Reinforcement Learning for Algorithmic Trading in Financial Markets

by Soheil Gityforoze

A.L.M. in Finance, November 2019, Harvard University
M.S. in Applied Analytics, May 2022, Columbia University

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

August 15, 2025

Praxis directed by

Aaron Focacci
Professorial Lecturer of Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Soheil Gityforoze has passed the Final Examination for the degree of Doctor of Engineering as of July 2, 2025. This is the final and approved form of the Praxis.

Reinforcement Learning for Algorithmic Trading in Financial Markets

Soheil Gityforoze

Praxis Research Committee:

Aaron Focacci, Professorial Lecturer in Engineering and Applied Science, Praxis Director

Vijay Raghavan, Professorial Lecturer in Engineering and Applied Science, Committee Chair

Hamza Alsarhan, Professorial Lecturer in Engineering and Applied Science, Committee Member

Copyright © 2025 by Soheil Gityforoze
All rights reserved

Acknowledgments

The author wishes to thank my advisor Dr. Aaron Focacci, for his guidance and mentorship throughout the development of this praxis.

Abstract of Praxis

Reinforcement Learning for Algorithmic Trading in Financial Markets

In multi-period algorithmic trading, determining algorithms that are ideal for risk-averse strategies is a challenging task. This study explored the application of model-free reinforcement learning (RL) in algorithmic trading and analyzed the relationship between risk-averse strategies and implementation of RL algorithms including Q-learning, Greedy-GQ, and SARSA. The data for this quantitative research included one year of the E-mini-NASDAQ-100 futures (2023-2024). Over 7,500 simulation results substantiated a proof of concept that Q-learning can successfully generate risk-adjusted trading signals in the highly liquid technology-focused futures market. With an optimized configuration of hyperparameters including look-back period, basis and reward functions, Q-learning delivered nearly twice the returns of the competing RL algorithms. Beyond absolute returns, Q-learning exhibited lower volatility across key risk metrics and outperformed the NASDAQ-100 benchmark by approximately 75 percentage points. These findings suggest reinforcement learning as a promising artificial intelligence and machine learning framework for alpha generating strategies in systematic trading.

Table of Content

Acknowledgement ……..…….. iv

Abstract of Praxis ……..... v

Table of Contents …… vi

List of Figures …….. xi

List of Tables ……... xii

List of Symbols …….. xiii

List of Equations …… xiv

List of Acronyms …… xv

List of Glossary of Terms …….. xvi

Introduction ……..... 1

1.1 Background …… 1

1.2 Research Motivation …….. 3

1.3 Problem Statement ……. 4

1.4 Thesis Statement …… 5

1.5 Research Objectives …….. 6

1.6 Research Questions and Hypotheses …… 6

1.7 Scope of Research …… 8

1.8 Research Limitations …… 9

1.9 Organization of Praxis ……. 10

Literature Review …… 12

2.1 Introduction ……. 12

2.2 Reinforcement Learning (RL) …… 13
2.2.1 RL Application in Financial Markets …… 13
2.2.2 Theoretical Foundations for RL Application in Financial Markets 14
2.2.3 Markov Decision Processes (MDPs) ……. 15
2.2.4 Bellman's Principles of Optimal Control …….. 15
2.2.5 Utility Theory …… 16
2.2.6 Modern Portfolio Theory (MPT) …….. 16
2.2.7 Convergence of Theoretical Foundations ……. 17
2.3 RL in Algorithmic Trading ……. 19
2.3.1 Comparative Performance of RL and Traditional ML Models …. 20
2.3.2 Human Trader Approaches …… 24
2.3.3 Categorization of RL Models …… 24
2.3.4 Model-Free, Critic-Only Approach …….. 25
2.3.5 Advantages and Disadvantages of Each Algorithm ……. 28
2.4 Discussion of Current Research ……..... 30
2.4.1 Discussion of Research Gap ……. 36
2.5 Summary and Conclusion …… 38
Methodology …….. 41
3.1 Introduction …….. 41
3.2 Data Collection ……. 42
3.2.2 Data Preprocessing …….. 43
3.2.3 Data Organization and Storage …… 44
3.3 System Architecture and Data Flow……... 45

3.4 Reinforcement Learning (RL) Environment …….. 46

3.4.1 Elaboration of Technical Indicators for State Representation …… . 47

3.4.1.1 Bollinger Bands …….. 48

3.4.1.2 Exponential Moving Average (EMA) ……. 49

3.4.1.3 Linearly Weighted Moving Average (LWMA) ……. 50

3.4.2 Technical Indicators in the Reinforcement Learning Framework.... 51

3.5 Reinforcement Learning (RL) Algorithms …… 56

3.5.1 Q-Learning Algorithm …… 57

3.5.2 SARSA Algorithm…….. 58

3.5.3 Greedy-GQ Algorithm …….. 58

3.5.4 Hyperparameter Optimization ……... 59

3.5.5 Hyperparameter Evaluation Process …….. 60

3.6 Simulation and Performance Evaluation ……... 61

3.6.1 Simulation Architecture and Execution Process ……. 61

3.6.2 Simulation Data Processing ……. 62

3.6.3 Performance Evaluation Methodology ……... 64

3.7 Model Selection and Performance Evaluation Metrics …… 67

3.7.1 Profitability Metrics …….. 67

3.7.2 Risk Metrics ……... 68

3.8 Summary ……. 68

Results …….. 70

4.1 Introduction …… 70

4.2 Data Exploration and Market Analysis …….. 70

4.2.1 E-mini Nasdaq-100 Futures Market Characteristics ……. 70
4.2.2 Overall Market Trend …… 71
4.2.3 Contract-Specific Analysis ……. 72
4.2.4 Price Distribution Analysis ……. 73
4.2.5 Volatility Progression ……. 74
4.2.6 Market Regimes Summary ……. 75
4.3 Hyperparameter Optimization Results……... 75
4.3.1 Individual Contract Results …… 76
4.3.2 Hyperparameter Patterns Across All Tickers …… 77
4.4 Hypothesis 1 Results ……. 77
4.5 Hypothesis 2 Results …… 80
4.5.1 Comparative Algorithm Performance …….. 81
4.6 Hypothesis 3 Results …….. 82
4.6.1 Overall Performance Summary……. 83
4.6.2 Risk-Adjusted Performance Metrics …… 83
4.7 Chapter Summary …….. 84
Discussion and Conclusion…….. 85
5.1 Discussion …….. 85
5.1.1 Discussion of Hypothesis 1 …… 86
5.1.2 Discussion of Hypothesis 2 …… 88
5.1.3 Discussion of Hypothesis 3 …… 89
5.2 Conclusion …….. 91
5.2.1 Lessons Learned …… 92

5.3 Contribution to the Body of Knowledge……. ... 93

5.4 Future Research Recommendations.……. ... 95

References …….. 97

List of Figures

Figure 2-1 RL Agent and the Market Environment …… 14

Figure 2-2 Theoretical Framework Integration in RL Trading System …….. 18

Figure 3-1 Data Directory Structure Example …… 45

Figure 3-2 System Architecture Workflow …… 46

Figure 3-3 Hyperparameter Optimization Workflow ……. 66

Figure 4-1 E-mini Nasdaq-100 Futures Market Trend (2023-2024) …… 71

Figure 4-2 Multi-Panel Price Action …….. 73

Figure 4-3 Price Density Distribution …… 74

Figure 4-4 P&L Distribution by Ticker ……. 77

Figure 4-5 Hyperparameter Selection Frequency …….. 79

Figure 4-6 Hyperparameter Impact on Trading Performance (500 Simulations) ……... 80

Figure 4-7 Model Performance Benchmark Comparison ……... 81

List of Tables

Table 3-1 Reward Functions …… 55

Table 3-2 Basis Functions …… 56

Table 3-3 Hyperparameter Search Space …… 60

Table 3-4 Data Flow and Simulation Run Process…… 63

Table 3-5 Profitability Metrics ……. 67

Table 3-6 Risk Metrics …….. 68

Table 4-1 Market Regimes (2023 –2024)…….. 72

Table 4-2 Summary of Returns…….. 82

Table 4-3 Risk Metrics for RL Models …….. 84

List of Symbols

A	Action space
k	Number of standard deviations
initType	Weight initialization method
LB	Lower Band (Bollinger Bands)
n	Lookback window size / time period
MB	Middle Band (Bollinger Bands)
Q	Q-value function
R	Reward
S	State space
t	Time step
UB	Upper Band (Bollinger Bands)
w	Weight vector
α (alpha)	Learning rate parameter / smoothing factor for EMA
γ (gamma)	Discount factor
ε (epsilon)	Exploration rate (SARSA)
η (eta)	Primary learning rate
ζ (zeta)	Secondary learning rate (Greedy-GQ)
σ (sigma)	Standard deviation / volatility
θ (theta)	Target value function
π (pi)	Policy
ϕ (phi)	Feature vector

List of Equations

Equation 3-1 Bollinger Bands Formula …….. 48

Equation 3-2 Exponential Moving Average (EMA) Formula …… 49

Equation 3-3 Linearly Weighted Moving Average (LWMA) Formula ……. 50

Equation 3-4 Q-Learning Weight Update Equation …….. 57

Equation 3-5 SARSA Target Value Function ……. 58

Equation 3-6 Greedy-GQ Target Value Function ……. 59

List of Acronyms

AI	Artificial Intelligence
CME	Chicago Mercantile Exchange
EMA	Exponential Moving Average
LWMA	Linearly Weighted Moving Average
MDP	Markov Decision Process
ML	Machine Learning
OHLCV	Open, High, Low, Close, Volume
P&L	Profit and Loss
RL	Reinforcement Learning
ROI	Return on Investment
SARSA	State-Action-Reward-State-Action
SMA	Simple Moving Average

List of Glossary of Terms

Action Space	The set of possible trading decisions available to an RL agent. It includes three discrete actions: Buy (1) to enter a long position, Hold (0) to maintain the current position, and Sell (-1) to enter a short position.
Actor-Critic Methods	RL approaches where the actor proposes actions and the critic evaluates them. They allow for the policy to adjust based on feedback.
Algorithmic Trading	Utilizes computer algorithms to execute automated trades based on mathematical models and predetermined rules.
Bear Market	A market condition characterized by falling prices and negative investor sentiment.
Bellman's Principles	Emphasize value functions that evaluate the long-term impact of decisions instead of instant gains, to help balance short-term profit opportunities with long-term strategic objectives.
Bollinger Bands	Measure price fluctuations relative to recent price levels, in order to provide insights about potential overbought or oversold states. They use upper and lower bands placed at standard deviation positions above and below a simple moving average.
Bull Market	A market regime identified by increasing prices and positive investor sentiment.
CALMAR Ratio	Compares returns against the largest decline in asset value (from its peak to lowest point). It evaluates both profit potential and the ability to minimize severe losses during market downturns.
Critic-Only Methods	Focus on learning value functions to estimate expected rewards for actions in given states. They guide decision-making based solely on value evaluations.

E-mini-NASDAQ-100 Futures	A futures contract available on the CME that tracks the NASDAQ-100 Index. It follows 100 of the major technology firms listed on the Nasdaq exchange.
Exponential Moving Average (EMA)	Computes a weighted price average that gives higher importance to newer observations. It responds more quickly to recent price changes than simple moving averages.
Feature Vector	A structured numerical representation of market data, formed by applying basis functions to selected technical indicators, and used as input to the Q-function approximation.
Greedy-GQ	Designed to mitigate bias in off-policy learning by introducing a secondary weight update mechanism refining the weight vector iteratively to improve convergence stability, particularly in environments with stochastic market dynamics.
Linearly Weighted Moving Average	Uses a step-by-step weighting system that gives progressively higher importance to newer prices, in order to offer a middle ground between EMA's quick response and SMA's steady performance.
Markov Decision Processes (MDPs)	Offer an organized structure for consecutive decision-making in algorithmic trading, in which market situations (states), trading choices (actions), market changes (transition probabilities), and trading outcomes (rewards) are included.
Model-Free RL	Unlike model-based methods, it does not require a predefined model in order to allow agents to directly connect actions with rewards through real-time experience. They have computationally simpler process and are well-suitable for volatile and dynamic markets.
Modern Portfolio Theory (MPT)	Suggests that combining diverse assets can lower the total portfolio risk, while sustaining expected returns through diversification.

Off-Policy Learning	Updates action-value estimates using the Bellman equation without following a specific policy. It promotes robustness in unpredictable environments by learning from maximum future rewards.
On-Policy Learning	Adjusts value estimates on the basis of the agent's current action to ensure consistency with ongoing actions and strategies.
Q-Learning	An off-policy reinforcement learning method that adjusts action-value estimates through the Bellman equation, for which it uses an ε-greedy policy for choosing actions. It is well-known for striking a balance between exploring new options and exploiting known strategies.
Reinforcement Learning (RL)	A field of machine learning in which agents discover the best strategies by interacting with their environment, and learning through rewards and penalties along the way. Unlike supervised and unsupervised learning, RL focuses on maximizing rewards through trial and error.
Reward Function	Evaluates trading performance by calculating profits and losses based on entry and exit prices. The three versions for this study include: *minusMean,* which subtracts average reward from the most recent reward, *Immediate,* which measures the raw profit/loss from the current trade, and *mean*, which averages all trade profits up to the current trade.
SARSA	An on-policy algorithm that adjusts value estimates based on agent's real-time actions. The learning process reflects the agent's current behavior in order to stay aligned with the on-going policy and its outcomes.
Sharpe Ratio	Measures the excess return of an investment per unit of risk taken, compared to a risk-free asset. This ratio provides a standardized way to evaluate and compare the risk/reward outlook of different investment strategies.

Sortino Ratio	Measures how much extra return an investment generates for each unit of downside risk. It only considers negative returns in the risk calculation. This ratio serves as a useful performance measure for conservative investors who focus on limiting losses and protecting their capital.
State Representation	Involves turning raw market data into a structured format with technical indicators and a sliding window of length n. It captures price patterns over time. This setup provides the agent with historical context, allowing it to learn more effectively than it would from isolated data points.
Utility Theory	Models how individual investors think about their preferences and risk tolerance. It uses a mathematical utility function to weigh the pursuit of higher returns against avoiding risk. This framework breaks down decision-making under uncertainty. It helps to understand and predict investor behavior. This makes it possible to study and forecast investment choices when outcomes remain unknown.

Introduction

1.1 Background

The application of artificial intelligence (AI) and machine learning (ML) tools for trading in financial markets has gained considerable momentum in recent years (Meng & Khushi, 2019). At the same time that more is expected of technology, reinforcement learning (RL) has emerged. This growing area of ML has proven to be more reliable than traditional methods. The concept of RL has enhanced the forecasting performance of trading models by effectively processing and handling highly complex financial data (Millea, 2021).

Traditional supervised learning is designed to recognize patterns. Unsupervised learning is directed to capture clusters. In contrast, RL relies on sequential decision-making through trial and error (Sutton & Barto, 2018). RL's decision-making mechanism includes an intelligent agent who directly interacts with an environment. This agent learns from experience and receives either rewards or penalties.

The agent aims to continually adjust its actions and learn an optimal policy that maximizes the reward function (Singh et al., 2022). This self-directed learning, which does not include the intervention of external labels or engineered designs, is the key feature that explains RL suitability for financial trading applications where algorithmic decision-making is influenced by constant uncertainty and volatility (Park et al., 2024).

In financial trading, market data serves as the environment for the RL agent. The agent directly interacts with the environment and learns from a real-time feedback loop. Accordingly, the agent continuously updates its decisions to minimize investor sentiment influence and establishes more of an objective and adaptive algorithmic trading process (Liu et al., 2022). The final goal is to maximize performance over time. Hence, in comparison with fixed, pattern-based methods, reinforcement learning keeps automated strategies dynamic and capable of promptly discovering and adapting to new data (Liu et al., 2022). These characteristics allow the trading algorithm to respond to the changes in the market and achieve maximum returns, which contributes to the attraction of RL as an impressive adaptive algorithmic trading tool.

Embracing the full potential of machine learning advancements, algorithmic trading applies computer algorithms for automatic trades on the basis of investment rules and formulas (Zhang et al., 2019). To execute trades quickly, diversify over a large pool of assets, and deal with extremely large datasets, algorithmic trading is now an integral part of operations for investment banks, mutual funds, and hedge funds. According to Yahoo Finance (2022), algorithmic trading constitutes approximately 60–73% of the U.S. equity market's total volume. Nonetheless, the volatile nature of financial markets presents ongoing challenges, which makes the design of resilient models pivotal for generating stable returns, particularly for conservative investors (Vijaya et al., 2024).

Model-free methods such as Q-learning, Greedy Gradient Q-Learning (Greedy-GQ), and State-Action-Reward-State-Action (SARSA) are among the common branches of reinforcement learning (Corazza et al., 2019). They have become popular in algorithmic trading given their versatility and adaptability (Corazza et al., 2019).

Featuring both on-policy and off-policy mechanisms, model-free RL algorithms enable agents to directly learn from the market dynamics without requiring a predefined model of the environment. Instead, they embrace a learning process through which they make decisions in real-time. The ability to learn and adapt quickly establishes model-free reinforcement learning as a highly relevant tool for solving financial and investing challenges in highly volatile and shifting market dynamics (Sutton & Barto, 2018).

By leveraging reinforcement learning techniques, traders can use real-time data to respond more promptly to emerging market trends. In this view, RL's responsive adaptation to volatile and unpredictable market conditions enables traders to systematically integrate these methods into their decision-making processes. Therefore, they can enhance returns, reduce risk, and improve efficiency in algorithmic trading designs (Pereira, 2022).

1.2 Research Motivation

Model-free RL methods can mainly be helpful for conservative investors that value stability above aggressive growth (Van den Bergh-Lindeque et al., 2022). What makes these methods so useful is that they can control the downside risk under almost all market conditions (Vu et al., 2023). To cater to such investors, RL systems may be optimized with proper optimization metrics, enabling them to learn directly from the market surroundings and aim for stable returns with restricted volatility exposure (Ritter, 2017). This functionality is of the utmost importance for strategies that aim for volatility reduction and consistent return achievement (Van den Bergh-Lindeque et al., 2022).

Pereira (2022) tested the performance of the three model-free methods of reinforcement learning, namely Q-learning, Greedy-GQ, and SARSA, on the data from

futures contracts traded on the Brazilian stock exchange under multiple market conditions and over various sets of trading intervals. While his work provides valuable insights on the potential of such models being applied to a wide variety of investor profiles, it also reveals limitations on the applicability and reliability of such methods to achieve consistent returns on different regional markets and asset classes for conservative investors. His work requires more research to confirm and adjust these approaches. Consistently, there exists a wide gap in the literature regarding the exploration of model-free methods of reinforcement learning that work most effectively for investors who value return stability and reduced volatility.

Building on existing research, this work uses an alternative dataset and a different trading period with special emphasis on their applicability for risk-averse traders. The research investigates the Q-learning, Greedy-GQ, and SARSA methods under the revised parameters to validate and continue Pereira's work. In doing so, this work seeks to bridge the gap in the literature by providing an evidence-based, practical method that delivers more consistent returns with mitigated volatility for risk-averse investors.

Reinforcement learning, in particular model-free approaches such as Q-learning, have presented the potential to be more profitable than traditional machine learning and deep learning models. Specifically, research has shown when applied to forecasting stock trends, they can suggest favorable trades without assuming a predetermined model of the market environment (Prasad & Seetharaman, 2021; Kolm & Ritter, 2024).

However, the performance of these approaches can diminish dramatically when hyperparameters are not chosen optimally, and may provide suboptimal returns accompanied by greater risk, especially for conservative investors.

1.3 Problem Statement

Some model-free reinforcement learning techniques can fail to generate consistent returns for risk-averse investors in algorithmic trading (Enkhsaikhan & Jo, 2024), resulting in up to 73.34% lower return over a one-year trading period (Pereira, 2022).

Although model-free reinforcement learning methods exhibit responsiveness to changes in dynamic environments, they have limitations when applied to algorithmic trading for risk-averse investors who expect stable returns. Enkhsaikhan and Jo (2024) note that some model-free RL methods underperform under the uncertainty of small data as well as unstable market environments that could interfere with their optimal convergence. Pereira (2022) illustrates that some model-free RL methods could result in 73.34% lower returns over a one-year trading period when not tailored for conservative investors, indicating that RL models might not adapt market environments when stable returns with lower volatility are the goal.

Without the specific optimization criteria, model-free RL methods such as Q-learning could not achieve an optimal balance between reward and risk, and would thus not be optimal for conservative behavior within trading environments (Ritter, 2017). This highlights the importance of deriving an optimized model-free RL method that will resist the volatility of the market and generate stable performance for conservative investors.

Reinforcement learning, especially model-free approaches such as Q-learning, has outshined conventional machine learning and deep learning approaches to forecasting stock trends with the capability for generating trading signals without using predefined market models (Prasad & Seetharaman, 2021; Kolm & Ritter, 2024). Nevertheless, when

the hyperparameters are poorly chosen, the performance of such approaches could significantly decline, which can cause unstable returns with higher risks for the conservative investor.

1.4 Thesis Statement

An optimal model-free reinforcement learning technique is required that can generate consistent profits with lower volatility for risk-averse investors in algorithmic trading.

This study develops a predictive RL model for risk-averse investors as an academic and industry benchmark for replication, stress testing, and optimization of financial trading strategies. By adapting dynamically to the conditions of the market, the RL methods presented here seek to maximize profitability with efficient management of risk. As a model-free RL tool, the implementation of this approach by risk-averse investors will allow for better management of volatility in a variety of asset classes and market states, with a stable and adaptive strategy for institutional and retail investors alike to pursue stable returns with minimized risk.

1.5 Research Objectives

This praxis has four primary objectives:

1. To determine the most predictive hyperparameters of future performance among model-free reinforcement learning methods, including Q-Learning, Greedy-GQ, and SARSA, for algorithmic trading.
2. To evaluate if the Q-learning algorithm can generate a positive return that exceeds the NASDAQ-100 index benchmark when making trades on E-mini-NASDAQ-100 futures through algorithmic trading.

3. To establish the most appropriate model-free method of reinforcement learning for generating low-volatility profits for risk-averse investors in algorithmic trading.
4. To confirm, compare, and draw conclusions from earlier research, using model-free reinforcement learning methods on different datasets and time periods. The goal is to identify the optimal set of hyperparameters in order to design algorithmic trading rules that are suitable for risk-averse investors.

1.6 Research Questions and Hypotheses

This research explores the application of model-free reinforcement learning approaches, including Q-Learning, Greedy-GQ, and SARSA, in algorithmic trading in order to improve the trading performance for risk-averse investors. As such, the study tests the following research questions:

RQ1: Which hyperparameters are the most indicative of future performance in model-free reinforcement learning techniques, including Q-Learning, Greedy-GQ, and SARSA, for algorithmic trading?

RQ2: Can the Q-learning algorithm generate a positive return that exceeds the NASDAQ-100 index benchmark when trading the E-mini-NASDAQ-100 futures through algorithmic trading?

RQ3: Which model-free reinforcement learning technique among Q-Learning, Greedy-GQ, or SARSA is the best at generating low-volatility profits for risk-averse investors through algorithmic trading?

The study also investigates various hyperparameter combinations and the reinforcement learning models to see which one of the settings provides the most stable performance in algorithmic trading for the conservative investor. Accordingly, the research hypotheses include:

H1: In model-free reinforcement learning techniques for algorithmic trading, the hyperparameters, including basis function (hypTanh), look-back period (5), and reward function (minusMean), are the most indicative of future performance.

H2: The Q-learning algorithm can generate a positive return that is at least 50% greater than the benchmark returns of the NASDAQ-100 index, when trading the E-mini-NASDAQ-100 futures through algorithmic trading.

H3: Q-learning, as a model-free reinforcement learning technique, is the best at generating low-volatility profits for risk-averse investors in algorithmic trading.

1.7 Scope of Research

The purpose of this study is to examine the application of model-free RL approaches including Q-Learning, Greedy-GQ, and SARSA for algorithmic trading that can be appropriate for risk-averse investors. The study focuses on the examination of these approaches under various market conditions with respect to their effectiveness in obtaining consistent and low-volatility returns.

The historical data from the E-mini-NASDAQ-100 futures are employed to train and test the models, with the high liquidity and volatility of these data providing an ideal dynamic environment for algorithmic strategies (Ates & Wang, 2005; Carlson, 2024). As

the representative of the largest 100 technology-focused stocks traded on the NASDAQ stock exchange, the NASDAQ-100 index is selected as the benchmark for comparison because it is a well-known and reliable indicator of the performance of technology-oriented portfolios (Banerjee, 2022).

Various hyperparameter choices are tested to determine which work best to achieve the most stable performance for the risk-averse methods. The one-year duration from May 2023 to May 2024 exposes the project to varied market conditions, including global economic disturbances and the technological AI-powered boom (Straits Research, 2023), and the performance of the algorithms under such conditions is tested for resilience.

Although additional data would be advantageous, with limited computing time and resources available, the study is centered on the investigation of the use of RL models to produce the lower volatility returns for risk-averse investors during this period of trading. The conclusions of this research will provide insights into the optimal RL models and hyperparameter choices that financial institutions and retail investors can use to maximize risk management in algorithmic trading. Future work can extend this framework to examine its performance across various asset classes and general market conditions.

1.8 Research Limitations

Some of the potential limitations of this research are that the analysis is limited to historical data from the Chicago Mercantile Exchange that were obtained from a vendor and does not take live market conditions into account. It is worth noting that live

trading environments could have variance and complexity introduced by market shocks and imbalances that do not come into play with this research (Lopez de Prado, 2018).

Secondly, this study is strategically limited to three specific model-free methods of reinforcement learning, namely Q-Learning, Greedy-GQ, and SARSA, because of their unique characteristics and applicability in the current literature for risk-averse algorithmic trading (Pereira, 2022; Corazza et al., 2019; Ritter, 2017). In order to take advantage of both exploration and exploitation features embedded in the functionality of models, these three models have been chosen to achieve a representative analysis of the well-known RL methods employed for algorithmic trading.

Third, this research only considers the asset class of E-mini-NASDAQ-100 futures for one year of trading. While this provides a perspective for measuring model performance, it limits the scope of the research to one asset class of financial instrument and one time horizon. Consequently, conclusions derived might not entirely generalize across asset classes or timeframes, e.g., equities, commodities, or longer-term investments.

The main purpose of this praxis is not the provision of a ready-to-use trading strategy. Instead, it presents a new data-driven analysis of the reliability, adoptability, and potential profitability of reinforcement learning models in risk-averse algorithmic trading for a less studied asset class in the literature. This work aims to narrow the gap between exploration in theory and implementation in practice and exemplifies the potential of the considered models. These models would require additional fine-tuning and validation for live trading implementation, but this research establishes the basis for subsequent development.

1.9 Organization of Praxis

The rest of this praxis is structured as follows: Chapter 2 provides an overview of literature that offers an extensive survey of the previous research on algorithmic trading and reinforcement learning. The historical development and background of these research areas and their applicability to this praxis topic are explored. Chapter 3 discusses the research methodology, including the sources of the data, the data-processing steps, development of the models, and hyperparameter tuning approach. It also discusses the process of measuring the performance of the models and the process of verifying the research hypotheses.

Chapter 4 presents findings from experiments that were carefully conducted in order to explore the best combinations of hyperparameters and their effects on the performance of the three model-free reinforcement learning methods. Chapter 5 summarizes key findings by combining all aspects of the research and evaluating the impact of reinforcement learning on algorithmic trading, particularly for risk-averse investors. The chapter then discusses how this praxis contributed to the existing body of knowledge. It concludes with future research recommendations, so that the methods discussed can be further fine-tuned for real-world trading.

Literature Review

2.1 Introduction

This chapter reviews recent research on reinforcement learning (RL) in financial market trading. As one of the most rapidly evolving fields in finance, algorithmic trading uses computer programs and algorithms to automate strategies, analyze data, and execute trades (Lopez de Prado, 2018). What's particularly evident is that recent advances in artificial intelligence (AI) and machine learning (ML) have essentially transformed these capabilities, which lead to more flexible trading strategies, to say the least (Millea, 2021).

RL is a type of machine learning that trains agents to maximize rewards by interacting directly with their environment through trial and error (Jeong & Gu, 2024). RL is especially effective for creating trading strategies that balance risk and reward. Its iterative process allows agents to respond to changing market conditions in real time (Jeong & Gu, 2024). However, selecting the ideal RL techniques and hyperparameters can help achieve stable, low-volatility returns, though this can be challenging.

Section 2.3 then explores the significance of RL in algorithmic trading, particularly highlighting the advantages and disadvantages of specific algorithms, namely Q-learning, SARSA, and Greedy-GQ for risk-averse investors—because these algorithms each handle risk-reward tradeoffs differently. Moving on, **Section 2.4** examines existing literature that has applied these models across various market environments, in the process uncovering key insights and, more importantly, identifying notable research gaps.

In light of these findings, **Section 2.5** brings everything together by summarizing the results and, in essence, connecting the identified gaps to the overarching goals of this praxis.

2.2 Reinforcement Learning (RL)

Reinforcement learning (RL) is a machine learning discipline in which the optimal strategies of the agent are learned when the agent interacts with the environment and receives rewards or penalties on the basis of the actions they undertake (Sutton & Barto, 2018). This approach differs from supervised learning, where instances are labeled, and from unsupervised learning, where patterns are learned without the provision of explicit feedback (Fischer, 2018).

Reinforcement learning (RL) operates on the fundamentals of critical components including a state space that reflects states of the environment, an action space that defines the possibilities of decisions, and value functions that assess options in order to make the next choices (Sutton & Barto, 2018). Conventionally, RL earned appreciation for its efficiency in resolving game-like challenges, for example, for playing Atari games where the aim is to maximize a given score (Hirsa et al., 2021). In such applications, optimal actions are learned by the agent through trial and error with accumulative repetitions, iteratively modifying their decisions with respect to rewards obtained (Jeong & Gu, 2024).

2.2.1 RL Application in Financial Markets

RL diversified from game applications to the financial trading world whose market conditions could be compared to dynamic environments with unstable states and unpredictable behavior (Hirsa et al., 2021). The volatile and dynamic nature of the

financial market makes RL's exploration and adaptation traits beneficial in the formulation of trades that could withstand market volatility and maintain their performance in the long term (Guo et al., 2024). RL treats investment decisions as problems of stochastic control, usually represented as Markov Decision Processes (Hirsa et al., 2021). Contrary to traditional models that use static predictions from historical or predicted data, RL-based strategies focus on learning from continuous market experience (Guo et al., 2024).

Figure 2-1 illustrates the interaction of an RL agent with the market environment in which the agent gathers market state (e.g., price) and reward (e.g., profit or loss) feedback, and responds with appropriate actions (e.g., buy, sell, or hold).

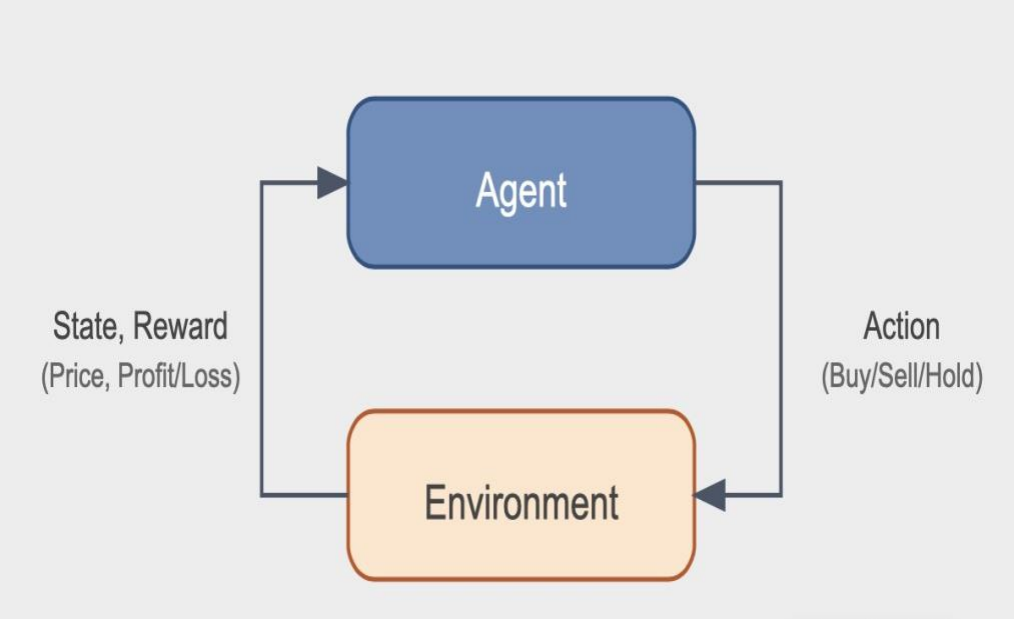

Figure 2-1 RL Agent and the Market Environment

2.2.2 Theoretical Foundations for RL Application in Financial Markets

RL in financial markets relies on a number of underpinning theories that supply the mathematics and concepts necessary for decision-making under uncertainty (Jeong &

Gu, 2024; Hirsa et al., 2021; Kolm & Ritter, 2024; Ritter, 2017). The most common and documented theories in the literature will be discussed. These theories, including Markov Decision Processes (MDPs), Bellman's Principles, Utility Theory, and Modern Portfolio Theory (MPT), offer significant determinants of the application of RL in the financial markets (Jeong & Gu, 2024; Hirsa et al., 2021; Kolm & Ritter, 2024; Ritter, 2017).

2.2.3 Markov Decision Processes (MDPs)

Markov Decision Processes (MDPs) include a formal model for sequential decision-making for algorithmic trading. Looking at the technicality of the model, it includes key components that are defined as states (market conditions), actions (trading decisions), probabilities of transition (market change), and rewards (trading returns) (Giorgi et al., 2023). According to the fundamentals of the model, an RL agent perceives the world as a state, chooses an action, receives a reward, and moves to the next state, basically repeatedly going through the cycle to form an interaction trajectory (Hirsa et al., 2021).

The aim is to maximize future rewards cumulatively. This structure does indeed accommodate well the notion of trading in liquid markets such as futures contracts, in which states can consist of historical price patterns, volume patterns, and technical indicators, whose actions typically comprise buy, sell, or hold choices (Hirsa et al., 2021). One of the main aspects of MDPs is that they support RL agents in dealing with the sequential aspect of trading. In this view, they capture the dynamics of the interactive behavior between immediate actions and subsequent opportunities within an unpredictable environment so that optimal decisions can be made over time (Giorgi et al., 2023).

2.2.4 Bellman's Principles of Optimal Control

Building on MDPs, Bellman's Optimal Control Principles (1957) introduced a new paradigm for sequential decision-making. Bellman's work emphasizes the application of value functions to evaluate the long-run implications of decisions versus immediate rewards. Applying the principles of Bellman in the context of trading, Bellman's principles of short-term profit opportunities equilibrating with long-term strategic objectives play significant roles in RL-based trading systems design (Giorgi et al., 2023).

While MDPs provide the structure, Bellman's principles supply the mathematics for the optimal realization of objectives through sequential decisions. Unlike the traditional approaches that make known relations between actions and effects, RL agents deduce these relations from experience, adapting Bellman's principles to the dynamic, complex nature of markets (Ritter, 2017).

2.2.5 Utility Theory

While MDPs and Bellman's principles establish the structure and the frameworks of evaluation for decision-making, Utility Theory introduces the behavioral aspect by capturing individual investor preferences and risk tolerance (Fishburn, 1979). The theory accounts for the manner in which investors weigh the desire for greater returns against aversion to risk, measuring this in mathematical utility functions (Levy et al., 2011; Wan et al., 2024). The functions provide the connection between theoretical models and actual decision-making by transferring human preferences for risk into algorithmic rules, keeping the trading strategies consistent with investor priorities even under unstable

market conditions (Fishburn, 1979). This aspect of the aversion to risk is essential for creating RL agents that act according to investor preferences (Wan et al., 2024).

2.2.6 Modern Portfolio Theory (MPT)

Modern Portfolio Theory (MPT), introduced in 1952 by Harry Markowitz, targets the maximization of the risk-return profile of a portfolio (Markowitz, 1952). MPT assumes that investors will diversify and thus reduce the overall risk without compromising the potential returns by investing in several asset classes. This theory has had a great influence on the management of investments, offering a source of dynamic trading approaches and algorithmic implementation intended to maximize the expected utility of wealth of risk-averse investors (Merli et al., 2021). It also affects industry practice and research in quantitative finance since it underlies the construction of models and approaches that mix return optimization with the management of risks (Levy et al., 2011; Ritter, 2017). MPT's principles extend to individual asset instruments, including E-mini-NASDAQ-100 futures (Fassas, 2021), in algorithmic trading. Investors make rational, risk-averse choices, Markowitz (1952) suggests, and while this may not always be the case, successful long-term investing suggests that this will generally work (Ritter, 2017).

2.2.7 Convergence of Theoretical Foundations

The combination of MDPs' structured decision-making, Bellman's optimal control, Utility Theory's modeling of preferences, and MPT's optimization of risk and return, creates a basis for the application of RL to financial markets (Wan et al., 2024; Giorgi et al., 2023; Ritter, 2017; Merli et al., 2021). All the theories provide an essential element to RL's structure:

- MDPs structure the RL agent's experience to an ordered state-action-reward cycle.
- Bellman's principles facilitate long-term decision analysis using value functions.
- Utility theory directs the RL agent's value estimation.
- MPT determines the basis for offsetting risks and returns.

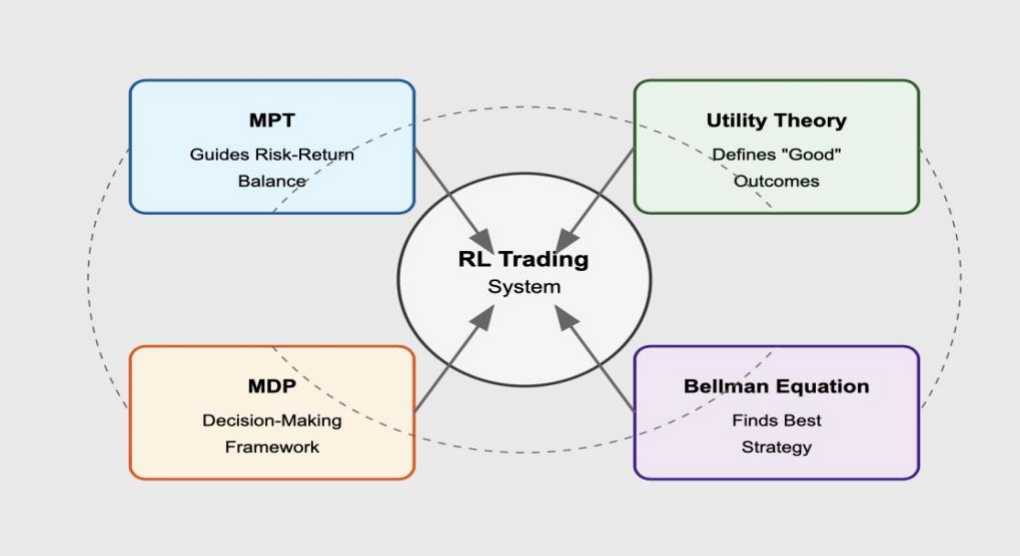

Figure 2-2 Theoretical Framework Integration in RL Trading System

RL agents learn optimal behavior through interaction with the market itself, with all of the theoretical constructs being incorporated into the process of learning (Wan et al., 2024). The state-action-reward cycle is formalized by MDP, Bellman's principles update learning control, the structure of reward is determined by utility functions, and control over the behavior of return and risk is provided by MPT (Giorgi et al., 2023; Ritter, 2017; Merli et al., 2021).

Theoretical underpinnings supply the foundations for interpreting decisions over investments and for managing risks, but implementation in contemporary markets calls

for high-level computation (Jeong & Gu, 2024). This especially goes for RL approaches in algorithmic trading since decisions must happen with speed in dynamically shifting market conditions so that the agent may modify and adjust on the basis of information derived in real time (Jeong & Gu, 2024).

The combination of classical theories of finance with contemporary machine learning methods poses the core question: Can reinforcement learning methods, rooted in those theoretical premises, replicate and even beat the choice patterns of rational, risk-averse investors in dynamic market conditions? The subsequent portion presents a comprehensive overview of RL in algorithmic trading and then comparatively discusses traditional ML as well as human trader methods. This then leads into RL classification and the application of particular models appropriate for risk-averse investors, with an analysis of the research gaps that lays the basis for this praxis.

2.3 RL in Algorithmic Trading

The application of RL in the financial markets has proven to yield positive results in algorithmic trading environments with regard to optimal timing of trades for maximum returns, minimizing risks, and maximizing efficiency (Jeong & Gu, 2024; Hirsa et al., 2021; Pereira, 2022). In this case, an RL agent decides whether to buy, sell, or hold according to the market states and responds with reward (profit or loss), thus learning and refining its trading strategy to effectively respond to dynamic market conditions (Sun et al., 2023). Using the RL power to adjust instantly, traders can achieve longevity of performance within a dynamic environment as well as strive for long-term profitability over an extended duration (Pereira, 2022).

RL traders form decisions in the same manner that an expert game player adjusts in response to changing circumstances (Hirsa et al., 2021). Trading decisions that work like successful game play, with the connotation of combining short-term actions with long-term goals, lead RL traders to continuously adjust their strategies during market interaction, learning to make the optimal decision for every time frame and for every trading cycle (Pereira, 2022; Ritter, 2017). Similar to the theoretical constructs of the above discussion, these kinds of agents employ value functions to estimate and achieve a balance among short-term trading opportunities and the overall aim of long-term profitability for algorithmic trading (Aloud & Alkhamees, 2021).

2.3.1 Comparative Performance of RL and Traditional ML Models

In comparison to traditional machine learning models, where the relationships between actions and outcomes are fixed, RL agents acquire these relationships directly from their interaction with the market data (Sutton & Barto, 2018; Ritter, 2017). RL tries to solve two main challenges: accurately representing market conditions and taking the optimal actions under those conditions (Hirsa et al., 2021). This is in fact consistent with RL's principle of maximum accumulated reward for intelligent decision-making in which there is no explicit supervision or hand-coded rules (Abu Mallouh et al., 2023; Pereira, 2022). Even though the exhaustive comparison of classical ML approaches with RL approaches is not within the scope of this praxis, recent studies which are examined in the subsequent section establish the feasibility of RL-based approaches when it comes to constructing dynamic algorithmic trading strategies.

Recent work by Ngo et al. (2023) offers compelling evidence that RL outperforms existing traditional portfolio optimization approaches as well as other machine learning

methods on a consistent basis. They compare RL with deep portfolio models. To prove their point, they use a variety of models including Long Short-Term Memory networks (LSTM), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Artificial Neural Networks (ANN).

In two rigorous experiments in which the authors took various market conditions with Vietnam's VN30 stocks and U.S. ETFs into account, they illustrate that RL-based models exhibit more favorable performance as well as risk control (Ngo et al., 2023). RL had the highest daily return of 0.32% against deep portfolio's 0.24% with almost similar volatility (2.94% vs. 2.69%), and this implies the better Sharpe ratio of 3.79 compared to 3.36 (the Sharpe ratio calculates risk-adjusted returns, with larger values denoting better performance compared to risk).

The Sharpe ratio, named after William Sharpe (1966) in his seminal work on "Mutual Fund Performance," is a well-established metric for investment analysis. The Sharpe ratio is a standardized measure of excess return per unit of risk that allows for proper comparison across investment strategies (Sharpe, 1994). The appropriateness of the Sharpe ratio for measuring return and volatility makes the Sharpe ratio an instrumental tool for the analysis of strategies intended to produce consistent, risk-adjusted performance (Hirsa et al., 2021).

RL generated a return of 0.07% with a standard deviation of 2.27% in the U.S. ETFs experiment (Ngo et al., 2023). Holt's smoothing posted the highest Sharpe ratio of 1.5392, yet RL provided stable performance with a Sharpe ratio of 1.3457, performing better than the majority of the other methods. Of particular mention, the deep portfolio strategy performed poorly under this kind of market environment with returns of -0.01%

and a negative Sharpe ratio of -0.1858. The stable performance of RL under all economic environments is attributed to its unique trait of maintaining well-diversified portfolios and rebalancing dynamically through interaction with the market (Jeong & Gu, 2024).

In addition, RL exhibited consistency under different market conditions, recording a total return of 67% compared to the 46% of deep learning, with the rest of the ML models performing poorly, especially under negative market conditions. This consistent superiority draws from RL's inherent characteristic of keeping well-diversified portfolios with dynamic reallocation during market interaction (Jeong & Gu, 2024).

Evidence of RL's enhanced financial trading performance is also provided by a systematic review of the literature conducted by Prasad and Seetharaman (2021), which canvassed over 50 papers from the literature on financial machine learning. The review revealed that although typical machine learning methods such as LSTM were good at the accuracy of the prediction, RL methods were better at profitability and Sharpe ratios compared to these methods. As a case in point, RL approaches had annual returns of 16.3% with more favorable stability. This indeed demonstrates the capability of the models for profit-oriented optimization, which makes them sensibly better than alternate approaches such as SVM and XGBoost that do not provide for real-time decision-making and that do not generalize well to out of sample instances (Prasad & Seetharaman, 2021; Carapuco et al., 2018). The study also emphasized the advantage of RL under different market environments, in which the deep RL models managed to deliver Sharpe ratios of above 5.34 (Yong et al., 2017), directional accuracy of over 76% (Zeng & Khushi, 2020), and net profits of 66% (Ponomarev et al., 2019).

Putting all the findings into perspective, the work emphasizes RL's inherent ability to learn directly from market experience and not from observing past patterns of trends as is typically the case in traditional ML approaches. This unique capability enables the algorithm to perform well regardless of adverse market conditions and volatility. That is to say, the study supports RL's effectiveness compared to traditional methods of ML under both stable and adverse market conditions, which allows the algorithm to maintain performance regardless of how the market behaves during stressful market conditions and high volatility (Prasad & Seetharaman, 2021).

The superiority of RL for financial applications is further supported by evidence from a thorough review by Rundo et al. (2019), analyzing the progression of quantitative finance from conventional statistical methods to state-of-the-art machine learning methods. From their analysis, they find that although traditional autoregressive methods such as ARIMA and initial machine learning methods performed modestly well under stable market conditions, they have not fared well with the nonlinear and non-stationary nature of financial markets, which usually consist of volatile and unpredictable patterns (Rundo et al., 2019).

In contrast, RL demonstrates a unique strength in handling these complexities. This is evident as the research highlights that RL implementations, such as those utilizing LSTM, achieved annualized risk-adjusted returns of over 5.8 on S&P 500 data (Rundo et al., 2019). What is important to note here is the fact that this measure significantly surpasses industry benchmarks, given that a ratio above 2.0 is typically considered very good, which essentially translates to highly favorable risk-adjusted performance (Corporate Finance Institute, 2025).

Accentuating a key point, the study showcases that RL's adaptability stems from its ability to learn directly from market interactions rather than relying on static historical patterns. This proves to be particularly effective for algorithmic trading environments, particularly for those in which rapid market shifts are the norm (Rundo et al., 2019). Being typically static in nature, traditional ML techniques tend to perform poorly in the financial markets, and therefore, are not adaptable to the quick changes in the environment (Ngo et al., 2023). Despite RL's proven potential to provide a dynamic alternative that could outperform both conventional ML and human-operated trading strategies, there remains a pressing need to further solidify its efficacy through additional research (Ngo et al., 2023; Rundo et al., 2019).

2.3.2 Human Trader Approaches

Human traders use two major methods: fundamental and technical analysis. Fundamental analysis estimates the inherent value of a stock on the basis of economic factors and financial data, whereas technical analysis relies on past price and volume patterns for anticipating the future (Shleifer, 2000; Murphy, 1999). These two human-oriented methods, though, have limitations in the fast-changing and complicated markets of the current era as they do not allow traders to respond quickly to abrupt market changes and fluctuating volatility (Lo et al., 2000).

Growing research on RL suggests that it can overcome the aforementioned constraints by learning from the market itself and adapting dynamically in real-time (Prasad & Seetharaman, 2021). The judicious choice of an RL model for risk-averse investors could increase profitability with low volatility, and RL can emerge as a desirable addition to algorithmic trading (Ritter, 2017).

2.3.3 Categorization of RL Models

The general classification differentiates between two approaches, model-based (MB) and model-free (MF) (Wyckmans et al., 2022; Sutton & Barto, 2018). Model-based RL implies learning an explicit model of the dynamics of the environment, such that the agent can simulate what will happen in the future and make decisions considering long-term effects. This makes goal-directed behavior and responsiveness to change possible (Sutton & Barto, 2018).

However, capturing the realism of financial markets becomes highly challenging as they are random and unpredictable (Prasad & Seetharaman, 2021). Model-free RL, on the other hand, eliminates the need for an explicit model because the agent can relate actions to rewards from experience in real-time (Sutton & Barto, 2018). This is less computationally intensive and appropriate for rapidly-changing, complicated markets (Hirsa et al., 2021).

Apart from the general classification of model-free and model-based methods, RL methods tend to be grouped into three categories depending on their structure: critic-only, actor-only, and actor-critic approaches (Fischer, 2018). The specification focuses on the various ways that RL methods learn and make decisions, especially in portfolio management and algorithmic trading. Critic-only approaches aim to learn value functions to make an estimation of the expectation of reward after an action takes place in a particular state, in order to inform the decision-making process according to the value estimates (Hirsa et al., 2021).

Actor-only approaches directly learn a policy that translates states into actions without actually estimating value functions, normally with methods such as policy gradients (Fischer, 2018). Actor-critic approaches blend both sides together, with the actor suggesting actions and the critic scoring them, enabling more subtle policy updates from feedback (Hirsa et al., 2021).

In this praxis, model-free, critic-only approaches are of special interest since they only aim to learn value functions and future reward estimates for actions in states without the requirement of an explicit environment model (Hirsa et al., 2021; Fischer, 2018). The agent, in these methods, criticizes possible actions by their expected future rewards (Rundo et al., 2019).

2.3.4 Model-Free, Critic-Only Approach

The use of the model-free, critic-only approach is justified on the basis of the complexity of the financial markets on the grounds that:

- Resilience and Simplicity: The algorithm would easily apply and resist instability and divergences that might beset complex algorithms (Giorgi et al., 2023).
- Flexibility of reward functions: Traders can shape reward functions to include their particular goals and constraints so that the agent's learning process better aligns with real-world trading considerations (Ritter & Kolm, 2024; Ritter, 2017).
- Adaptability: The agent may adjust to new environments by repeatedly updating its value estimates from feedback derived from the rewards that have actually occurred, without depending on an often flawed representation of the world (Giorgi et al., 2023).

- Avoidance of Model Bias: Since the agent does not rely on an explicit model of market behavior, it does not incur biases that may stem from flawed or overly simplified models (Giorgi et al., 2023).

In summary, the model-free, critic-only approach is suitable for the formulation of financial market trading strategies. Its inherent merits serve the needs of financial applications, where there exists a requirement for flexibility, reliability, and adaptability (Rundo et al., 2019; Giorgi et al., 2023). Fundamental to this approach are model-free methods such as Q-learning, SARSA, and Greedy-GQ since they learn from experience through the environment directly and not from a given model (Giorgi et al., 2023; Pereira, 2022; Corazza et al., 2019).

- **Q-learning** is a model-free and off-policy algorithm for learning the optimal action to pursue in every situation by modifying its value estimates from the rewards it receives, and the maximum estimated future rewards (Sutton & Barto, 2018). The policy of an agent refers to the rules governing the decisions of what actions to take. Action values in Q-learning are determined independently of the followed policy so that the algorithm can learn from experience without the influence of the employed strategy (Sutton & Barto, 2018).
- **SARSA (State-Action-Reward-State-Action)** is an on-policy, model-free algorithm that updates value estimates from the actions being performed (Sutton & Barto, 2018). The learning process of SARSA reflects the current behavior of the agent directly in comparison with the Q-learning process (Sutton & Barto, 2018).

- **Greedy-GQ** improves on Q-learning by solving issues in high-dimensional environments, including approximating values in high-dimensional states and stability issues in off-policy learning (Sutton & Barto, 2018). Its structure also ensures that large numbers of actions and variables are handled, so it can work well with many options and continuous data (Sutton & Barto, 2018; Pereira, 2022; Corazza et al., 2019).

Overall, all these three approaches (Q-learning, SARSA, and Greedy-GQ) can be employed for financial trading. That is, they can learn from experience, adapt to the nature of the market, and make decisions about the available actions, which make them valuable tools for the development of trading strategies that respond to the volatility and uncertainty of financial markets (Giorgi et al., 2023; Corazza et al., 2019).

Nonetheless, one of the significant limitations of these model-free methods is that they tend to use discrete action spaces, which become computationally intensive when the possible actions increase, especially when it comes to managing multiple assets or implementing advanced strategies (Aloud & Alkhamees, 2021). Notwithstanding such a shortcoming, these algorithms continue to provide a potential alternative for the construction of trading algorithms that compare outcomes on the basis of learned value functions, without the extensive modeling of market dynamics (Giorgi et al., 2023; Aloud & Alkhamees, 2021; Corazza et al., 2019). The subsequent part of this work outlines the strengths and weaknesses of each algorithm.

2.3.5 Advantages and Disadvantages of Each Algorithm

- **Q-Learning**

  Advantages (Sutton & Barto, 2018):

- Flexible in handling many conditions without fixed commitment to a specific policy, which allows for resilience under unstable settings.
- Updates from maximum future rewards, which may help to capture potential gains when conservatism needs to be applied, with no additional risk from the impact of the policy.

Disadvantages (Hilpisch, 2024):

- Off-policy behavior can impose inconsistency when there's a high departure of conditions from the conditions of training because it may overfit to conditions that don't represent real-world behavior of the agent.
- Overestimation of value functions under unstable conditions, which may lead to misaligned decisions or untrustworthy value estimations.

- **SARSA (State-Action-Reward-State-Action)**

  Advantages (Yao et al., 2024):

  - It keeps the congruence with the actual behavior of the agent, which renders SARSA congruent with actual actions and policies.
  - Reduced overestimation errors because it develops policies from actual experience from actions taken, which yields stable performance in anticipated situations.

  Disadvantages (Zhao et al., 2016):

    - May miss high-reward opportunities because of learning only from experience, limiting exploration of potentially rewarding options.
    - Slower adaptation to quickly evolving conditions since updating only occurs with the current policy and can hinder rapid adjustments to new situations.

- **Greedy-GQ**

  Advantages (Ma et al., 2021):

    - Suitable for stable learning in high-dimensional spaces, therefore ideal for situations with multiple variables or dimensions.
    - Continuous action space management reduces the likelihood of overfitting, which can yield stable behavior under diverging conditions.

  Disadvantages (Sutton & Barto, 2018):

    - Dependent on tuning of parameters, which may become challenging in highly random environments.
    - Computationally expensive, potentially limiting its use for real-time decisions, especially when there are large action spaces or updates that come continuously.

The next section will review existing research that has utilized Q-Learning, Greedy-GQ, and SARSA, elucidating their individual contributions to the formulation of trading strategies and the forecasting of the markets. The next section will also shed light on the limited research that has employed the three methods, laying the groundwork for the purpose of this praxis. It is important to note that there exist contending schools of

thought on the appropriateness of the given models for risk-averse investors that support the rationale for verification of existing knowledge (Giorgi et al., 2023; Pereira, 2022; Aloud & Alkhamees, 2021; Corazza et al., 2019).

While literature that directly estimates the risk-averse applicability of every model continues to be limited, there exists a common opinion that Q-Learning would better suit risk-averse approaches (Aloud & Alkhamees, 2021; Pereira, 2022; Ritter, 2017). This assumption, however, leaves room for future work, reflective of the importance of comparative analysis to confirm the current viewpoints and complement the analysis with extra measures of risk, another gap that this research endeavors to fill.

2.4 Discussion of Current Research

Aloud and Alkhamees (2021) applied Q-learning to the major US stock indices such as the S&P 500, NASDAQ, and Dow Jones for a period of five years from 2015 to 2020. The goal of the study was to optimize trading strategies by dynamically responding to the shifts in the market through Q-learning. Additionally, a standard Direct Reinforcement Learning (Direct RL) model without any special Q-learning adaptations was employed as a baseline for benchmarking.

The Q-learning algorithm learned the market states that dictated buy, sell, and hold choices, allowing it to properly respond to multiple market states (Aloud & Alkhamees, 2021). Outputs revealed that Q-learning consistently had higher returns as well as Sharpe Ratios than those of Direct RL. This highlights the efficiency of Q-learning in managing the risk and reward in fluctuating trading markets. With the use of immediate reward function, the study underscored the key role of reward functions in improving overall trading performance (Aloud & Alkhamees, 2021).

In an earlier study, Kim and Yang (2007) employed Q-learning in their MQ-Trader model on the Korea Composite Stock Price Index (KOSPI 200), with emphasis on major stocks listed on the Korean stock exchange. This work offered insights into the suitability and applicability of Q-learning for risk-averse investors when trading in stock markets. They used asset growth as a measure of return, analyzing the asset increase during the test period compared to a benchmark market index, the KOSPI index, in order to compare the MQ-Trader performance. The study showed that MQ-Trader recorded a 1138.7% asset growth over the course of the experiment (Kim & Yang, 2007).

In addition, MQ-Trader's multi-agent system had separate buy and sell decision-making agents with representations of states. These representations included the long-term price history and profit/loss factors (Kim & Yang, 2007). Using independent buy and sell decision-making, the agents ensured that profit margins on sales are considered during selling, which in return reduced the impact of losses in volatile markets (Kim & Yang, 2007). The empirical findings were quite promising as they showed an asset increase even though bear markets persisted. Needless to say, this is precisely the kind of outcome that would appeal to risk-averse investors.

Ritter (2017) offers a framework for implementing Q-learning in multi-period trading. As shown by his study, Q-learning can efficiently manage risk-averse trading by optimizing expected final wealth utility while avoiding mere profit maximization. A customized reward function, incorporating both risk and profit considerations, frames the model according to the goals of a risk-averse investor so that it can modify its strategy to optimize utility despite the presence of transaction costs (Ritter, 2017). Through a simulated market setting, Ritter demonstrates how a Q-learning agent can learn

adaptively to have statistically preferable results honoring the complexity of risk-averse preferences.

In a subsequent refinement, Kolm and Ritter (2024) introduced a new reward function for striking a balance between returns and risk volatility to satisfy risk-averse investor requirements in complex market environments. Being mean-variance based, this new reward function penalizes high variance and enables the Q-learning agent to target stable, risk-adjusted returns. With this systematic reward approach, Kolm and Ritter (2024) highlight Q-learning's applicability in administering risk-adjusted strategies within financial decision making, demonstrating its efficiency in harmonizing reward and risk (Kolm & Ritter, 2024). Their study improves on previous findings and explores the possibilities of RL in trading, particularly by demonstrating how Q-learning, with an appropriate reward function, can meet risk-averse investor needs.

In their study, Giorgi et al. (2023) introduce the implementation of SARSA, specifically, for trading commodities. According to the study, SARSA algorithm adjusts to non-linear price behaviors. This adjustment was especially useful for trading West Texas Intermediate (WTI) oil whose price pattern exhibits non-linear trends (Giorgi et al., 2023). Using SARSA, trading decisions were optimized within a continuous state-action space, which allowed for higher flexibility of asset positioning (Giorgi et al., 2023). SARSA showed less variability of wealth than the Markowitz strategy when faced with negative price movement (Giorgi et al., 2023). Crucially, when compared to Markowitz models, SARSA, besides exhibiting adaptability when faced with high volatilities, was also better at controlling downside risks, making it an attractive candidate for the risk-averse case (Giorgi et al., 2023).

Corazza et al. (2019) compared SARSA, Q-Learning, and Greedy-GQ on the Italian stock market, utilizing five large company daily prices from January 2000 to October 2018. The study examined three look-back horizons to see how far back the model should consider historical data for making trading decisions. These were look-back periods of 5 days, 10 days, and 20 days. Their findings revealed that the look-back period of 20 days performed better on all measures of performance than the other look-back periods. This means this time period provided the model with sufficient historical context to make knowledgeable trading decisions (Corazza et al., 2019).

The study revealed that Greedy-GQ outperformed SARSA and Q-Learning with higher returns on a consistent basis over time (Corazza et al., 2019). The Greedy-GQ combined with a look-back window of 20 days proved to be the most effective in laying out positive returns on the risky Italian stock market. However, the research primarily aimed at maximizing cumulative returns and suggests future research for determining an appropriate model for risk-averse strategies (Corazza et al., 2019).

Pereira (2022) extended the prior study, testing Q-Learning, SARSA, and Greedy-GQ on the Brazilian mini-index futures market across 60-min and 500k trading intervals. The research provides valuable insights into the relative performance of different basis and reward functions, as well as look-back periods in algorithmic trading applications (Pereira, 2022).

As for basis function selection, they explored three methods including sigmoid, sigmoid 123, and hyperbolic tangent (hypTanh) function. The hypTanh was more stable and outperformed the others, particularly for 500k tick intervals (Pereira, 2022). Theoretical assumptions are fulfilled in this research since the hyperbolic tangent output

is bounded; it avoids explosive value estimations when market conditions are volatile (Sutton & Barto, 2018). The sigmoid function, while commonly used in machine learning applications, showed concerning volatility in their tests, frequently generating substantial losses while only occasionally producing significant gains (Pereira, 2022). The sigmoid123 function, producing discrete outputs, showed intermediate performance but lacked the consistency of the hyperbolic tangent approach (Pereira, 2022).

The research utilized three reward functions, among which MinusMean approach (computing immediate reward minus average reward) performed better than others all along (Pereira, 2022). MinusMean approach, along with hypTanh basis function and smaller look-back windows (n=5), provided the best performance. As for look-back windows, shorter windows were more effective. n=5 performed better than larger windows like n=25 or n=50. The study determined that larger look-back windows added noise to decision making, likely because they adhered to less pertinent historical data (Pereira, 2022). The discovery indicates that RL algorithms for futures trading would be more likely to benefit from recent price behavior rather than extended historical patterns.

The analysis identified distinctive patterns among algorithms and time frames (Pereira, 2022). Q-Learning best maintained its performance for both time frames for trading. During the 60-minute window, it recorded its lowest volatility ratio of 0.39 and highest annual return value of 57.40%. The 500k tick implementation recorded equal stability, posting its lowest volatility ratio value of 0.53 and an annual return value of 130.98%. The performance monitored for both SARSA and Greedy-GQ had more volatility for both trading windows: SARSA recorded a 0.44 value for 60-minute

windows and a 0.55 value for 500k ticks, and Greedy-GQ recorded 0.48 and 1.16 values, respectively.

One key limitation of this research is that it does not include additional measures of risk. Although the existing measure, defined in terms of standard deviation of returns, provides some indication of various algorithm configurations' risk-return profiles, insufficient risk measures, notably the Sharpe ratio, restrict the validity of these results for risk-averse investors (Hirsa et al., 2021). This is an area for additional research where more extensive risk measures are used to increase the applicability of the findings for risk-averse investors.

2.4.1 Discussion of Research Gap

While earlier research has shown that RL can be effective for algorithmic trading using Q-learning, SARSA, and Greedy-GQ algorithms, a research gap remains in assessing these models' fitness for risk-averse investors across various asset types and markets. In addition, previous research tends to utilize limited measures of risk, typically relying mostly on standard deviation, which limits their usability for risk-averse investment purposes. This praxis fills this gap through an expanded risk assessment framework and cross-validating previous findings based on E-mini-NASDAQ-100 futures. The assessment utilizes three complementary measures for risk-adjusted performance:

- The Sharpe Ratio, which captures excess return for each unit of risk relative to the risk-free rate (Hirsa et al., 2021).

- The Sortino Ratio quantifies extra return for each unit of adverse risk, taking into account just negative returns when computing its risk and not all volatility (Hirsa et al., 2021).
- The Calmar Ratio measures returns versus greatest decline in portfolio value (from peak to trough), assessing both potential for gains and potential for limiting extreme losses during times of market decline (Hirsa et al., 2021).

The choice of E-mini-NASDAQ-100 futures as the main trading vehicle is an important contribution to the literature, given that prior research has largely examined either stocks or non-US asset classes. The instrument possesses a number of unique qualities for algorithmic trading research: First, its large turnover during the day, typically in excess of 200,000 contracts, provides a good price discovery mechanism (Ates & Wang, 2005; Carlson, 2024).

Second, the fairly low margin requirements allow for effective capital usage, along with access to the technology-focused NASDAQ-100 index (Banerjee, 2022). Third, the standard contract size and near 24-hour trading offer liquidity and continuity of price, which are all necessary for reinforcement learning model training and validation (Ates & Wang, 2005; Carlson, 2024).

Hyperparameter selection for reinforcement learning in financial domains is a necessary but less explored area of research (Sutton & Barto, 2018; Ritter, 2017; Pereira, 2022). This research seeks to examine three basis functions: sigmoid123, sigmoid, and hyperbolic tangent (hypTanh), which all possess unique value in approximating non-linear relations in financial data (Sutton & Barto, 2018). The sigmoid function scales

input values to a bounded scale and is well suited for simulating probability-like values in financial markets (Sutton & Barto, 2018; Pereira, 2022).

The sigmoid123 version builds upon this feature through the additional utilization of three discrete output values (1, 2, and 3) in order to classify market states, enabling more versatile mapping of input characteristics (Sutton & Barto, 2018). The hyperbolic tangent function has similar advantages but a varied output range and thereby is able to better capture both positive and negative sentiments in a market environment (Sutton & Barto, 2018; Pereira, 2022).

The analysis for windows of varying sizes (5, 10, and 20 days) addresses the time dependency problem in financial markets (Pereira, 2022). The short windows could be more useful in capturing market behavior in real time, but they could be subject to noise. The long windows could be more useful in tapping into underlying patterns, though they could be slower in reacting to market corrections (Corazza et al., 2019). Likewise, various reward functions provide unique modes for optimization: Immediate rewards concentrate on current outcomes from trading, Mean rewards consider average performance, and MinusMean rewards examine deviations from average returns (Pereira, 2022). This praxis will continue to examine these parameters to test and ensure optimal setup for risk-averse trading in the futures markets.

This comprehensive analysis of Q-Learning, SARSA, and Greedy-GQ and their hyperparameters, with further risk metrics and emphasis on E-mini-Nasdaq-100 futures, provides a solid groundwork for evaluating model-free RL in a scenario directly applicable to risk-averse investors. Additionally, the systematic comparison to the

NASDAQ-100 benchmark and buy-and-hold strategies is made to offer insights into real-world usability in actual trading scenarios (Banerjee, 2022).

2.5 Summary and Conclusion

Even though Q-learning, Greedy-GQ, and SARSA are theoretically relevant, there remains a gap in the literature where all three are used in one study for a direct comparison. Although some research has considered these algorithms individually for various applications in finance, research applying Q-learning, Greedy-GQ, and SARSA together for comparison is scarce. This insufficient amount of comparative research raises questions around what each algorithm's strengths and weaknesses are in different trading conditions, such as the futures markets. Moreover, most studies assess risk based on standard deviation of returns, a limited measure that inadequately captures all that is needed in risk-averse investing. What is necessary is more in-depth research using more advanced measures of risk to gain deeper insight into which algorithm is best suited for risk-averse investors.

This praxis aims to fill these gaps by building on earlier research and implementing Q-learning, Greedy-GQ, and SARSA in another market context: the E-mini-NASDAQ-100 futures. The E-mini-NASDAQ-100 is a futures contract representing the value of the top 100 technology entities listed on the NASDAQ exchange, such as well-known tech companies like Apple, Microsoft, and Amazon (Ates & Wang, 2005; Carlson, 2024). The instrument enables investors to expose themselves to these tech companies for significantly less than it would cost to enter into a standard contract, making it more accessible for both individuals and institutions (Kurov, 2021). The popularity of this instrument grows out of its liquidity, flexibility, and responsiveness to

movement in the market — conditions amenable to efficient trading and diverse investment strategies. By testing these algorithms in the E-mini-NASDAQ-100 futures market, this research aims to offer greater insight into their comparative performance, particularly in situations calling for stability, flexibility, and effective risk management.

While there may not be a one-size-fits-all approach, the literature considered in this research shows strong evidence that RL can not only replicate but also outperform rational investment choices in financial markets and potentially enhance risk-averse trading. This praxis seeks to further examine and validate these findings. Additionally, most of the rigorous scientific work in algorithmic trading is not made publicly available. As Theate and Ernst (2021) explain, the substantial financial stakes involved ensure that proprietary fintech companies do not disclose their most recent research breakthroughs. Thus, many of the most recent advancements in reinforcement learning for trading are developed behind closed doors and are not made publicly available for further analysis (Theate & Ernst, 2021).

Finally, fair comparisons between various trading strategies are difficult to make due to the lack of a common framework for performance metrics (Theate & Ernst, 2021). Although researchers have demonstrated RL's value in different market conditions, the diversity in methodologies and evaluation measures makes objective judgment challenging. This praxis aims to narrow the gap on these challenges by implementing most frequently cited metrics. Through this systematic approach, this research aims to provide clear, comparable insights into the effectiveness of RL strategies for risk-averse trading in modern market environments.

Methodology

3.1 Introduction

Chapter 3 details the research methodology for fulfilling research questions and hypotheses designed in Chapter 1. The chapter provides a holistic framework for data acquisition, data preprocessing methods, model development, performance measures, and hypothesis testing methods. The chapter lays out an explicit methodology for designing and testing trading algorithms based on empirical work and theory. The algorithmic models transform market data into optimal trading choices based on unique learning paradigms, which will be studied deeply in later sections.

Section 3.2 covers data collection and the characteristics of the E-mini-NASDAQ-100 Futures market. **Section 3.3** outlines the preprocessing pipeline for transforming raw data into well-structured trading intervals. **Section 3.4** introduces the market simulation environment for reinforcement learning. **Section 3.5** covers the three reinforcement learning algorithms including Q-learning, Greedy-GQ, and SARSA, their mathematical foundation, and implementation. **Section 3.6** defines details for simulation and performance measures in order to find optimal algorithmic configurations. **Section 3.7** lays out performance metrics in order to assess trading strategy performance for risk-averse investors, covering measures for profitability and risk profiles. **Section 3.8** summarizes the methodological framework and key concepts discussed in this chapter.

3.2 Data Collection

The dataset comprises high-frequency data for the E-mini-NASDAQ-100 futures contracts bought and sold on the Chicago Mercantile Exchange (CME). The data were initially made available through the exchange and were available through DataBento (2024), a data vendor offering hedge fund quality financial market data. This dataset can be downloaded publicly through this data vendor for a fee. The original dataset contains:

- Nanosecond-recorded timestamps
- Approximately 102,978,608 rows of transaction data
- Five E-mini-NASDAQ-100 futures contracts spanning May 2023 to May 2024:
    - **NQM3** (March 2023 to June 2023 Contracts)
    - **NQU3** (June 2023 to September 2023 Contracts)
    - **NQZ3** (September 2023 to December 2023 Contracts)
    - **NQH4** (January 2024 to March 2024 Contracts)
    - **NQM4** (March 2024 to May 2024 Contracts)

The futures contracts under consideration are highly liquid, electronically traded instruments that mirror the NASDAQ-100 index and thus are appropriate instruments for testing RL-based trading strategies (CME Group, 2025; Charles Schwab, n.d.). Traded hundreds of thousands of times daily, they form a deep and liquid market that minimizes slippage during order execution, a favorable requirement for algorithmic trading (DataBento, 2024). The dataset utilized required very little cleanup, with no missing values or large anomalies to be fixed.

The data were originally stored in .csv.zst format, processed to .zip, and finally converted into ordinary CSV format for analysis. The selection of E-mini-NASDAQ-100 futures contracts during the period of May 2023 to May 2024 provides a balanced and realistic test scenario for RL-based algorithmic trading. Compared to the excessive volatility during the COVID-19 period (2020–2021) or distortions fueled by inflation during 2021–2022, the period includes both phases of upward surge fostered by the technology boom and AI advances (Bradshaw, 2024), and correction periods shaped by geopolitical tensions (Berger, 2024). The mixture of bullish and bearish periods occurring side by side offers a more comprehensive test of RL algorithm versatility than periods in which only one trend prevails. Finally, the liquid nature of the E-mini- NASDAQ-100 contracts and access to granular transaction data through DataBento (2024) support a scientifically valid test of trading strategies under conditions that more realistically replicate common market cycles.

3.2.2 Data Preprocessing

The raw data follow a sequence of transformations in order to create a well-organized format suitable for reinforcement learning in algorithmic trading:

1. **Format Conversion:** The raw data are converted from .zip to standard CSV format.
2. **Tick Aggregation:** Individual transactions are aggregated into 500,000-tick intervals.
3. **OHLCV Generation:** The values for **Open, High, Low, Close**, and **Volume** for each interval are computed.

The OHLCV representation is a standard technique for summarizing market activity (Mettenheim & Breitner, 2012). It includes the opening price (the price of the first trade in the interval), highest price, lowest price, closing price (the price of the last trade), and total volume transacted within the period. Rather than using a time-based interval, this research uses a tick-based aggregation approach, where each interval consists of exactly 500,000 trades.

The 500,000-tick interval provides a favorable compromise between having sufficient market microstructure and excessive noise created by time-based transactions (Lopez de Prado, 2018). The scale of this interval is also compatible with previous studies on high-frequency futures data, allowing for easier comparison of outcomes while each bar still reflects statistically significant trading activity (Pereira, 2022). The 500,000-tick OHLCV data are used to represent the state for the RL models in this research, so that the models are tested on effective, market-driven price movement rather than arbitrary time periods (Lopez de Prado, 2018). This makes the technique more relevant for algorithmic trading, where tick-based analysis of market dynamics is more effective than time-interval partitioning.

3.2.3 Data Organization and Storage

The E-mini-NASDAQ-100 futures data were hierarchically structured for streamlined preprocessing and model development. Raw transaction data were stored in compressed files, extracted and processed into CSVs that retained transaction detail, and then aggregated into 500,000-tick-interval files for direct loading into the reinforcement

learning platform. The hierarchy, shown in **Figure 3-1**, ensured data integrity and made raw and intermediary formats accessible for verification.

Figure 3-1 Data Directory Structure Example

3.3 System Architecture and Data Flow

The reinforcement learning trading system contains key components regulating the data flow pipeline: **Data Preprocessing**, where raw market data are converted into 500k tick intervals in OHLCV format; **State Representation**, where technical indicators are added as state variables; **Model Training and Execution**, where Q-Learning, SARSA, and Greedy-GQ make optimal trading calls; and **Performance Evaluation**, wherein models are tested across various simulations and parameter configurations. **Figure 3-2** depicts this process, from raw data conversion to making trading calls and verifying performance.

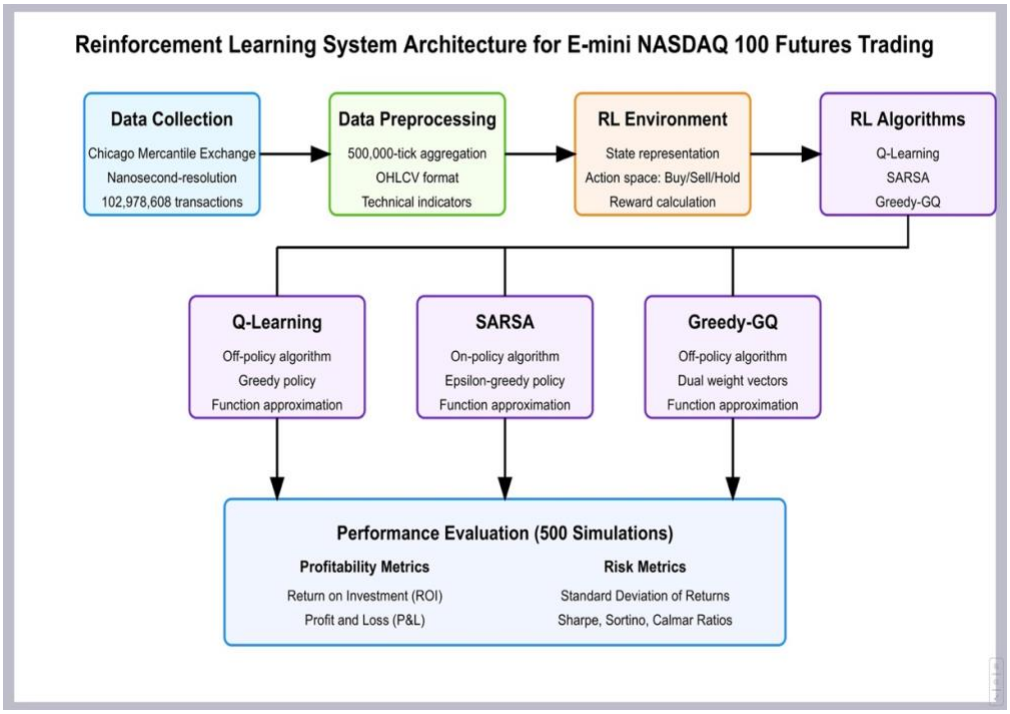

Figure 3-2 System Architecture Workflow

3.4 Reinforcement Learning (RL) Environment

The reinforcement learning environment for trading in the E-mini-NASDAQ-100 futures market implements sophisticated market simulations that integrate financial data processing with sequential decision-making (Moriyama et al., 2012; Wang et al., 2022). The RL environment comprises three fundamental elements: state representation, action space, and reward calculation (Sutton & Barto, 2018):

State Representation: The environment creates a pertinent state representation from raw market data using technical indicators. A sliding window of size n sees temporal patterns in price, allowing the agent to learn from prior context rather than from isolated data points. The state includes Bollinger Bands, which assess price volatility using standard deviation-based upper and lower bands around a moving average

(Vergura, 2020); the Exponential Moving Average (EMA), which emphasizes recent price movement (Vergura, 2020); and the Linearly Weighted Moving Average (LWMA), which focuses on recent data through a linear weighting scheme (Zulkifli et al., 2024). Together, these indicators provide insights into market trends, volatility, and relative price movement, while reducing dimensionality to mitigate overfitting.

The trading environment is managed through the environment class, and it serves as the interface between market data and the reinforcement learning algorithm (Pereira, 2022). The market states are stored, the value and status of positions are tracked, the trades are executed, and the rewards are computed. Additionally, the environment allows for a dynamic and well-structured interaction between market conditions and the RL agent through the generation of the next state and reward based on the agent's actions, following the Markov Decision Process (MDP) framework (Puterman, 1994). The framework provides a mathematical description for making decisions in stochastic conditions in which outcomes are based on randomness and the agent's actions.

3.4.1 Elaboration of Technical Indicators for State Representation

The reinforcement learning environment requires a meaningful representation of market conditions in order to make informed trading decisions (Ansari et al., 2024). Technical indicators are transformations of price data that extract potentially useful patterns for trading algorithms, enabling more efficient decisions through computational algorithms and statistical inference (Lo et al., 2000). This praxis employs three distinctive technical indicators: Bollinger Bands, Exponential Moving Average (EMA), and Linearly Weighted Moving Average (LWMA) (Vergura, 2020; Zulkifli et al., 2024).

3.4.1.1 Bollinger Bands

Bollinger Bands measure price volatility relative to recent price levels, providing information about potential overbought or oversold conditions (Vergura, 2020). Bollinger Bands consist of three lines:

a. Middle Band (Simple Moving Average):

$$MB_t = SMA_t = \frac{1}{n}\sum_{i=0}^{n-1} Price_{t-i}$$

b. Upper Band (UB):

$$UB_t = MB_t + k.\sigma_t$$

c. Lower Band (LB):

$$LB_t = MB_t - k.\ \sigma_t$$

Equation 3-1 Bollinger Bands Formula

where,

- MBt is the middle band (simple moving average of prices over period n).
- UBt is the upper band.
- LBt is the lower band.
- k is the number of standard deviations (commonly set to 2).
- σt is the rolling standard deviation of the price over the same period n, calculated as:

$$\sigma_t = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(Price_{t-i} - MB_t)^2}$$

The implementation consists of a simple moving average (SMA) with upper and lower bands placed at standard deviation levels above and below this average. Bollinger Bands provide context to state representation in order to provide insights into price volatility and relative positioning within recent trading ranges. This information is valuable for the identification of potential regime changes and extreme market conditions. In the reinforcement learning context, these indicators help the agent distinguish between normal market movements and significant deviations that might warrant action, particularly the case when the price crosses above the upper band or below the lower band (Yan et al., 2022).

3.4.1.2 Exponential Moving Average (EMA)

The Exponential Moving Average, as shown in **Equation 3-2**, calculates a weighted average of prices that assigns greater importance to recent observations (Vergura, 2020):

$$EMA_t = \alpha \cdot Price_t + (1 - \alpha) \cdot EMA_{t-1}$$

where,

$$\alpha = \frac{2}{n+1}, \quad n \text{ is the time period}$$

Equation 3-2 Exponential Moving Average (EMA) Formula

Unlike simple moving averages, EMA responds more quickly to recent price changes (Vergura, 2020). It contributes to the identification of directional price trends while filtering out short-term noise. Because of its responsive nature, the indicator helps the reinforcement learning agent to detect trend changes earlier than with traditional moving averages (Vergura, 2020). This becomes particularly relevant for model states

when deciding whether to enter or exit positions based on emerging trends. The EMA values, when incorporated into the feature vector through the basis functions, allow the agent to evaluate whether the current price is above or below its recent trend (Schwarzer et al., 2020).

3.4.1.3 Linearly Weighted Moving Average (LWMA)

The Linearly Weighted Moving Average applies a linear weighting scheme in which it assigns incrementally higher weights to more recent prices, calculated as (Zulkifli et al., 2024):

$$LWMA_t = \frac{\sum_{i=0}^{n-1}(n-i) \cdot Price_{t-i}}{\sum_{i=0}^{n-1}(n-i)}$$

Equation 3-3 LWMA Formula

where,

- n is the time period (window size).
- Price t−i is the price at time t−i.
- The weights decrease linearly from n (most recent data) to 1 (oldest data).
- The denominator ensures the sum of weights is normalized.

LWMA provides a balance between the responsiveness of EMA and the stability of SMA (Zulkifli et al., 2024). By weighting observations in a linear fashion, LWMA creates a smoothed representation of price action so that recent movements are emphasized while historical context is still considered. The reinforcement learning agent uses this indicator to assess intermediate-term momentum and potential support/resistance levels (Huang et al., 2022).

3.4.2 Technical Indicators in the Reinforcement Learning Framework

Technical indicators are integrated into the state representation. The combination of these indicators creates a multi-dimensional view of the market in order to capture trend, volatility, and relative positioning information (Yan, et al., 2022; Zulkifli et al., 2024).

When processed through basis functions (e.g., sigmoid, hyperbolic tangent, or sigmoid123), the indicators form the feature vector so that they serve as input to the Q-function approximation (Sutton & Barto, 2018). Integrating established technical indicators into the state representation prevents overfitting. The indicators act as dimensionality reduction mechanisms that assist in transforming high-dimensional raw price sequences into lower-dimensional, information-dense features (Lo et al., 2000). This transformation simplifies the learning task for reinforcement learning agents, reducing the risk of overfitting by focusing on the most relevant aspects of the data.

Second, using technical indicators allows reinforcement learning models to incorporate domain knowledge from financial markets so that they focus on patterns with demonstrated theoretical and empirical relevance (Deep et al., 2024). Third, the well-defined mathematical properties of these indicators create a framework in which interpretability is enhanced compared to black-box models that rely on raw data, thereby improving transparency and trust in financial applications (Pippas et al., 2024).

The selected technical indicators provide sufficient market information for the reinforcement learning algorithms while avoiding excessive feature engineering that could lead to overfitting (Lo et al., 2000; Yan et al., 2022; Deep et al., 2024). This approach strikes a balance between model simplicity and expressiveness, in order to

allow the algorithms to capture meaningful market patterns without becoming overly sensitive to noise or artifacts in the historical data (Pippas et al., 2024; Zulkifli et al., 2024).

Action Space: The action space, which defines the agent's possible trading decisions, consists of three discrete actions: **Buy (1)** to enter a long position, **Hold (0)** to maintain the current position, and **Sell (-1)** to enter a short position. In order to ensure logical consistency, the action space adapts to the agent's current trade status, preventing invalid transitions such as selling without holding a position or buying while already in a long position. This tripartite action framework enables the agent to establish directional market positions or maintain neutrality (Pereira, 2022). The environment tracks position status throughout the trading session, which comes with appropriate state transitions when actions change the current position.

The reinforcement learning (RL) framework is modeled as a discrete-time stochastic control process where an algorithmic trader interacts with a market environment (Corazza et al., 2019; Pereira, 2022). This approach, as discussed in **2.2.3** and **2.2.4** follows the Markov Decision Process (MDP) paradigm that incorporates Bellman's theory for optimal decision-making.

The state representation consists of a structured observation matrix containing historical OHLCV (Open, High, Low, Close, Volume) price values over a fixed window of n+1 time steps, in order to ensure that the RL agent learns from market-driven price dynamics (Pereira, 2022; Sutton & Barto, 2018; Lopez de Prado, 2018).

The agent's decision-making process is based on its current trading status. If the agent holds a short position, it can either maintain its position or switch to a long

position. In the event there is no active position, it can choose to go long, go short, or remain neutral (Pereira, 2022). Conversely, if the agent is already in a long position, the available options are to hold the position or switch to a short position. These discrete actions define the agent's ability to adapt to evolving market conditions, in order to ensure trading decisions, align with the observed state representation (Sutton & Barto, 2018; Lopez de Prado, 2018).

Reward Calculation: The reward mechanism evaluates trading performance in which it calculates profits and losses based on position entry and exit prices (Pereira, 2022). Profit and loss (P&L) refers to the financial outcome of executed trades, representing the difference between an asset's entry and exit prices (Pereira, 2022). A trade results in a profit when an asset is sold at a higher price than its purchase price, while a loss occurs when it is sold at a lower price.

In short selling, the calculation is reversed: a profit is generated when the asset is repurchased at a lower price than its initial selling price (Pereira, 2022). For buy positions, the reward is determined as the difference between the exit and entry prices, while for sell positions, it is computed as the inverse (Pereira, 2022). In order to ensure consistency across different price scales, the environment applies logarithmic transformations to price ratios, through which they normalize returns and mitigate distortions caused by absolute price differences (Pereira, 2022).

Cumulative performance is tracked through the investment balance, which updates dynamically after each action based on price movements. This environment design establishes a sequential decision-making framework, so that the agents learn to map market states to optimal trading actions by maximizing expected returns. The

environment employs a trading system by which rewards associated with the initial investment capital and trading actions are monitored. Since the objective is to maximize trading profits, the reward represents the profitability balance attained from market interactions. The algorithmic trader starts with an initial capital, and as market time progresses, the environment updates the investment surplus based on each action taken.

For example, if the initial balance is 33,600 points and a trade results in a profit of 1,000 points, the updated balance becomes 34,600 points. This reward evolution process continues at each time step, in order to record all accumulated rewards in an environment variable. In simple terms, this variable stores the sequence of historical rewards, by which the agent is able to track performance trends rather than treating each reward in isolation.

This study implements three reward functions to evaluate and optimize trading performance. These three reward functions shape the agent's learning process, in order to influence how it evaluates past trading outcomes. The minusMean approach normalizes rewards, the immediate approach focuses solely on current trades, and the mean approach provides a smoothed long-term reward signal (**Table 3-1**) (Sutton & Barto, 2018; Pereira, 2022).

All three algorithms rely on linear function approximation to estimate action-value functions, following the principles of linear reinforcement learning (Corazza & Sangalli, 2015; Baird, 1995). This method avoids the need to estimate the probability transition of the environment, which constitutes a model-free approach. Since financial markets are complex and unpredictable, the model-free learning is well-suited for adapting to changing market conditions without relying on predefined transition dynamics (Baird, 1995).

Table 3-1 Reward Functions

Reward Function	Description
minusMean	Subtracts the average of all past rewards from the current reward. Focuses on deviations from past performance rather than absolute values. Makes the model sensitive to changes in profitability and aligns with reinforcement learning frameworks for continuing tasks.
immediate	Uses raw profit or loss at the current time step without historical adjustments. Agent learns based solely on immediate outcomes of trades. More reactive to short-term market fluctuations, potentially leading to volatile learning behavior.
mean	Computes reward as the average of all historical rewards accumulated to the current time step. Smooths short-term fluctuations and provides a more stable learning signal. Encourages trading strategies that optimize consistent long-term performance.

In reinforcement learning, the action-value function is approximated using a weight vector and a corresponding feature representation of the state-action pair (Sutton & Barto, 2018; Baird, 1995). Instead of maintaining a large table of values for every possible state-action combination, the function approximator generalizes from past experiences and makes learning more efficient (Corazza & Sangalli, 2015). The weight vector is continuously updated as the agent interacts with the environment, by which it refines its understanding of optimal trading decisions (Baird, 1995).

Since financial markets do not have well-defined transition probabilities, there is a requirement for encoding environmental states as feature vectors for effective learning (Baird, 1995; Sutton & Barto, 2018). The choice of basis functions determines how raw market data is transformed into meaningful representations that the learning algorithm can process, allowing the reinforcement learning agent to generalize from past

experiences without relying on predefined transition dynamics (Corazza & Sangalli, 2015). As shown in **Table 3-2**, three basis functions are used in this study to map states to feature values.

Table 3-2 Basis Functions

Basis Function	Description
Sigmoid	Maps real-valued inputs to range [0,1]. Useful for normalizing market features and ensuring stable learning. Monotonic with well-defined derivative allowing smooth updates.
Hyperbolic Tangent (hypTanh)	Maps inputs to range [-1,1]. Enables symmetric representation of market states, better distinguishing between positive and negative changes.
Sigmoid123	Modified sigmoid that provides discrete outputs categorized into three distinct values based on predefined thresholds. Reduces noise in feature space and ensures similar market conditions yield consistent representations.

3.5 Reinforcement Learning (RL) Algorithms

This study implements three model-free reinforcement learning algorithms including Q-Learning, SARSA, and Greedy-GQ to optimize trading decisions in the E-mini-NASDAQ-100 futures market. These algorithms approximate action-value functions using linear function approximation, in order to allow for continuous state representation and adaptive decision-making (Corazza & Sangalli, 2015). They were selected because they are suitable for discrete action spaces and are able to handle stochastic reward environments, both of which align well with the structure of financial markets (Kim & Yang, 2007).

3.5.1 Q-Learning Algorithm

As shown in **Equation 3-4**, Q-Learning is an off-policy reinforcement learning algorithm that updates action-value estimates using the Bellman equation (Sutton & Barto, 2018).

$$\overrightarrow{w_{t+1}} = \overrightarrow{w_t} + \eta_t \left[R_{t+1} + \gamma \max_{a' \in A(tradingStatus)} Q\left(S_{t+1}, a'; \overrightarrow{w_t}\right) - Q(S_t, A_t; \overrightarrow{w_t}) \right] \overrightarrow{\phi_t}$$

Equation 3-4 Q-Learning Weight Update Equation

where,

- η_t is the learning rate,
- γ is the discount factor,
- ϕ_t is the feature vector,
- a' is the action chosen using a greedy policy,
- $Q(S_t, A_t)$ represents the estimated action-value function.

The update rule modifies the weight vector based on the learning rate (η), discount factor (γ), and feature vector (ϕ), so that it ensures the agent refines its decision-making by incorporating observed rewards and estimated future returns (Hilpisch, 2024). The algorithm follows an ϵ-greedy policy for action selection by which it balances exploration and exploitation, ensuring that the agent continues to explore new actions while favoring those that have yielded higher rewards in past interactions (Pereira, 2022).

3.5.2 SARSA Algorithm

SARSA (State-Action-Reward-State-Action) is an on-policy algorithm that updates value estimates based on the agent's actual action sequence (Corazza & Sangalli, 2015). The weight update follows the Bellman optimality principle, as shown in **Equation 3-5**.

$$\theta_t = R_{t+1} + \gamma Q(S_{t+1}, a'; \overrightarrow{w_t})$$

Equation 3-5 SARSA Target Value Function

where,

- R_{t+1} is the observed reward at time $t + 1$,
- γ is the discount factor,
- $Q(S_{t+1}, a'; \overrightarrow{w_t})$ is the estimated action-value function for the next state-action pair,
- a' is the action chosen using an ϵ-greedy policy to balance exploration and exploitation

The ϵ-greedy action selection strikes a balance between exploration and exploitation, which accordingly, allows SARSA to adapt dynamically to market conditions (Sutton & Barto, 2018). Unlike Q-Learning, which updates based on the best possible future action, SARSA updates using the actual action taken, which typically leads to a smoother and more stable learning trajectory (Corazza & Sangalli, 2015).

3.5.3 Greedy-GQ Algorithm

Greedy-GQ is designed to mitigate bias in off-policy learning by introducing a secondary weight update mechanism (Ma et al., 2021). As outlined in **Equation 3-6**, this

algorithm refines the weight vector iteratively in order to improve convergence stability, particularly in environments with stochastic market dynamics.

$$\theta_t = R_{t+1} + \gamma \max_{a' \in A(tradingStatus)} Q\left(S_{t+1}, a'; \overrightarrow{w_t}\right)$$

Equation 3-6 Greedy-GQ Target Value Function

where,

- R_{t+1} is the observed reward at time $t+1$,
- γ is the discount factor,
- $Q(S_{t+1}, a'; \overrightarrow{w_t})$ is the estimated action-value function for the next state-action pair,
- a' is the action chosen using a greedy policy, selecting the action with the highest estimated Q-value.

3.5.4 Hyperparameter Optimization

The performance of reinforcement learning algorithms in financial trading is highly sensitive to their hyperparameter settings (Sutton & Barto, 2018). This study employs a grid search approach in order to systematically identify optimal configurations. The evaluation process explores a range of hyperparameters that include:

- Lookback window size (n)
- Basis function type (basisFctType)
- Reward function variant (rewardType)
- Learning rate (η) and secondary learning rate (ζ, for Greedy-GQ)
- Discount factor (γ)

- Exploration-exploitation probability (ϵ, for SARSA)
- Learning rate scheduling method (lrScheduler)
- Weight initialization technique (initType)

Each parameter combination is evaluated through 500 independent simulations, in order to ensure robustness by varying 50 random seeds within these runs. **Table 3-3** presents the hyperparameter search space explored for each algorithm.

Table 3-3 Hyperparameter Search Space

Hyperparameter	Description	Search Values
n	State window size	[5, 10, 15]
basisFctType	Activation function	["sigmoid", "sigmoid123", "hypTanh"]
rewardType	Reward calculation method	["minusMean", "immediate", "mean"]
η (eta)	Primary learning rate	[0.1, 0.01]
γ (gamma)	Discount factor	[1.0, 0.95]
ϵ (epsilon)	Exploration rate (SARSA)	[0.15, 0.1]
ζ (zeta)	Secondary learning rate (Greedy-GQ)	[0.1, 0.01]
initType	Weight initialization	["uniform01", "zeros"]
lrScheduler	Learning rate decay interval	[0, 200]
seeds	Random number generator	[1, 2, 3, ..., 51]

3.5.5 Hyperparameter Evaluation Process

A grid search evaluates all hyperparameter combinations. The top configurations, ranked by mean profit/loss across 500 simulations, are selected for further analysis through a selection process which is specifically designed to identify the most

robust and consistently performing hyperparameter sets. Thus, the goal of this process is to identify hyperparameters that most influence performance within the given trading period.

Grid search was employed due to its exhaustive evaluation of hyperparameter combinations in order to ensure that the best-performing configurations are identified with statistical confidence (Gupta, 2020). While alternative approaches such as Bayesian Optimization or Random Search could reduce computational complexity, grid search guarantees that all possible parameter combinations are systematically explored (Gupta, 2020). Accordingly, this optimization methodology ensures:

- **Statistical robustness** by evaluating performance across 500 independent simulations.
- **Reduced overfitting risk** by testing a variety of model configurations on the dataset.
- **Reliability** in identifying hyperparameters that contribute to the best results.
- **Consistency** across various contract periods (e.g., NQM3, NQU3) of the same asset, in order to make sure the selected configurations are not overly dependent on a specific time frame.

3.6 Simulation and Performance Evaluation

The simulation framework implements a systematic approach to evaluate reinforcement learning algorithms across multiple configurations and contract periods. The methodology follows a two-phase design with the intention of efficiently exploring the hyperparameter space while ensuring statistical robustness.

3.6.1 Simulation Architecture and Execution Process

For each algorithm (Q-Learning, SARSA, and Greedy-GQ), the simulation process follows a structured pipeline:

1. **Data Preparation:** First, the environment loads extensive market data—specifically, 500,000 ticks of OHLCV information from five E-mini-NASDAQ-100 futures contracts (NQM3, NQU3, NQZ3, NQH4, NQM4).
2. **Environment Initialization:** Each simulation initializes a trading environment with a specific set of hyperparameters and a unique random seed.
3. **Algorithm Configuration:** Next, the selected reinforcement learning algorithm is set up with specific hyperparameters chosen from the grid search space, in order to make sure each configuration gets tested systematically.
4. **Simulation Execution:** The agent interacts with the environment sequentially, making trading decisions at each time step based on the current state, until reaching the terminal state (end of dataset).
5. **Performance Recording:** Finally, while running through each simulation, the system records key performance metrics, including profit and loss (P&L), investment balance trajectories, and trade statistics.

3.6.2 Simulation Data Processing

Each simulation processes the input data as follows:

1. **Data Source Selection:** The environment loads contract-specific OHLCV data at 500,000-tick intervals.
2. **Technical Indicator Application:** For generating the state representation, the original data passes through three technical indicators: Bollinger Bands,

Exponential Moving Averages (EMA), and Linear Weighted Moving Averages (LWMA).

3. **Feature Vector Construction:** The transformed data is passed through a selected basis function (sigmoid, sigmoid123, or hypTanh) to create feature vectors for the agent.
4. **Sequential Processing:** In the next step, the agent processes each bar sequentially, through which state transitions occur after each trading action that is driven by the policy of the selected RL algorithm.

Table 3-4 details the specific data flow process and highlights the key differences between simulation runs, illustrating how hyperparameters map to each simulation execution.

Table 3-4 Data Flow and Simulation Run Process

Process Steps	Implementation Steps
Data Loading	Each simulation begins by loading 500,000-tick OHLCV data for a specific E-mini-NASDAQ-100 contract (e.g., NQM3). The environment class reads this data from CSV files using pandas.
Technical Indicator Application	The Environment.applyTA() method transforms raw data by calculating Bollinger Bands, EMA, and LWMA for the specified lookback period, through which it creates a meaningful state representation.
Agent Initialization	For each simulation, the system initializes the chosen algorithm (QLearning, SARSA, or GreedyGQ) with its respective hyperparameter settings. Each execution is assigned a different random seed.
Simulation Loop	The Environment.runNext() method processes each time step, with the Agent.run() method selecting actions and propagating weights according to algorithm-dependent update rules.
Differences Between Runs	1. Algorithm choice drives the learning mechanics. 2. Each basis function offers its own data lens. 3. Seeds randomize the starting point. 4. Lookback window size affects how much historical data is used for decision-making.
Execution Process	Phase 1: Each of the 1,512 parameter combinations is tested with 50 random seeds. Phase 2: Top-performing configurations undergo extended testing with 500 random seeds. All configurations are tested across 5 contract periods to ensure generalizability.

3.6.3 Performance Evaluation Methodology

The simulation methodology consists of two distinct phases:

Phase 1: Hyperparameter Grid Search

Each unique hyperparameter configuration is evaluated using 50 random seeds (1–50) so as to mitigate initialization bias and stochastic variance. The full grid search covers 1,512 unique configurations distributed across the three algorithms.

For **Q-Learning**, there are 216 possible combinations stemming from 3 lookback periods, 3 basis functions, 3 reward types, 2 learning rates, 2 discount factors, and 2 learning rate schedulers. Since Q-Learning follows a straightforward off-policy update rule, it needs only these essential hyperparameters.

SARSA expands to 864 configurations by adding epsilon values and initialization types to the same base structure. This makes sense because SARSA's on-policy nature requires explicit exploration parameters and benefits from different weight initialization strategies.

Greedy-GQ uses 432 configurations, incorporating an additional secondary learning rate for its zeta parameter—the dual weight vector that handles off-policy correction. This extra complexity makes sense because Greedy-GQ needs sophisticated stabilization to handle off-policy scenarios properly.

Each configuration is evaluated across five contract periods (NQM3, NQU3, NQZ3, NQH4, NQM4) to assess performance consistency and generalizability. The mean return is computed across the 50 seed-based simulations for each configuration to identify top-performing combinations.

Phase 2: Extended Robustness Testing

The top hyperparameter configurations discovered within Phase 1 are re-evaluated. Each of the best-picked configurations is tested using 500 independent simulation evaluations with random seeds ranging from 1 to 500, such that the results are

statistically valid. This step is designed so that high-performing configurations are not statistical outliers but, rather, those that consistently produce better results with large sample sizes. The process of evaluation is composed of the following steps:

1. **Initial Screening:** Mean return is calculated across the 50 simulations for each configuration during Phase 1.
2. **Configuration Ranking:** The configurations are sorted according to mean return. The best-performing top 50 and worst-performing bottom 50 configurations are determined to be analyzed further.
3. **Hyperparameter Sensitivity Analysis:** The frequency distribution of hyperparameter values within the top and bottom groups is analyzed in order to identify parameters whose values most strongly correlate with performance outcomes.
4. **Focused Evaluation:** There will be a special focus on the basis function, reward function, and lookback period since these hyperparameters serve as the most predictive of trading performance.
5. **Model Validation:** In Phase 2, the top configurations are re-evaluated with 500 simulations to validate their performance consistency and stability under varying stochastic conditions.

This approach directly supports a core objective of this study: identifying the key hyperparameter choices that most influence trading performance.

Figure 3-3 illustrates the hyperparameter optimization methodology, including sequential execution of grid search, simulation runs, and result aggregation.

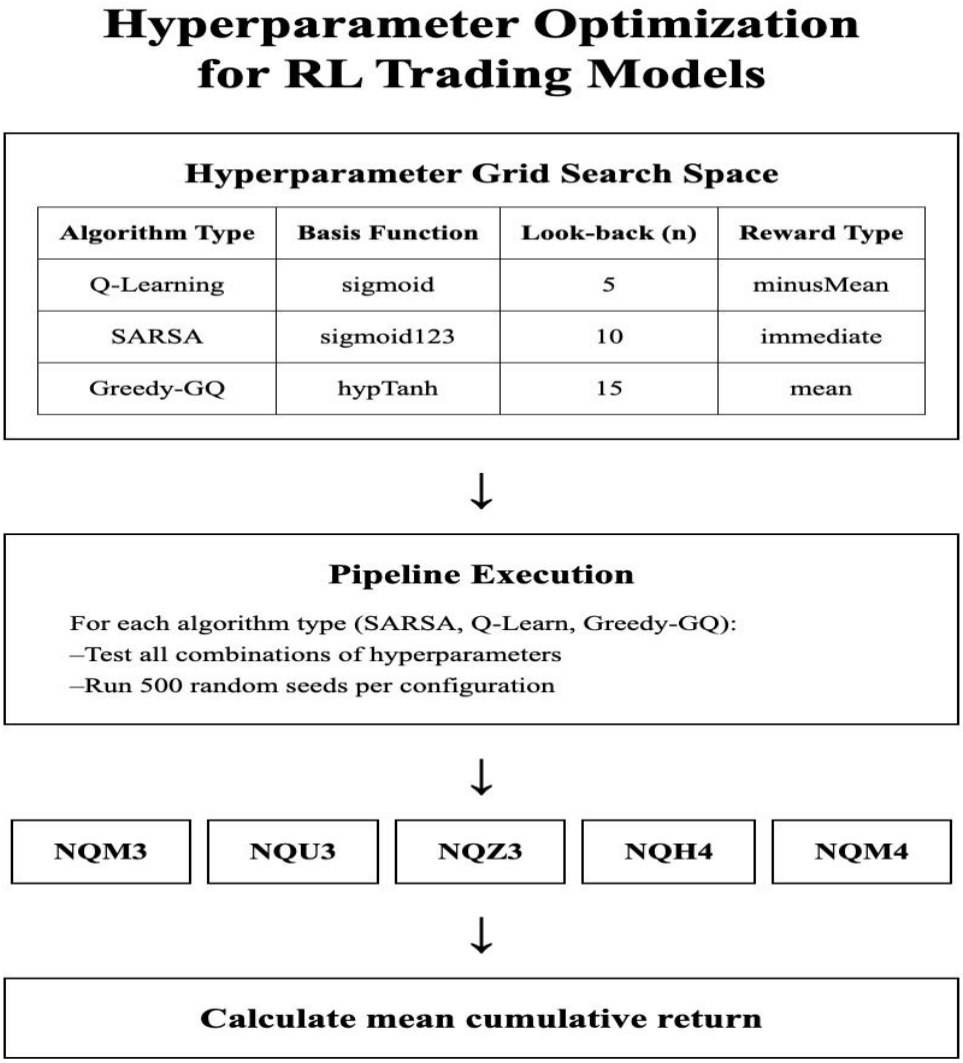

Figure 3-3 Hyperparameter Optimization Workflow

Aggregation of results across contract periods and simulations allows for a rigorous evaluation of trading performance. This facilitates model selection by identifying models that outperform benchmarks, including the NASDAQ Index, followed by the determination of models that maximize returns while controlling for volatility.

3.7 Model Selection and Performance Evaluation Metrics

Model performance is compared using a profitability-risk framework that prioritizes strategies achieving the highest ROI/P&L while maintaining favorable risk characteristics. Evaluation metrics include the Sharpe Ratio (reward-to-risk trade-off), Sortino Ratio (downside risk sensitivity), and Calmar Ratio (long-term return stability relative to drawdown), in order to ensure a comprehensive assessment of both profitability and risk exposure (Hirsa et al., 2021).

3.7.1 Profitability Metrics

Table 3-5 includes the profitability indicators used to measure model performance. Profit and Loss (P&L) indicates the absolute change of trade value against initial investment, while Return on Investment (ROI) is computed to indicate the percentage yield against initial capital (Pereira, 2022).

Table 3-5 Profitability Metrics

Profitability Metric	Description
Profit and Loss (P&L)	The absolute change in portfolio value from the initial investment. $P\&L = G_T - Initial\ Investment$ where G_T is the final portfolio value at the terminal state.
Return on Investment (ROI)	The percentage return relative to the initial investment. $ROI = (P\&L / Initial\ Investment) \times 100\%$

3.7.2 Risk Metrics

Table 3-6 summarizes the risk assessment metrics. Standard deviation of returns quantifies volatility, indicating how much the model's returns fluctuate over time. A

lower standard deviation suggests more consistent performance. Additional risk metrics as discussed in **2.4.1** further evaluate the stability of each model.

Table 3-6 Risk Metrics

Risk Metric	Description
Standard Deviation of Returns	Measures the volatility of returns. $\sigma = \sqrt{[1/n \sum(R_i - \bar{R})^2]}$ where R_i is the return for simulation i and $\bar{R}$ is the mean return.
Sharpe Ratio	Measures risk-adjusted return relative to risk-free rate. $Sharpe\ Ratio = (R_p - R_l)/\sigma_p$ where R_p is the portfolio return, R_l is the risk-free rate, and σ_p is the standard deviation of portfolio returns.
Sortino Ratio	Similar to Sharpe ratio but focuses only on downside risk. $Sortino\ Ratio = (R_p - R_l)/\sigma_d$ where σ_d is the standard deviation of negative returns only.
Calmar Ratio	Measures return relative to maximum drawdown. *Calmar Ratio = Annualized Return/Maximum Drawdown*

3.8 Summary

The chapter provided a detailed methodology including the ways by which reinforcement learning-based models are built, trained, and evaluated, particularly when trading the E-mini-NASDAQ-100 futures. In doing so, the chapter effectively covered the steps including the dataset acquisition, preprocessing, description of state, algorithm implementation, hyperparameter optimization, model selection, and performance assessment. Subsequently, leveraging a modular system architecture enables structured deployment and evaluation of reinforcement learning models for trading the E-mini-NASDAQ-100 futures, so that statistical analysis of model outcomes can be effectively conducted across different configurations and contract periods.

Furthermore, several safeguards are taken into account for the methodology in order to ensure reliable model outcomes. First, the use of multiple contract periods effectively prevents asset-specific overfitting, while the inclusion of technical indicators increases signal quality. Moreover, the implementation of 500 independent simulations is designed in such a way that it ensures stability across various market conditions. This is then followed by aggregation of results across different contract periods, which consequently enables the selection of models with strong generalization capabilities. These protective measures are meant to ultimately provide a minimal chance of overfitting, thereby enhancing the generalizability of reinforcement learning-based models when it comes to deploying them in various market conditions.

By combining high-quality data along with model-free reinforcement learning techniques, not to mention systematic hyperparameter grid search as well as comprehensive evaluation metrics, this framework accordingly establishes a foundation for the purpose of engineering and deploying RL-based algorithmic trading strategies. The following chapter presents the results obtained from these simulations and analyzes their broader implications.

Results

4.1 Introduction

This chapter presents results obtained from the three variants of reinforcement learning (RL) algorithms, including Q-Learning, Greedy-GQ, and SARSA, on E-mini-NASDAQ-100 futures trading. In this study, the comparison is conducted using the datasets listed in **Section 3.2** and the methodologies described in **Section 3.6** to provide a systematic evaluation of these variants of the RL algorithms.

Section 4.2 outlines the datasets and their key characteristics. **Section 4.3** identifies each algorithm's optimal hyperparameter configuration with 50-seed optimization. **Section 4.4** presents results for Hypothesis 1 with 500-simulation validation tests. **Section 4.5** compares Q-Learning's performance with the NASDAQ benchmark and presents results equivalent to Hypothesis 2. **Section 4.6** analyzes risk-adjusted performance measures in order to address Hypothesis 3. **Section 4.7** synthesizes key results from the preceding analyses.

4.2 Data Exploration and Market Analysis

4.2.1 E-mini-Nasdaq-100 Futures Market Characteristics

The dataset used in this study includes price data for E-mini-Nasdaq-100 futures contracts spanning from May 2023 to May 2024. This period captures price movements across five consecutive futures contracts: NQM3, NQU3, NQZ3, NQH4, and NQM4. The data consists of 500,000-tick interval aggregated prices containing open, high, low, close, and volume information, in order to provide a comprehensive view of market behavior throughout the study period.

4.2.2 Overall Market Trend

The E-mini-NASDAQ-100 futures market experienced significant growth during the study period, in which there was an overall price increase of 37.74% from May 2023 to May 2024. This upward trajectory was not uniform as the market progressed through distinct phases of price action and volatility. The transition through different market regimes provides an ideal environment in order to test and evaluate the adaptive capabilities of reinforcement learning algorithms (Corazza & Sangalli, 2015). The market progressed from approximately 13,024 points to 17,939 points through distinct phases of varying momentum and volatility. **Figure 4-1** displays the complete price trajectory, showing the market's overall upward trend with notable periods of consolidation and acceleration across all five futures contracts. In the candlestick chart, green bars represent bullish movements (close price > open price) while red bars indicate bearish movements (close price < open price).

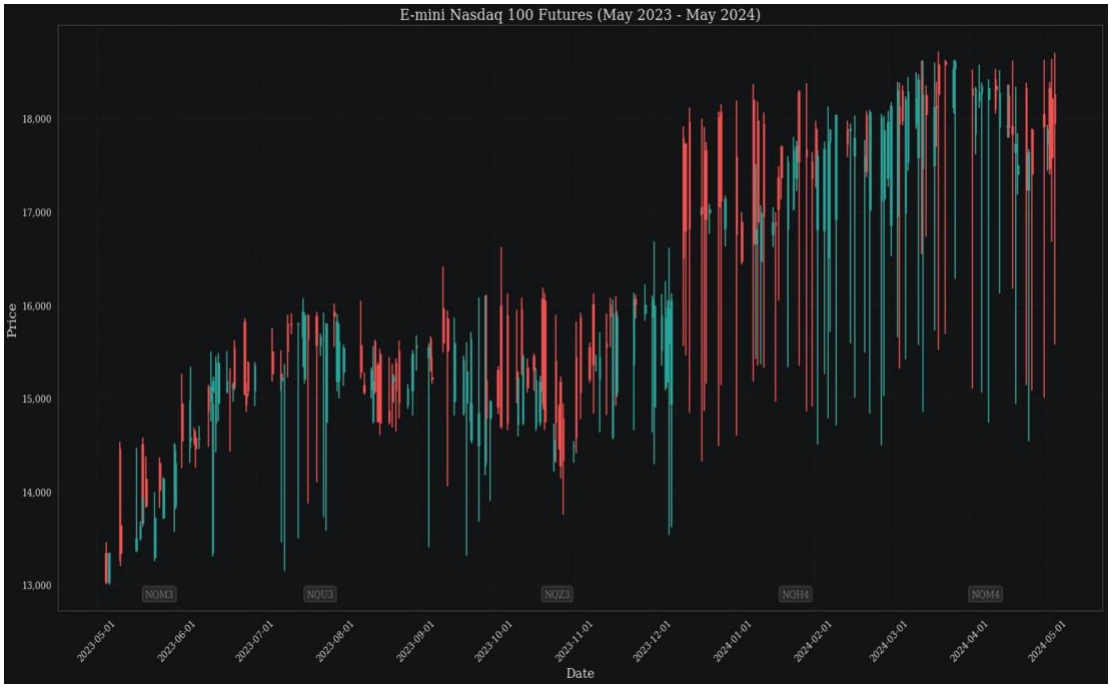

Figure 4-1 E-mini-Nasdaq100 Futures Market Trend (2023-2024)

4.2.3 Contract-Specific Analysis

Each futures contract exhibited distinct characteristics, which are presented in the following section. **NQM3 (May 2023 – June 2023)**: Strongly bullish with +14.71% return, low volatility (4.71%), and 66.7% positive trading days. Price range: $13,001 to $15,324.**NQU3 (June 2023 – Sep 2023)**: Neutral phase with +2.32% performance, minimal volatility (5.25%), however fewer favorable days (44.9%). Price range: $13,150 to $16,062. **NQZ3 (Sep 2023 – Dec 2023)**: Continued neutral trend with +2.90% return but increased moderate volatility (7.17%) and 47.2% positive days. Price range: $13,310 to $16,674. **NQH4 (Dec 2023 – Mar 2024)**: Return to strong bullish trend with +10.83% return, high volatility (12.26%), and 63.3% positive days. Price range: $14,324 to $18,436. **NQM4 (March 2024 – May 2024)**: Bearish trend with -1.49% return, high volatility (10.98%), and only 39.4% positive days. Price range: $14,537 to $18,709. **Table 4-1** contains market regimes covering the trading period from 2023 to 2024.

Table 4-1 Market Regimes (2023 –2024)

Contract	Period	Price Range	Start → End Price	Return (%)	Up Days (%)	Avg. Volatility (%)	Market Regime
NQM3	May 4 - Jun 14, 2023	13,001 - 15,324	13,024 → 14,940	+14.71	66.7	4.71	Strongly Bullish, Low Volatility
NQU3	Jun 12 - Sep 7, 2023	13,150 - 16,062	14,856 → 15,201	+2.32	44.9	5.25	Neutral, Low Volatility
NQZ3	Sep 11 - Dec 8, 2023	13,310 - 16,674	15,588 → 16,040	+2.90	47.2	7.17	Neutral, Moderate Volatility
NQH4	Dec 12 - Mar 8, 2024	14,324 - 18,436	16,500 → 18,287	+10.83	63.3	12.26	Strongly Bullish, High Volatility
NQM4	Mar 11 - May 3, 2024	14,537 - 18,709	18,210 → 17,939	-1.49	39.4	10.98	Bearish, High Volatility
Overall	May 4, 2023 - May 3, 2024	13,001 - 18,709	13,024 → 17,939	+37.74	52.3	8.07	Bullish, Moderate Volatility

Figure 4-2 presents the detailed price action for each individual contract, which includes the distinct trading environments encountered during the study period.

Figure 4-2 Multi-Panel Price Action

4.2.4 Price Distribution Analysis

The price behavior across the five futures contracts shows distinct distribution patterns that reflect the market's progression over the study period. The price distribution analysis reveals the following:

1. **NQM3** operated in the 13,000-14,500 range.
2. **NQU3** had a concentrated distribution around 15,000.
3. **NQZ3** showed a broader, more dispersed distribution spanning 14,000-16,000.
4. **NQH4** displayed a bimodal distribution centered around 17,000.
5. **NQM4** was concentrated at high-end price points around 18,000.

Figure 4-3 illustrates the clear progression and separation of price distributions across the five contracts, in which each contract occupies a distinct price range. This progression reflects the overall upward trend of the market with minimal overlap between contracts, which demonstrates that the algorithms were tested across diverse price environments.

Figure 4-3 Price Density Distribution

4.2.5 Volatility Progression

Average volatility increased progressively throughout the study period:

- **NQM3**: 4.71%
- **NQU3**: 5.25%
- **NQZ3**: 7.17%
- **NQH4**: 12.26%
- **NQM4**: 10.98%

This pattern shows a substantial increase in market volatility from the initial contract to the fourth contract, with a slight moderation in the final period.

4.2.6 Market Regimes Summary

The data has four distinctly defined market regimes:

1. **Initial Strong Bullish Phase** (NQM3): Strong bullish trend with low volatility.
2. **Consolidation Phase** (NQU3, NQZ3): Price action within range.
3. **Renewed Bull Run** (NQH4): Strong bullish movement with high but stable volatility.
4. **Transitional Correction** (NQM4): Relatively bearish conditions with high volatility.

Average volatility across all contracts was 8.07%, with a sizable range during the period. This provides the market backdrop against which to evaluate the three reinforcement learning algorithms.

4.3. Hyperparameter Optimization Results

The hyperparameter optimization was conducted using a fixed initial capital of 33,600 points (double the minimum margin requirement of $16,800) through a 50-seed

grid search to ensure statistical reliability and avoid overfitting to single runs. Each point is worth $20. That is, Q-Learning's 698-point increase in NQM3 is equivalent to $13,960 profit beyond initial capital.

4.3.1 Individual Contract Results

- **NQM3:** Q-Learning achieved optimal performance with 698 points (n=5, sigmoid, minusMean), outperforming SARSA (469 points) and GreedyGQ (193 points).
- **NQU3:** GreedyGQ achieved the best performance with 497 points (n=10, sigmoid, minusMean), followed by SARSA (248 points) and Q-Learning (155 points).
- **NQZ3:** Q-Learning dominated with 821 points (n=5, sigmoid123, minusMean), followed by GreedyGQ (644 points) and SARSA (353 points).
- **NQH4:** Q-Learning led with 812 points (n=10, sigmoid, minusMean), followed by SARSA (317 points) and GreedyGQ (311 points).
- **NQM4:** Q-Learning achieved 403 points (n=10, hypTanh, minusMean), followed by SARSA (379 points) and GreedyGQ (325 points).

The optimization outcomes show mixed levels of performance with respect to different combinations of parameters and market situations. **Figure 4-4** reveals the distribution of profit and loss (P&L) outcomes for each futures contract under the optimization tests.

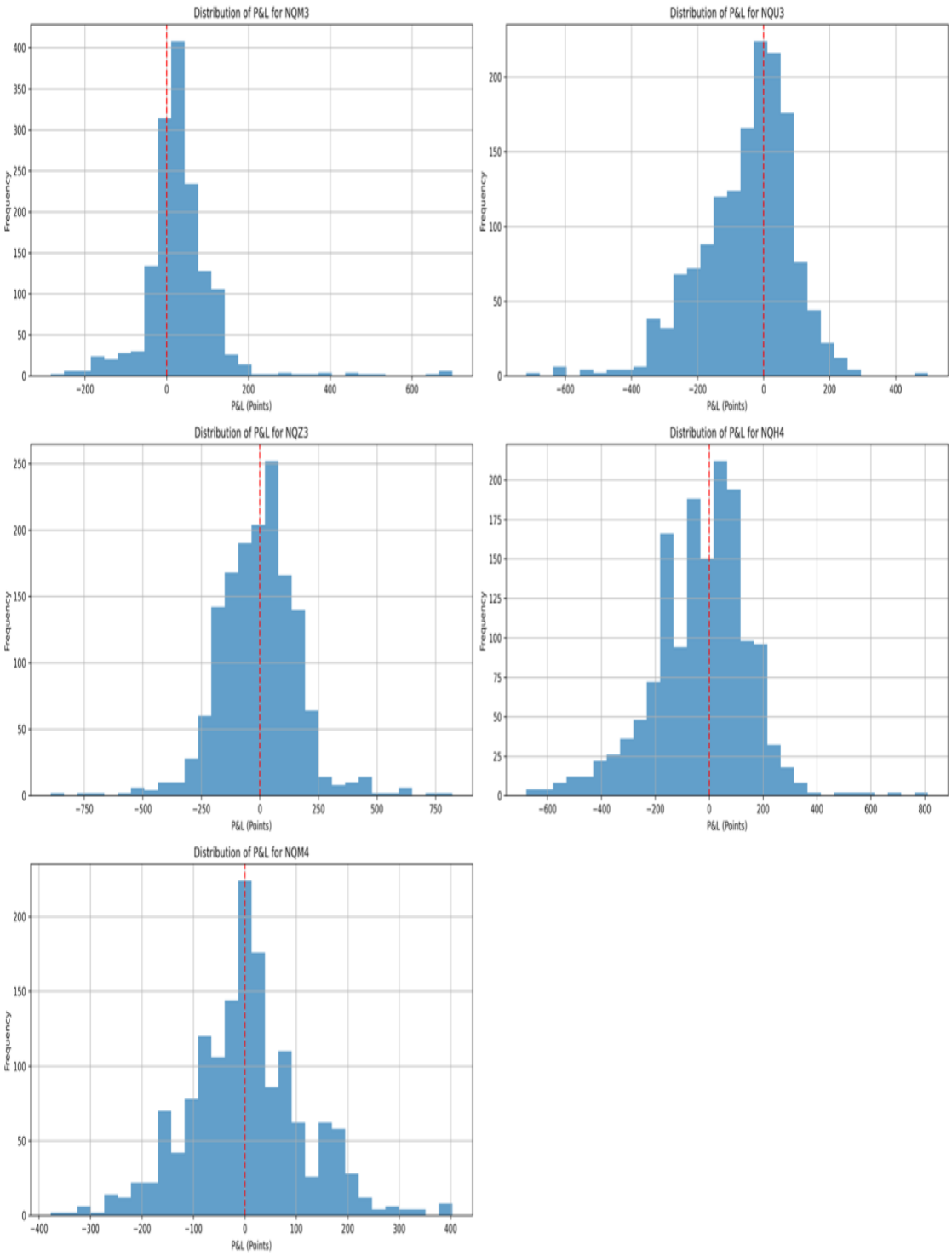

Figure 4-4 P&L Distribution by Ticker

4.3.2 Hyperparameter Patterns Across All Tickers

The 50-seed optimization step systematically checked all potential parameter combinations for each algorithm such that the number of combinations was the product of all choices available. This gave rise to 864, 432, and 216 combinations that corresponded to SARSA, GreedyGQ, and Q-Learning, respectively, making a total of 1,512 unique hyperparameter configurations combined across all algorithms and contracts. In order to identify the most effective parameters, analysis of these combinations revealed:

- **Reward Function:** The immediate reward achieved the highest performance ratio (66.7%), followed by mean reward (59.3%) and minusMean (46.7%).
- **Basis Function:** HypTanh achieved the highest performance with a success ratio of 68.75%, whereas the sigmoid and sigmoid123 achieved about 43%.
- **Look-back Period:** n=10 showed optimal performance (64.7% success ratio), compared to n=5 (46.5%) and n=15 (40%).

4.4 Hypothesis 1 Results

H1: *In model-free reinforcement learning techniques for algorithmic trading, the hyperparameters, including basis function (hypTanh), look-back period (5), and reward function (minus mean), are the most indicative of future performance.*

To validate this hypothesis, the optimal hyperparameter configurations identified through the 50-seed optimization were tested using 500 independent simulations with compounding capital. This production-level testing evaluates performance under realistic trading conditions where profits and losses accumulate over time. Among the 15 optimal

configurations (3 algorithms × 5 tickers) tested with 500 simulations each, the most frequently selected hyperparameters were:

- **Reward function minusMean:** 73.3% frequency
- **Look-back period 5:** 46.7% frequency
- **Basis function hypTanh:** 46.7% frequency

Figure 4-5 demonstrates the selection frequency for each hyperparameter category, showing the dominance of the hypothesized parameters in the 500-simulation configurations.

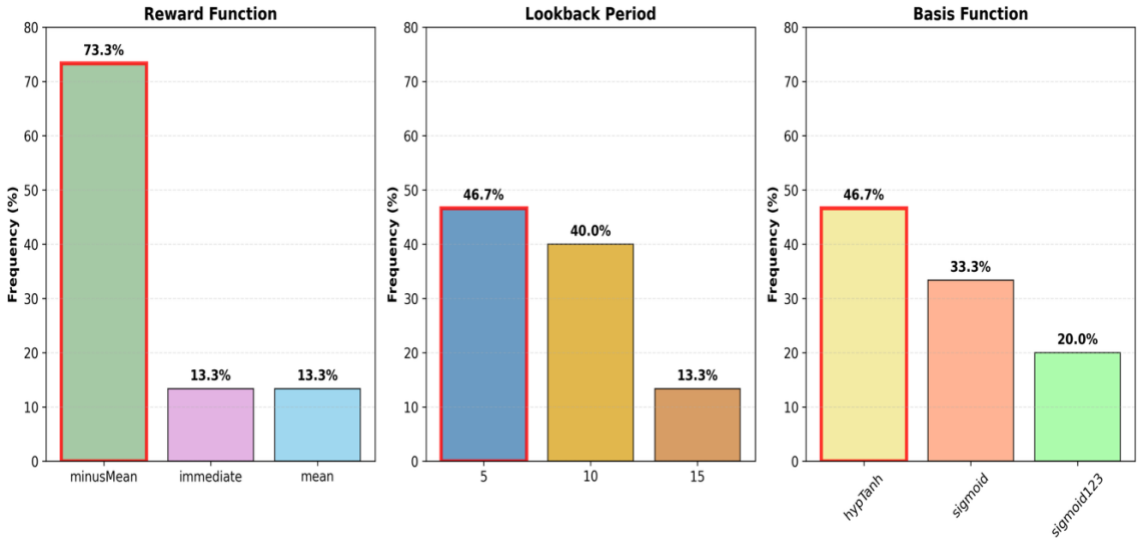

Figure 4-5 Hyperparameter Selection Frequency

The results support the hypothesis that these hyperparameters are most indicative of performance. Look-back period 5 and minusMean reward function emerged as the most frequently selected parameters, while hypTanh represented the plurality among basis functions at 46.7%. The 500-simulation production testing created conditions in which the hypothesized hyperparameters successfully translated from optimization to real

trading performance, so that their predictive value for algorithmic trading applications was validated. The implementation results showed performance variability across tickers (ranging from 699 to -8 points), which indicates that hyperparameter effectiveness is context-dependent. While the specific optimal configurations varied by contract to adapt to different market conditions, the overall dominance of hypTanh (46.7%), lookback period 5 (46.7%), and minusMean reward function (73.3%) across all implementations confirms the hypothesis that these hyperparameters are the most indicative of future performance. **Figure 4-6** illustrates the performance distributions and impact of each hyperparameter category in the 500-simulation production testing.

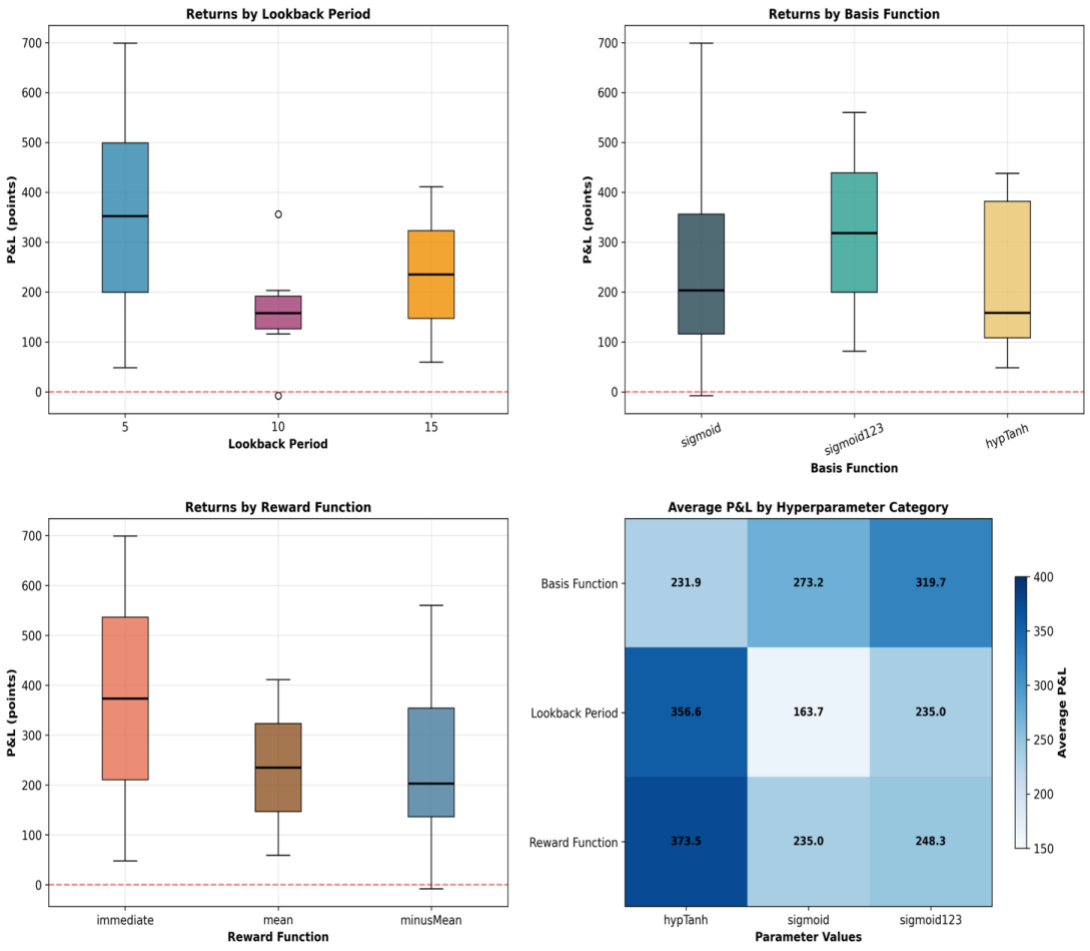

Figure 4-6 Hyperparameter Impact on Trading Performance (500 Simulations)

4.5 Hypothesis 2 Results

H2: *The Q-learning algorithm can generate a positive return that is at least 50% greater than the benchmark returns of the NASDAQ-100 index, when trading the E-mini-NASDAQ-100 futures through algorithmic trading.*

The 500-simulation validation demonstrates that Q-Learning significantly exceeds the performance benchmark. Starting with an initial capital of $33,600, the Q-Learning algorithm achieved a year-to-date return of 113.25% through sequential trading across five E-mini-NASDAQ-100 futures contracts, with which profits and losses from each contract period accumulate into the starting capital for the subsequent period, that is, creating a compounding effect. This performance substantially surpasses the NASDAQ index, which recorded a 37.8% return during the same period, from $12,982.48 on May 4, 2023, to $17,890.80 on May 3, 2024 (NASDAQ, 2024). Q-Learning's 113.25% return exceeds both the benchmark and the hypothesis target of 50% greater returns (56.7%), which demonstrates outperformance of 75.45 percentage points above the index. **Figure 4-7** illustrates the cumulative performance comparison across all contract periods.

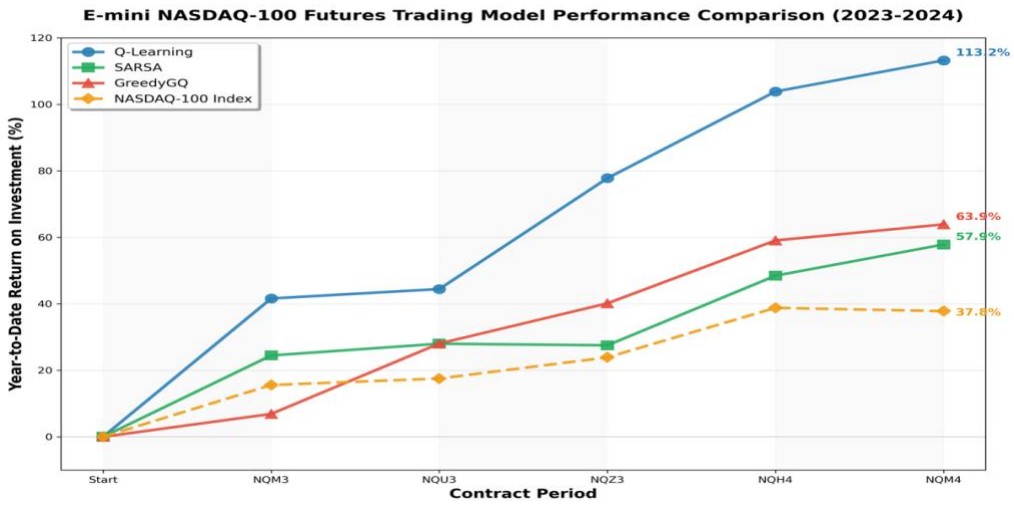

Figure 4-7 Model Performance Benchmark Comparison

Q-Learning's performance across the five contract periods demonstrates consistent profitability with compounding capital. Starting with $33,600, the algorithm achieved its highest return of 41.61% in NQM3 using a 5-day lookback period with sigmoid basis function and immediate reward. This was followed by strong performance in NQZ3 (23.09%) using a 5-day lookback period with sigmoid123 basis function and minusMean reward, and NQH4 (14.66%) employing a 5-day lookback period with hypTanh basis function and minusMean reward. Even the lowest performing period, NQU3 (2.01%), remained profitable using a 5-day lookback period with hypTanh basis function and immediate reward. The final period NQM4 achieved 4.60% return with a 10-day lookback period, hypTanh basis function, and minusMean reward. These varying hyperparameter combinations across contracts demonstrate the algorithm's adaptability to different market conditions, contributing to the cumulative 113.25% year-to-date return and final balance of $71,651.

4.5.1 Comparative Algorithm Performance

All three reinforcement learning algorithms substantially outperformed the 37.8% NASDAQ benchmark. Both Q-Learning and GreedyGQ demonstrated consistent performance with positive returns across all contracts, while SARSA experienced only one minor loss (- 0.38% in NQZ3).

- **Q-Learning:** 113.25% return ($71,651 final balance)
- **GreedyGQ:** 63.91% return ($55,074 final balance)
- **SARSA:** 57.86% return ($53,040 final balance)

Looking at the comprehensive results in **Table 4-2**, all three algorithms together achieved 78.35% performance—clear evidence that model-free RL approaches can handle futures trading effectively.

Table 4-2 Summary of Returns

Model	Ticker	Starting Capital	P&L $	Ending Balance	ROI (%)	YTD ROI (%)
GreedyGQ	NQM3	$33,600	$2,313	$35,913	6.88	6.88
Q-Learning	NQM3	$33,600	$13,980	$47,580	41.61	41.61
SARSA	NQM3	$33,600	$8,222	$41,822	24.47	24.47
GreedyGQ	NQU3	$35,913	$7,113	$43,027	19.81	28.05
Q-Learning	NQU3	$47,580	$956	$48,536	2.01	44.45
SARSA	NQU3	$41,822	$1,187	$43,009	2.84	28.00
GreedyGQ	NQZ3	$43,027	$4,064	$47,090	9.44	40.15
Q-Learning	NQZ3	$48,536	$11,208	$59,745	23.09	77.81
SARSA	NQZ3	$43,009	-$163	$42,846	-0.38	27.52
GreedyGQ	NQH4	$47,090	$6,360	$53,450	13.51	59.08
Q-Learning	NQH4	$59,745	$8,757	$68,502	14.66	103.88
SARSA	NQH4	$42,846	$7,040	$49,886	16.43	48.47
GreedyGQ	NQM4	$53,450	$1,624	$55,074	3.04	63.91
Q-Learning	NQM4	$68,502	$3,149	**$71,651**	4.60	**113.25**
SARSA	NQM4	$49,886	$3,154	$53,040	6.32	57.86

4.6 Hypothesis 3 Results

H3: *Q-learning, as a model-free reinforcement learning technique, is the best at generating low-volatility profits for risk-averse investors in algorithmic trading.*

The aggregated performance analysis across all five NASDAQ-100 futures contracts demonstrates that Q-learning is the optimal choice for risk-averse investors, outperforming competing reinforcement learning models in risk-adjusted returns. The results are based on 2,500 independent simulation outcomes (500 simulations per contract) using a compounding capital strategy.

4.6.1 Overall Performance Summary

- Q-learning achieved a year-to-date ROI of 113.25%.
- $33,600 initial capital grew to a final portfolio value of $71,651.
- Total profit made for Q-learning was $38,051 across all contract periods.
- Cumulative ROI across all three models was 78.35%.

4.6.2 Risk-Adjusted Performance Metrics

The favorable performance of Q-learning for risk-averse investors is demonstrated with four leading risk measures:

Standard Deviation: Q-learning registered the lowest portfolio volatility at 746.86, significantly lower than SARSA (817.82) and GreedyGQ (1,086.49), confirming the hypothesis of low volatility profit generation.

Sharpe Ratio: Q-learning achieved the optimal risk-adjusted returns with a Sharpe ratio of 0.55, outperforming SARSA (0.26) and GreedyGQ (0.23), which implies it generated approximately 2 times more return per unit of risk compared to other models.

Sortino Ratio: Q-learning demonstrated better control over downside risk with a Sortino ratio of 3.14, more than double that of SARSA (0.51) and GreedyGQ (0.37), with better downside protection and the ability to limit losses during market declines.

CALMAR Ratio: Q-learning exhibited the strongest risk-adjusted performance with a CALMAR ratio of 0.14, substantially outperforming both SARSA (0.05) and GreedyGQ (0.07), demonstrating superior return relative to maximum drawdown.

Table 4-3 Risk Metrics for RL Models

Model	Mean Return	Std Dev	Sharpe	Sortino	CALMAR
Q-Learning	51973.26	746.86	0.5488	3.138	0.1364
SARSA	42427.03	817.82	0.2556	0.5078	0.0549
GreedyGQ	42867.55	1086.49	0.2316	0.372	0.0676

4.7 Chapter Summary

This chapter provided empirical findings from implementing all three reinforcement learning algorithms on E-mini-NASDAQ-100 futures trading for a one-year period. All three research hypotheses were supported, with the results showing that Q-learning was the best- performing algorithm on a number of performance metrics.

Discussions and Conclusions

5.1 Discussion

The goal of this research was to determine whether model-free reinforcement learning techniques could effectively generate consistent profits with lower volatility for risk-averse investors in algorithmic trading. Previous research has explored the application of RL approaches in financial markets, achieving varying degrees of success across different market environments and asset classes (Fischer, 2018; Hilpisch, 2021; Meng & Khushi, 2019; Song et al., 2023). Researchers have demonstrated the potential of Q-Learning, SARSA, and Greedy-GQ algorithms in trading applications, in which practical implementations have shown promise (Corazza et al., 2019; Giorgi et al., 2023; Pereira, 2022; Ritter, 2017).

However, a side-by-side comparative analysis specifically tailored for risk-averse investment strategies in technology-focused US futures markets across varying market regimes and hyperparameter configurations remains limited. That being the case, this study addressed these gaps by systematically evaluating all three algorithms in the E-mini-NASDAQ-100 futures market, in order to provide comprehensive insights for risk-averse traders.

The remainder of this chapter presents the findings related to the hypotheses introduced in Chapter 1, along with insights addressing the corresponding research questions. The results and predictions of the model discussed in Chapter 4 have been interpreted in **Section 5.1.1** through **Section 5.1.3**. A brief overview of key conclusions and lessons learned from empirical results is provided in **Section 5.2**. **Section 5.3** contains the research's contribution to the existing literature on reinforcement learning and algorithmic trading. **Section 5.4** introduces directions for areas of future research that

can potentially extend the understanding of the applications of reinforcement learning methods for algorithmic trading.

5.1.1 Discussion of Hypothesis 1

H1: In model-free reinforcement learning techniques for algorithmic trading, the hyperparameters, including basis function (hypTanh), look-back period (5), and reward function (minusMean), are the most indicative of future performance.

This research employed 50-seed optimization across 1,512 parameter combinations to test Hypothesis 1, so that thorough parameter exploration could take place. This step was followed by 500-simulation validation with compounding capital, through which key insights for risk-averse trading strategies were uncovered.

Reward Function Analysis: As revealed, **minusMean reward** was dominant among reward functions (73.3% frequency, 248.27 P&L). It calculates reward by subtracting average profit from current trade profit (Pereira, 2022; Sutton & Barto, 2018), which teaches algorithms relative performance. That is to say, if a trade makes $100 but the average is $120, the -$20 reward signals underperformance, so as to prevent overconfidence during periods of consecutive profitable trades. **Mean reward**, on the other hand, (13.3% frequency, 235.00 P&L) smooths learning through running averages but creates sluggish adaptation that struggles to capitalize on emerging opportunities (Pereira, 2022; Sutton & Barto, 2018). In fact, **Immediate reward** achieved the highest P&L of 373.50 with 13.3% frequency by rewarding direct trade profits (e.g., $100 trade = $100 reward), which creates explosive performance in favorable conditions but erratic behavior across different market regimes, thereby advocating high-risk behavior unsuitable for conservative strategies (Pereira, 2022; Sutton & Barto, 2018).

Lookback Period Analysis: The impact of lookback window selection on the efficacy of the RL algorithms has been documented in prior studies (Corazza et al., 2019; Pereira, 2022). In this particular study, Lookback n=5 achieved both the highest frequency (46.7%) and P&L (356.57) by capturing immediate market momentum without noise from distant price history, so as to enable rapid adaptation that is essential in volatile futures markets where conditions change within hours (CME Group, 2025; Pereira, 2022). Lookback n=10, on the other hand, achieved moderate performance (163.67 P&L, 40% frequency) by balancing recent trends with broader context, but this compromise often resulted in delayed responses to market shifts. Lookback n=15 showed sporadic success (235.00 P&L, 13.3% frequency). It performed well in strongly bullish conditions (411 P&L in NQM3) but struggled in neutral markets (59 P&L in NQU3), demonstrating that extensive historical data works better during clear directional trends but becomes less useful when markets lack strong momentum (Corazza et al., 2019).

Basis Function Analysis: The fact that the choice of basis functions significantly impacts the performance of the RL algorithms in financial applications has been established in some RL papers (Sutton & Barto, 2018; Pereira, 2022; Corazza et al., 2019). As revealed, **HypTanh** led in frequency (46.7%, 231.86 P&L) through its bounded output range (-1 to +1) and smooth gradients, which provide mathematical stability across varying market volatility levels (Sutton & Barto, 2018). **Sigmoid** achieved moderate performance (33.3% frequency, 273.20 P&L) with 0-1 bounded output, but its steeper gradient transitions occasionally caused the algorithm to rapidly flip between buy and sell signals when market prices hit critical thresholds, creating

inconsistent trading behavior during market regime shifts where adaptation would be necessary.

Sigmoid123, on the other hand, achieved the highest returns (319.67) but low frequency (20%) due to extreme regime sensitivity, which showed 560 P&L in neutral NQZ3 versus 81 in bearish NQM4 (Pereira, 2022). This reveals that its discrete categories (1, 2, 3) work brilliantly for clear trends but fail catastrophically when market conditions fall between categories, so that performance becomes unpredictable. That is to say, if sigmoid123 classifies market states as 1='bearish', 2='neutral', 3='bullish', it forces complex market conditions into these three boxes only. As a result, when markets show mixed signals or transitional periods, the algorithm must still pick one of these three categories even when the market doesn't clearly fit any of them, therefore creating erratic trading decisions when market reality doesn't match its simplified classification (Sutton & Barto, 2018).

Practical Implications: As demonstrated, these findings validate that consistency trumps peak performance for risk-averse strategies. As a result, the dominance of hypTanh, n=5 lookback, and minusMean rewards creates a reliable framework for algorithmic trading, thereby serving as a benchmark for further trading strategy development.

Hyperparameter Selection Results: Hypothesis confirmed.

5.1.2 Discussion of Hypothesis 2

H2: *The Q-learning algorithm can generate a positive return that is at least 50% greater than the benchmark returns of the NASDAQ-100 index, when trading the E-mini-NASDAQ-100 futures through algorithmic trading.*

Hypothesis 2 was validated through comprehensive testing across five consecutive futures contracts using compounding capital strategy, so that robust performance evaluation could be achieved. Q-Learning achieved a 113.25% year-to-date return, exceeding the hypothesized 50% outperformance threshold above the NASDAQ-100 benchmark's 37.8% return, accordingly, transforming $33,600 into $71,651. In fact, this 75.45 percentage point outperformance validates Q-Learning's superior adaptive learning mechanism as well as its off-policy exploration capabilities.

Algorithm Performance Analysis: Dominating with 113.25% over GreedyGQ (63.9%) and SARSA (57.9%), Q-Learning demonstrates superior performance of the foundational off-policy approach, so that the benefits of this methodology become evident (Sutton & Barto, 2018).

Due to the fact that Q-Learning's off-policy learning excels in volatile markets, it can explore the potential outcomes of selling at market peaks without executing the risky sell position. Therefore, Q-learning can build superior market intuition through consequence-free experimentation. Also, because Greedy-GQ's stabilization features are designed to prevent erratic trading decisions, they can hinder rapid adaptation to market changes. However, SARSA's on-policy constraint forces slower learning since it cannot explore alternatives without real consequences. This proves financial markets reward swift adaptation over unduly cautious stability, explaining why Q-Learning's "learn from everything" approach outperformed both Greedy-GQ's "learn carefully with safeguards" and SARSA's "learn from experience only" methodologies.

Comparing these results with Pereira's (2022) findings where SARSA achieved 173.30% returns in the Brazilian mini-index futures market with Q-Learning ranking

second, it is imperative to note that the optimal algorithm selection is highly dependent on specific market microstructures and asset class characteristics. This sheds light on market variation and complexity. Stating it differently, algorithms that excel in emerging markets with different volatility patterns and trading volumes may not translate effectively to the highly liquid, tech-driven NASDAQ-100 futures environment with its distinct market features and institutional participation.

Practical Implications: Q-Learning's superior cumulative performance across all market regimes positions it as a potentially optimal choice for institutional deployment in the US derivatives markets. That said, practitioners must take into account careful risk management practices tailored to specific investor risk profiles, particularly when the compounding approach can amplify both gains and losses in real-world applications.

Q-learning Performance Results: Hypothesis confirmed.

5.1.3 Discussion of Hypothesis 3

H3*:* *Q-learning, as a model-free reinforcement learning technique, is the best at generating low-volatility profits for risk-averse investors in algorithmic trading.*

Hypothesis 3 was evaluated using a risk-adjusted performance analysis across multiple metrics. These were designed so that different dimensions of investment risk could be captured (Sharpe, 1994; Sortino & van der Meer, 1991). The results in **Section 4.6** provide compelling evidence which indeed supports Q-Learning's advantage. In particular, the results revealed that Q-Learning, while generating the highest return of 113.25%, showed the lowest return volatility at 746.86. SARSA achieved 817.82 and Greedy-GQ reached 1,086.49. This establishes Q-Learning as the most stable performer among the three algorithms.

Risk Metrics Analysis: The Sharpe ratio analysis was instrumental in revealing Q-Learning's risk-adjusted performance at 0.55, generating approximately 2.1 times more return per unit of risk than SARSA (0.26) and 2.4 times more than Greedy-GQ (0.23) (Sharpe, 1994). This superior risk-adjusted performance indicates that Q-Learning not only generates higher absolute returns but does so with more favorable risk characteristics, a critical requirement for conservative investment strategies.

Establishing even more compelling evidence for risk-averse investors, the Sortino ratio result for Q-learning is significantly better than the other two models. Q-Learning achieved 3.14 compared to SARSA's 0.51 and Greedy-GQ's 0.37. In this analysis, higher ratios are indicative of superior performance (Sortino & van der Meer, 1991). The results show the Sortino ratio's focus on downside deviation makes it particularly relevant for conservative investors who prioritize capital preservation over aggressive growth. That being the case, Q-Learning's strong Sortino performance demonstrates its ability to minimize negative volatility while maintaining strong positive returns, so as to address the core concerns of risk-averse market participants.

The CALMAR ratio analysis further reinforces Q-Learning's superiority, with a ratio of 0.14 substantially outperforming both competing algorithms (Young, 1991). This metric emphasizes maximum drawdown recovery. As a result, it becomes particularly valuable for risk-averse investors. These are investors whose portfolios must maintain consistent performance through adverse market conditions, so that their portfolios remain stable over time. When looking at Q-Learning's superior CALMAR performance, it becomes apparent that the algorithm's ability to recover quickly from temporary setbacks is better than other models.

Practical Implications: Generating comparative absolute and relative returns with low volatility characteristics, Q-Learning becomes uniquely suited for conservative investment strategies. These strategies prioritize consistent performance over speculative gains, ensuring that risk is minimized. The superior risk-adjusted metrics support their adoption in institutional portfolios. That said, tailored implementation remains important for portfolios that focus on capital preservation and steady growth.

Q-learning Model Selection Results: Hypothesis Confirmed.

5.2 Conclusion

This praxis research aimed to address the research questions outlined in Chapter 1:

RQ1: Which hyperparameters are the most indicative of future performance in model-free reinforcement learning techniques, including Q-Learning, Greedy-GQ, and SARSA, for algorithmic trading?

RQ2: Can the Q-learning algorithm generate a positive return that exceeds the NASDAQ-100 index benchmark when trading the E-mini-NASDAQ-100 futures through algorithmic trading?

RQ3: Which model-free reinforcement learning technique—Q-Learning, Greedy-GQ, or SARSA—is the best at generating low-volatility profits for risk-averse investors through algorithmic trading?

Based on the systematic evaluation across diverse market conditions, it is important to acknowledge that there is no one-size-fits-all approach to RL algorithm and hyperparameter selection in algorithmic trading. Optimal model selection depends on numerous factors, including but not limited to asset class characteristics, trading periods,

specific market microstructures, capital allocation strategies, and investor risk tolerance profiles. When all factors are considered, this praxis draws the following primary conclusions:

1. **Algorithm Selection**: Q-Learning worked best for risk-averse strategies in the E-mini-NASDAQ-100 futures study. However, other algorithms might exhibit superior performance in different market environments.
2. **Hyperparameter Significance**: The combination of minusMean rewards, n=5 lookback, and hypTanh basis functions performed well across different market conditions and contributed to the positive performance of all three models, in particular, Q-learning. This provides practitioners with a data-driven comparative baseline.
3. **Benchmark Outperformance**: All three algorithms beat the NASDAQ-100 benchmark despite its unusually high return. They achieved 78.35% total returns, which shows that RL approaches work better than just buying and holding in this market.
4. **Risk-Return Balance**: It is evident that the RL algorithms managed to generate profits while keeping risks under control. Q-Learning, in particular, maintained the balance between risk and return, through which it could generate solid returns without taking on excessive risk when trading E-mini futures.

5.2.1 Lessons Learned

The implementation of this research across varying market conditions sheds light on several practical insights for algorithmic trading deployment and enables practitioners to benefit from these findings.

1. **Compounding Capital Strategy:** Although the compounding capital approach amplified both gains and potential losses throughout the study period, all algorithms managed to maintain positive profitability. As capital increased from successful trades, it is evident that the risk of larger losses grew proportionally, but the reward potential also scaled accordingly. This proves the viability of this approach for moderately risk-tolerant implementations in financial trading systems.
2. **Market Regime Adaptability**: Overall model performance remained favorable despite individual period fluctuations, during which transitions from bullish trends to bearish corrections confirmed that properly configured RL-based trading systems can operate efficaciously across different market environments.
3. **State Representation Design**: The integration of technical indicators (Bollinger Bands, EMA, LWMA) matters for how well the RL algorithm performs. Structured market features such as these help RL models learn more effectively than when they only consider raw price data.
4. **Off-Policy Learning Advantage**: Q-Learning's off-policy learning mechanism consistently outperformed the on-policy approach (SARSA) across risk-adjusted metrics, from which it is evident that exploration of hypothetical scenarios without execution provides superior learning capabilities in volatile financial markets.

5.3 Contribution to the Body of Knowledge

This research advances the existing literature on reinforcement learning applications in financial markets through several contributions that are discussed in the following section:

1. **E-mini-NASDAQ-100 Futures Systematic Comparison**: Conducting a novel systematic comparison of model-free Q-Learning, SARSA, and Greedy-GQ algorithms within the US-based E-mini-NASDAQ-100 futures market, this study provided a performance evaluation in a highly liquid, institutionally dominated, and technology-driven environment. Accordingly, the findings provide insights into RL efficacy for practical trading.
2. **Robust Simulation Methodology**: The 500-simulation validation protocol employs varied random seeds, by which statistical robustness is ensured while overfitting risks are mitigated. This methodology eliminates "lucky winner" outcomes in which single-run models might produce misleading results. Such methodological rigor surpasses typical approaches and strengthens the model's reliability for practical deployment.
3. **Compounding Capital Framework**: Implementing compounding capital strategies through which profits and losses accumulate over time, this study implements a framework that reflects real-world trading dynamics, unlike the fixed capital approaches prevalent in some of the existing research. Transaction costs were excluded to maintain consistency with prior literature and focus on algorithmic reliability. The compounding approach still mimics the behavior of live markets.

4. **Risk-Adjusted Performance Framework**: The integration of multiple risk-adjusted metrics including Sharpe, Sortino, and CALMAR ratios establishes a comprehensive evaluation framework where risk considerations complement return analysis. This addresses the identified gap in the literature, wherein emphasis on absolute returns often overshadows the risk dimensions, which leaves risk-averse investors without reliable assessment tools.
5. **Performance Across Distinct Market Regimes**: The evaluation of algorithms' performance across distinct market regimes during the 2023–2024 period provides unique insight into RL adaptability under changing market conditions in futures trading. The study encompassed five different market environments: strongly bullish low-volatility conditions (NQM3: 4.71% volatility, +14.71% return), neutral low-volatility periods (NQU3: 5.25% volatility, +2.32% return), neutral moderate-volatility conditions (NQZ3: 7.17% volatility, +2.90% return), strongly bullish high-volatility environments (NQH4: 12.26% volatility, +10.83% return), and bearish high-volatility periods (NQM4: 10.98% volatility, -1.49% return). This praxis explored the application of model-free RL algorithms across a range of market conditions, with volatility levels from 4.71% to 12.26%. The fact that these algorithms performed reliably across all trading cycles provides valuable insights for practitioners who face challenges meeting their investment goals in volatile markets.
6. **Theoretical Foundation Validation**: This research connected classical financial theory, discussed in **Section 2.2.2**, with modern reinforcement learning

applications. It shows that foundational theories remain relevant while improving their practical use in today's algorithmic trading systems.

5.4 Future Research Recommendations

The findings of this research suggest several promising directions for future investigation through which the application of reinforcement learning in algorithmic trading could advance further:

1. **Time-Based Intervals**: Future research can look into experimenting with time-based intervals such as high frequency and intraday trading in order to obtain complementary insights about the application of RL models in financial markets.
2. **Asset Class Expansion**: Testing the same algorithms across assets such as fixed income, foreign currencies, and commodities would reveal asset-specific performance characteristics and demonstrate hyperparameter configuration generalizability across different market microstructures.
3. **Enhanced State Representation**: Incorporating additional technical indicators beyond Bollinger Bands, EMA, and LWMA, future research could explore momentum, trend, volatility, and volume indicators in order to enhance algorithm performance through more sophisticated market feature capture.
4. **Ensemble Algorithm Development**: Although the models in this study were trained independently, the collective 78.35% return achieved across Q-learning, SARSA, and Greedy-GQ suggests that exploring ensemble reinforcement learning strategies in future research may also hold promise.
5. **Transaction Cost Integration**: Future research should evaluate how transaction costs affect RL algorithm performance. Specifically, commission structures, bid-

ask spreads, and market impact costs across different trading volumes and market conditions could be further explored.

6. **Extended and Alternative Time Periods:** Evaluating algorithm performance across extended training periods (e.g., 3-5 years) and different market cycles, future research could assess temporal robustness and generalizability. Testing across varying historical periods (e.g., pre-COVID, COVID period, post-COVID recovery) would reveal how algorithms perform under different macroeconomic conditions. This would facilitate the investigation of concept drift effects, in which evolving market dynamics may require algorithm retraining and recalibration to maintain performance consistency.

These research directions can expand the current understanding of reinforcement learning in algorithmic trading and wealth creation strategies. As artificial intelligence enhances the use of technology in financial markets, reinforcement learning emerges as an important force for innovation. It fuels the development of the next generation of adaptive, risk-aware trading systems.

References

Abu Mallouh, A., Qawaqneh, Z., Abuzaghleh, O., & Al-Rababa'a, A. (2023). A new efficient method for refining the reinforcement learning algorithm to train deep Q agents for reaching a consensus in P2P networks. *IEEE Access*. https://doi.org/10.1109/ACCESS.2023.3268283

Aloud, M. E., & Alkhamees, N. (2021). Intelligent algorithmic trading strategy using reinforcement learning and directional change. *IEEE Access*, *9*, 114659-114671. https://doi.org/10.1109/ACCESS.2021.3105259

Ansari, Y., Gillani, S., Bukhari, M., Lee, B., Maqsood, M., & Rho, S. (2024). A multifaceted approach to stock market trading using reinforcement learning. *IEEE Access, 12*, 1087–1101. https://doi.org/10.1109/ACCESS.2024.3418510

Ates, A., & Wang, G. H. K. (2005). Information transmission in electronic versus open-outcry trading systems: An analysis of U.S. equity index futures markets. *Journal of Futures Markets, 25*(7), 679–715. https://doi.org/10.1002/fut.20160

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In A. Prieditis & S. Russell (Eds.), *Machine learning proceedings 1995* (pp. 30–37).

Banerjee, A. (2022). An empirical risk and return analysis of select stocks in NASDAQ 100. *Accounting and Finance Research*. https://doi.org/10.5430/afr.v11n2p1

Bellman, R. (1957). *Dynamic programming*. Princeton University Press.

Berger, P. (2024). New Disruptions, Geopolitics Hang Over 2024 Supply Chains. *The Wall Street Journal*.

Bergh-Lindeque, A., Ferreira-Schenk, S., Dickason-Koekemoer, Z., & Habanabakize, T. (2022). What makes risk-averse investors tick? A practitioners guide. *Cogent Economics & Finance, 10*(1), 2111786.

Bradshaw, T. (2024). Nvidia and the AI boom face a scaling problem. *Financial Times*.

Carlson, D. (2024). The growth of tech and 25 years of Nasdaq futures. *Institutional Investor.*

Carapuço, J., Neves, R., & Horta, N. (2018). Reinforcement learning applied to FOREX trading. *Applied Soft Computing*, *73*, 783–794.

Charles Schwab. (n.d.). *NASDAQ-100 E-mini-Futures*. Retrieved from https://www.schwab.com/futures/nasdaq-100-emini

CME Group. (2025). *E-mini-NASDAQ-100 futures*. Retrieved from https://www.cmegroup.com/markets/equities/nasdaq/e-mini-nasdaq-100.html

Corazza, M., Fasano, G., Gusso, R., & Pesenti, R. (2019). A comparison among reinforcement learning algorithms in financial trading systems. Department of Economics, Ca' Foscari University of Venice.

Corazza, M., & Sangalli, A. (2015). Q-learning and SARSA: A comparison between two intelligent stochastic control approaches for financial trading. *Working Papers 2015:15*, Department of Economics, University of Venice "Ca' Foscari".

Corporate Finance Institute (2025). *Sharpe ratio*. Corporate Finance Institute. https://corporatefinanceinstitute.com/resources/valuation/sharpe-ratio/

DataBento. (2024). *Equity futures data*. Retrieved from https://databento.com/

Deep, A., Monico, C., Shirvani, A., & Fabozzi, F. J. (2024). Assessing the impact of technical indicators on machine learning models for stock price prediction. *arXiv preprint arXiv:2412.15448.*Retrieved from 10.48550/arXiv.2412.15448

Dolan, R. J., & Dayan, P. (2013). Goals and habits in the brain. *Neuron*, *80*(2), 312–325.

Enkhsaikhan, B., & Jo, O. (2024). Risk-averse reinforcement learning for portfolio optimization. *ICT Express*.

Fassas, A. P. (2021). Price discovery in a new futures market: Micro E-mini-index futures.

Fischer, T. G. (2018). Reinforcement learning in financial markets—a survey. *FAU Discussion Papers in Economics*, No. 12/2018.

Fishburn, P. C. (1979). *Utility theory for decision making*. Krieger.

Gupta L. (2020). *Comparison of hyperparameter tuning algorithms: Grid search, random search, Bayesian optimization*. Analytics Vidhya.

Giorgi, F., Herzel, S., & Pigato, P. (2023). A reinforcement learning algorithm for trading commodities. *Applied Stochastic Models in Business and Industry*. https://doi.org/10.1002/asmb.2825

Guo, M., Liu, J., Luo, Z., & Han, X (2024). Deep reinforcement learning for pairs trading: Evidence from China black series futures. *International Review of Economics and Finance*. Retrieved from https://ideas.repec.org/a/eee/reveco/v93y2024ipbp981-993.html

Hilpisch, Y. (2024). *Reinforcement learning for finance*. O'Reilly Media, Inc.

Hilpisch, Y (2021). *Artificial intelligence in finance: A Python-based guide*. O'Reilly Media.

Hirsa, A., Osterrieder, J., Misheva, B. H., & Posth, J.-A. (2021). Deep reinforcement learning on a multi-asset environment for trading.

Huang, H., Gao, T., Gui, Y., Guo, J., & Zhang, P. (2022). Stock trading optimization through model-based reinforcement learning with resistance support relative strength. *arXiv preprint arXiv:2205.15056*. Retrieved from https://arxiv.org/abs/2205.15056

Jeong, D. W., & Gu, Y. H. (2024). Pro Trader RL: Reinforcement learning framework for generating trading knowledge by mimicking the decision-making patterns of professional traders. *Expert Systems with Applications, 254*, 124465. https://doi.org/10.1016/j.eswa.2024.124465

Kim, K., & Yang, J. (2007). A multi-agent system for automated stock trading using Q-learning. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *37*(6), 864-877. https://doi.org/10.1109/TSMCA.2007.904825

Kolm, P. N., & Ritter, G. (2024). Modern perspectives on reinforcement learning in finance. *Courant Institute of Mathematical Sciences, New York University*.

Kurov, A. (2021). Tick size reduction, execution costs, and informational efficiency in the regular and E-mini-NASDAQ-100 index futures markets. *Business History*, *63*(7), 1197–1221. https://doi.org/10.1080/00076791.2019.1676231

Lee, J. W., Park, J., O, J., Lee, J., & Hong, E. (2007). A multiagent approach to Q-learning for daily stock trading. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 37*(6), 864–877. https://doi.org/10.1109/TSMCA.2007.904744

Levy, H., De Giorgi, E. G., Hens, T. (2011). Two paradigms and Nobel Prizes in economics: A contradiction or coexistence? *European Financial Management*, *18*(2), 163–182. https://doi.org/10.1111/j.1468-036X.2011.00617.x

Liu, P., Zhang, Y., Bao, F., Yao, X., & Zhang, C. (2022). Multi-type data fusion framework based on deep reinforcement learning for algorithmic trading. *Applied Intelligence, 53*(1683–1706)

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

Lopez de Prado, M. (2018). *Advances in financial machine learning*. Wiley.

Ma, S., Chen, Z., Zhou, Y., & Zou, S. (2021). Greedy-GQ with variance reduction: Finite-time analysis and improved complexity. *arXiv*. https://arxiv.org/abs/2103.16377

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77–91.

Mettenheim, H.-J., & Breitner, M. H. (2012). Forecasting and trading the high-low range of stocks and ETFs with neural networks. In *Communications in Computer and Information Science* (Vol. 311, pp. 423–432).

Meng, T. L., & Khushi, M. (2019). Reinforcement learning in financial markets. *Data*, *4*(3), 110. https://doi.org/10.3390/data4030110

Merli, M., Parent, A., & Edlinger, C. (2021). Portfolio advice before modern portfolio theory: The Belle Époque of French analyst Alfred Neymarck. *Business History*, *63*(7), 1197–1221. https://doi.org/10.1080/00076791.2019.1676231

Millea, A. (2021). Deep reinforcement learning for trading—a critical survey. *Data Review*.

Packt Publishing. (2018). *Machine learning for trading*. Packt Publishing.

Moriyama, K., Matsumoto, M., Fukui, K., & Numao, M. (2012). Reinforcement learning on a futures market simulator. In *Communications in Computer and Information Science: Vol. 449. Agent and Multi-Agent Systems: Technologies and Applications* (pp. 119–128).

Murphy, J. J. (1999). Technical analysis of the financial markets: A comprehensive guide to trading methods and applications. New York Institute of Finance.

Nasdaq. (n.d.). *Nasdaq official website*. https://www.nasdaq.com

Ngo, V. M., Nguyen, H. H., & Nguyen, P. V. (2023). Does reinforcement learning outperform deep learning and traditional portfolio optimization models in frontier and developed financial markets? *Research in International Business and Finance*.

Park, J.-H., Kim, J.-H., & Huh, J.-H. (2024). Deep reinforcement learning robots for algorithmic trading: Considering stock market conditions and U.S. interest rates. *IEEE Access*, *12*, 20705-20725.

Pereira, F. R. (2022). Trading financial markets using reinforcement learning: Application and analysis. *Department of Mathematics, University of Oslo*.

Pippas, N., Turkay, C., & Ludvig, E. A. (2024). The evolution of reinforcement learning in quantitative finance. *arXiv preprint arXiv:2408.10932*. Retrieved from https://arxiv.org/html/2408.10932v1

Ponomarev, E. S., Oseledets, I. V., & Cichocki, A. S. (2019). Using reinforcement learning in the algorithmic trading problem. *Journal of Communications Technology and Electronics*, *64*(12), 1450–1457.

Prasad, A., & Seetharaman, A. (2021). Importance of machine learning in making investment decisions in the stock market. *Vikalpa: The Journal for Decision Makers*, *46*(4), 209–222.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.

Ritter, G. (2017). Machine learning for trading. New York University, Courant Institute of Mathematical Sciences. Available at SSRN: https://ssrn.com/abstract=3015609

Rundo, F., Trenta, F., di Stallo, A. L., & Battiato, S. (2019). Machine learning for quantitative finance applications: A survey. *Applied Sciences, 9*(24), 5574. https://doi.org/10.3390/app9245574

Schwarzer, M., Anand, A., Goel, R., Hjelm, R. D., Courville, A., & Bachman, P. (2020). Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*. Retrieved from https://arxiv.org/abs/2007.05929

Sharpe, W. F. (1966). Mutual Fund Performance. The Journal of Business, 39(1), 119-138.

Sharpe, W. F. (1994). The Sharpe Ratio. The Journal of Portfolio Management, 21(1), 49-58.

Shleifer, A. (2000). *Inefficient markets: An introduction to behavioral finance*. Clarendon Press.

Singh, V., Chen, S.-S., Singhania, M., Nanavati, B., Kar, A. K., & Gupta, A. (2022). 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, 2*(2), 100094. https://doi.org/10.1016/j.jjimei.2022.100094

Song, Z., Wang, Y., Qian, P., Song, S., Coenen, F., Jiang, Z., & Su, J. (2023). From deterministic to stochastic: An interpretable stochastic model-free reinforcement learning framework for portfolio optimization. *Applied Intelligence, 53*(12), 15188–15203. https://doi.org/10.1007/s10489-022-04217-5

Sortino, F. A., & van der Meer, R. (1991). Downside risk. *Journal of Portfolio Management, 17*(4), 27–31. https://doi.org/10.3905/jpm.1991.409343

Straits Research. (2023). Key events of 2023 and their impact. *Straits Research*. https://straitsresearch.com/statistic/key-events-of-2023-and-their-impact

Sun, Q., Gong, X., & Si, Y.-W. (2023). Transaction-aware inverse reinforcement learning for trading in stock markets. *Applied Intelligence*, *53*, 28186–28206. https://doi.org/10.1007/s10489-023-04959-w

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.

Thakar, C., & Pachanekar, R. (2023). *Bollinger Bands explained: Trading strategy, formula, calculation and more*. QuantInsti. Retrieved from https://blog.quantinsti.com/bollinger-bands/

Theate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications, 173*, 114632. https://doi.org/10.1016/j.eswa.2021.114632

Van den Bergh-Lindeque, A., Ferreira-Schenk, S., Dickason-Koekemoer, Z., & Habanabakize, T. (2021). What makes risk-averse investors tick? A practitioner's guide. *Cogent Economics & Finance*, *10*, 2111786.

Vergura, S. (2020). Bollinger Bands based on Exponential Moving Average for statistical monitoring of multi-array photovoltaic systems. *Energies, 13*(15), 3992. Retrieved from https://doi.org/10.3390/en13153992

Vijaya, C., Hati, K., & Thenmozhi, M. (2024). Developing a security risk assessment-based smart beta portfolio model for robo advising. *Australasian Accounting, Business & Finance Journal, 18*(3), 7-25.

Vu, M. N., Nguyen, H. H., & Nguyen, P. V. (2023). Does reinforcement learning outperform deep learning and traditional portfolio optimization models in frontier and developed financial markets? *Research in International Business and Finance, 64*, 101750.

Wang, C. D., Liu, X. Y., Yang, L., Zhang, R., Xiao, B., Liu, Q., ... Numao, M. (2022). FinRL: A deep reinforcement learning library for automated stock trading. *arXiv preprint arXiv:2011.09607.* Retrieved from https://arxiv.org/abs/2011.09607

Wan, Z., Cho, J.-H., Zhu, M., Anwar, A. H., Kamhoua, C., & Singh, M. P (2024). Decision theory-guided deep reinforcement learning for fast learning. *arXiv preprint arXiv:2402.06023v1*.

Wyckmans, F., Banerjee, N., Saeremans, M., Otto, R., Kornreich, C., Vanderijst, L., Gruson, D., Carbone, V., Bechara, A., Buchanan, T., & Noël, X. (2022). The modulation of acute stress on model-free and model-based reinforcement learning in gambling disorder. *Journal of Behavioral Addictions*, *11*(3), 831–844. https://doi.org/10.1556/2006.2022.00059

Yahoo Finance. (2022). *Algorithmic trading market – Growth, trends, COVID-19 impact, and forecasts (2022–2027).* Yahoo Finance. https://finance.yahoo.com/news/algorithmic-trading-market-growth-trends-112400870.html

Yan, K., Wang, Y., & Li, Y. (2022). Enhanced Bollinger Band stock quantitative trading strategy based on random forest. *Artificial Intelligence Evolution, 3*(2). Retrieved from https://doi.org/10.37256/aie.4120231991

Yao, G., Zhang, N., Duan, Z., & Tian, C. (2024). Improved SARSA and DQN algorithms for reinforcement learning. *Theoretical Computer Science*. https://doi.org/10.1016/j.tcs.2024.115025

Yong, B. X., Rahim, M. R. A., & Abdullah, A. S. (2017). A stock market trading system using deep neural network. In Asian Simulation Conference (pp. 356-364).

Young, W. (1991). Calmar ratio: A smoother tool. *Futures.*

Zhang, Z., & Khushi, M. (2020). GA-MSSR: Genetic algorithm maximizing Sharpe and Sterling ratio method for robotrading. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE.

Zhang, Z., Zohren, S., & Roberts, S. (2019). Deep reinforcement learning for trading. *Department of Engineering Science, Oxford-Man Institute of Quantitative Finance, University of Oxford*.

Zhao, D., Wang, H., Shao, K., & Zhu, Y. (2016). Deep reinforcement learning with experience replay based on SARSA. *IEEE*.

Zulkifli, Z. S., Zamri, N., Mohammad, H., & Al-Saadi, R. A. A. R. (2024). Design algorithmic trading strategies with expert advisor using Linear Weighted Moving Average (LWMA) and Stochastic Oscillator Technical Indicators. https://doi.org/10.56427/jcbd.v3i2.404