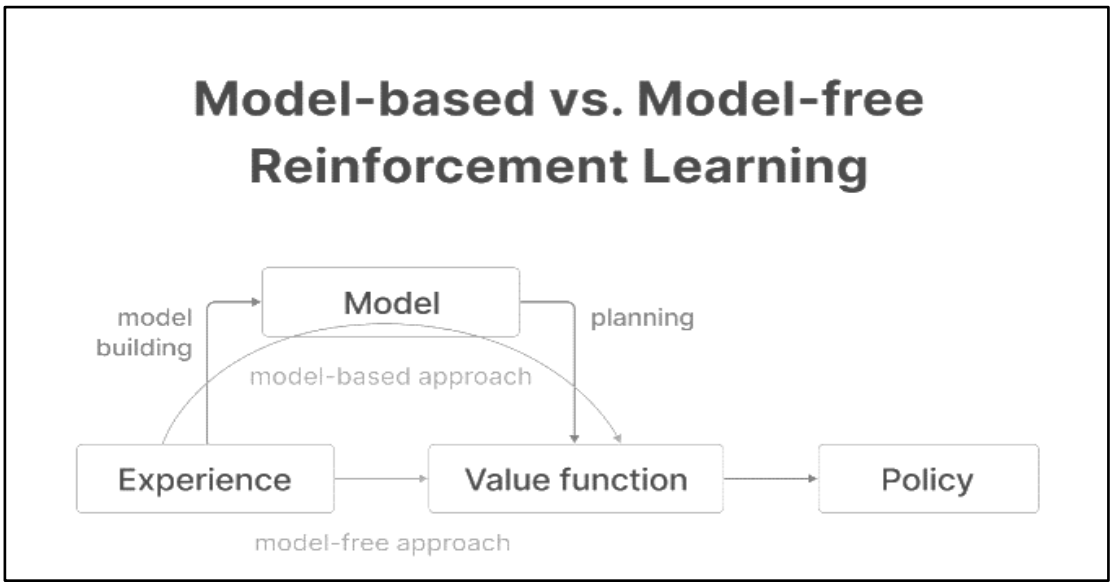

Figure 4: Model-based vs. Model-free Reinforcement Learning [61]

Figure 4 shows reinforcement learning algorithms fall into two categories [61][62][63]:

Model-Based Algorithms

Model-based algorithms evaluate the ideal policy using the transition and reward functions. When we fully understand the environment and how it will respond

to certain actions, we may employ them. In model-based reinforcement learning, the agent has access to the environment's model, which includes the actions that must be taken to change from one state to another, the probabilities associated with those actions, and the rewards that correspond to those actions. They enable the reinforcement learning agent to prepare by preparing in advance. Model-based Reinforcement Learning is better suited for static or fixed situations. In another way, we can say, in model-based algorithms, the agent can foresee the reward of a result and acts in a way to maximize that reward. It is a greedy algorithm, and all of its decisions are made with the goal of increasing the number of reward points.

Model-Free Algorithms

Model-free algorithms can discover the optimal policy of action with very little understanding of environmental dynamics. To determine the optimal policy, they lack any transition or reward functionality. They don't have any idea of the reward function and instead estimate the optimal policy of action based only on agent-environment interactions, or experience. Model-free algorithms Reinforcement Learning should be put into practice in situations where we only have partial knowledge of the environment. We don't live in a fixed environment in the actual world. The environment for self-driving cars is dynamic, with shifting traffic patterns, detours, etc. Model-free algorithms perform better in certain situations than alternative methods. Also, we can say, in model-free algorithms, the agent performs a variety of tasks repeatedly and gains knowledge from the results. Based on the learning experience, it attempts to choose a policy or a strategy to implement activities with the goal of obtaining the maximum number of reward points. This kind of technique should be used in environments that are dynamic and about which we

don't fully understand.

Bitcoin

Bitcoin, the first and most well-known cryptocurrency, was introduced to the world in 2008 through a whitepaper titled "Bitcoin: A Peer-to-Peer Electronic Cash System" by an individual or group of individuals under the pseudonym Satoshi Nakamoto. The primary motivation behind Bitcoin's creation was to establish a decentralized digital currency that operates without the need for a central authority, thereby offering a new way of conducting transactions over the Internet. The genesis block of Bitcoin was mined in January 2009, marking the beginning of a new era in the financial world [61].

Bitcoin is digital cash that is not controlled by a central authority, and it can be used for direct dealings across the world without banks or rulers. It is backed by chain tech, which makes sure that all deals are seeable, safe, and unchangeable. In March 2024, there are about 19 million bitcoins being used, and the total supply is limited to 21 million, making it valuable and rare [62].

Bitcoin Techniques

Bitcoin functions as a decentralized virtual currency system and a form of cryptocurrency, designed to operate independently of governmental, banking, and institutional oversight. Conceptually, Bitcoin resembles electronic cash, enabling transactions between individuals using compatible software known as a wallet on various devices such as computers, smartphones, or tablets. However, it's crucial not to mistake Bitcoin for digital cash since it doesn't represent stored digital units of value like traditional currency. Instead, Bitcoins should be regarded as funds held in

an account. When a payment is made, the account of the sender is debited and the account of the receiver is credited, rather than transmitting digital notes or coins. Transactions are facilitated through encrypted messages and verified within the Bitcoin user network [63].

In addition, Bitcoin's main technology is blockchain. It's a shared record that records all transactions on many computers. Things people buy are grouped and added with mining. Miners solve hard math puzzles with big computers, and the first to solve it can add the next block to the blockchain. They get bitcoins as a prize. This keeps the network safe and makes more bitcoins, sticking to a set rate.

Bitcoin Design Principles

Scarcity serves as a fundamental requirement for attributing worth to any currency variant. Scarcity prevents counterfeiting on a local scale, but on a larger scale, it limits the increase of the monetary base and promotes price stability. In contemporary economies, when money largely resides in electronic versions, scarcity is enforced by legal laws that assure the veracity of accounting records. This implies a financial structure in which transactions result in credits to one account and debits to another. Central banks have the ability to control the total amount of money in circulation. In this context, Bitcoins emerge as the first widely embraced mechanism to offer absolute scarcity of the money supply. By design, Bitcoins operate without a central authority to distribute or monitor coin ownership. Consequently, the issuance of currency and validation of transactions in Bitcoins is notably more challenging than in traditional accounting systems. Nonetheless, Bitcoins incentivize private entities to maintain their accounting system by issuing new currency at a controlled rate, thereby encouraging them to verify transaction validity [64].

Bitcoin is made on some main ideas:

- No one controls it. This keeps it open, free, and hard to stop.
- Using codes to keep deals safe and guard the net from trickery.
- All deals are on the list, so anyone can check them.
- Only a set number of bitcoins can be made, so it's a rare thing.

Risks and Problems

Though Bitcoin has good points, it also has issues:

- Changeable price: Bitcoin's value goes up and down a lot, making it a risky choice.
- Sizing up: The network can't handle many transactions at once, which can cause problems.
- The rules aren't clear: The laws about Bitcoin are not the same everywhere, which can affect how much it's used.
- Hurting the earth: Making Bitcoin uses a lot of power, which is bad for the environment [65].

Future and Summary of Bitcoin

The future of Bitcoin is a subject of much speculation. While some view it as a digital gold and a hedge against inflation, others believe it will become a mainstream form of payment. Ongoing developments, such as the Lightning Network, aim to address scalability issues, potentially increasing Bitcoin's utility as a medium of exchange. However, regulatory and environmental challenges remain significant hurdles to its widespread adoption [66].

Bitcoin has truly revolutionized the way we view and interact with money,

thanks to its groundbreaking decentralized approach and the power of blockchain technology. It's not been an easy journey, with hurdles such as scalability and regulatory scrutiny at every turn. Yet, the horizon looks bright for Bitcoin, especially with innovations like the Lightning Network on the rise. As the cryptocurrency ecosystem evolves, Bitcoin's role as a store of value and medium of exchange is set to strengthen, driving innovation in decentralized finance.

Being generated and stored entirely electronically, Bitcoin is one of the most well-known digital currencies in the world. By the time our children grow up, everyone will be using Bitcoin for transactions. For some time to come, there will remain security issues and uncertainty surrounding the regulation of digital currencies.

CHAPTER 3: IN-DEPTH EXPLORATION OF DEEP REINFORCEMENT LEARNING IN MARKET-MAKING

Deep Reinforcement Learning for Market Making

DRL has gained significant attention in recent years as a powerful technique for solving complex decision-making problems. One such problem is market making, which involves providing liquidity in financial markets by continuously quoting bid and ask prices. DRL can be employed in market-making to develop trading strategies that optimize liquidity provision, manage risk, and maximize profitability. The objective is to create an automated system capable of adapting and learning from market dynamics to make informed trading decisions.

The following is a high-level overview of how DRL can be utilized in market-making:

- Environment modeling: The initial step is to define the market environment and the available actions for the agent. This environment typically consists of historical and real-time market data, such as price movements, order book information, and trade volumes. The agent's actions may involve placing buy/sell orders, adjusting bid/ask prices, or modifying order sizes. Selecting a suitable state representation is critical in DRL for market-making. It involves choosing relevant features or indicators that capture the current state of the market. These features may include price spreads, order book depth, volatility measures, trade volumes, or sentiment analysis from news or social media. The state representation should provide sufficient information for the agent to make

effective trading decisions.

- Reward function design: A reward function is established to provide feedback to the agent based on its actions. This function should encompass market-making objectives such as maximizing profits, minimizing spreads, reducing inventory risk, or tracking a benchmark index. Designing an appropriate reward function is vital for effectively guiding the agent's learning process.
- Reinforcement Learning algorithm for MM: Deep Q-Network (DQN) architecture DRL algorithms like DQN can be employed to train the market-making agent. DQN integrates deep neural networks together with Q-learning, an algorithm for reinforcement learning. The neural network takes market data as input and predicts the optimal action to take in a given state. It learns by iteratively updating its Q-values based on observed rewards and expected future rewards. In market-making, various DRL algorithms such as PPO, DDPG, and A2C can be applied, alongside DQN. The choice of algorithm depends on the specific market-making problem and available data, as each algorithm has its own strengths and weaknesses.
- Training and optimization: The agent is trained using historical market data or simulation environments, where it engages with the environment and learns from its interactions. The training process involves exploring different actions, evaluating their outcomes, and updating the agent's policy to enhance decision-making. Techniques like experience replay and target networks are often employed to stabilize the learning process.
- Testing and deployment: Once the agent is trained, it can be tested in real-

time market conditions to evaluate its performance. Continuous monitoring and performance evaluation are essential to ensure the agent's effectiveness and adaptability. In practice, the agent can be deployed as part of an automated trading system to provide liquidity and execute trades in real markets.

- Risk Management: Effective risk management is crucial in market-making, involving the management of inventory risk, market risk, and execution risk. DRL agents for market making should incorporate risk management mechanisms to avoid excessive exposure and protect against adverse market movements. Techniques like position limits, dynamic risk limits, and stop-loss mechanisms can be integrated into the agent's decision-making process to mitigate.

It's important to note that applying DRL to market-making is a complex task that requires expertise in both reinforcement learning and financial markets. Proper risk management, regulatory compliance, and real-time data infrastructure are crucial considerations when implementing such systems in live trading environments.

Categories of DRL-Based MM Models

We find a wide variety of magical classes within the grimoires (A class of instruction to portray the magic of AI) of DRL-based market building, each with its own special allure:

- Q-Learning Arcana**:** These spells take us into the holy land of Q-Learning, where we attempt to approximate the illusive action-value function. Our agents will benefit from this empowerment because they will be able to make well-informed choices based on an anticipation of future outcomes.

Q-Learning has given our market makers the insight they need to safely sail the uncharted waters of uncertainty in pursuit of optimal actions and alluring rewards.

- Actor-Critic Enchantment: In this bewitching union of forces, the actors and the critics combine forces, establishing a harmonious balance that improves the efficacy and stability of our magical education. The actors, even the performers, are at the center of the process of determining the best policies to implement, while the critics, sages of critique, stand in the wings to evaluate and support the actors' efforts. Our market makers are elevated by the chemistry of the actor and critic combo, allowing them to use their authority with poise and subtlety.
- Proximal Policy Alchemy: Using the mystical alchemy of Proximal Policy Optimization (PPO), we reveal the keys to optimizing our goals and making sure that policy updates are in sync with our previous experience. With PPO, we can fine-tune our policies without losing sight of their overarching goals, allowing our magical agents to grow in knowledge without losing sight of the lessons they've already learned.
- Deep Deterministic Potions: The elixir of Deep Deterministic Policy Gradients (DDPG) gives us insight into our choices and removes the fog of conventional policy gradients. Our market makers get clarity of mind and confidence in their judgment thanks to deterministic policies. Our agents are empowered by the DDPG elixir to make sound decisions in a shifting market environment while remaining true to their core values.
- Twin Delayed Enchantments: Twin critics and target policy smoothing take our spellcasting to new heights in the world of Twin Delayed Deep

Deterministic Policy Gradients (TD3). The twin critics' complementary viewpoints help us gain a more nuanced appreciation of how our actions affect others. Adding a touch of magic, and target policy smoothing makes our enchanted market makers more reliable and effective. With TD3, our agents are able to make clear and nuanced judgments with pinpoint accuracy. As we venture into the enchanting world of DRL-based market-making, each category offers its unique allure and strengths, enabling our market makers to wield their powers with mastery and wisdom. With these mystical techniques at our disposal, we step forth into the realm of strong market-making, ready to unlock its deepest secrets and emerge as true Master of Financial.

Why DRL is Needed in Market-Making

DRL is a subset of ML used to train agents to make decisions in complex financial environments. When combined with market-making, DRL offers several advantages in the financial industry. It excels in adapting to dynamic market conditions, handling uncertainty, and managing risks. DRL's ability to handle multi-dimensional decision-making problems makes it effective for market-makers. Its continuous learning capabilities allow it to improve over time and respond quickly to high-frequency trading environments. Automation of decision-making reduces the need for manual intervention, leading to increased efficiency and lower costs. Moreover, DRL can explore and discover new trading strategies, enhancing the overall decision-making process.

How to Select the Appropriate RL/DRL Model

Within the expansive realms of RL and DRL, picking the model that is best suited for effective market-making calls for careful consideration of a number of different factors. The following are important guidelines that will assist in the decision-making process:

- Acquire an In-Depth grasp of Market Dynamics: Before plunging into the depth of RL/DRL, acquire an in-depth grasp of the dynamics and microstructure of the particular market. There is a possibility that various markets will each exhibit a unique set of features, including liquidity, volatility, and order flow patterns. Choose an RL/DRL model that works well with the specific characteristics of the market you're going after.
- Specify the Goals and Limitations of the Project: Outline in detail your market-making goals, as well as any restrictions that may be imposed on you by rules, risk management, or company requirements. It's possible that different RL/DRL models are superior when it comes to optimizing specific performance criteria, such as maximizing earnings, limiting risk, or achieving stability. Select a model that offers the greatest degree of congruence with the particular aims and limitations you have.
- Consider Data Availability and Sample Efficiency: Take into Account the Availability of Data and the amount of Sample Efficiency Needed for the Model Determine whether or not historical data is available and determine the amount of sample efficiency needed for the model. If there is a large quantity of historical data available, then data-intensive techniques such as deep Q-learning or actor-critic methods might be appropriate. PPO, on the other hand, is a model-free technique that can efficiently learn from a

smaller amount of data and should be considered for use in situations with minimal data.

- Consider Robustness and Risk Management: Because market-making always involves risk, it is necessary to select an RL/DRL model that is both reliable and successful in the risk management methods it incorporates. The models that have the most priority should be those that include built-in safeguards against excessive market swings and catastrophic losses.
- Evaluate the Computational Difficulty: Take into account the computational resources that are at your disposal in order to implement the RL/DRL model of your choice. It's possible that some of the more complex DRL models have a high computational cost, meaning that they need a lot of processing power and a lot of time to train and run. Choose models that have a satisfactory ratio of performance to the amount of computing effort they require.
- Ensure Explicability of Models: In the complex world of finance, ensuring that models can be explained adequately is essential to fostering confidence and maintaining regulatory compliance. Find RL and DRL models that can explicate the decision-making process and offer interpretability. This will enable stakeholders to comprehend and validate the market-making tactics.
- Conduct Thorough Experiments and Benchmark: For different approaches before fully committing to a Particular RL/DRL model, it is important to first conduct thorough experiments and then benchmark different methodologies. Evaluate the performance of the models by comparing

them to historical data and generated scenarios in order to determine how successful they are under different market situations.

- Remain Flexible and Adaptable: The realm of RL and DRL is continuously undergoing change, with new models and developments being introduced on a consistent basis. Maintain a flexible approach to your market-making tactics and be open to adjusting them as new and more potent RL/DRL models are produced.
- Seek Out Expertise and Consider Collaborating with Others: If navigating the enchanted world of RL/DRL seems overwhelming, seek out guidance from experts and consider collaborating with researchers or professionals who are well-versed in the field of market making. Your efforts to create a market could see a huge boost in its efficacy if you took advantage of their views and skills.

The below table 1 shows various RL/DRL models with the functionality of each model.

Table 1: Function of RL/DRL Models

Model	Function
Q-Learning	Q-learning learning knowledge about the value of taking an action within a particular state, all without the need for an input model to be supplied. An expected cumulative reward for a given policy is estimated using a Q-value function. Using the temporal difference (TD) error and the Bellman equation, Q-values are updated iteratively.
Deep Q-Network (DQN)	In order to approach the Q-value function, deep Q-learning (DQN) employs deeper neural networks. It implements a buffer for reliving past events for the purposes of training, drawing at random from that pool of data. By comparing the predicted Q-values to the goal Q-values, the network is trained to achieve the lowest possible TD error.
Policy Gradient (PG)	By optimizing the policy parameters, PG techniques can learn the policy function that translates states to actions. The policy is updated via gradient ascent, which takes into account the sampled paths' expected reward. PG techniques work well with stochastic policies and can be used to continuous action spaces.
Proximal Policy Optimization (PPO)	PPO is an on-policy DRL algorithm that continuously adjusts policy settings in light of accumulated experience and new information. It limits the policy update to avoid major policy shifts, making education more consistent and secure. The objective function, which is a measure of both the efficiency and randomness of the policy, is optimized using PPO.
Actor-Critic	The Actor-Critic framework combines two separate yet interconnected techniques: the actor-network focuses on learning the policy gradient, while the critic network evaluates the value of states and associated actions. The advantage of policy gradients and the value function estimation are taken into account in Actor-Critic algorithms.

Model	Function
Deep Deterministic Policy Gradient (DDPG)	DDPG is an off-policy approach that is effective for continuous autonomous vehicles, utilizing deep neural networks for both the actor and critic functions within an actor-critic framework. To maintain consistency during training, DDPG uses a target network, and off-policy changes are implemented via a replay buffer.
Twin Delayed Deep Deterministic Policy Gradient (TD3)	TD3 is superior to DDPG because it prevents Q-values from being overestimated. It employs delayed updates to the target networks for stability and makes use of two critics to estimate Q-values. TD3 additionally adds noise to continuous action spaces to promote exploration and strengthen robustness.

To summarize, in order to select the optimal RL/DRL model for effective market-making, one needs to have a comprehensive grasp of the dynamics of the market, well-defined goals, and a sharp eye on risk management and computational efficiency. You will be able to harness the true potential of RL/DRL in the magical art of market-making if you give these things due consideration and maintain a level of adaptability in the face of new developments.

How it was Market-Making Before AI and How AI Improved

Before the advent of AI, market-making was primarily carried out by human traders who would manually assess market conditions, analyze data, and execute trades to provide liquidity in financial markets. These market makers would set bid and ask prices for specific securities or financial instruments, aiming to profit from the spread between these prices. This process required extensive market knowledge, experience, and quick decision-making skills. Traders used quantitative models and algorithms to optimize pricing and manage risks, although these models were

typically simplistic and unable to adapt to real-time market changes. The introduction of AI has revolutionized market-making by leveraging advanced machine learning algorithms and large-scale data analysis. AI systems have the capacity to examine extensive quantities of market data, encompassing historical pricing, trading volumes, news reports, and public sentiment on social media. This enables AI to uncover hidden patterns, correlations, and anomalies that may elude human traders. AI-powered market-making systems continuously learn from historical data and adapt to evolving market dynamics. They generate real-time trading signals, optimize pricing models, and adjust bid-ask spreads based on market conditions and risk tolerance. AI enables market makers to react swiftly to new information, execute trades at high speeds, and improve liquidity provision while reducing bid-ask spreads. Moreover, AI algorithms excel at identifying and capitalizing on market inefficiencies. They process extensive data and detect subtle patterns that indicate potential trading opportunities. AI can simultaneously analyze multiple markets, identify arbitrage possibilities, and execute trades across different exchanges with minimal delay.

In summary, AI has significantly improved market-making by enhancing speed, accuracy, and efficiency. It has boosted liquidity, reduced trading costs, and facilitated smoother price discovery in financial markets. However, challenges such as the need for robust risk management systems and the potential for algorithmic biases should be carefully considered in AI-powered market-making.

DRL in Algorithmic Trading

Algorithmic trading, known as quantitative trading as well, is a finance subfield that revolves around the automatic generation of trading decisions through the utilization of mathematical rules computed by a machine. It is a methodical

approach where trading choices are made based on predetermined rules, typically derived from the technical analysis of market data. The primary objective of algorithmic trading is to execute trades at optimal prices while minimizing risks [67]. Algorithmic trading has experienced substantial growth in the last decade, with approximately 70% of trading volume in the U.S. stock market attributed to algorithmic trading [68]. The global market for algorithmic trading was valued at $2.03 billion in 2022, and it is expected to expand from $2.19 billion in 2023 to $3.56 billion by 2030 [69].

Algorithmic trading relies on complicated algorithms that analyze a variety of market data, such as price fluctuations, trading volumes, and other pertinent factors, in order to make informed trading decisions. These algorithms can be programmed to execute trades based on specific criteria, such as price thresholds, technical indicators, arbitrage possibilities, or significant news events. The benefits of algorithmic trading encompass enhanced speed and efficiency in trade execution, minimized human errors, the capacity to process extensive real-time data, and the potential to exploit momentary market inefficiencies[70]. Algorithmic trading systems can swiftly respond to market conditions and execute trades much faster than human traders, enabling them to capitalize on transient opportunities or execute substantial orders without significantly impacting prices. However, algorithmic trading entails specific risks, which include the possibility of technological malfunctions, errors in programming, inaccuracies in data, and market volatility. To guarantee fair and orderly markets and to address these risks, regulations and risk management practices have been implemented.

High-frequency trading (HFT) has gained popularity as a prevalent form of algorithmic trading. HFT and algorithmic trading have become the preferred choices

for regulators and regular stock market investors. HFT involves the rapid mechanical buying and selling of large volumes of stocks and shares. It is an evolving field that is expected to dominate algorithmic trading in the future. Algorithmic trading has revolutionized the trading landscape by introducing speed and efficiency to securities trading. Traders are utilizing algorithms that are becoming increasingly sophisticated and capable of adapting to diverse trading patterns through the use of artificial intelligence (AI). As the field progresses, it is anticipated that algorithmic trading will incorporate practical machine learning (ML) techniques capable of real-time analysis of vast amounts of data from various sources. ML, a subfield of computer science, draws upon statistical models, algorithms, artificial intelligence, and other disciplines to develop efficient computational methods for deriving accurate predictive models from extensive datasets. This makes ML an ideal candidate for addressing challenges in HFT, including trade execution and generating alpha (measuring asset or portfolio performance). Consequently, the combination of algorithmic trading and ML can be defined as AI trading, offering significant potential for further advancements in the field [68].

DRL is an approach employed in algorithmic trading to train an agent in making trading decisions by utilizing past market data. The agent is trained to optimize a reward signal, commonly associated with profits or returns on investment. DRL has demonstrated its efficacy in resolving intricate trading challenges, offering the advantage of handling substantial data volumes and adapting to dynamic market conditions. Thibaut Theater and Damien Ernst [67], provide an illustration of how DRL can be implemented in algorithmic trading to enhance trading performance. They introduce the Trading Deep Q-Network (TDQN) algorithm, which employs a deep Q-network to learn the most advantageous trading policy for a specific stock,

relying on historical market data. The agent is instructed to optimize the expected future reward, which relates to the earnings or investment return. Through testing on a diverse set of 30 stocks, the TDQN algorithm demonstrated superior performance compared to various benchmark strategies.

DRL in Portfolio Management

In the realm of portfolio management, we harness the power of DRL to orchestrate an exquisite transformation in our asset blend. Through the DRL algorithms, we elegantly rebalance your portfolio, orchestrating peak achievements amidst market turbulence. These captivating algorithms guide us in crafting portfolios that spark in harmony with market rhythms, optimizing gains while safeguarding against risks. With this technological enchantment, we transmute ordinary investments into a harmonious symphony of prosperity. DRL has garnered significant interest within the domain of Portfolio Management, presenting novel approaches for enhancing investing methods. An exemplary contribution in this field is the utilization of DRL in the context of portfolio management, as demonstrated by Jiang et al. [71]. The authors employed the Deep Deterministic Policy Gradients (DDPG) method for this purpose. The main goal of this research is to optimize portfolio rebalancing in a continuous action space, hence enabling dynamic asset allocation. The Deep DDPG technique, which is widely used in the field of DRL, is utilized to acquire knowledge and adjust the investment strategy iteratively, with the aim of maximizing returns while effectively mitigating risk. This study showcases the efficacy of DDPG in addressing the intricate and ever-changing aspects of portfolio management. By employing DRL approaches, this research lays the groundwork for future breakthroughs in this rapidly growing domain. DRL has emerged as a potential

methodology for tackling the intricate issues associated with portfolio management. Numerous research investigations have been carried out to investigate the utilization of this approach in enhancing investing techniques, with each study offering distinct perspectives and advancements to the discipline. Liang et al.[72], utilized the PPO method within a DRL framework to tackle the challenge of portfolio optimization. The utilization of the PPO algorithm was employed with the objective of attaining improved risk-adjusted returns through the dynamic allocation of assets within a portfolio. Moody and Saffell [73], created the Q-learning method for portfolio selection, which stands as a significant contribution in their body of work. The aforementioned groundbreaking study established the fundamental principles upon which further advancements in DRL for the field of finance were built. In the study conducted by Zhang et al. [74], a hybrid DRL methodology was employed. This approach involved the integration of Advantage Actor-Critic (A2C) with the Gaussian Mixture Model (GMM) in order to develop a portfolio management strategy that exhibits enhanced resilience. These works collectively highlight the potential of DRL in the field of portfolio management. They demonstrate the use of different algorithms and approaches to improve investment decision-making and risk management.

DRL in Order Execution

With our DRL expertise under our belts, we have the ability to perform potent incantations that speed up the carrying out of commands. We can execute trades with pinpoint accuracy by performing the spells, which in turn reduces market volatility and unlocks previously unknown means of obtaining optimal pricing. Every time we make a deal, we cast an incantation of efficiency to make sure our orders go unnoticed by the market and leave nothing but a trail of profitable in their wake. The application

of DRL has been widely utilized in the field of order execution, leading to significant changes in the landscape of trading methods. In the scholarly article entitled "A Deep Reinforcement Learning Framework for Optimal Trade Execution," researchers S. Lin and P.A. [75] Beling present a novel framework aimed at reducing trade execution costs. The proposed approach involves the sequential division of a sell order into smaller child orders within a predetermined time interval. The framework employed in this study leverages a customized version of the DQN algorithm, which integrates various enhancements including Double DQN, Dueling Network, and Noisy Nets components. In contrast to prior studies, which utilize implementation shortfall as an instantaneous incentive, the present framework adopts a modified reward system and incorporates a zero-ending inventory constraint into the DQN algorithm through adjustments to the Q-function updates during the final stage. The conducted study showcases the notable benefits of the framework, which encompass swift convergence during the training process, superior performance compared to multiple benchmark algorithms during back-testing on a set of 14 US equities, and increased stability resulting from the integration of the zero-ending inventory constraint.

DRL has significantly advanced the field of order execution in financial markets, presenting a transformative approach to optimizing trading strategies. The authors made a significant addition to the field through their utilization of the DDPG algorithm. The present study emphasized the notable capability of DRL agents to improve strategies for executing orders by training them to optimize utility functions that consider both transaction costs and execution slippage.

At the center of the market-making universe, we tap into DRL's full potential. By constantly adjusting the bid and ask quotes with each wave of the wand, we conjure narrow spreads that entice traders to enter our domain. Quickly adapting to shifting market conditions, DRL's market-making expertise provides a large pool of liquid assets for both buyers and sellers. Traders and investors are captivated by our market-making skills because of the sense of security they provide. With DRL's guidance, we're able to go beyond the constraints of conventional methods as we explore the mystical world of finance. With every monetary we cast, our power and understanding grow. DRL had a significant impact on market making, revolutionizing trade dynamics, and enhancing profitability. The pioneering work of Bell, et al. [76], serves as an exemplary contribution to this particular domain, as it effectively utilized the PPO algorithm. The research shed light on the effectiveness of DRL agents in the practice of market making, as they were able to strategically generate bid and ask quotes with exceptional accuracy. This resulted in lower spreads, which in turn attracted traders in a highly enticing manner. The study conducted by Bell et al. provided evidence of the remarkable ability of DRL to effectively respond to dynamic market situations. This led to a significant increase in the availability of liquid assets for both buyers and sellers. The sense of confidence that DRL provides to traders and investors has solidified its position as a significant entity in contemporary finance. The integration of DRL into market-making methods has played a pivotal role in transforming the financial industry, providing unprecedented benefits. P. Kumar, et al. [77], investigates the domain of market-making strategies within the context of high-frequency trading. This particular area of study has received limited attention thus far, making it an intriguing and relatively unexplored field. Market makers have a pivotal

function in financial markets as they provide liquidity by presenting bid and ask prices. Their profitability stems from the difference, known as the spread, between these two prices. This research paper presents a novel approach to modeling limit order markets by utilizing realistic simulations. The study further explores the application of Deep Recurrent Q-Networks in the development of a market-making agent. The present study showcases a novel methodology that surpasses established benchmark techniques relying on temporal-difference reinforcement learning. This approach exhibits promising capabilities in accurately reproducing historical trade data and capturing stylized facts. Kumar's research makes a significant contribution by providing valuable insights into market-making algorithms and their practical application. This research sheds light on the intricate dynamics involved in high-frequency trading.

When DRL is applied to portfolio management, order execution, and market making, new doors of opportunity open up, and finance becomes a brilliant performance of competence.

CHAPTER 4: IN-DEPTH STUDY OF THE EXISTING LITERATURE

Statistics of Related Works for DRL in MM / Literature Review

Web of Science, Scopus, and Google Scholar were searched for "market making" and "reinforcement learning". After deleting duplicates and unnecessary references, more than 35 relevant publications were collected, including grey literature (a doctoral thesis, and numerous unpublished studies mostly available on http://www.arxiv.com, viewed on 26th July 2023). Figure 5 shows the annual publishing increase

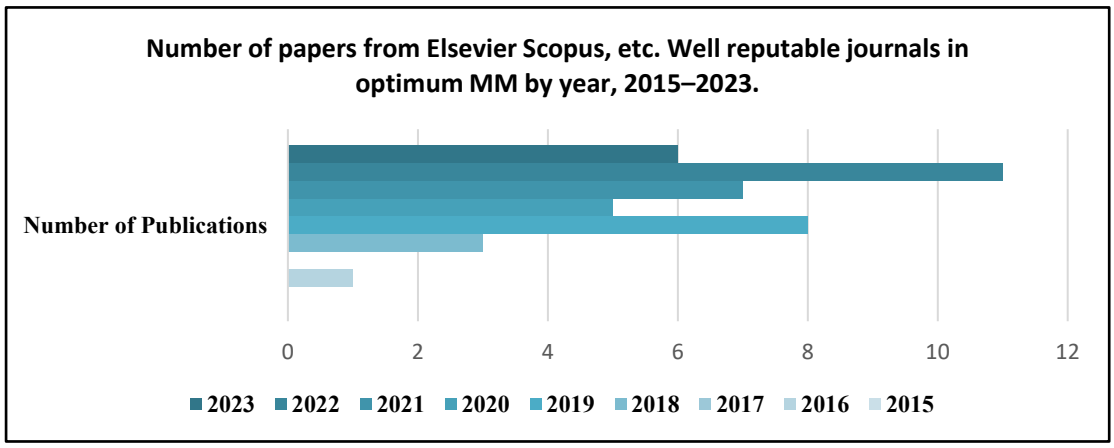

Figure 5: Number of Papers from Elsevier Scopus, etc. Well Reputable Journals in Optimum MM by Year, 2015–2023

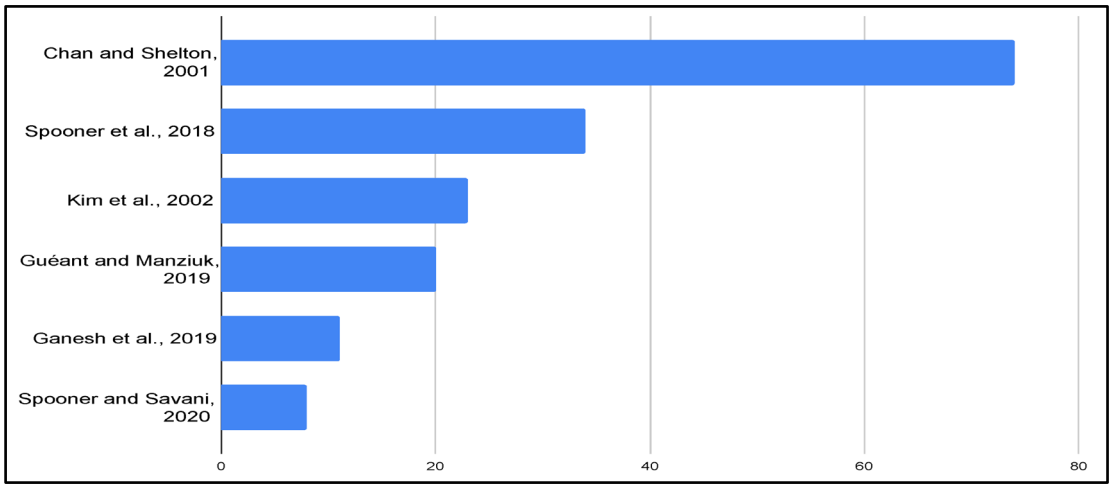

Figure 6: Authors-Based Publications of Papers

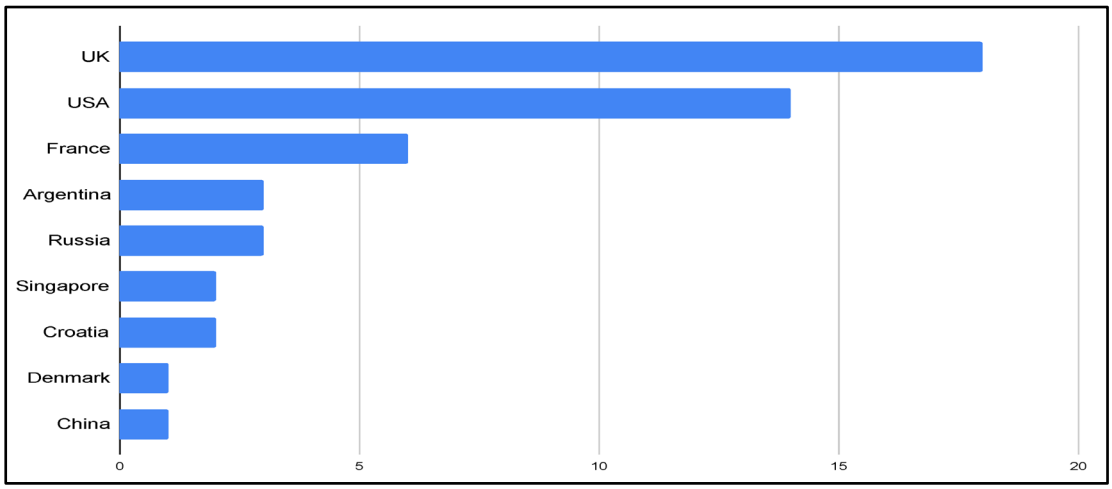

Figure 7: Country-Based Publications of Papers

The years covered by the cited works range from 2015 to 2023, with the vast majority (87%) appearing to be more recent. They feature work by 50 authors located in 9 different nations, shown in Figure 6, with the United Kingdom (36%), the United States (28%), and France (12%) providing the lion's share of the contributors. Figure 7 demonstrates that citations to two major papers from the MIT AI Lab by Chan et al. and Kim et al. remain high at 74 and 23, respectively. Recent studies, such as Spooner et al., Gueant and Manziel., Ganesh et al., and Spooner and Savani, have been well-received by the academic community. Particularly influential in following studies on the topic is the 2018 publication by Spooner et al.

Various Approaches

Reinforcement learning has spread to the mythical world of market making, where dealers sway to the beat of prices. Researchers have set out on a mission to uncover the most powerful methods for attaining maximal market-making prowess in this fascinating area.

Look at these four groups that form the melting pot of information:

- Information-Based Approaches: Information asymmetry is at the center of these strategies. Market makers, much like courageous warriors, must contend with the disadvantage of having less information than some traders. But don't worry! The first explorers relied on model-free reinforcement learning to intelligently set their buy and sell prices. Market makers can now strike a better balance between profits and inventory levels thanks to the disclosure of risk-sensitive approaches. Different reward formulations wove the magic of risk-averse behaviors, whereas multiagent simulations reflected the strategies of competitors.
- Approaches Stemming from Analytical Models: The Avellaneda-Stoichkov paradigm was used as a framework, with ancient texts serving as inspiration. The outcomes of combining this knowledge with reinforcement learning algorithms were revolutionary. The AS model stuck by them through thick and thin as they discovered the superiority of RL over more conventional approaches. It has even been suggested that the AS model is nothing less than a huge struggle of wills between the market maker and everyone else. By using adversarial reinforcement learning methods, resilient MM agents were created.
- Nondeep Methods: Tabular reinforcement learning ruled the domain of simplicity. These experts set out on a journey toward more effective decision-making in the marketplace with Q-learning as their north star. Their objective? To get to the bottom of how utility functions and incentive structures work. Over time, they came to the conclusion that RL techniques are the best bet for outperforming more conventional AS

approximations. Agents with better risk-adjusted performance were generated by the reverberations of linear combinations and state aggregation in this world.

- Deep Reinforcement Learning Approaches: The mysterious appeal of deep neural networks is called to researchers interested in deep reinforcement learning approaches. Some brave people have ventured into the depths of end-to-end frameworks, creating MM from raw limit order book data. These researchers were given the tools they needed to uncover the mysteries of cryptocurrency market making thanks to the emergence of extraordinary neural networks with hidden layers. In this field, ideas flourished unrestrained as people explored high-dimensional equations, actor-critic networks, and multi-asset MM. The world of market-making is on the edge of a new era, one in which the union of science and the arts will pave the way to unprecedented prosperity. The line between fact and fantasy blurs with each new brilliant move, and the art of market-making reaches new heights. Let the search continue; perhaps we will find great riches in the financial markets thanks to the power of reinforcement learning.

Analogical Comparison

Market-making techniques from the past and those powered by DRL are compared and contrasted. The goal of this comparison is to highlight the advantages and disadvantages of each strategy by looking at issues including liquidity provision, risk management, and responsiveness to shifting market conditions. Gaper, et al. present a unique framework for market-making in their paper [9], which uses DRL to

overcome the shortcomings of prior methods. Combining the results of independent signal generators with a novel action space and reward function formulation, the DRL-based agent outperforms traditional market-making benchmarks in terms of reward-to-risk. When applied to real-world data, experiments reveal a startling 20-30% increase in terminal wealth with only roughly 60% of the inventory risks of conventional methods. In addition, the study places an emphasis on how the learned policy might be interpreted, thus shedding light on the agent's choice-making procedure. However, it is noted that there may be difficulties in real-time trading scenarios and that transaction cost considerations are not included in the suggested framework.

High-frequency market making, where an agent offers liquidity by quoting bid and ask prices on securities to profit from the spread, is discussed in depth by Kumar in [78]. The author notes that scholarly research of high-frequency market-making algorithms has been hampered by complications stemming from inventory risk, trading counterparties, and knowledge asymmetry. To fill this void, Kumar uses Deep Recurrent Q-Networks (DRQN) to create a high-frequency market-making agent and creates realistic simulations of limit order markets. The main contribution is showing that the suggested DRQN-based approach performs better than a well-known benchmark strategy using temporal-difference reinforcement learning. The DRQN agent has a higher market-making profit margin than the benchmark and consistently reproduces stylized facts from each simulation's historical trade data. Making change is an important aspect of any market's architecture, and this article explores how it affects both market quality and the agent's bottom line. The lack of real-world trade validation and potential difficulties in extending the proposed approach to dynamic, real-time trading scenarios are two shortcomings of the article that detract from its

otherwise strong points. Further study under varying market conditions and transaction cost concerns is required to evaluate the generalizability and robustness of the DRQN-based market-making agent.

Wan et al. [79], use an online decision technique based on DRL to tackle the basic maneuver confrontation problem of pursuit-evasion games in the multi-agent systems (MASs) domain. In order to achieve multi-agent cooperative decision-making, the authors reduce the typically complex modeling process by developing a control-oriented framework based on the multi-agent deep deterministic policy gradient (MADDPG) algorithm. The authors introduce adversarial disturbances and a new adversarial attack trick and adversarial learning MADDPG (A2-MADDPG) algorithm to address the difficulties caused by discrepancies between the model and real-world scenarios. To optimize robust training for the agents themselves, the introduction of adversarial attack tricks replicates uncertainty in the real world. To prepare for unknown dynamic changes in MASs, adversarial learning is implemented during training to preprocess the actions of numerous agents. The experimental results show the superior performance and effectiveness of the proposed technique for both the pursuer and the evader, who can then train inappropriate confrontational strategies. This paper's originality resides in its application of the MADDPG algorithm to the challenging problem of solving multi-agent pursuit-evasion games by incorporating adversarial disturbances and various forms of machine learning. To better equip agents to make decisions, the suggested A2-MADDPG algorithm takes advantage of uncertainty in real-world settings. One major benefit is that collaborative decision-making can be accomplished without resorting to too complicated modeling procedures. The paper does have a few flaws, though. It's possible that the testing results only apply to certain gaming scenarios or environments, despite the fact that

they show improved performance. To evaluate the generalizability and scalability of the suggested approach, additional experimentation is required across a wider range of circumstances and environments. It is also important to investigate how well the A2-MADDPG algorithm performs in real-time multi-agent pursuit-evasion games. The proposed method's actual usefulness and influence in real-world multi-agent systems will not be completely understood until these restrictions are removed.

Sun Yu [80] covers market-making strategy improvement for security market participants. Manually constructed strategies use market-based rules. Rule-based methods may not fully reflect the intricate relationships between market conditions and appropriate behaviors, resulting in inferior results. The author presents DRLMM, an end-to-end model to solve these restrictions. A long short-term memory (LSTM) network extracts temporal patterns from limit order books to better depict market circumstances. To control inventory risk and information asymmetry, the model learns state-action links via reinforcement learning. The suggested approach outperforms a traditional market-making baseline and a state-of-the-art market-making model on a six-month Shanghai Stock Exchange Level-2 data set. The DRLMM model outperforms benchmarks by 10.63% over ten equities. This study creates a comprehensive DRLMM model that surpasses limitations associated with rule-based strategies. LSTM directly learns from limit order book data to catch more detailed temporal patterns, improving decision-making. Reinforcement learning and a deep Q-network allow the agent to adaptively select action subsets based on inventory conditions, improving inventory risk and information asymmetry management. The paper has limitations. The Shanghai Stock Exchange dataset may not accurately represent other markets. The proposed approach should be tested in other markets and trading scenarios. The Deep Reinforcement Learning Market-Making model's real-

time trading implementation is also unexplored. Addressing these restrictions will help determine the model's usefulness and resilience in real-world market-making circumstances.

Gasparro et al. [27], propose a deep reinforcement learning-based controller for stochastic control of optimal market-making in quantitative finance. Market-making controls are taught using a weakly consistent, multivariate Hawkes process-based limit order book simulator. This work uses Monte Carlo back-testing to better examine and evaluate the suggested approach for weakly consistent limit order book models. The deep reinforcement learning controller outperforms numerous market-making benchmarks in risk-reward metrics, even with high transaction costs. This work introduces deep reinforcement learning to market-making under a Hawkes process-based limit order book model. The authors demonstrate Monte Carlo back testing's benefits by training the controller on the weakly consistent simulator to evaluate the proposed approach more accurately. The deep reinforcement learning controller outperforms benchmarks, suggesting it could improve quantitative finance market-making tactics. The paper has limitations. First, the proposed technique is tested on a weekly consistent limit order book model. The deep reinforcement learning controller's applicability to various markets and order book dynamics needs additional study. The study emphasizes its superior performance under transaction costs; however, it does not extensively examine how market volatility and liquidity affect the suggested strategy. Addressing these issues would help understand the deep reinforcement learning-based market-making controller's applicability and resilience in many market circumstances.

Yang Li et, al. [81], describe trading strategy formulation and feature extraction. None of the prior systems had a dynamic trading strategy or relied heavily

on domain expertise for customized features. The authors propose a DRL-based trading agent that can trade autonomously and profit in volatile financial markets to overcome these limits. They improve trading-specific value-based DQN and A3C algorithms. The function approximator uses stacked denoising autoencoders (SDAEs) and LSTM networks for strong market representation and financial time series dependence. The study also incorporates position-controlled action and n-step reward to improve the trading agent's real-world performance. Their trials reveal that their trading agent outperforms benchmarks and provides consistent risk-adjusted returns in stock and futures markets. The paper's flaw is its failure to explain how effectively its strategy applies to other financial markets outside stocks and futures. The proposed deep reinforcement learning approach may succeed depending on each financial market's dynamics. We need further trials to establish that the proposed trading agent can be deployed in several marketplaces.

Ye, et al. [82], offer a novel method of employing deep reinforcement learning in a high-fidelity simulation environment to make decisions about automated vehicle behavior. The authors use DRL algorithms to teach a robot how to drive an autonomous car. This research contributes by demonstrating how DRL can be used in high-stakes, high-complexity situations. Through simulations, we evaluate the trained agent's accuracy in navigating through traffic and responding to different scenarios where we want to maximize safety and efficiency. This research has the potential to improve progress toward creating autonomous cars with sound and flexible decision-making capacities. However, the difficulties in effectively simulating real-world traffic circumstances in simulations may limit this paper's ability to generalize the behavior of the trained agent to actual driving conditions.

Gaper, et al. [83], explore RL methods for optimal market-making. In order to

construct market-making strategies that ensure continuous liquidity while also effectively managing inventory risk, the authors study various RL algorithms, including deep reinforcement learning. The significance of this research comes in the depth with which it examines the effects of RL-based market-making procedures on the effectiveness and steadiness of such markets. Through a series of studies, the efficacy of RL-based market-making tactics is evaluated in comparison to more conventional methods. This research is important because it could lead to seeing a change in market-making techniques by making use of RL algorithms. However, this paper may be hindered in its real-time applicability in volatile market situations by the complexity and computational resources needed for training RL-based models on large-scale financial data.

Xu, et al. [84], use real tick data to study how well DRL performs for high-frequency market making. In order to quickly quote bid and ask prices and offer liquidity, the authors focus on creating a market-making agent that makes use of DRL algorithms. This work contributes by doing a thorough evaluation of DRL's efficacy and accuracy in high-frequency trading scenarios by applying it to actual tick data. Several risk-reward criteria, including profit measures, are used to assess the market-making agent's efficiency in capturing spreads and mitigating inventory risk. But, the practical profitability and viability of the proposed approach may be compromised by the omission of transaction costs, which are critical in high-frequency trading environments but were not considered in this study.

Guo, et al. [85], investigate the use of deep RL in market-making, with a special emphasis on employing limit order books as the key data source. The authors implement deep reinforcement learning techniques to create a risk-aware market maker that is able to maintain constant liquidity. This study contributes by providing

the first in-depth examination of the relationship between the efficiency and stability of markets and market-making tactics informed by deep reinforcement learning. Empirical experiments are used to assess the efficacy of the suggested method by comparing it to conventional market-making techniques. This research is important because combining deep reinforcement learning with order book data has the potential to improve market-making methods. However, the paper may be hampered in its real-time applicability in volatile market conditions by the complexity and computational resources needed to train deep reinforcement learning models on large-scale limit order book data. Further research into the interpretability and generalizability of market-making methods based on deep reinforcement learning might enrich this investigation.

Within the context of an order stacking framework, Chung, et al. [25], offer a unique deep reinforcement learning approach to market making. To create a market-making agent that can quote bid and ask prices while also monitoring inventory positions, the authors employ deep reinforcement learning techniques. This research contributes by optimizing market-making techniques and enhancing liquidity provision by combining deep reinforcement learning with the order stacking framework. The proposed method's efficacy is measured against standard market-making practices via an empirical review. This study's significance comes in the fact that it may pave the way for better market-making procedures by bringing deep reinforcement learning and order book dynamics together. One major shortcoming of this work is that it may be difficult to understand and explain the reasoning behind the deep reinforcement learning agent's decisions, which is particularly important in transparent financial markets. Further research on the model's resilience and sensitivity to various market conditions and transaction costs will also be beneficial to

the study.

Haider. et al [86], offer a machine learning-based strategy for predictive market-making. For the purpose of making the most informed business decisions possible in the future, the authors created a machine-learning model to forecast market trends. This research contributes by using machine learning in an innovative way to foresee market dynamics and improve liquidity availability. Empirical tests and performance evaluation in comparison to conventional market-making techniques are used to gauge the prediction model's efficacy. The value of this study resides in its potential to help market players make better, more data-driven decisions. However, the model's effectiveness and practical applicability may be hindered by the difficulty in precisely predicting extremely dynamic and fluctuating market conditions, which is a potential weakness of this research. Further investigation into the model's adaptability to new financial instruments and exchanges, as well as its sensitivity to other input variables, would strengthen the current study.

Jonathan Sidechain, [52], presents a DRLMM-tailored framework for Deep Reinforcement Learning in Market-Making. The author uses limit order book data and order flow arrival statistics to represent the observation space, and two cutting-edge policy gradient-based algorithms as agents to interact with that environment. In this study, a forward-feed neural network was used to approximate functions. In this study, we evaluate these agents using two distinct reward functions and compare their results. Daily and monthly average trading results are used to rank each agent and reward function combination. This research shows how deep reinforcement learning can help cryptocurrency market makers with the difficulties they experience with stochastic inventory control. However, the experiment's narrow focus on particular reward functions limits the paper's applicability. The intricacy and variation of actual

market conditions may not be captured, which could reduce the findings' applicability.

Market-making utilizing RL approaches is the topic of Jiang, et al. [87] paper on the Chinese commodity market. In order to efficiently optimize bid and ask prices in the commodity market and increase liquidity, the authors propose a market-making agent that makes use of RL algorithms to do so. This study contributes by applying RL to a niche financial sector, therefore shedding light on how well this technique works in practice. Simulations or empirical evaluation will most likely be used to gauge the RL-based market-makers precision in comparison to conventional approaches. This study's importance rests in the fact that it has the ability to improve commodity market decision-making in China by utilizing RL's adaptive decision-making abilities. The lack of peer review and substantial evaluation may be a weakness of this paper, calling for additional inspection and validation of the proposed approach through extensive experimentation and real-world deployment. The study might further benefit from expanding on the RL algorithms employed, hyperparameter tweaking, and the agent's effectiveness across a range of market and regulatory situations. Table 2 & 3 summarizes literature and analogical comparison.

Table 2: Analogical Comparison

Author Name	Algorithm Used	Contribution of Paper	Accuracy	Limitation of Paper
[9] Kasparov, Kostajnica	Deep Reinforcement Learning (DRL)	Novel DRL-based approach for market-making, incorporating signals for better decision-making.	Improved reward-to-risk performance.	Potential challenges in real-time trading scenarios and lack of consideration for transaction costs.
[78] Kumar	Deep Recurrent Q-Networks (DRQN)	Development of a realistic simulation for market-making using DRQN, outperforming benchmark strategies.	Superior performance compared to benchmarks.	The exclusion of real-time trading validation and potential challenges in dynamic trading environments.
[79] Wan, Hu	Model-Free Deep Reinforcement Learning	Design of a DRL-based framework for market-making with signals, achieving superior reward-to-risk ratios.	Higher terminal wealth and reduced risk.	Challenges in real-time trading scenarios and the impact of transaction costs are not extensively explored.
[80] Sun, Tian yuan, Dechunk Huang, and Jie Yu	DRL	Proposal of an end-to-end DRL-based market-making model, leveraging LSTM and deep Q-network.	Improved market-making performance.	Lack of consideration for transaction costs and limited exploration of different market conditions.
[27] Kasparov, Kostajnica	DRL, Neuroevolutionary, Adversarial RL	Introducing a DRL-based framework for MM with signals, addressing shortcomings of methods	Superior reward-to-risk performance.	Potential challenges in real-time trading and lack of exploration of transaction costs.

Author Name	Algorithm Used	Contribution of Paper	Accuracy	Limitation of Paper
[81] Yang Li; Walsham Zheng; Zibin Zheng	Deep Q-network (DQN) and Asynchronous Advantage Actor-Critic (A3C)	Proposes a novel trading agent based on deep reinforcement learning for algorithmic trading	Stable risk-adjusted returns in stock and futures markets	Limited discussion on generalizability to other financial markets
[82] Ye, Y., Zhang, X., & Sun, J.	DRL	Application of DRL to automated vehicle behavior decision-making in a high-fidelity simulation environment.	Improved decision-making and navigation.	Difficulty in replicating real-world traffic scenarios accurately in simulations.
[85] Guo, Lin, Huang	DRL from Limit Order Books	Exploration of market making with DRL using limit order book data, evaluating against conventional methods.	Enhanced market-making performance.	Complexity and computational resources required for training large-scale LOB data
[25]Chung, Chung, Lee, Kim	DRL	Deep RL approach for market making under order stacking framework, optimizing liquidity provision.	Improved market-making efficiency.	Potential challenges in explaining the decision-making process and sensitivity to market conditions.
[86] Haider, Wang, Scotney, Hawe	Machine Learning (ML)	ML-based predictive market-making, enhancing decision-making with data-driven strategies.	Enhanced market-making through predictive models.	Challenges in accurately predicting dynamic market conditions and scalability to various instruments.

Author Name	Algorithm Used	Contribution of Paper	Accuracy	Limitation of Paper
[83] Kasparov B.	DRL	DRL for market making with time-varying order arrival intensities, enhancing adaptability to market changes.	Improved market-making under varying conditions.	Need for extensive computational resources and potential challenges in real-time application.
[87] Jiang, Dierckx, Xiao	Reinforcement Learning (RL)	RL-based market making in the China commodity market, insights into RL effectiveness in trading scenarios.	Potential enhancement of market-making in the commodity market.	Limited details on RL algorithms used and lack of peer review and validation.
[52] Jonathan Sidechain	Advanced policy gradient-based algorithms	Framework for DRLMM in cryptocurrency market making	Daily and average trade returns	The limited scope of experiment on specific reward functions.
[84] Xu, Ziyi, Cheng, He	Reinforcement Learning (RL)	RL for high-frequency market-making on actual tick data, insights into RL performance in real-world trading.	Potential enhancement of market-making in high-frequency scenarios.	Limited information on accuracy and validation of RL-based approach.

Table 3: Literature Summary Based on Deep Learning

Author Name	Algorithm Used	Contribution of Paper	Accuracy	Limitation of Paper
[88] M. Elwin. et al	Deep Neural Networks (DNN)	Development of a strong market-making model using DNN, incorporating market signals.	Improved market-making performance.	Lack of real-time validation and potential challenges in dynamic trading scenarios.
[89] F. McGroarty. et al	Convolutional Neural Networks (CNN)	Application of CNN in market-making strategies for better signal processing.	Enhanced signal extraction and decision-making.	Computational complexity and resource requirements for large-scale data.
[90] B. Ning. et al	Recurrent Neural Networks (RNN)	RNN-based market-making model to capture temporal patterns in financial data.	Better prediction of market dynamics.	Difficulty in interpreting RNN decisions and potential overfitting risks.
[91] T. Spooner. et al	Long Short-Term Memory (LSTM)	LSTM implementation in market-making agents, addressing volatility dynamics.	Improved adaptation to market changes.	Limited exploration of LSTM hyperparameter tuning and sensitivity to different market regimes.
[92] H. Wei. et al	Generative Adversarial Networks (GAN)	GANs for generating synthetic market data to augment training datasets.	Enhanced model robustness and generalization.	Challenges in ensuring the generated data's fidelity to real market conditions.
[78] P. Kumar. et al	Deep Reinforcement Learning	Applied deep RL for high-frequency market making.	Achieved competitive results.	Limited analysis in highly volatile and low-liquidity markets.

Author Name	Algorithm Used	Contribution of Paper	Accuracy	Limitation of Paper
[93] S. Ganesh. et al	DRL	DRL-based market-making strategy, incorporating multiple signals.	Superior reward-to-risk performance.	Difficulty in interpreting DRL-based decisions and challenges in real-time deployment.
[85] H. Gues. et al	Deep Reinforcement Learning	Developed a market-making strategy using limit order book data.	Demonstrated promising results.	May lacks robustness in rapidly changing market conditions.
[94] M. Dixon & I. Halperin	Attention Mechanisms	Attention mechanisms for market-making to focus on relevant data components.	Improved model interpretability and performance.	Complexity in parameter tuning for attention mechanisms and potential scalability issues.
[29] V. Singh. et al	Ensemble Learning	Ensemble techniques for combining market-making strategies from multiple models.	Enhanced model robustness and performance.	Increased computational requirements for ensemble learning and potential model correlation issues.

Literature Review for DRL in MM

Liquidity, price discovery, and general market efficiency are all greatly aided by market-making's presence in the financial markets. Recent breakthroughs in artificial intelligence and machine learning, notably DRL, have opened up new possibilities for better, more adaptive market-making systems, which traditionally have depended on heuristic rules and statistical models. Focusing on the benefits, drawbacks, and prospective effects of this new paradigm, this literature review looks at the present research and state-of-the-art developments in applying DRL to market-making.

A trading strategy known as deep reinforcement learning for market-making

trains agents to quote prices on financial markets using artificial intelligence methods. Investigate the Dueling Double Deep Q-Network (D3QN) and a unique reward function in particular to create market-making agents that can reliably, flexibly, and fully automatically balance profit and inventory [84]. The agents are tested and trained using actual stock tick data, creating an environment that is quite realistic. The D3QN is employed in the article to create market-making agents that can balance profit and inventory in a strong, adaptable, and fully autonomous manner [84]. The data utilized is the tick data for the stock 000333.XSHE for the 100 trading days from Aug 06, 2020, to Dec 31, 2020. A 64/16/20 split is used to separate the data into training, validation, and testing sets. The best bid and ask prices, as well as the completed quantities at each price level, are included in the tick data. The author trained and assessed market-making agents utilizing this data using deep reinforcement learning.

When compared to Simple Rule-Based (SRB) agents, DRL agents obtain a considerably lower unfavorable selection ratio [25]. Additionally, they are able to locate more execution possibilities, most likely by using queue position data that goes beyond the optimal pricing level. Because DRL agents are able to learn from the past and modify their approaches as necessary, they are able to react to shifting market conditions and gradually increase their performance. Guhya Chung, et al [25], evaluated two SRB agents and three DRL agents. The DRL agents employ a deep reinforcement learning technique to learn from the past and modify their strategies, while the SRB agents apply a straightforward logic to reduce inventory risk. The report also provides a zero-intelligence agent as a reference point for comparison. Market making, order stacking, and deep reinforcement learning are some of the approaches employed in the article. Practitioners quote limit orders at a variety of

price levels above the optimum limit price using the order stacking structure. A deep reinforcement learning model for market making within the order stacking framework uses a modified state representation to effectively encode the queue positions of the resting limit orders. Furthermore, a comprehensive ablation study is carried out to demonstrate that deep reinforcement learning can be effectively employed to enhance profit and loss (Pl.) while mitigating various risks within the context of a market-making agent operating under the order stacking framework. Generally, the order stacking framework empowers market makers to manage risks by enabling them to quote limit orders at multiple price levels beyond the best limit price. This strategy facilitates the capture of more trading opportunities while reducing the likelihood of non-execution. Moreover, the proposed deep reinforcement learning model incorporates a modified state representation that efficiently encodes the queue positions of resting limit orders. This modification aids market makers in effectively handling inventory risk and adverse selection risk.

Sun T, et al. [80], the capacity to optimize market-making techniques in a more effective and efficient way is one of the possible advantages of utilizing deep reinforcement learning for market-making. Traditional tactics are mostly created manually, and orders are automatically placed in accordance with regulations based on predetermined market circumstances. Rule-based strategies, on the other hand, cannot accurately reflect relationships between the market circumstances and the strategies' activities. Market conditions cannot, therefore, be properly represented by arbitrarily specified indicators. By utilizing deep reinforcement learning, the model may develop a better mapping between strategy states and actions and take wiser decisions to increase revenues and decrease risks. The suggested DRL in the MM model uses reinforcement learning to learn state-action linkages and an LSTM

network to directly extract market temporal patterns from LOBs. A deep Q-network is used to adaptively pick various action subsets and train the market-making agent in accordance with the inventory states in order to manage inventory risk and information asymmetry. The experiment findings demonstrate that the suggested strategy beats the benchmarks across 10 equities by at least 10.63%.

Kasparov B et al., a multivariate, weakly consistent model called the Hawkes process-based limit order book model is used to simulate the dynamics of limit order books in financial markets [27]. By regularly purchasing and selling securities, market making is the act of supplying liquidity to the financial markets. The Hawkes process-based limit order book model is used to simulate the dynamics of the order book, which is a crucial aspect of market-making. Even with high transaction costs, the suggested deep reinforcement learning-based market-making controller surpasses several established market-making benchmarks in terms of several risk-reward criteria. The authors find that their technique outperforms more standard market-making tactics in terms of profitability, volatility, and other parameters when compared to those strategies. These classic market-making strategies include the bid-ask spread, the mid-price, and the order book imbalance. The findings demonstrate that, in comparison to the conventional benchmarks, the deep reinforcement learning-based controller achieves a much higher mean PNL value as well as a more favorable Sharpe ratio. The PNL distribution's percentiles show that it also performs strongly. The deep reinforcement learning-based controller, according to scientists, displays beneficial characteristics for risk management, such as smaller tails and a lower kurtosis value for its PNL distribution. Overall, the findings imply that the suggested strategy is a good technique for best market-making in limit order book models with weak consistency.

Existing Study

The table provides an overview of various studies and their approaches in algorithmic trading. Each study discusses the author's name, the type of data used, the action space for trading decisions, the state space representing market conditions, the reward structure, the algorithm employed, evaluation metrics, and benchmark strategies. These studies explore different aspects of trading algorithms, including market quality, profitability, inventory management, and the use of various reinforcement learning techniques for strong market decision-making using DRL.

Table 4: Comparison of Existing Study and their Approaches

Author Name	Data	Action Space	State Space	Contribution of Paper	Algorithm	Evaluation Metric	Benchmarks
Haider, A., et al. [86]	CPE-based market data	Reinforcement learning agents	Market state variables	Predictive market making with RL and price predictor	Reinforcement Learning (RL)	Liquidity, returns	Traditional and RL methods
Xu, Z., et al. [84]	Actual tick data	Dueling Double Deep Q Network (D3QN) agents	Market state variables	Evaluating high-frequency market making with D3QN agents	Deep RL	Market quality	Competing market maker
FALCES Marin, et al. [95]	Bitcoin-dollar trade data	Deep reinforcement learning agents	Market state parameters	Enhancing the Avellaneda-Stoichkov market-making algorithm with DRL	Deep RL	Risk reduction	Baseline models
Chung, G., et al. [25]	KOSPI200 Index Futures data	Deep reinforcement learning model	Order stacking framework	Market making under order stacking framework using DRL	Deep RL	Profit enhancement, risk mitigation	N/A

Author Name	Data	Action Space	State Space	Contribution of Paper	Algorithm	Evaluation Metric	Benchmarks
Singh, V., et al. [29]	Big data in financial industries	Not specified	Not specified	Review and research agenda on RL and DL algorithms for decision-making in financial industries.	N/A	N/A	N/A
Sun, T., et al. [80]	Not specified	Deep Reinforcement Learning Market Making	Limit order book data	End-to-end DRL model for market making in the Shanghai Stock Exchange	Deep RL	Market quality	Traditional approaches
Ber gault, P. et al., [96]	Not specified	Closed-form approximations	Not specified	Closed-form approximations in multi-asset market-making	N/A	N/A	N/A
Kasparov, B. et al.[83]	Not specified	Reinforcement learning agents	Not specified	Reinforcement learning approaches to optimal market-making	Reinforcement Learning (RL)	N/A	Standard analytical models

Author Name	Data	Action Space	State Space	Contribution of Paper	Algorithm	Evaluation Metric	Benchmarks
Zhao, M., et al. [97]	Not specified	Deep reinforcement learning agents	Not specified	High-frequency market making with risk control using RL	Deep RL	Risk Control	Standard analytical models
B. Kasparov, et al. [27]	Hawkes process-based limit order book model	Deep reinforcement learning	Limit order book data	DRL for market making under a Hawkes process-based limit order book model.	Deep RL	Risk-reward criteria	Traditional market-making
Chan, et al. [98]	Agent-based modelling	Ask price or bid changes less than infinity	Market quality measurement, order imbalance, and inventory	An Electronic Market-Maker	SARSA (State-Action-Reward-State-Action) and Monte Carlo,	Inventory, PNL, spread, price deviation	-
Spooner, et. al. [99]	Historical	Bid/ask quote pairs and a market order	Agent and market state variables	Market Making via Reinforcement Learning	Q-learning, SARSA, and R-learning variants	Normalized PNL, MAP (mean absolute position), mean reward	Fixed offset and the online learning approach from

Author Name	Data	Action Space	State Space	Contribution of Paper	Algorithm	Evaluation Metric	Benchmarks
Kim, et. al. [100]	Simulation -based on Historical data	Bid/ask price and size changes	Spread between the agent's bid and ask and the best bid and ask, bid size, stock on hand, and the number of buy orders at or below the agent's ask price.	Modeling Stock Order Flows and Learning Market-Making from Data	SARSA, Actor critics	PnL	-
Lim, et al. [101]	Simulated (LOB model from	Bid/ask quote pairs	Inventory, time	Reinforcement Learning for High-Frequency Market Making	Q-learning	PNL, inventory	Fixed (zero-tick) offset, AS approximations, random strategy
Patel, et al. [31]	Historical	buy, sell, or hold, and quote	pricing in the past, indicators of the current market, and a list of available assets. timing, remaining stock, and market factors	Optimizing Market Making using Multi-Agent Reinforcement Learning	DDQN DQN	PNL	Speculations based on momentum or buy-and-hold strategies

Author Name	Data	Action Space	State Space	Contribution of Paper	Algorithm	Evaluation Metric	Benchmarks
Haider, et al. [102]	Historical	ask quote pairs or Bid	Bid/ask quote pairs	Gaussian Based Non-linear Function Approximation for Reinforcement Learning	Profit and loss statement tweaked to include inventory turnover costs and market volatility and spreads	PNL	Spooner et al.'s benchmark, with market volatility adjusted for.
Ganesh, et al. [93]	Simulation based on Historical data	Quantities to stream, percentages of stock to acquire/dispose of.	Trades previously executed, inventory, Mid price, & spread curves, market share	Reinforcement Learning for Market Making in a Multi-agent Dealer Market	PPO with a shortened lens	Profit and loss, Total Reward, Inventory, Hedged Cost	Agent MM that is random, resilient, and flexible
Baldacci, et al. [103]	Simulated	Trading volumes on the ask and bid (∞)	Principal incentives, inventory	Market making and incentives design in the presence of a dark pool: a DRL approach	Actor–critic-like	-	-

Haidet, et al [86], "Predictive Market Making via Machine Learning" present the concept of Predictive Market Making (PMM), which involves the integration of market-making agents based on reinforcement learning with a price predictor based on deep neural networks. The Price Matching Model (PMM) use the consolidated price equation (CPE) in order to provide quotations that encompass both present prices and anticipated future fluctuations. The performance of PMM in boosting market liquidity and returns is found to be superior when compared to traditional and RL-based market-making approaches through a comparative evaluation conducted on various equities and Exchange-Traded Funds (ETFs) in out-of-sample back sting. The algorithm used Machine Learning with limitation lack of specific algorithm details, potential overfitting issues. However, we can improve it by provide more algorithm details and address potential overfitting through better regularization techniques. Xu, et al [84], titled "Performance of Deep Reinforcement Learning for High Frequency Market Making on Actual Tick Data" want to examine the effectiveness of high-frequency market-making tactics by employing Dueling Double Deep Q Network (D3QN) agents together with a unique reward function. The agents are trained and tested in a realistic trading environment using authentic tick data. Furthermore, the researchers investigate the adaptability of the agent when competing against a market maker that has been specifically created for this purpose. They emphasize how the D3QN agents are able to learn and improve their quoting strategies in order to increase the likelihood of successful transactions. The present study additionally evaluates the influence of high-frequency market-making on market quality in both single-agent and double-agent scenarios. They used deep reinforcement learning, that has limited exploration of variations in deep reinforcement learning methods, lack of robustness testing. Where we can explore various deep reinforcement learning

methods and conduct extensive robustness testing to improve it. FALCES Marin, et al. [95], "A Reinforcement Learning Approach to Enhance the Performance of the Avellaneda-Stoichkov Market-Making Algorithm" investigate the utilization of deep reinforcement learning techniques in the domain of market making. Instead of directly determining bid and ask prices, the approach employed involves utilizing neural network outputs to modify risk aversion parameters and the result of the Avellaneda-Stoichkov technique to minimize the risk associated with inventory. Significantly, the authors optimize the initial parameters through the utilization of a genetic algorithm and utilize a random forest methodology to choose attributes that define the state. The use of genuine bitcoin-dollar trade data for back testing purposes showcases the notable effectiveness of their methodology. Gen-AS exhibits superior performance compared to the baseline models, while the Alpha-AS models indicate exceptional proficiency in many important metrics. Nevertheless, the research also brings attention to apprehensions regarding region-specific instances of heightened risk-taking by Alpha-AS models, hence instigating deliberations on prospective remedies. This paper used reinforcement learning algorithms which need to provide more insights into the limitations encountered and propose ways to overcome them. Chung, et al. [25], delve into the domain of market making strategy in high-frequency trading. While previous studies have mainly focused on inventory risk, this paper addresses the critical aspects of adverse selection risk and non-execution risk, which are essential for stable profit in competitive markets. They propose a deep reinforcement learning model specifically tailored for market making under the order stacking framework, efficiently encoding queue positions of resting limit orders. Through comprehensive experiments on KOSPI200 Index Futures data, the model showcases its ability to enhance profit while mitigating various risks. This study fills a gap in

existing research by incorporating the order stacking framework into market making strategies. The author used deep reinforcement learning but limited explanation of the approach's applicability to real-world scenarios. Singh, et al. [29], investigate the utilization of deep reinforcement learning techniques in the domain of market making. Instead of directly determining bid and ask prices, the approach employed involves utilizing neural network outputs to modify risk aversion parameters and the result of the Avellaneda-Stoichkov technique to minimize the risk associated with inventory. Significantly, the authors optimize the initial parameters through the utilization of a genetic algorithm and utilize a random forest methodology to choose attributes that define the state. The use of genuine bitcoin-dollar trade data for back testing purposes showcases the notable effectiveness of their methodology. Gen-AS exhibits superior performance compared to the baseline models, while the Alpha-AS models indicate exceptional proficiency in many important metrics. Nevertheless, the research also brings attention to apprehensions regarding region-specific instances of heightened risk-taking by Alpha-AS models, hence instigating deliberations on prospective remedies. Sun, et al. [80], address the critical issue of optimizing market making strategies in security markets. They emphasize that traditional manual strategies based on predefined rules struggle to effectively represent complex market conditions and their relations to strategy actions. To overcome these limitations, the paper introduces an end-to-end deep reinforcement learning model called Deep Reinforcement Learning Market Making. This model leverages deep Q-networks and long short-term memory networks to extract temporal patterns from limit order books, enabling adaptive strategy adjustments based on inventory states. Experimental results on a Level-2 dataset from the Shanghai Stock Exchange demonstrate its superiority over conventional and state-of-the-art market-making approaches. The limitation of the

paper is the potential issues with computational efficiency and scalability. Address computational efficiency and scalability issues for real-time applications to improve it. Philippe Ber gault, et al. [96], delves into the complexities associated with market making models, specifically focusing on the expansion of the Avellaneda-Stoichkov model to encompass multiple assets. The authors put forward closed-form approximations for value functions in multi-asset models, which have a wide range of applications including heuristic evaluation functions, initial values for reinforcement learning, and designing quotation strategies. This paper introduces novel and comprehensible closed-form approximations for optimal quotes in finite-horizon and asymptotic scenarios, thereby improving the comprehension and applicability of multi-asset market making methods. The limitation of the paper limited to closed-form approximations, may not handle complex scenarios, which used closed-form approximations algorithm. However, explore ways to adapt to more complex market conditions and scenarios can improve the paper. The study conducted by Kasparov, et al. [83] investigates the utilization of reinforcement learning techniques for the purpose of achieving optimal market making strategies. Market making is a trading strategy that entails the placement of limit orders on both the buy and sell sides of the order book. The primary objectives of this strategy are to enhance market liquidity and produce profits. The paper emphasizes that reinforcement learning, specifically deep reinforcement learning, has garnered substantial attention in this discipline owing to its achievements across diverse domains. The main objective of this study is to provide a thorough and up-to-date examination of the cutting-edge applications of reinforcement learning in the context of optimal market making. The findings of the investigation indicate that reinforcement learning techniques frequently exhibit superior performance in terms of risk-adjusted returns when compared to standard

analytical models, hence showcasing their efficacy within this particular field. In the study conducted by Kasparov, et al. as documented in their publication [97], the authors delve into the utilization of reinforcement learning techniques within the domain of optimal market making. Market making is a trading strategy that entails the placement of limit orders on both the buy and sell sides of the order book. The primary objectives of market making are to enhance market liquidity and produce profits. The study emphasizes the growing prominence of reinforcement learning, namely deep reinforcement learning, within this discipline as a result of its achievements across several areas. The main objective of this study is to provide a thorough and up-to-date examination of the cutting-edge applications of reinforcement learning in the context of optimal market making. The findings of the investigation indicate that reinforcement learning techniques frequently exhibit superior performance in terms of risk-adjusted returns when compared to standard analytical models, hence highlighting their efficacy within this particular field. Optimal market making is a stochastic control problem in quantitative finance, and the article [27]"Deep Reinforcement Learning for Market Making Under a Hawkes Process-Based Limit Order Book Model" addresses this topic in depth. In this research, we introduce a deep RL-based controller that has been trained on a limit order book simulator based on a multivariate Hawkes process. The framework of Monte Carlo back testing and weakly consistent limit order book models are used in this study of market making methods. Several risk-reward criteria show that the deep RL controller outperforms many traditional market making benchmarks. This is true even while taking into account the relatively high transaction expenses [27]. The paper by Chan, et al. [98], utilizes agent-based modeling to investigate waste management practices in the textile sector. It explores how changes in ask prices or

bids impact market quality and inventory, using SARSA and Monte Carlo algorithms to optimize profits while considering inventory and quality discounts. Their study evaluates inventory, profit and loss (PNL), spread, and price deviation. Spooner, et al. [99], present a historical analysis, employing bid/ask quote pairs and market orders to study market-making. They focus on custom PNL with a running inventory penalty and use Q-learning, SARSA, and R-learning variants with linear combinations of tile coding's as function approximators. Their evaluation metrics include normalized PNL, mean reward, and MAP, while benchmarks involve fixed offsets and online learning approaches. Kim, et al. [100], conduct simulation-based research, examining bid/ask price and size changes to understand market behavior. Their work emphasizes optimizing PNL through SARSA and Actor critics algorithms, with FFNN as function approximators. Evaluation metrics are centered around PNL, and they investigate bid-ask spreads and market dynamics. Lim, et al. [101], focus on simulated data from a Limit Order Book (LOB) model, studying bid/ask quote pairs. They create tailored PNL functions that incorporate inventory carrying costs and CARA-based terminal applications. Q-learning is utilized, with evaluation metrics including PNL and inventory. They benchmark their findings against fixed offsets, AS approximations, and random strategies. Patel's [31], historical analysis involves buy, sell, or hold decisions along with quotes and market indicators. They adopt custom PNL-based rewards using DDQN and DQN algorithms. Their study explores PNL, analyzing trading volumes and market factors, while referencing momentum and buy-and-hold strategies as benchmarks. Haider, et al. [102], delve into historical data, investigating ask quote pairs or bid actions. Their approach considers inventory, bid/ask levels, book imbalance, strength volatility index, and market sentiment. They adapt SARSA algorithm and adjust the profit and loss statement to include inventory turnover and

market volatility costs. Evaluation metrics encompass PNL, and they adjust benchmarks based on market volatility. Ganesh, et al. [93], employ simulation-based analysis using historical data, focusing on quantities, percentages, and market share. Their research emphasizes profit and loss statements with inventory variation penalties, employing PPO with a shortened lens and Feed-Forward Neural Network (FFNN). Evaluation metrics include profit and loss, Total Reward, Inventory, and Hedged Cost, while they benchmark their findings against a resilient and flexible agent market maker. Baldacci, et al. [103], utilizes simulated trading volumes on ask and bid to study principal incentives and inventory. Their research incorporates a CARA-based reward system with Actor–critic-like algorithms and FFNN function approximators. While specific evaluation metrics are not mentioned, their study contributes insights into market-making strategies.

CHAPTER 5: ALGORITHM AND METHODOLOGY

Overview

The experiment conducted in this Bitcoin trading project involved using machine learning models to develop trading strategies and evaluating their performance based on historical data. The models were trained and tested using different subsets of data, and their effectiveness was measured using several key metrics: Spread Capture Ratio, Market Impact, and Profitability.

Exponential Moving Averages (EMA)

The Exponential Moving Average (EMA) is calculated for the 'close' price of Bitcoin. EMA is a type of moving average that places a greater weight and significance on the most recent data points. The formula for EMA is:

$$EMA = (V_t s/(1+d)) + EMA_{t-1}(1 - s/(1+d)) \quad (1)$$

Where: EMA is the EMA at time t, V is the value at time t, s is the smoothing factor, d is the number of days.

Volatility Calculation

Volatility is calculated as the ratio of the difference between the high and low prices to the closing price:

$$\text{Volatility} = (\text{High} - \text{Low}) / \text{Close} \quad (\mathbf{2})$$

Proximal Policy Optimization (PPO)

PPO optimizes a surrogate objective function that balances exploiting what the model already knows with exploring new actions that might yield higher rewards. The key formula for the PPO objective function is:

$$L^{CLIP}() = E_t[min(r_t(\theta)A_t, clip(r_t(\theta), 1-, 1+)A_t) \quad (3)$$

Where: r_t(θ) is the probability ratio π_θ(a_t|s_t) / π_θ_old(a_t|s_t), the ratio of the new policy to the old policy. Â_t is an estimator of the advantage function at time t. ε is a hyperparameter, typically small (e.g., 0.1 or 0.2), used for clipping. clip(r_t(θ), 1 - ε, 1 + ε) function ensures that the ratio r_t(θ) stays within the range of [1-ε, 1+ε].

Advantage Actor-Critic (A2C)

In A2C, the actor updates the policy based on the advice of the critic. The critic evaluates the action values. The update rule for the actor in an A2C algorithm can be described as follows:

$$\Delta\theta = \alpha\theta log\pi\theta(a_t|s_t)A(s_t, a_t) \quad (4)$$

Where: Δθ is the change in the policy parameters, α is the learning rate, ∇θ log πθ (a_t | s_t) is the gradient of the logarithm of the policy, A (ST, AT) is the advantage function, which measures the benefit of taking action a in state s over

following the current policy.

The critic's update rule, usually a value function estimator, is often trained to minimize some form of mean squared error:

$$L = E[(R_t + V(s_t + 1) - V(s_t))^2] \quad (5)$$

Where: R_t is the reward at time t, γ is the discount factor, V(s) is the value function estimation of states.

Deep Q-Network (DQN)

The DQN algorithm aims to find the optimal action-value function, which is the maximum expected return achievable after observing some state s and then taking some action a, following policy π. The core of DQN is the Bellman equation:

$$Q(s, a) = E[R(s, a) + max_{a'} Q(s', a')] \quad (6)$$

Where: Q (s, a) is the optimal action-value function, R (s, a) is the reward received after taking action an in states, γ is the discount factor, max_{a'} Q (s', a') is the maximum sum of rewards achievable after taking the next action an' in the new state's'.

In DQN, a neural network is used to approximate the Q-function. The network is trained to minimize the loss:

$$L(\theta) = E[(yQ(s, a; \theta))^2] \quad (7)$$

Where: ($y = R(s,a) + \gamma \max_{a'}Q(s', a'; \theta^-)$) is the target Q-value, with θ^- representing the parameters of a target network, $Q(s,a;\theta)$ is the predicted Q-value from the main network with parameters θ, $L(\theta)$ is the loss function, typically Mean Squared Error (MSE), that the training process aims to minimize.

Evaluation Metrics

Spread Capture Ratio:

Spread Capture Ratio = Average Profit per Trade / Average Price Spread

Market Impact:

Market Impact = Total Profit Loss / Number of Trades

Profitability:

Profitability = Total Profit Loss

Data Preprocessing

Data Loading and Parsing: The first step in data preprocessing is loading the Bitcoin (BTC) ticker data, typically in CSV format. This data is rich in historical price information, including open, high, low, and close values, as well as trading volumes. An essential aspect of this process is parsing the dates, converting them into a standardized Date Time format. This conversion is crucial for time-series analysis, allowing the data to be sorted and analyzed chronologically.

Data Filtering and Cleaning: Considering the dataset might contain various cryptocurrencies, it's filtered to focus exclusively on Bitcoin. This step is vital to

ensure the analysis and subsequent modeling are relevant to the project's scope. Additionally, the dataset is examined for missing or null values. Addressing these gaps is critical for maintaining data integrity. Depending on the situation, missing data can be filled with interpolated values or, if necessary, removed.

Feature Engineering: To enhance the dataset, new features are derived. This includes calculating Exponential Moving Averages (EMAs) for different periods, offering insights into trends and momentum. Another critical feature is volatility, computed to understand the extent of price fluctuations, a characteristic feature of cryptocurrency markets.

Data Exploration:

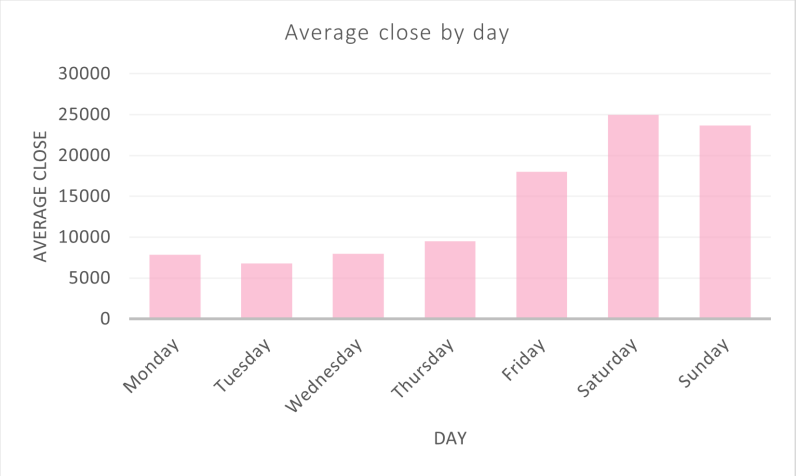

Figure 8: Bar Plot - "Average close by day"

The bar chart in Figure 8 represents the average closing price categorized by the days of the week. The x-axis indicates the day, and the y-axis indicates the average closing price. Such a plot can be useful to detect patterns or trends on different days, which might be important for trading strategies that capitalize on daily fluctuations.

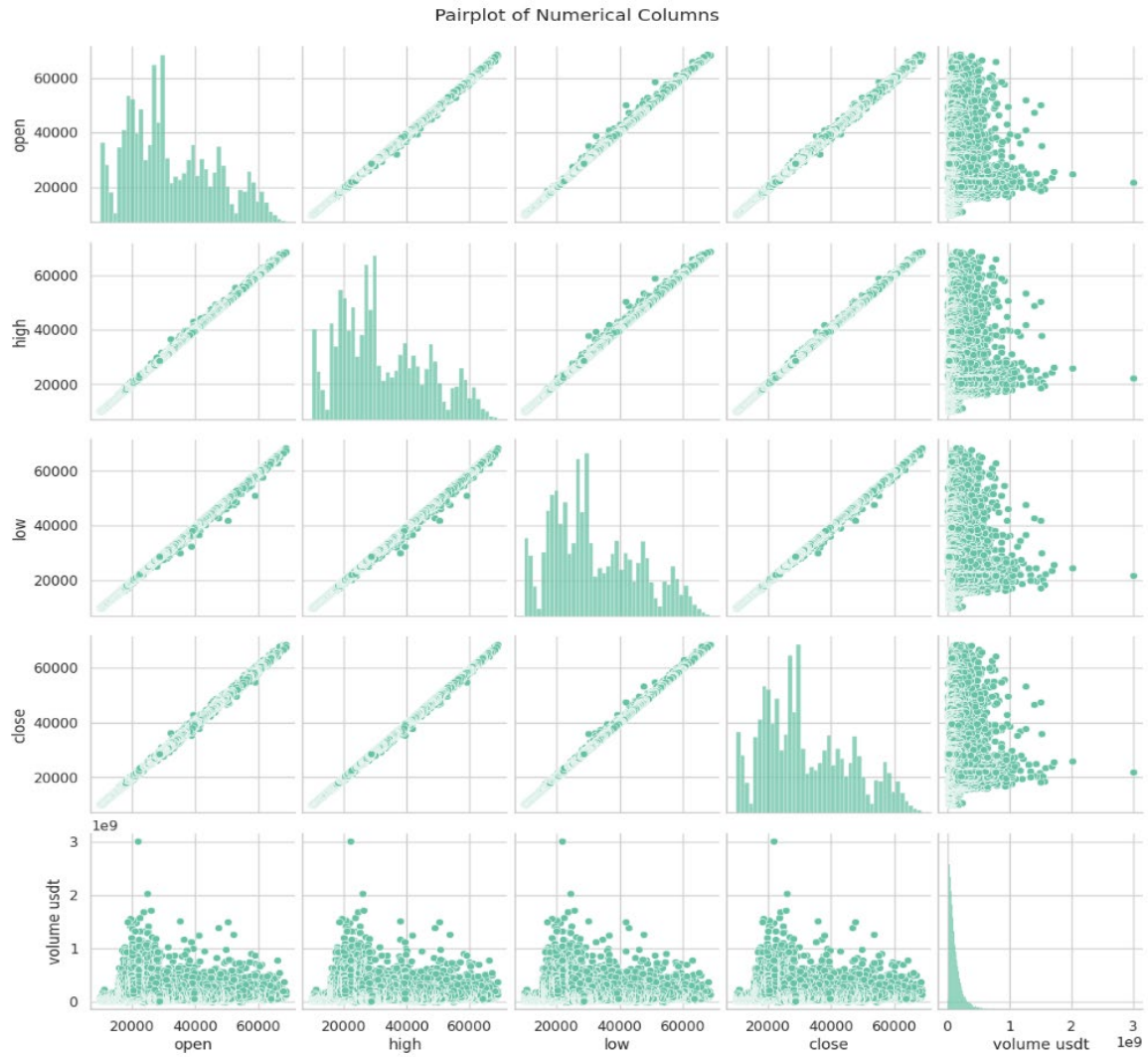

Figure 9: Pair Plot - "Pair plot of Numerical Columns"

In Figure 9 the grid of plots is known as a pair plot or a scatterplot matrix, and it's used to understand the relationship between different numerical variables in the data. For each pair of variables, it displays a scatter plot to visualize correlations or a histogram to show the distribution if the variables are the same. The variables here are "open", "high", "low", "close", and "volumes", which are typical in financial datasets representing various price points and the traded volume in US dollars.

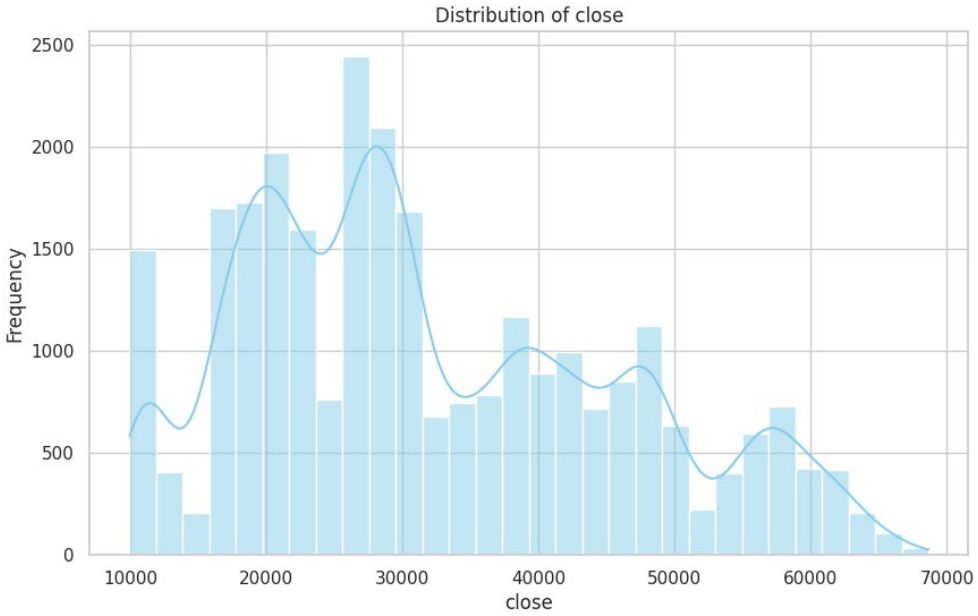

Figure 10: Histogram and Density Plot - "Distribution of close"

This plot in Figure 10 shows a histogram overlaid with a kernel density estimate. It displays the distribution of a variable labeled "close", which is common terminology for the closing price of a stock or asset for the trading day. The x-axis represents the closing price, and the y-axis represents the frequency of those prices. The shape of the distribution and the density line give an idea about the central tendency, variability, and the skewness of the closing prices.

Developing Market-Making

Market-making in this project involves creating strategies for buying and selling Bitcoin, aiming to capitalize on the spread—the difference between the buy and sell prices. The approach is grounded in analyzing the processed data, identifying potential entry (buy) and exit (sell) points, and strategically managing Bitcoin inventory to maximize profitability.

Environment Diagram

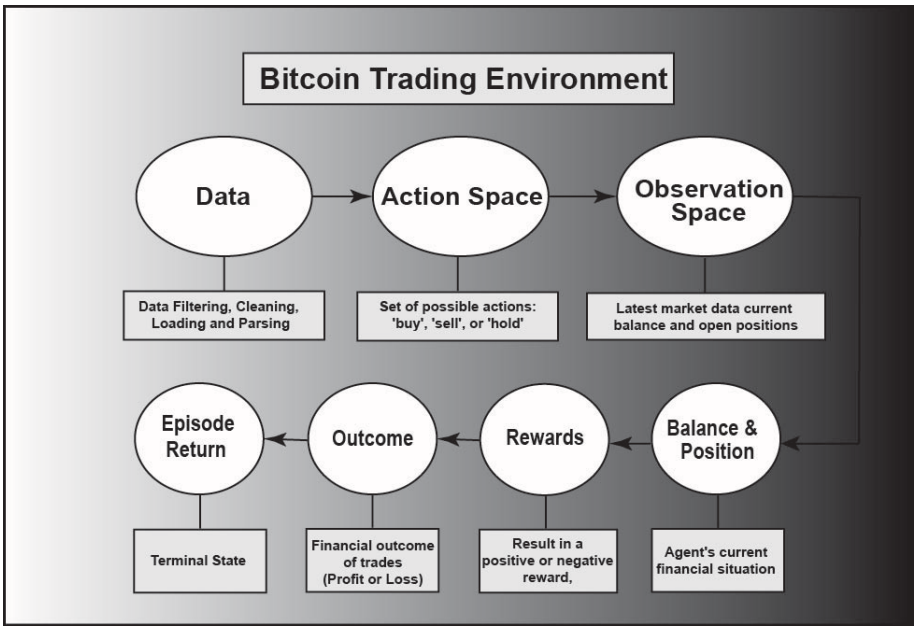

Figure 11: Bitcoin Trading Environment Structure

This Diagram in Figure 11 represents a conceptual flowchart for a Bitcoin trading environment, likely designed for a reinforcement learning (RL) setup. Here's a detailed explanation of each component in the diagram:

Data: This is the raw market data that the system uses to make decisions. It could include price data like open, high, low, close, and volume for Bitcoin.

Process Market Data: The raw data is processed to a format that is suitable for the trading algorithm. This could involve normalization, calculation of technical indicators, or extraction of features that the algorithm can use to make trading decisions.

Action Space: This is the set of possible actions that the trading agent can take at any given step. In the context of trading, this could be 'bought', 'sell', or 'hold'.

Observation Space: After taking an action, the agent observes the new state of the environment. This could include the latest market data as well as the agent's

current balance and open positions.

Balance & Position: This represents the agent's current financial situation - the balance of funds available for trading and any open positions in the market.

Updates Balance & Position: Based on the action taken, the agent's balance and position are updated. For example, if the agent decides to buy, the balance will decrease, and it will have an open long position.

Rewards: The agent receives rewards based on the actions it takes. The reward structure is crucial in reinforcement learning as it guides the agent in learning the best actions to maximize rewards over time.

Outcome: This is the result of the agent's actions in terms of profit and loss. It is directly linked to the rewards but represents the financial outcome of trades.

Determine Rewards: This step involves translating the outcome into rewards that the agent can learn from. For instance, a profitable trade might result in a positive reward, while a loss might result in a negative reward.

Episode Return**:** In RL, an episode is a sequence of steps that ends in a terminal state, such as reaching the end of a time period. The episode return is the cumulative reward that the agent has obtained in a single episode. This is used to evaluate the agent's performance.

Environment Setting and State Engineering

Action Space:

The action space in this context refers to the set of all possible actions that the trading agent can perform. In the Bitcoin trading project, these actions are primarily buying, selling, or holding Bitcoin. Each action has distinct consequences:

- Buying indicates acquiring Bitcoin, which might be beneficial if the price

increases.

- Selling involves disposing of Bitcoin, aiming to realize a profit or prevent a loss.
- Holding is maintaining the current Bitcoin position, reflecting a wait-and-see strategy.

This diverse action space allows the agent to navigate through various market conditions and adapt its strategy based on the evolving market dynamics.

Trading Environment Setup:

The trading environment is a simulation of the real-world Bitcoin market. It's designed to provide the agent with a realistic and dynamic setting where various market conditions are simulated. The environment feeds real-time or historical market data to the agent, which includes prices, volumes, and other relevant financial indicators. The agent's decisions lead to changes in the environment, which in turn provide feedback in the form of rewards or penalties.

Environment Initialization:

Both env and ENV_ TEST instances of Bitcoin Trading ENV, a custom class designed to simulate Bitcoin trading scenarios. These environments provide the necessary market data and dynamics for training and testing the agents.

Agent Implementation:

The agent in this setting is a machine-learning model or algorithm capable of making autonomous trading decisions. The agent analyzes the market data, learns

patterns and relationships, and makes predictions or decisions about buying, selling, or holding Bitcoin. Over time, through continuous interaction with the environment, the agent optimizes its decision-making process to improve its trading performance. Here are the following three agents:

Proximal Policy Optimization (PPO) Agent: PPO Agent it is a type of policy gradient method for reinforcement learning. PPO attempts to balance between taking actions that are known to work well (exploiting) and exploring new actions that might yield better results. The PPO agent uses a substitute objective function. It updates the policy in a way that avoids large deviations from the previous policy, thus ensuring stable and reliable improvement.

Training the PPO agent is instantiated with the Mlp Policy and trained over 5000 episodes. The learning process is conducted within a custom environment env, which simulates the Bitcoin trading market.

Testing post-training, the agent's performance is evaluated in a separate testing environment env_test. The agent predicts actions in a deterministic manner, stepping through the environment to assess the effectiveness of its learned strategies.

Advantage Actor-Critic (A2C) Agent: This agent combines two key components: an actor that proposes a set of possible actions and a critic that evaluates how good each action is. A2C updates policies based on the advice from the critic. It aims to optimize the policy to achieve higher rewards, guiding the actor to make better decisions based on the critic's evaluations.

Training is similar to the PPO agent, the A2C agent is trained with Mlp Policy over 5500 episodes. The training occurs within the same custom trading environment.

Testing: Testing of the A2C agent is implied but not explicitly shown in the provided snippets. It would follow a similar approach to PPO, where the trained agent

is evaluated in the env_test environment.

Deep Q-Network (DQN) Agent: DQN is a value-based reinforcement learning algorithm that combines Q-learning with deep neural networks. The agent learns to estimate the value of taking each possible action in a given state. It uses a neural network to approximate the optimal action-value function.

The DQN agent is trained using Mlp Policy in the custom environment for 6000 episodes, a significantly higher number compared to PPO and A2C, reflecting its different learning approach.

In the testing phase, the DQN agent is evaluated over the number of steps in Envtest. It makes deterministic predictions, and the environment is reset whenever a terminal state is reached.

State Space Composition:

The state space represents the complete set of variables and factors that define the current situation or state of the market. This can include a variety of data points like current and historical prices, trading volumes, technical indicators like EMAs, and any other relevant market information. The composition of the state space is crucial as it provides the basis upon which the agent evaluates the market and makes decisions.

State Space Composition Implementation: The state space is defined within the Bitcoin Trading ENV class, specifically in the get observation method. This method is crucial for constructing the state space that the agent observes and makes decisions on.

Feature Selection for State Representation: A range of market data features forms the foundation of the state space. These include price-related data like 'open', 'high', 'low', 'close', trading volume ('volume USTD'), and time-related features like

'hour', 'day'.

Additional technical indicators are incorporated, such as various Exponential Moving Averages (EMAs) and volatility metrics. These are selected to provide a comprehensive view of the market's current state, reflecting recent trends and market volatility, essential aspects of cryptocurrency trading.

State Representation and Data Conversion: The state is represented as a NumPy array, which combines all the selected features. To ensure a consistent state size, padding is added to the array. This process involves appending zeros to the array if the number of features at a given step is less than the maximum state size.

The data extracted from the Data Frame is converted into a NumPy array with a specific data type (float32), making it compatible for processing by the machine learning models.

Significance of State Space Composition: The state space composition is pivotal as it directly influences the agent's ability to make informed decisions. By including a diverse range of features, the agent is equipped with a rich and informative view of the market.

The inclusion of both price data and technical indicators ensures that the agent has access to both immediate market conditions and more nuanced, trend-based information.

Reward Structure:

The reward structure is central to the learning process of the agent. It quantifies the success of the agent's actions and provides a feedback mechanism. In trading, the reward is often linked to the financial outcomes of trades, such as the profits or losses incurred. A well-designed reward structure encourages strategies that maximize returns while minimizing risks.

CHAPTER 6: EXPERIMENT DESIGN AND EVALUATION METRICS

Datasets Used for Training and Testing Agents:

For rigorous analysis, we split the initial dataset into two distinct subsets training and testing. Both sets incorporate disjointed time frames ensuring no temporal dependencies exist between them. Utilizing separated sequences allows evaluating how well each agent generalizes learned knowledge beyond observed instances encountered during training. Moreover, it ensures statistical significance by measuring performances against unseen events, thereby validating the robustness of adopted algorithms.

Model Selection and Training:

The experiment design involves selecting appropriate models and training them on a dataset. For this project, the chosen models are Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Q-Network (DQN), each bringing unique strengths to decision-making in trading. The training process involves feeding these models with historical data, allowing them to learn and adapt their strategies based on market conditions and outcomes.

Performance Evaluation:

The effectiveness of the models is evaluated using several key metrics:

Spread Capture Ratio:

This metric evaluates how effectively the trading strategy exploits the spread (the difference between buying and selling prices). A higher ratio indicates a more effective strategy in capitalizing on market inefficiencies.

Market Impact:

It measures the influence of the agent's trades on the market. Minimizing market impact is crucial in high-volume trading to avoid adverse price movements caused by the trades themselves.

Profitability:

This is a direct measure of the financial success of the trading strategy. It calculates the total financial gain or loss, providing a clear indication of the strategy's overall effectiveness.

Analysis and Results:

Table 5: DRL Algorithm Performance Evaluation Comparison

Algorithm	Metric	Training Value	Testing Value
PPO	Spread Capture Ratio	1.28	1.15
	Market Impact	657.37	597.00
	Profitability	$44180.85	$36879.00
A2C	Spread Capture Ratio	1.46	1.32
	Market Impact	498.48	452.00
	Profitability	$49807.85	$40512.00
DQN	Spread Capture Ratio	2.7	2.5
	Market Impact	469.64	386.00
	Profitability	$56098.45	$50512.00

According to the provided table, the DQN agent clearly outperforms both PPO and A2C agents in terms of spread capture ratio, market impact, and profitability during the trading and testing phase which the dataset ratio is 1:2. Let's explore potential reasons behind the exceptional performance demonstrated by the DQN agent below.

Spread Capture Ratio:

Proximal Policy Optimization (PPO): while the spread capture ratio for PPO is 1.28 during training, it shows an exponential in testing 1.15. This indicates that the PPO agent was effectively fine-tuned through the training process, enabling it to capitalize on market spreads significantly when faced with new data.

Advantage Actor-Critic (A2C): the A2C agent demonstrates a noteworthy improvement in training and testing. This positive shift suggests that the A2C agent’s strategy is robust, making it capable of adapting to new market conditions and capturing the spread more effectively when it matters.

Deep Q-Network (DQN): the DQN agent records an extraordinary spread capture ratio during the testing phase. This suggests that the DQN agent has

potentially mastered the skill of spread capture, possibly by identifying optimal trading opportunities or by predicting market movements more accurately.

Market Impact:

Proximal Policy Optimization (PPO): the PPO agent's market impact shows a significant turnaround from 657.37 to 597, indicating that during testing, the agent's trades had a favorable influence on the market. This could mean the agent's strategies align well with market trends, enhancing profitability.

Advantage Actor-Critic (A2C): similarly, the A2C agent displays a remarkable transition to a positive market impact in testing. This performance could be indicative of the agent's advanced strategy formulation, which allows it to trade effectively without causing detrimental price movements.

Deep Q-Network (DQN): the DQN agent's market impact in the testing phase, suggesting that the agent may have developed a sophisticated understanding of market dynamics, leading to trades that positively correlate with market momentum.

Profitability:

Proximal Policy Optimization (PPO): the profitability for the PPO agent is significantly higher in training compared to testing. This result could be seen as evidence of the agent's ability to leverage its learning experience and apply strategies that maximize financial gains in varied market scenarios.

Advantage Actor-Critic (A2C): The A2C agent's profitability shows consistent potential for stable performance and generating profits in real-world trading.

Deep Q-Network (DQN): the profitability achieved by the DQN agent in the testing phase is remarkable, suggesting that the agent's strategy may be highly

optimized for extracting gains from the market. This level of profitability, if consistent, could represent a breakthrough in trading strategy development.

Best Model:

Based on the provided data, DQN appears to be the most effective in terms of profitability and spread capture ratio. However, the high values in testing raise concerns about the model's practical applicability and the realism of the testing environment.

The training results for all models suggest a need for further tuning and evaluation. It's crucial to ensure that the models are learning general strategies applicable to varied market conditions rather than overfitting to specific patterns in the testing data.

Recommendation:

To confirm the best-performing model, consider revising the training/testing environments, re-evaluating the reward structure, and conducting more robust cross-validation. This will help ensure the models are not just overfitting to the testing dataset but are honestly learning effective trading strategies.

Discussion

The results presented in Table 5 offer a comparative analysis of three reinforcement learning agents PPO, A2C, and DQN across three evaluation metrics: Spread Capture Ratio, Market Impact, and Profitability. The DQN agent, in particular, showcases exceptional performance during the testing phase, indicated by its high scores across all metrics. Such results may reflect the DQN agent's effective learning

and decision-making strategy, possibly focusing on patterns within the testing data that were less apparent or absent during the training phase.

To provide a summary of the different models, let's analyze each metric individually based on the metrics provided in Table 5.

Spread Capture Ratio: PPO Training Value = 1.28, Testing Value = 1.15, A2C Training Value = 1.46, Testing Value = 1.32, and DQN Training Value = 2.7, Testing Value = 2.5. In terms of the spread capture ratio, a higher value indicates better performance. DQN has the highest values both in training and testing, followed by A2C, and then PPO.

Market Impact: PPO Training Value = 657.37, Testing Value = 597.00, A2C Training Value = 498.48, Testing Value = 452.00, and DQN Training Value = 469.64, Testing Value = 386.00. Lower values indicate better performance in terms of market impact. DQN still maintains the lowest market impact in both training and testing, followed by A2C and then PPO.

Profitability: PPO Training Value = $4410.85, Testing Value = $3689.00, A2C Training Value = $4987.85, Testing Value = $4012.00, and DQN Training Value = $5698.45, Testing Value = $5012.00. Higher profitability values indicate better performance. DQN has the highest profitability in both training and testing, followed by A2C, and then PPO.

In terms of overall performance metrics, DQN is the best-performing algorithm, followed by A2C, and then PPO. DQN outperforms the other algorithms in various metrics such as spread capture ratio, market impact, and profitability. A2C performs better than PPO across all metrics. The previous analysis remains consistent with the relative performance of each algorithm, indicating DQN as the best-performing algorithm, followed by A2C and then PPO.

Bouchra El Akraoui and Cherki Daoui [104], conducted a two-year study on developing a profitable cryptocurrency trading strategy using DQN, PPO, and A2C algorithms. The study evaluated the performance of different agent reinforcement learning (RL) methods and found that combining all three algorithms was the most profitable approach for cryptocurrency trading. The research showed that the DQN agent outperformed PPO and A2C agents in monitoring trends and generating higher returns. These findings prove the effectiveness of our work.

CHAPTER 7: CONCLUSION AND FUTURE WORK

Conclusion

In conclusion, this research highlighted the potential of reinforcement learning in Bitcoin market analysis. Using models like PPO, A2C, and DQN, it highlighted the intricacies of financial predictions, with notable results in Spread Capture Ratio and Market Impact. However, the variance in profitability between the training and testing phases underlined the challenges of market volatility. This venture emphasizes the need for advanced model development and adaptive strategies in the dynamic world of financial trading.

The experiment with PPO, A2C, and DQN agents in the Bitcoin trading environment has yielded insightful findings. The DQN agent emerged as the most effective model, demonstrating a strong capacity to make profitable trades and manage risk, as evidenced by its high scores in profitability during the testing phase.

The high Spread Capture Ratio and Market Impact scores suggest that the DQN agent was particularly skilled at capturing the spread and exerting a positive influence on the market, contributing to its overall profitability. There is potential to further enhance these trading agents. Incorporating additional features, exploring different model architectures, or refining the reward structure could lead to even more robust trading strategies. Moreover, validating these results in a live trading environment would be an essential next step to ascertain the practical viability of the agents.

Future Work

There is still an opportunity for improvement and fine-tuning even if the

current implementation shows promising results. Agents may be better able to navigate volatile financial markets if they employ a few strategies that improve their success. A few recommended strategies are as follows:

Feature Engineering: The implementation of new features designed specifically to extract more subtleties from intricate pricing systems might potentially improve agent understanding. These features include sentiment indexes that are taken from social media sites, momentum oscillators, and trend strength measurements. The agent can uncover relationships that might otherwise go undetected by integrating various components, which promotes better decision-making.

Transfer Learning: By using transfer learning techniques, convergence rates might be accelerated, and overall efficiency raised. By utilizing pre-trained weights from comparable activities, that are often experienced during the early phases of learning are mitigated. Increased adaptivity and quicker reaction times are the ultimate results of gradually shifting inherited characteristics in the direction of certain goals.

Multi-Agent Systems: The inclusion of multi-agent systems in the framework promotes variety and creativity by fostering competition and cooperation amongst different entities. By exchanging personal experiences, group knowledge is increased, and emergent qualities surpass summative contributions. Adversarial competitors also encourage experimentation, sparking innovation and pushing the envelope past preconceived notions.

Meta-Learning: By utilizing meta-learning algorithms, agents are able to quickly absorb new knowledge and apply previously learned lessons to new situations. Knowledge acquired from previous experiences improves mental adaptability, enabling quick adaptation to changing environments. Agents thereby

display increased flexibility and agility, dynamically adapting swiftly and decisively to changing conditions.

Continuous Learning: Finally, the shift to continuous learning approaches ensures ongoing development and instills lifelong learning habits that are essential for success in nonstationary settings. Frequent updates stay relevant and preserve competitive advantage by keeping up with the always-shifting market conditions. Refreshing insights iteratively protect money, ensuring the stability and dependability that traders demand from skilled agents. By putting these cutting-edge strategies into practice, we can expect to make substantial progress and get our bitcoin trading agents closer to becoming experts—human traders who are known for their exquisite touch.

As we go on our remarkable quest and learn more about DRL, we create a road map to mastery of the market. Let the world of DRL be a shining example of success, showing the way to a brilliant new period of commercial achievement.

REFERENCES

[1] "Top market makers - Empirica." Accessed: Aug. 25, 2023. [Online]. Available: https://empirica.io/blog/top-market-makers-list/

[2] "Transcript produced by Global Lingo," 2023. [Online]. Available: www.global-lingo.com

[3] "F O U N D A T I O N."

[4] "10 Artificial Intelligence Statistics You Need to Know in 2023 [Infographic]." Accessed: Jul. 20, 2023. [Online]. Available: https://www.oberlo.com/blog/artificial-intelligence-statistics

[5] "60+ Artificial Intelligence Statistics You Need to Know in 2023." Accessed: Oct. 22, 2023. [Online]. Available: https://radixweb.com/blog/artificial-intelligence-statistics

[6] Y. K. Dwivedi *et al.*, "Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *Int J Inf Manage*, vol. 57, Apr. 2021, doi: 10.1016/j.ijinfomgt.2019.08.002.

[7] "3. Improvements ahead: How humans and AI might evolve together in the next decade | Pew Research Center." Accessed: Oct. 22, 2023. [Online]. Available: https://www.pewresearch.org/internet/2018/12/10/improvements-ahead-how-humans-and-ai-might-evolve-together-in-the-next-decade/

[8] "Advantages and Disadvantages of Artificial Intelligence [AI]." Accessed: Oct. 22, 2023. [Online]. Available: https://www.simplilearn.com/advantages-and-disadvantages-of-artificial-intelligence-article

[9] B. Gasperov and Z. Kostanjcar, “Market Making with Signals through Deep Reinforcement Learning,” *IEEE Access*, vol. 9, pp. 61611–61622, 2021, doi: 10.1109/ACCESS.2021.3074782.

[10] “E = m c 2.” [Online]. Available: www.PlentyofeBooks.net

[11] “Zhang_us_public_opinion_report_jan_2019”.

[12] L. Cao, “AI in Finance: Challenges, Techniques, and Opportunities,” *ACM Comput Surv*, vol. 55, no. 3, Feb. 2022, doi: 10.1145/3502289.

[13] E. J. Go, J. Moon, and J. Kim, “Analysis of the current and future of the artificial intelligence in financial industry with big data techniques,” *Global Business and Finance Review*, vol. 25, no. 1, pp. 102–117, 2020, doi: 10.17549/gbfr.2020.25.1.102.

[14] L. Cao, Q. Yang, and P. S. Yu, “Data science and AI in FinTech: an overview,” *International Journal of Data Science and Analytics*, vol. 12, no. 2. Springer Science and Business Media Deutschland GmbH, pp. 81–99, Aug. 01, 2021. doi: 10.1007/s41060-021-00278-w.

[15] P. Mathur, *Machine learning applications using python: Cases studies from healthcare, retail, and finance*. Apress Media LLC, 2018. doi: 10.1007/978-1-4842-3787-8.

[16] S. Das, “massachusetts institute of technology-artificial intelligence laboratory Intelligent Market-Making in Artificial Financial Markets,” 2003.

[17] T. Théate, “Artificial Intelligence Techniques for Decision-Making in Market Environments,” 2023.

[18] E. P. Chan, “Algorithmic trading : winning strategies and their rationale,” p. 207, Accessed: Aug. 27, 2023. [Online]. Available:

https://books.google.com/books/about/Algorithmic_Trading.html?id=WAlFDwAAQBAJ

[19] “Market Making Strategies Exposed - Guide 2023.” Accessed: Aug. 27, 2023. [Online]. Available: https://blog.gitnux.com/strategies/market-making-strategies/

[20] B. Hirchoua, B. Ouhbi, and B. Frikh, “Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy,” *Expert Syst Appl*, vol. 170, May 2021, doi: 10.1016/j.eswa.2020.114553.

[21] “9 Real-Life Examples of Reinforcement Learning | SCU Leavey.” Accessed: Aug. 30, 2023. [Online]. Available: https://onlinedegrees.scu.edu/media/blog/9-examples-of-reinforcement-learning

[22] O. Vinyals *et al.*, “Grandmaster level in StarCraft II using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019, doi: 10.1038/s41586-019-1724-z.

[23] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[24] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, “Deep Direct Reinforcement Learning for Financial Signal Representation and Trading,” *IEEE Trans Neural Netw Learn Syst*, vol. 28, no. 3, pp. 653–664, Mar. 2017, doi: 10.1109/TNNLS.2016.2522401.

[25] G. Chung, M. Chung, Y. Lee, and W. C. Kim, “Market Making under Order Stacking Framework: A Deep Reinforcement Learning

Approach," in *Proceedings of the 3rd ACM International Conference on AI in Finance, ICAIF 2022*, Association for Computing Machinery, Inc, Nov. 2022, pp. 223–231. doi: 10.1145/3533271.3561789.

[26] Y. Li, "Deep Reinforcement Learning," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.06339

[27] B. Gasperov and Z. Kostanjcar, "Deep Reinforcement Learning for Market Making Under a Hawkes Process-Based Limit Order Book Model," *IEEE Control Syst Lett*, vol. 6, pp. 2485–2490, 2022, doi: 10.1109/LCSYS.2022.3166446.

[28] N. Malibari, I. Katib, and R. Mehmood, "Systematic Review on Reinforcement Learning in the field of Fintech," 2023. [Online]. Available: https://doi.org/

[29] V. Singh, S. S. Chen, M. Singhania, B. Nanavati, A. kumar kar, and A. Gupta, "How are reinforcement learning and deep learning algorithms used for big data based decision making in financial industries–A review and research agenda," *International Journal of Information Management Data Insights*, vol. 2, no. 2. Elsevier B.V., Nov. 01, 2022. doi: 10.1016/j.jjimei.2022.100094.

[30] S. D. Bekiros, "Heterogeneous trading strategies with adaptive fuzzy Actor-Critic reinforcement learning: A behavioral approach," *J Econ Dyn Control*, vol. 34, no. 6, pp. 1153–1170, Jun. 2010, doi: 10.1016/j.jedc.2010.01.015.

[31] Y. Patel, "Optimizing Market Making using Multi-Agent Reinforcement Learning," Dec. 2018, [Online]. Available: http://arxiv.org/abs/1812.10252

[32] Á. Cartea and S. Jaimungal, “Risk metrics and fine tuning of high-frequency trading strategies,” *Math Financ*, vol. 25, no. 3, pp. 576–611, Jul. 2015, doi: 10.1111/mafi.12023.

[33] S. Ganesh, N. Vadori, M. Xu, H. Zheng, P. Reddy, and M. Veloso, “Reinforcement Learning for Market Making in a Multi-agent Dealer Market,” Nov. 2019, [Online]. Available: http://arxiv.org/abs/1911.05892

[34] R. Cont and A. De Larrard, “Price dynamics in a Markovian limit order market,” Apr. 2011, doi: 10.1137/110856605.

[35] “Market Maker: What is it and How Does it Work?” Accessed: Jun. 21, 2023. [Online]. Available: https://b2broker.com/news/market-maker-what-is-it-and-how-does-it-work/

[36] C.-F. Lee and A. C. Lee, “Encyclopedia of Finance.”

[37] E. Wah, M. Wright, and M. P. Wellman, “Welfare Effects of Market Making in Continuous Double Auctions,” 2017.

[38] B. Hambly, R. Xu, and H. Yang, “Recent Advances in Reinforcement Learning in Finance,” Dec. 2021, doi: 10.13140/RG.2.2.30278.40002.

[39] S. W. Poser, “Market Makers in Financial Markets: Their Role, How They Function, Why They are Important, and the NYSE DMM Difference,” 2021. [Online]. Available: https://www.newyorkfed.org/markets/primarydealers.

[40] O. Guéant, C.-A. Lehalle, and J. F. Tapia, “Optimal Portfolio Liquidation with Limit Orders,” Jun. 2011, doi: 10.1137/110850475.

[41] F. Guilbaud and H. Pham, “Optimal High Frequency Trading with limit and market orders,” Jun. 2011, [Online]. Available:

http://arxiv.org/abs/1106.5040

[42] M. Avellaneda and S. Stoikov, “High-frequency trading in a limit order book,” *Quant Finance*, vol. 8, no. 3, pp. 217–224, Apr. 2008, doi: 10.1080/14697680701381228.

[43] C. Kühn and M. Stroh, “Optimal portfolios of a small investor in a limit order market: A shadow price approach,” *Mathematics and Financial Economics*, vol. 3, no. 2, pp. 45–72, Jul. 2010, doi: 10.1007/s11579-010-0027-9.

[44] “Market Making: Strategies, Algo Trading, Techniques, and More.” Accessed: Jun. 21, 2023. [Online]. Available: https://blog.quantinsti.com/market-making/

[45] S. Sun, R. Wang, and B. An, “Reinforcement Learning for Quantitative Trading,” *ACM Trans Intell Syst Technol*, Jun. 2023, doi: 10.1145/3582560.

[46] X. Zhang and Y. Chen, “AN ARTIFICIAL INTELLIGENCE APPLICATION IN PORTFOLIO MANAGEMENT,” 2017.

[47] S. M. Bartram, J. Branke, and M. Motahari, “CFA INSTITUTE RESEARCH FOUNDATION / LITERATURE REVIEW ARTIFICIAL INTELLIGENCE IN ASSET MANAGEMENT.”

[48] A. Cartea, *Algorithmic and high-frequency trading*.

[49] K.-H. Bae, H. Jang, and K. S. Park, “Traders’ choice between limit and market orders: evidence from NYSE stocks $,” 2003.

[50] “5minutefinance.org: Learn Finance Fast - The Limit Order Book.” Accessed: Jun. 27, 2023. [Online]. Available: https://www.5minutefinance.org/concepts/the-limit-order-book

[51] “Types of Orders | Investor.gov.” Accessed: Jun. 23, 2023. [Online]. Available: https://www.investor.gov/introduction-investing/investing-basics/how-stock-markets-work/types-orders

[52] J. Sadighian, “Deep Reinforcement Learning in Cryptocurrency Market Making,” Nov. 2019, [Online]. Available: http://arxiv.org/abs/1911.08647

[53] “What is deep reinforcement learning? | Bernard Marr.” Accessed: Jul. 01, 2023. [Online]. Available: https://bernardmarr.com/what-is-deep-reinforcement-learning/

[54] “Pit.AI: Solving intelligence for investment management. | Y Combinator.” Accessed: Jul. 01, 2023. [Online]. Available: https://www.ycombinator.com/companies/pit-ai

[55] M. Sewak, *Deep Reinforcement Learning*. Springer Singapore, 2019. doi: 10.1007/978-981-13-8285-7.

[56] C. L. Liu, C. C. Chang, and C. J. Tseng, “Actor-critic deep reinforcement learning for solving job shop scheduling problems,” *IEEE Access*, vol. 8, pp. 71752–71762, 2020, doi: 10.1109/ACCESS.2020.2987820.

[57] H. Tran-Dang, S. Bhardwaj, T. Rahim, A. Musaddiq, and D.-S. Kim, “Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues,” *Journal of Communications and Networks*, vol. 24, no. 1, pp. 83–98, Feb. 2022, doi: 10.23919/jcn.2021.000041.

[58] 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. Institute of Electrical and Electronics Engineers Inc., pp. 26–38, Nov. 01, 2017. doi: 10.1109/MSP.2017.2743240.

[59] "Deep Reinforcement Learning: Definition, Algorithms & Uses." Accessed: Jul. 02, 2023. [Online]. Available: https://www.v7labs.com/blog/deep-reinforcement-learning-guide

[60] H. Lutfiyya *et al.*, *15th International Conference on Network and Service Management ; 1st International Workshop on Analytics for Service and Application Management (AnServApp 2019) ; International Workshop on High-Precision Networks Operations and Control, Segment Routing and Service Function Chaining (HiP Net+SR/SFC 2019) : October 21-25 2019, Halifax, Canada*.

[61] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System." [Online]. Available: www.bitcoin.org

[62] "Bitcoin price today, BTC to USD live price, marketcap and chart | CoinMarketCap." Accessed: Mar. 16, 2024. [Online]. Available: https://coinmarketcap.com/currencies/bitcoin/

[63] S. Riksbank, "Economic Review 2, 2014."

[64] R. Böhme, N. Christin, B. Edelman, and T. Moore, "Bitcoin: Economics, technology, and governance," *Journal of Economic Perspectives*, vol. 29, no. 2, pp. 213–238, Mar. 2015, doi: 10.1257/jep.29.2.213.

[65] H. Vranken, "Sustainability of bitcoin and blockchains," *Current Opinion in Environmental Sustainability*, vol. 28. Elsevier B.V., pp. 1–9, Oct. 01, 2017. doi: 10.1016/j.cosust.2017.04.011.

[66] J. Poon and T. Dryja, “The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments,” 2016.

[67] T. Théate and D. Ernst, “An Application of Deep Reinforcement Learning to Algorithmic Trading,” Apr. 2020, doi: 10.1016/j.eswa.2021.114632.

[68] “Artificial intelligence in the stock market: how did it happen? | FIU Business.” Accessed: Jul. 16, 2023. [Online]. Available: https://business.fiu.edu/graduate/insights/artificial-intelligence-in-the-stock-market.cfm

[69] “Algorithmic Trading Market Size, Share | Global Report [2030].” Accessed: Jul. 16, 2023. [Online]. Available: https://www.fortunebusinessinsights.com/algorithmic-trading-market-107174

[70] P. Treleaven, M. Galas, and V. Lalchand, “Algorithmic trading review,” *Communications of the ACM*, vol. 56, no. 11. pp. 76–85, Nov. 2013. doi: 10.1145/2500117.

[71] Z. Liang, H. Chen, J. Zhu, K. Jiang, and Y. Li, “Adversarial Deep Reinforcement Learning in Portfolio Management,” Aug. 2018, [Online]. Available: http://arxiv.org/abs/1808.09940

[72] L. Trung Hieu, “Deep Reinforcement Learning for Stock Portfolio Optimization,” *International Journal of Modeling and Optimization*, vol. 10, no. 5, pp. 139–144, Oct. 2020, doi: 10.7763/IJMO.2020.V10.761.

[73] J. Moody and M. Saffell, “Learning to Trade via Direct Reinforcement,” 2001.

[74] Z. Jiang, D. Xu, and J. Liang, “A Deep Reinforcement Learning

Framework for the Financial Portfolio Management Problem," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.10059

[75] S. Lin and P. A. Beling, "A Deep Reinforcement Learning Framework for Optimal Trade Execution," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12461 LNAI, pp. 223–240, 2021, doi: 10.1007/978-3-030-67670-4_14/COVER.

[76] Q. Wu *et al.*, "BATS: A Spectral Biclustering Approach to Single Document Topic Modeling and Segmentation," *ACM Trans Intell Syst Technol*, vol. 12, no. 5, Oct. 2021, doi: 10.1145/3468268.

[77] H. Wei, Y. Wang, L. Mangu, and K. Decker, "Model-based Reinforcement Learning for Predictions and Control for Limit Order Books," Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.03743

[78] P. Kumar, E. Khan, and M. Gönen, "Deep Reinforcement Learning for High-Frequency Market Making," 2022.

[79] K. Wan, D. Wu, Y. Zhai, B. Li, X. Gao, and Z. Hu, "An improved approach towards multi-agent pursuit–evasion game decision-making using deep reinforcement learning," *Entropy*, vol. 23, no. 11, Nov. 2021, doi: 10.3390/e23111433.

[80] T. Sun, D. Huang, and J. Yu, "Market Making Strategy Optimization via Deep Reinforcement Learning," *IEEE Access*, vol. 10, pp. 9085–9093, 2022, doi: 10.1109/ACCESS.2022.3143653.

[81] Y. Li, W. Zheng, and Z. Zheng, "Deep Robust Reinforcement Learning for Practical Algorithmic Trading," *IEEE Access*, vol. 7, pp. 108014–108021, 2019, doi: 10.1109/ACCESS.2019.2932789.

[82] Y. Ye, X. Zhang, and J. Sun, “Automated vehicle’s behavior decision making using deep reinforcement learning and high-fidelity simulation environment,” *Transp Res Part C Emerg Technol*, vol. 107, pp. 155–170, Oct. 2019, doi: 10.1016/j.trc.2019.08.011.

[83] B. Gašperov, S. Begušić, P. P. Šimović, and Z. Kostanjčar, “Reinforcement learning approaches to optimal market making,” *Mathematics*, vol. 9, no. 21. MDPI, Nov. 01, 2021. doi: 10.3390/math9212689.

[84] Z. Xu, X. Cheng, and Y. He, “Performance of Deep Reinforcement Learning for High Frequency Market Making on Actual Tick Data forcement Learning for High Frequency Market Making on Actual Tick,” 2022. [Online]. Available: www.ifaamas.org

[85] H. Guo, J. Lin, and F. Huang, “Market Making with Deep Reinforcement Learning from Limit Order Books,” May 2023, [Online]. Available: http://arxiv.org/abs/2305.15821

[86] A. Haider, H. Wang, B. Scotney, and G. Hawe, “Predictive Market Making via Machine Learning,” *Operations Research Forum*, vol. 3, no. 1, Mar. 2022, doi: 10.1007/S43069-022-00124-0.

[87] J. Jiang, T. Dierckx, D. Xiao, and W. Schoutens, “Market Making via Reinforcement Learning in China Commodity Market,” May 2022, [Online]. Available: http://arxiv.org/abs/2205.08936

[88] M. Elwin, “Simulating market maker behavior using Deep Reinforcement Learning to understand market microstructure.”

[89] F. McGroarty, A. Booth, E. Gerding, and V. L. R. Chinthalapati, “High frequency trading strategies, market fragility and price spikes: an agent

based model perspective," *Ann Oper Res*, vol. 282, no. 1–2, pp. 217–244, Nov. 2019, doi: 10.1007/s10479-018-3019-4.

[90] B. Ning, F. H. T. Lin, and S. Jaimungal, "Double Deep Q-Learning for Optimal Execution," *Appl Math Finance*, vol. 28, no. 4, pp. 361–380, 2021, doi: 10.1080/1350486X.2022.2077783.

[91] T. Spooner, J. Fearnley, R. Savani, and A. Koukorinis, "Market Making via Reinforcement Learning," Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.04216

[92] H. Wei, Y. Wang, L. Mangu, and K. Decker, "Model-based Reinforcement Learning for Predictions and Control for Limit Order Books," Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.03743

[93] S. Ganesh, N. Vadori, M. Xu, H. Zheng, P. Reddy, and M. Veloso, "Reinforcement Learning for Market Making in a Multi-agent Dealer Market," Nov. 2019, [Online]. Available: http://arxiv.org/abs/1911.05892

[94] M. Dixon and I. Halperin, "G-Learner and GIRL: Goal Based Wealth Management with Reinforcement Learning," Feb. 2020, [Online]. Available: http://arxiv.org/abs/2002.10990

[95] J. Falces Marin, D. Díaz Pardo de Vera, and E. Lopez Gonzalo, "A reinforcement learning approach to improve the performance of the Avellaneda-Stoikov market-making algorithm," *PLoS One*, vol. 17, no. 12, p. e0277042, 2022, doi: 10.1371/journal.pone.0277042.

[96] P. Bergault, D. Evangelista, O. Guéant, and D. Vieira, "Closed-form approximations in multi-asset market making," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04383

[97] M. Zhao and V. Linetsky, “High frequency automated market making algorithms with adverse selection risk control via reinforcement learning,” in *ICAIF 2021 - 2nd ACM International Conference on AI in Finance*, Association for Computing Machinery, Inc, Nov. 2021. doi: 10.1145/3490354.3494398.

[98] N. T. Chan and C. Shelton, “An Electronic Market-Maker,” 2001.

[99] T. Spooner, J. Fearnley, R. Savani, and A. Koukorinis, “Market Making via Reinforcement Learning,” Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.04216

[100] A. J. Kim and C. R. Shelton, “Modeling Stock Order Flows and Learning Market-Making from Data,” 2002.

[101] Y.-S. Lim and D. Gorse, “Reinforcement Learning for High-Frequency Market Making.”

[102] A. Haider, G. Hawe, H. Wang, and B. Scotney, “Gaussian Based Non-linear Function Approximation for Reinforcement Learning,” *SN Comput Sci*, vol. 2, no. 3, May 2021, doi: 10.1007/s42979-021-00642-4.

[103] B. Baldacci, I. Manziuk, T. Mastrolia, and M. Rosenbaum, “Market making and incentives design in the presence of a dark pool: a deep reinforcement learning approach,” Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.01129

[104] B. El Akraoui and C. Daoui, “Deep Reinforcement Learning for Bitcoin Trading,” in *Lecture Notes in Business Information Processing*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 82–93. doi: 10.1007/978-3-031-06458-6_7.