DEEP OPTIONS: TRADING OPTIONS, STRATEGY SELECTION, AND RISK MANAGEMENT FRAMEWORK WITH MULTI-AGENT AND DEEP REINFORCEMENT LEARNING

Presented

by

Ayushman Gupta, M.TECH, Computer Science

DISSERTATION

Presented to the Swiss School of Business and Management Geneva
In Partial Fulfilment
of the Requirements

for the Degree

DOCTOR OF BUSINESS ADMINISTRATION

SWISS SCHOOL OF BUSINESS AND MANAGEMENT 2025

DEEP OPTIONS: TRADING OPTIONS, STRATEGY SELECTION, AND RISK MANAGEMENT FRAMEWORK WITH MULTI-AGENT AND DEEP REINFORCEMENT LEARNING

by

Ayushman Gupta, M.TECH, Computer Science

APPROVED BY

Prof.dr.sc. Saša Petar, Ph.D., Chair

RECEIVED/APPROVED BY:

SSBM Representative

Acknowledgements

This thesis on the Deep Options is the culmination of a challenging yet immensely rewarding journey, made possible through the support and encouragement of many individuals and institutions.

I would like to extend my sincere gratitude to 100X SPACE Investments Technologies PVT LTD for providing an inspiring and dynamic environment that not only facilitated my professional growth but also enriched my research with practical insights from the world of option trading. The opportunity to work within such a forward-thinking organization allowed me to integrate real-world experiences with academic inquiry, resulting in a robust and application-oriented study.

My deepest appreciation goes to my supervisor, Dr. Anuja Shukla, whose expertise, insightful guidance, and unwavering support were pivotal throughout this research. Her keen observations and constructive feedback not only refined the scope of this work but also encouraged me to approach challenges with a critical and innovative mindset.

I am also grateful to my colleagues, friends, and family for their constant encouragement and understanding, especially during the demanding phases of this research. Their support has been a steady source of motivation and has played a crucial role in the successful completion of this thesis.

Finally, I thank all those who have contributed to this journey, directly or indirectly, for helping me achieve this important milestone in my academic and professional career.

Abbreviations

- **RL:** Reinforcement Learning
- **MDP:** Markov Decision Process
- **POMDP:** Partially Observable Markov Decision Process
- **TD:** Temporal Difference
- **Q-learning:** Q-learning (a specific TD control algorithm)
- **SARSA:** State-Action-Reward-State-Action (another specific TD control algorithm)
- **DQN:** Deep Q-Network
- **DDQN:** Double Deep Q-Network
- **PG:** Policy Gradient
- **REINFORCE:** REINFORCE (a specific policy gradient algorithm)
- **A2C:** Advantage Actor-Critic
- **A3C:** Asynchronous Advantage Actor-Critic
- **PPO:** Proximal Policy Optimization
- **TRPO:** Trust Region Policy Optimization
- **SAC:** Soft Actor-Critic
- **DDPG:** Deep Deterministic Policy Gradient
- **TD3:** Twin Delayed Deep Deterministic Policy Gradient
- **RNN:** Recurrent Neural Network
- **CNN:** Convolutional Neural Network
- **LSTM:** Long Short-Term Memory
- **GRU:** Gated Recurrent Unit
- **ER:** Experience Replay
- **PER:** Prioritized Experience Replay
- **IRL:** Inverse Reinforcement Learning
- **MBRL:** Model-Based Reinforcement Learning
- **MFRL:** Model-Free Reinforcement Learning.
- **POMDP:** Partially Observable Markov Decision Process.
- **HRL:** Hierarchical Reinforcement Learning.
- **HFT:** High-Frequency Trading. This is the most prevalent abbreviation.

- **Algo Trading:** Algorithmic Trading (a broader term that encompasses HFT).
- **DMA:** Direct Market Access.
- **Latency:** The delay in data transfer or execution.

Abstract

Achieving consistent profitability and outperforming the underlying index are long-standing challenges in financial markets. While most AI-driven research focuses on stock trading and portfolio optimization, options trading—comprising nearly 90% of total exchange-traded volume—has remained relatively underexplored. Options trading strategies involve complex, sequential decision-making steps, from gauging market direction, volatility, and momentum to selecting strikes, managing risk, sizing positions, and determining entry/exit timing. Recent advances in Agentic AI and Deep Reinforcement Learning (DRL) have shown significant potential for tackling such high-dimensional, dynamic problems.

In this thesis, we propose an autonomous framework for options trading built on Agentic AI, comparing two distinct approaches. The first is a multi-agent collaborative system that orchestrates five specialized agents: a Generative Adversarial Network (GAN) for strategy generation, a dedicated strategy selection module, a Transformer-based market regime prediction agent, a risk management agent, and a data acquisition and technical analysis agent. The second approach leverages a DRL-driven pipeline to dynamically learn and execute options strategies. Both methods are benchmarked against 15 different option strategies across various market conditions. Experimental results demonstrate that our proposed framework consistently delivers robust performance and significantly outperforms the underlying index. This research closes a critical gap in AI-based decision-making for options trading and provides a scalable, adaptable, and empirically validated solution for real-world market environments.

Keywords: Deep Reinforcement Learning, Multi-agent System, Option Trading, Options Strategy, Automated Trading

Table of Contents

ABBREVIATIONS 4

ABSTRACT 6

Keywords: Deep Reinforcement Learning, Multi-agent System, Option Trading, Options Strategy, Automated Trading 6

LIST OF TABLES 12

LIST OF FIGURES 12

1 NTRODUCTION 14

1.1 Introduction 14

1.2 Challenges in Applying Artificial Intelligence techniques to Financial Markets . 10

1.3 Comparison of Autonomous AI driven; Rule based Vs Human Traders 12

1.4 Growth of Options Market in Indian 21

1.5 Significance of Study 24

1.6 Research Purpose and Questions 24

1.6.1 Research Purpose **26**

1.6.2 Research Questions **27**

1.7 Scope of the Study 27

1.8 Background and Motivation 29

1.8.1 Options **29**

1.8.2 Types of Equity Options **30**

1.8.3 Parties to an Option Contract **30**

1.8.4 American Options and European Options **30**

1.8.5 Characteristics of a Stock Option Contract **31**

1.8.5 Option pricing **31**

1.8.6 Option Strategies **34**

1.8.7 Agents **60**

1.8.8 Multi-Agent System **62**

1.8.10 Agentic AI Frameworks **76**

1.8.11 Generative Adversarial Networks **76**

1.8.12 Temporal Fusion Transformer **79**

2. LITERATURE REVIEW: 82

2.1 Introduction 82

2.1.1 Reinforcement Learning in Financial Trading **83**

2.1.2 Reinforcement Learning Agents with Deep Neural Networks in Options Trading **85**

2.1.3 Options Trading with Reinforcement Learning **86**

2.1.4 Options Trading **87**

2.1.5 Binomial Options Pricing Model **88**

2.1.6 Monte Carlo Simulation in Financial Engineering **88**

2.1.7 Multi-Agent Systems **90**

2.1.8 Markov Decision Processes **92**

Policy Gradient Methods **92**

2.1.9 Temporal Difference (TD) Learning **93**

2.1.10 Generative Adversarial Networks **93**

2.1.11 Temporal Fusion Transformer **94**

2.1.12 Policy Optimization Methods in Reinforcement Learning **95**

2.1.13 Regime Changes in Stock Trading **96**

2.1.14 Q-Learning Algorithms in Reinforcement Learning for Options Trading **97**

2.1.15 Backtesting in Reinforcement Learning-Based Options Trading **97**

2.1.16 ARIMA Models in Time Series Analysis **98**

2.1.17 Agentic AI **99**

2.1.18 Generic Agentic AI **101**

2.1.19 Stock Trading **101**

2.2 Conclusion of Literature Review 102

2.3 Research Gap 103

3. METHODOLOGY 105

3.1 Introduction 105

3.2 Overview of the Research Problem: 106

3.3 Traditional Challenges in Options Trading 106

3.4 The Role of Multi-Agent Systems in Overcoming These Challenges 107

3.5 The Role of Reinforcement Learning Systems in Overcoming These Challenges 107

3.6 Comparing the Multi Agent Framework with Reinforcement Learning.....108
Conclusion.....109
3.7 Multi-Agent System Development:.....109
3.7.1 Key Constructs 111
3.7.2 Observable Indicators.....111
3.7.3 Develop Measurement Tools 112
3.7.4 Establish Relationships Between Constructs 114
3.7.5 Ensure Validity and Reliability.....114
3.8 Deep Reinforcement Learning (DRL) for Options Trading 115
3.8.1 Proposed Solution Architecture of DRL Option Trading Framework
Figure 21: Architecture of DRL Option Trading Framework.....116
3.8.2 Problem Formulation as a Markov Decision Process (MDP) 116
3.8.3 DRL Algorithms Exploration.....117
3.8.4 Pseudocode for the Autonomous Option Trading DRL Framework 121
3.8.5 Data Preprocessing and Feature Engineering 122
3.8.6 Environment and Simulation 122
3.8.7 Training the DRL Agent.....122
3.8.8 Reward Function Design 123
3.8.9 System Architecture.....124
3.9 Research Purpose and Questions.....126
3.9.1 Research Questions 126
3.10 Research Design 127
3.10.1 Overall Framework Overview 127
3.10.2 Experimental Setup.....128
3.10.3 Alignment with Research Questions.....134
3.11 Research Design Limitations.....134
3.12 Experimental Setup.....135
3.12.1 Multi Agent Interaction 135
3.12.2 Reinforcement Learning.....145

4. RESULTS.....154

4.1 Research Question 1..... 154

4.2 Research Question 2 ..... **158**
4.3 Research question 3 ..... **161**

5 ..... **DISCUSSION** 165

5.1 Research Question 1 166
5.1.1 Strategy-by-Strategy Highlights **166**
5.1.2 Possible Explanations for Performance Differences **167**
5.2 Research Question 2 168
5.2.1 Strategy-by-Strategy Highlights **168**
5.3 Research Question 3 169
5.3.1 Strategy-by-Strategy Highlights **170**
5.3.2 Reflections and Future Directions **172**

6 ..... **SUMMARY, IMPLICATIONS, AND RECOMMENDATIONS** 173

6.1 Summary 173
6.2 Key Contributions and Implications 177
6.3 Recommendations For Future Research 178
6.4 Conclusion 179

APPENDIX A 181

Strategy wise Output of the Framework **181**
Strategy: Bear Call Spread Testing Year: 2024 ..... **181**
Strategy: Bear Condor Testing Year: 2021 ..... **185**
Strategy: Risk Reversal Testing Year: 2021 ..... **189**
Strategy: Jade Lizard Testing Year: 2021 ..... **193**
Strategy: Batman Testing Year: 2022 ..... **199**
Strategy: Range Forward Testing Year: 2021 ..... **202**
Strategy: Long Iron Butterfly Testing Year: 2021 ..... **206**
Strategy: Call Short Testing Year: 2022 ..... **211**
Strategy: Long Iron Condor Testing Year: 2021 ..... **214**
Strategy: Bull Call Spread Testing Year: 2021 ..... **218**
Strategy: Short Iron Condor Testing Year: 2021 ..... **222**
Strategy: Reverse Jade Lizard Testing Year: 2021 ..... **226**

Strategy: Short Strangle Testing Year: 2021.....230
Strategy: Short Straddle Testing Year: 2021.....234
Strategy: Put Short Testing Year: 2021.....238

List of Tables

Table 1 Index Options Trading Volume 25
Table 2 Result: Traditional Approach vs Multi-Agent Autonomous Framework Performance.158
Table 3 Results: Traditional Approach vs Deep Reinforcement Learning System Performance162
Table 4 Result of Comparative Performance of Multi-Agent Framework Vs DRL.....167

List of Figures

Figure 1 Rise of India's Options Market (in Bn$). 22
Figure2 Short Iron Condor… 35
Figure 3 Long Iron Condor… 37
Figure 4 Long Iron Butterfly…..... 38
Figure 5 Short Iron Butterfly…..... 40
Figure 6 Call Short 42
Figure 7 Put Short 43
Figure 8 Jade Lizard..... 44
Figure 9 Reverse Jade Lizard..... 46
Figure 10 Short Strangle 47
Figure 11 Short Straddle 49
Figure 12 Bull Call Spread..... 50
Figure 13 Range Forward..... 51
Figure 14 Risk Reversal 54
Figure 15 Batman..... 56
Figure 16 Bear Call Spread 58
Figure 17 Agentic AI Framework 61
Figure 18 Agentic Functionalities 62
Figure 19 Generative Adversarial Networks..... 77
Figure 20 Transformer Architecture..... 80
Figure 21 Multi Agent System Flow Diagram..... 110
Figure 22 Architecture of DRL Option Trading Framework 116
Figure 23 DQN Training Loop…..... 118
Figure 24 Double DQN 119
Figure 25 Proximal Policy Optimization..... 119
Figure 26 Asynchronous Advantage Actor-Critic 120
Figure 27 Soft Actor-Critic 120
Figure 28 DRL Agent Framework… 124
Figure 29 Multi Agent Interaction..... 136
Figure 30 Technical Analysis Agent Workflow..... 136
Figure 31 Temporal Fusion Transformer for Trend Analysis..... 138
Figure 32 Trend Analysis Agent Workflow..... 128
Figure 33 Generative Adversarial Network for Strategy Generation..... 140
Figure 34 Strategy Generator Agent Workflow 140
Figure 35 Strategy Selection Agent Workflow 142

Figure 36 Risk Management Agent Workflow 143

CHAPTER I:

1. Introduction

1.1 Introduction

In financial literature, option trading strategies have been an interesting topic for research (Black and Scholes, 1973; Merton, 1973; Moody and Saffell, 2001).
Several researchers have proved abnormal returns can be earned by implementing such option strategies (Tan, Quek, and Cheng, 2011).
In Trading option strategies are defined as combination of buying and selling call and put at different sticks of a securities or index (Black and Scholes, 1973; Cox, Ross, and Rubinstein, 1979). The basic objective of trading strategies/rules is to maximize the returns with the given level of risk.
Researchers and investors all over the world are very much interested in finding that if abnormal returns could be earned by implementing these trading strategies (Yang et al., 2022).

This research sits at the intersection of finance and artificial intelligence, aiming to enhance trading strategies and risk management through advanced machine learning techniques (Wen, 2021; Sutton and Barto, 1998; Schulman et al., 2017).

Options have become increasingly important in the world of finance in terms of their functions and volumes traded (Black and Scholes, 1973; Merton, 1973). Options are distinguished from other equity derivatives in the sense that options confer a right upon the buyer, but not an obligation to exercise a “call” or “put” contract. This right is not without a cost and requires the payment of a “premium”. In contrast, the seller earns this premium and is obligated to deliver or purchase the underlying asset on contract expiry.

Options trading is a sophisticated financial practice that allows investors to speculate on the future price movements of the underlying assets or to hedge existing positions (Cox, Ross, and Rubinstein, 1979).

1.2 Challenges in Applying Artificial Intelligence Techniques to Financial Markets

Implementing AI in financial markets presents several challenges (Busoniu et al., 2008; Sutton and Barto, 1998):

Machine learning for options trading faces numerous challenges stemming from the dynamic nature of financial markets and the inherent complexity of derivatives (Yang et al., 2022).

First, data is non-stationary and evolving, with markets transitioning unpredictably between bull, bear, sideways, high-volatility, and low-volatility regimes, and concept drift arising from new regulations, macroeconomic events, or technological shifts. Second, data quality and availability present obstacles: financial time-series data is often noisy due to microstructure effects, bid-ask spreads, and short-term volatility; survivorship bias and look-ahead bias can skew historical

datasets; and complete historical options data, particularly at the intraday level, can be difficult or expensive to obtain.

Third, the complexity of options and derivatives adds another layer of difficulty, given the multidimensional nature of contracts (strike prices, maturities, implied volatility, and Greeks) and the broader action space involved in multi-leg strategies such as spreads, straddles, and strangles (Cox, Ross, and Rubinstein, 1979). Fourth, the high-dimensional action and state spaces—encompassing technical indicators, macroeconomic variables, order book data, and the vast array of potential option trades—require sophisticated feature engineering and efficient handling of combinatorial explosions in strategy selection.

Fifth, the risks of overfitting and poor generalization are high, especially when significant market events like crashes or volatility spikes are rare in historical data, and when deep learning models with extensive hyperparameters undergo extensive tuning (Bradtke and Barto, 1996). Sixth, risk management must be deeply integrated, ensuring that position sizing, drawdown control, and regulatory or capital constraints are reflected in model decisions (Peng et al., 2024; Wu and Jaimungal, 2023).

Seventh, computational constraints are non-trivial, as large-scale or high-frequency data demands considerable processing power, and models may need to update or adapt online in near real-time (Haarnoja et al., 2018).Eighth, interpretability and explainability become critical because black-box models can be difficult to trust or justify to regulators and stakeholders, who need clarity on how decisions are made (Castelfranchi, 1998).

Ninth, market impact and liquidity considerations mean that even accurate predictions may fail to translate into profits if large trades move prices adversely, while slippage and execution costs can further erode returns.

Finally, ethical and regulatory concerns loom large, with algorithmic bias, market manipulation rules, and strict compliance standards requiring that AI-driven strategies adhere to legal and ethical guidelines in heavily regulated financial markets (Ali et al., 2020).

1.3 Comparison of Autonomous AI-Driven; Rule-Based Vs Human Traders

A. Speed and Reaction Time

Human Traders:

Rely on manual execution. Even skilled traders can only process a limited amount of information in real time. Reaction times can range from seconds to minutes, making it difficult to capitalize on rapid market movements.

Algo (Rule-Based) Trading:

Executes pre-programmed rules instantly, often in milliseconds or microseconds. Can scan multiple markets or instruments simultaneously without fatigue, making it highly effective for high-frequency or event-driven trading (Tan, W.L., Roberts, and Zohren, 2024).

AI-Driven Trading:

Combines automated execution with machine learning-based decision-making. Like traditional algorithmic trading, AI systems react extremely quickly, but they also adapt their rules and strategies based on data-driven insights rather than relying on static, hand-coded logic (Gupta, Abbeel, and Levine, 2018).

B. Adaptability and Learning

Human Traders:

Rely on experience, intuition, and continuous learning through trial and error. Their ability to adapt to new market regimes depends on personal skill, discipline, and emotional control.

Algo (Rule-Based) Trading:

Follows fixed rules or logic. While parameter tuning is possible, the core strategy typically remains unchanged unless a human updates it. The system does not "learn" unless reprogrammed or recalibrated.

AI-Driven Trading:

Uses machine learning models (e.g., Deep Learning, Reinforcement Learning) that can automatically adapt and learn from new data. This adaptability allows AI-based strategies to identify patterns or market shifts without explicit human intervention, though it can also lead to model overfitting if not carefully managed (Sutton, 1988).

C. Emotional and Cognitive Bias

Human Traders:

Susceptible to psychological biases like fear, greed, loss aversion, and overconfidence.

Emotions can drive suboptimal decisions, such as holding onto losing trades or chasing momentum too late (Bryzgalova and Pavlova, 2022).

Algo (Rule-Based) Trading:

Follows a predetermined set of rules and is not influenced by emotions. However, the system can still be biased if the rules themselves are based on flawed assumptions or incomplete data.

AI-Driven Trading:

Eliminates human emotional biases at the execution level. However, biases can creep into AI models through data selection, model design, or historical market anomalies. The "bias" in this case is more about data-driven distortions rather than human psychology (Castelfranchi, 1998).

D. Complexity of Strategy

Human Traders:

Can develop sophisticated discretionary strategies, combining fundamental, technical, and macroeconomic insights. However, executing extremely complex strategies in real time is challenging, especially under stress.

Algo (Rule-Based) Trading:

Can implement complex multi-factor strategies (e.g., pairs trading, statistical arbitrage) with relative ease, provided the rules are well-defined. However, it may struggle with market scenarios not explicitly covered by the rules (Brim, 2019).

AI-Driven Trading:

Capable of uncovering highly complex, non-linear relationships in the data that humans or rule-based systems might miss. Methods like deep neural networks or reinforcement learning can optimize large action spaces (e.g., option strikes, maturities) and adapt to changing conditions (Taghian, 2023).

E. Data Processing and Analysis

Human Traders:

Typically rely on chart analysis, news, and fundamental data. Even with tools, the volume of data a single individual can process is limited.

Algo (Rule-Based) Trading:

Efficiently processes large data sets for signals, such as technical indicators or order book depth. The scope is confined by the programmed logic (e.g., specific indicators or conditions).

AI-Driven Trading:

Excels at ingesting massive amounts of structured (price, volume, technical indicators) and unstructured data (news, social media sentiment) using nlp or advanced time-series modelling. AI can automatically discover hidden features and correlations, provided there is sufficient computational power and data quality (Liu et al., 2023).

F. Risk Management and Execution

Human Traders:

Implement risk controls manually—setting stop losses, position limits, etc. They may adjust these on the fly, but emotional and cognitive biases can interfere.

Algo (Rule-Based) Trading:

Strictly enforces predefined risk rules (e.g., position sizing, stop-loss triggers). However, it lacks adaptive risk management unless explicitly coded for different market regimes.

AI-Driven Trading:

Integrates adaptive risk management strategies, potentially learning to reduce exposure under higher volatility or unfavourable conditions. Reinforcement Learning, for example, can learn policies that maximize reward while penalizing excessive drawdowns (Peng et al., 2024; Wu and Jaimungal, 2023).

G. Transparency and Explainability

Human Traders:

Decisions can be explained through subjective reasoning (e.g., "the market felt overextended"). However, consistency and reproducibility can vary greatly.

Algo (Rule-Based) Trading:

Rules are explicitly defined. Traders and compliance officers can audit logic step-by-step. Easy to explain because the code or logic directly translates to actions.

AI-Driven Trading:

Often treated as a "black box," especially in deep learning models. Interpreting why a neural network made a certain trade can be difficult. This can pose challenges for compliance, regulatory oversight, and internal risk committees (Castelfranchi, 1998).

H. Scalability and Resource Requirements

Human Traders:

Limited by personal capacity and time. A single trader can only manage a finite number of instruments or strategies effectively.

Algo (Rule-Based) Trading:

Highly scalable—once developed, the same code can be deployed across multiple markets or instruments, provided the data feed and infrastructure can handle the volume.

AI-Driven Trading:

Also highly scalable but with potentially higher computational and data requirements. Training advanced models (e.g., DNN) on high-frequency data can be resource-intensive. Deploying such models in real-time trading systems may require specialized hardware (GPUs, TPUs, etc.) and robust data pipelines (Haarnoja et al., 2018; Gupta, Abbeel, and Levine, 2018).

I. Regulatory and Ethical Considerations

Human Traders:

Must follow regulations, but discretionary decisions are usually easier to audit. Compliance violations often hinge on personal misconduct.

Algo (Rule-Based) Trading:

Regulations often require clear documentation of trading logic and fail-safe mechanisms. Algorithms can inadvertently create "flash crashes" or abnormal market movements if poorly tested.

AI-Driven Trading:

Adds a layer of complexity for regulators, as the system "learns" over time and may evolve beyond its original programming. Auditing becomes more complex, raising questions about responsibility if the model's decisions lead to market anomalies (Ali et al., 2020; Clatterbuck et al., 2024).

J. Potential for Innovation

Human Traders:

Creativity and intuition can drive unique strategies (e.g., new approaches to fundamental analysis, contrarian plays). However, it's hard for a single individual to keep pace with markets that operate nearly 24/7.

Algo (Rule-Based) Trading:

Offers systematic, repeatable methods that can outperform humans in speed and consistency. Innovation is often limited to novel rule sets, factor combinations, or execution algorithms.

AI-Driven Trading:

Opens up new frontiers in pattern discovery and strategic adaptation. With techniques like RL, AI can discover strategies or micro-structure signals that neither human intuition nor straightforward algorithms might find (Acharya et al., 2025; Shavandi, 2023).

Conclusion

Each of the three paradigms—human, rule-based algorithmic, and AI-driven trading—has distinct advantages and limitations. Humans excel at intuitive leaps and creativity but are prone to emotional bias and limited by cognitive capacity. Traditional algorithmic trading systems can execute strategies rapidly and consistently, yet they lack adaptability unless reprogrammed. AI-

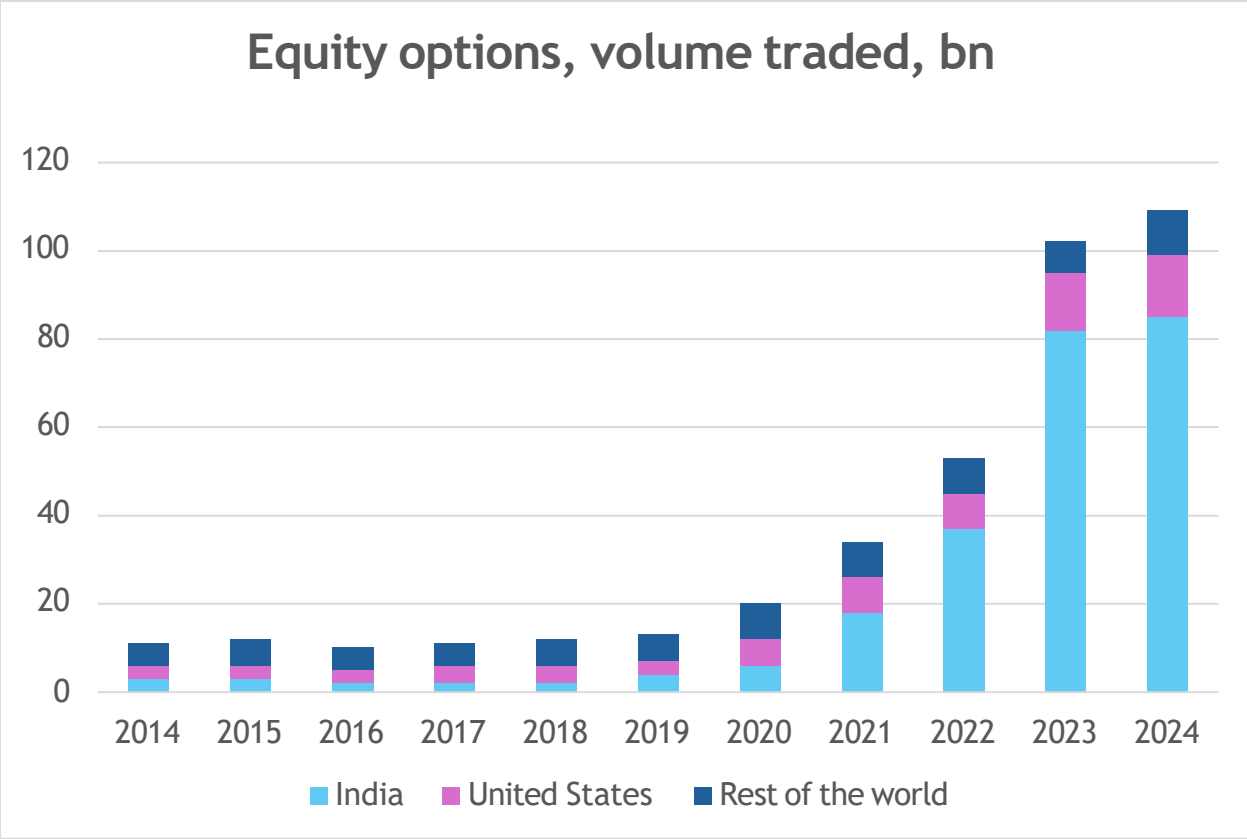
driven trading combines the speed and consistency of automated execution with the adaptability of machine learning, potentially discovering complex patterns and responding dynamically to evolving market conditions. However, AI systems can be difficult to interpret and require careful data management, robust risk controls, and continuous monitoring to avoid pitfalls like overfitting and unexpected market impacts (Guo et al., 2022; Zhang et al., 2022).

1.4 Growth of Options Market in Indian

In 2004, a turning point emerged in the global equity options arena as both retail and institutional investors began to drive unprecedented trading volumes, spurred by the advent of electronic and algorithmic trading (Bryzgalova and Pavlova, 2022; Tan, W.L., Roberts, and Zohren, 2024). While the Americas and parts of Europe traditionally dominated the trading landscape, emerging markets in China, Japan, and India were rapidly gaining ground. That year also marked a significant milestone for the National Stock Exchange (NSE) of India: it broke into the top 10 futures exchanges globally, although initially ranking 17th in terms of overall F&O volumes (Bryzgalova and Pavlova, 2022). Over time, the NSE's trading activity in futures and options has grown remarkably.

A landmark development occurred in 2016 with the introduction of weekly expiries in options trading (Bryzgalova and Pavlova, 2022). This innovation quickly resonated with market participants, particularly retail investors and proprietary traders, who were attracted by the lower premiums required to enter these contracts (Tan, W.L., Roberts, and Zohren, 2024). In that year alone, Bank Nifty options reached a staggering volume of approximately 3 billion contracts, while Nifty Index options saw over 1.6 billion contracts traded (Bryzgalova and Pavlova, 2022). Despite this high volume, the year-end open interest for Bank Nifty options was recorded at 704,000 contracts, indicating that many positions were held only briefly a trend that was similarly observed in Nifty Index options (Bryzgalova and Pavlova, 2022).

Figure 1: Rise of India’s Options Market (in Bn$)



Weekend Investing. (2024)

In recent years, the NSE has consistently ranked among the top three exchanges in the world by annual F&O volume (Bryzgalova and Pavlova, 2022). The momentum was particularly evident in 2018 and 2019 when India's index options market led the globe in trading volume (Bryzgalova and Pavlova, 2022). In 2018, the index options segment experienced an annual growth rate of 63%, one of the highest among the world's top exchanges (Bryzgalova and Pavlova, 2022). Meanwhile, stock options recorded a growth rate of 46%, second only to Brazil's impressive 75% growth rate (Bryzgalova and Pavlova, 2022).

The year 2020 further underscored the NSE's dominance, as it reported a total of 8.85 billion contracts traded in the F&O segment. Among these, Nifty Index options accounted for 2.37 billion contracts, while Bank Nifty options remained the most actively traded with 4.29 billion contracts. This growth is not just a story of increasing volumes—it is also a narrative of shifting participant dynamics. Since fiscal year 2016, proprietary traders and individual investors have

steadily become the primary drivers of the equity derivatives market, jointly representing around 60% of trading turnover. By 2020, their combined share had surged to over 70%, with individual investors alone contributing 32.7% to the F&O market.

For index options specifically, the combined participation of proprietary traders and retail investors has hovered around 70% until October 2020. Notably, while individual investor participation has seen a continuous rise since 2016, the share of proprietary traders has gradually declined from 53% in 2016 to 38% by December 2020, with Foreign Institutional Investors (FIIs) emerging as the dominant institutional group in FY 2020.

The surge in retail investor involvement can largely be attributed to the rapid adoption of internet-based trading, which gained remarkable momentum following the nationwide lockdown in March 2020. This digital transformation has bolstered all segments of trading, as evidenced by a 35% year-on-year increase in the average premium turnover for equity derivatives. Furthermore, the index options premium turnover reached an all-time high of Rs 171 billion on December 22, 2020, while the NSE's monthly turnover in the equity derivatives segment grew by 29.4% year-on-year by the end of 2020.

Together, these developments underscore a dynamic evolution in India's options market, reflecting both technological advancements and shifting market participation that have propelled the NSE to the forefront of global trading platforms.

The National Stock Exchange (NSE) in Mumbai is India's largest stock exchange. Established in 1992 as the country's first demutualised electronic exchange, the NSE pioneered an automated trading platform that offers a stable and secure trading environment for investors and traders nationwide. By April 2018, the NSE had become the 11th largest stock exchange globally, with an aggregated market capitalization exceeding US$2.27 trillion. Its flagship index, the NIFTY 50 which comprises 50 stocks serves as a benchmark for the Indian capital markets and is widely referenced by both domestic and international investors. In addition, the CNX Nifty Index was introduced in 1996, further highlighting the exchange's innovative approach. According to the World Federation of Exchanges (WFE) 2018 derivatives report, the NSE ranked first among the

top 10 stock exchanges for stock index options trading volume, recording 2,214,848,247 deals, with the CNX Nifty Index accounting for 622,118,790 of those trades.

Table 1: Index Options Trading Volume

Year	Index Futures		Index Options		
	No of Contracts (₹. cr.)	Turnover (₹. cr.)	No of contracts (₹. cr.)	Notional Turnover (₹. cr.)	Premium Turnover (₹. cr.)
2018-19	69824522	5568914.47	2652457487	203302404.9	654099.95
2017-18	57674584	4810454.34	1515034222	134921876.5	460653.71
2016-17	66535070	4335940.78	1067244916	72797287.69	350021.53
2015-16	140538674	4557113.64	1623528486	48951930.6	351221.01
2014-15	129303044	4107215.2	1378642863	39922663.48	265315.63
2013-14	105252983	3083103.23	928565175	27767341.25	244090.71
2012-13	96100385	2527130.76	820877149	22781574.14	184383.24
2011-12	146188740	3577998.41	864017736	22720031.64	253068.22
2010-11	165023653	4356754.53	650638557	18365365.76	192637.87
2009-10	178306889	3934388.67	341379523	8027964.2	124416.58
2008-09	210428103	3570111.4	212088444	3731501.84	91715.58
2007-08	156598579	3820667.27	55366038	1362110.88	29286.09
2006-07	81487424	2539574	25157438	791906	17650.87
2005-06	58537886	1513755	12935116	338469	5770.52
2004-05	21635449	772147	3293558	121943	2356.98
2003-04	17191668	554446	1732414	52816	991.48
2002-03	2126763	43952	442241	9246	112.7
2001-02	1025588	21483	175900	3765	1299
2000-01	90580	2365	-	-	-

Sources: Data retrieved from NSE website.

1.5 Significance of Study

This study holds considerable significance in both academic research and practical financial applications (Wen, 2021; Yang et al., 2022). By integrating advanced AI methodologies into options trading, the research aims to bridge the gap between traditional human-driven trading and the emerging realm of autonomous decision-making (Sutton and Barto, 1998;

Schulman et al., 2017). The following points outline the key contributions and impacts of the study:

The proposed approaches comparing a multi-agent collaborative system with a deep reinforcement learning (DRL) pipeline push the boundaries of current algorithmic trading frameworks (Busoniu et al., 2008; Haarnoja et al., 2018). By examining these distinct methodologies, the study contributes to a deeper understanding of how AI can be leveraged to make **real-time, high stakes trading decisions** (Gupta, Abbeel, and Levine, 2018).
With the potential to execute call and put options more dynamically, the reinforcement learning model may not only match but potentially surpass human trading performance (Tan, W.L., Roberts, and Zohren, 2024; Jin, 2022). Moreover, the study explores how these models can adapt to **varying market regimes**, offering insights into designing systems that remain robust under fluctuating market conditions (Yang et al., 2022; Huang et al., 2023).

By focusing on aspects such as **position sizing, entry/exit timing**, and the integration of protective closing strategies, the research emphasizes the importance of risk management in options trading (Peng et al., 2024; Wu and Jaimungal, 2023). The findings could lead to the development of more resilient trading strategies that mitigate losses while capitalizing on market opportunities (Tan, W.L., Roberts, and Zohren, 2024).

The multi-agent framework introduces a novel collaborative approach where specialized agents handle strategy generation, market regime prediction, technical analysis, and risk management (Wooldridge and Jennings, 1995; Shoham and Leyton-Brown, 2008). This decentralized decision-making process may enhance the speed and accuracy of strategy selection, particularly in high-frequency trading environments, offering a fresh perspective compared to traditional **single-agent models** (Busoniu et al., 2008).

The insights from this research can be directly applicable to financial institutions, hedge funds, and retail traders (Tan, W.L., Roberts, and Zohren, 2024). The potential to outperform market benchmarks and enhance trading performance positions this study as a valuable resource for those looking to harness AI in financial decision-making (Guo et al., 2022).
Overall, the significance of this study lies in its potential to revolutionize options trading by demonstrating **how autonomous AI frameworks** can improve efficiency, decision-making accuracy, and adaptability in complex and dynamic financial markets (Acharya et al., 2025; Guo et al., 2022). In essence, this research aims to push the boundaries of autonomous options trading

by providing empirical evidence and practical insights into the application of advanced AI techniques (Ferdowsi and Saad, 2020).

The findings of this study will be valuable to researchers, practitioners, and financial institutions seeking to leverage AI for improved trading outcomes and risk management (Yang et al., 2022; Peng et al., 2024).

1.6 Research Purpose and Questions

1.6.1 Research Purpose

The overarching purpose of this thesis is to explore and evaluate the efficacy of autonomous agentic AI frameworks for options trading, specifically focusing on two distinct approaches: a multi-agent collaborative system and a Deep Reinforcement Learning (DRL)-driven pipeline (Sutton and Barto, 1998; Schulman et al., 2017). This investigation aims to determine the potential of these advanced AI methodologies to navigate the complexities of options markets and achieve superior trading performance (Wen, 2021; Yang et al., 2022).

This thesis is designed to:

- **Develop a Multi-Agent Framework:** Construct a system comprising five specialized agents:
    - A Generative Adversarial Network (GAN) for strategy generation.
    - A dedicated strategy selection module.
    - A Transformer-based agent for predicting market regimes.
    - A risk management agent.
    - A data acquisition and technical analysis agent (Wooldridge and Jennings, 1995; Shoham and Leyton-Brown, 2008).
- **Implement a DRL-Driven System:** Create a system that uses deep reinforcement learning to dynamically learn and execute options trading strategies (Schulman et al., 2017; Haarnoja et al., 2018).
- **Compare Methodologies:** Assess the effectiveness of both the multi-agent system and the DRL approach in executing options trades, particularly in relation to human trading benchmarks and overall market performance.

- **Evaluate Adaptability:** Investigate how each approach recognizes and adapts to varying market regimes, ensuring robust performance across different market conditions.

1.6.2 Research Questions

1.6.2.1 Effectiveness of Multi-Agent Coordination:

Question 1: Can the coordination among specialized agents combined with decentralized decision-making within a multi-agent system enhance both the selection and execution of options trading strategies compare to traditional approach?

1.6.2.2 Effectiveness of Reinforcement Learning in Options Trading:

Question 2: Can Deep Reinforcement Learning models be developed to autonomously execute different option strategies in real time—aligning with human trading timeframes—and can these models outperform the underlying market index?

1.6.2.3 Comparative Analysis of Reinforcement Learning and Multi-Agent Systems:

Question 3: Can the adaptive, decentralized framework of multi-agent systems lead to superior trading performance compared to Deep Reinforcement learning based system under dynamic market conditions?

By addressing these questions, this thesis seeks to contribute to the understanding of how advanced AI techniques can be applied to develop sophisticated and effective autonomous options trading systems. We will evaluate the performance of our proposed frameworks through rigorous empirical testing, providing insights into the strengths and limitations of each approach.

1.7 Scope of the Study

This study focuses on exploring and evaluating autonomous options trading strategies within the Indian stock market (Bryzgalova and Pavlova, 2022; Cao, 2019). Specifically targeting the Indian stock market, the research concentrates on options trading instruments available on major exchanges (Bryzgalova and Pavlova, 2022). This localized focus ensures that the findings are highly relevant to the unique dynamics and regulatory environment of India’s financial markets

(Bryzgalova and Pavlova, 2022; Cao, 2019). Only call and put options are examined, excluding other financial instruments such as equities or futures, thereby maintaining a clear focus on the derivatives market (Black and Scholes, 1973; Merton, 1973).

Two distinct autonomous frameworks form the core of the study. The first is a multi-agent collaborative system that includes specialized agents (GAN for strategy generation, a strategy selection module, a Transformer-based market regime predictor, a risk management agent, and a data acquisition and technical analysis agent). The second framework is a deep reinforcement learning (DRL) system designed to dynamically learn and execute trading strategies. These two approaches are compared based on their decision-making processes, risk management capabilities, and adaptability to varying market regimes.

Historical and real-time data from the Indian stock market will be utilized for simulations and back-testing (Bryzgalova and Pavlova, 2022). While a rich dataset is leveraged, the analysis is bounded by the availability and quality of historical options trading data specific to India (Cao, 2019). An important component of the research is the assessment of risk management practices and strategy optimization (Peng et al., 2024). This involves examining techniques such as position sizing, timing for trade entries and exits, and the use of protective closing strategies, all tailored to the conditions of the Indian market (Wu and Jaimungal, 2023).

Limitations:

- The scope is confined to options trading and does not generalize to other trading instruments or international markets (Black and Scholes, 1973).
- The research emphasizes simulation and back-testing; real-time trading executions and live market interventions are beyond the current study (Bryzgalova and Pavlova, 2022).
- Findings may be influenced by market-specific factors, such as local regulatory frameworks and market volatility unique to the Indian context.

In summary, this study is designed to provide an in-depth analysis of autonomous, AI-driven trading systems within the Indian options market, highlighting both the innovative potential and practical challenges of deploying such technologies in a dynamic trading environment (Wen, 2021; Yang et al., 2022).

1.8 Background and Motivation:

1.8.1 Options

Options trading involves buying and selling options contracts on financial instruments like stocks, commodities, or indices (Black and Scholes, 1973; Merton, 1973). Engaging in options trading allows investors to speculate on the future price movements of the underlying asset, hedge existing positions, or generate additional income through strategies like writing options (Black and Scholes, 1973). However, it's important to note that options trading carries significant risks and complexities, necessitating a thorough understanding before participation (Tan, Quek, and Cheng, 2011).

In options trading, an option is a financial contract that grants the holder the right, but not the obligation, to buy or sell a specific quantity of an underlying asset at a predetermined price, known as the strike price, on or before a specified expiration date (Black and Scholes, 1973). These contracts are typically linked to various financial instruments, such as stocks, commodities, or indices (Black and Scholes, 1973). Options are categorized into two primary types:

- **Call Option**: This type of option gives the holder the right to purchase the underlying asset at the strike price within the specified timeframe. Investors typically buy call options when they anticipate that the price of the underlying asset will rise (Black and Scholes, 1973).
- **Put Option**: This type of option grants the holder the right to sell the underlying asset at the strike price within the specified period. Investors generally purchase put options when they expect the price of the underlying asset to decline (Black and Scholes, 1973).

Options are considered **derivatives**, meaning their value is derived from the price of the underlying asset. They are utilized for various purposes, including hedging against potential price fluctuations, speculating on market movements, and enhancing portfolio returns through strategic income-generating techniques. However, it's important to note that options trading involves significant risks and complexities, necessitating a thorough understanding before engaging in such transactions.

1.8.2 Types of Equity Options

- **Equity Index Options**: These options have an equity index, such as the S&P 500, as the underlying asset They can be either European-style or American-style European-style options can only be exercised at expiration, while American-style options can be exercised at any time before expiration Like index futures contracts, index options are typically cash-settled, meaning that upon exercise, the difference between the strike price and the market value of the index is paid in cash, and no physical delivery of stocks occurs (Cox, Ross, and Rubinstein, 1979).
- **Equity Stock Options**: Stock options are options on individual stocks. Currently, options trade on over 500 stocks in the United States. A standard contract gives the holder the right to buy or sell 100 shares of the underlying stock at the specified strike price, within a set time frame. These options are generally American style, allowing exercise at any time before expiration. They can be used for various strategies, including hedging, speculation, and income generation (Cox, Ross, and Rubinstein, 1979).

1.8.3 Parties to an Option Contract

- **Buyer of an Option (Holder):** The buyer, or holder, acquires the right—without the obligation to exercise the option. By paying the option premium, the holder gains the right to buy (in the case of a call option) or sell (in the case of a put option) the underlying asset at the specified strike price, within a defined period (Black and Scholes, 1973; Merton, 1973). The maximum loss for the holder is limited to the premium paid for the option (Black and Scholes, 1973).
- **Writer of an Option (Seller):** The writer, or seller, receives the option premium and, in return, assumes the obligation to buy or sell the underlying asset if the holder exercises the option (Black and Scholes, 1973; Cox, Ross, and Rubinstein, 1979). For call options, the writer must sell the asset at the strike price; for put options, the writer must buy the asset at the strike price. The writer's potential loss can be substantial, especially if the market moves unfavourably, as losses are theoretically unlimited for call options and significant for put options (Black and Scholes, 1973).1.8.4 American Options and European Options

- **American Options**: American options are options that can be exercised at any time up to and including the expiration date. This flexibility allows the holder to exercise the option whenever it is advantageous before expiration. Most exchange-traded options are American-style (Cox, Ross, and Rubinstein, 1979).
- **European Options**: European options are options that can be exercised only on the expiration date itself. This means the holder must wait until the specified expiration date to exercise the option, regardless of favourable market conditions before that time. European options are often considered easier to analyse due to their simpler exercise structure, and properties of American options are frequently deduced from those of their European counterparts. Additionally, European-style options are typically traded over the counter (OTC) rather than on exchanges.

1.8.5 Characteristics of a Stock Option Contract

- **Option Price/Premium**: The option price, also known as the option premium, is the amount the options buyer pays to the option seller for acquiring the right to buy or sell the underlying assets. This premium is influenced by various factors, including the underlying asset's current price, the strike price, time until expiration, and market volatility. It's important to note that the option premium is non-refundable, regardless of whether the option is exercised (Black and Scholes, 1973; Cox, Ross, and Rubinstein, 1979).
- **Expiration Date**: The expiration date, sometimes referred to as the exercise date, strike date, or maturity date, is the last day on which the option can be exercised. After this date, the option becomes void, and the holder loses the right to exercise it. The specific expiration date is defined in the option contract and varies depending on the type of option and the exchange on which it's traded (Black and Scholes, 1973; Cox, Ross, and Rubinstein, 1979).
- **Strike Price**: The strike price, or exercise price, is the predetermined price at which the holder of the option can buy (for call options) or sell (for put options) the underlying asset. This price is established at the time the option contract is created and remains fixed throughout the life of the option. The relationship between the strike price and the

underlying asset's market price at expiration significantly impacts the option's profitability (Black and Scholes, 1973; Merton, 1973).

- **Intrinsic Value of an Option:** The option premium can be broken down into two components - intrinsic value and time value. The intrinsic value of a call is the amount by which the option is ITM, if it is ITM. If the call is OTM, its intrinsic value is zero. Putting it another way, the intrinsic value of a call is $Max\ [0, (S_t — K)]$ which means the intrinsic value of a call is the greater of 0 or $(S_t — K)$. Similarly, the intrinsic value of a put is $Max\ [0, K — S_t]$, i.e. the greater of 0 or $(K — S_t)$. Here, K is the strike price and S_t is the spot price (Black and Scholes, 1973; Tan, Quek, and Cheng, 2011).
- **Time Value of an Option:** The time value of an option represents the portion of the option's premium that exceeds its intrinsic value. It reflects the potential for the option to become more profitable before its expiration date. Both call and put options possess time value, which is influenced by several factors.

  Time to Expiration: The longer the time remaining until the option's expiration, the greater the time value. This is because a longer duration increases the likelihood of the option becoming profitable as market conditions fluctuate (Black and Scholes, 1973).

  Volatility of the Underlying Asset: Higher volatility in the underlying asset's price enhances the time value. Increased volatility amplifies the potential for the option to move into a favourable position before expiration (Black and Scholes, 1973; Cox, Ross, and Rubinstein, 1979).

  Intrinsic Value: The intrinsic value is the difference between the underlying asset's current price and the option's strike price, provided this difference is favourable to the option holder. The time value is calculated by subtracting the intrinsic value from the option's total premium.

1.8.5 Option pricing

Option pricing is the process of determining the fair value of an options contract, considering various factors that influence its potential profitability. Accurate pricing is essential for both buyers and sellers to make informed decisions in the options market.

1.8.5.1 Key Factors Affecting Option Pricing:

1. **Underlying Asset Price**: The current market price of the asset underlying the option significantly impacts its value. For call options, as the underlying asset's price increases, the option's value typically rises. Conversely, for put options, an increase in the underlying asset's price usually decreases the option's value [Black and Scholes (1973); Tan, W. L., Roberts, and Zohren (2024)].
2. **Strike Price**: This is the price at which the option holder can buy (for call options) or sell (for put options) the underlying asset. The relationship between the strike price and the underlying asset's current price determines the option's intrinsic value [Merton (1973)].
3. **Time to Expiration**: The duration remaining until the option's expiration date affects its time value. Longer timeframes provide more opportunities for the option to become profitable, thereby increasing its value [Cox, Ross, and Rubinstein (1979)].
4. **Volatility**: Higher volatility in the underlying asset's price leads to greater potential for profit or loss, influencing the option's value. Increased volatility generally raises the option's price due to the higher risk associated with larger price swings [Leisen and Reimer (2006)].
5. **Interest Rates**: Changes in prevailing interest rates can affect option pricing. For instance, higher interest rates might increase call option values and decrease put option values, as they influence the cost of carry and the present value of the option's strike price [Black and Scholes (1973)].
6. **Dividends**: Expected dividends can impact on option pricing. When a company pays a dividend, the underlying asset's price typically drops by the dividend amount, affecting the option's value [Merton (1973)].

1.8.5.2 Common Option Pricing Models:

1. **Black-Scholes Model**: This model provides a theoretical estimate of European-style option prices, considering factors such as the underlying asset's price, strike price, time to expiration, volatility, and risk-free interest rates. It's widely used for pricing options on stocks that do not pay dividends.
2. **Binomial Options Pricing Model**: This model uses a discrete-time framework to model the possible price movements of the underlying asset over time. It constructs a binomial

tree of possible future prices and calculates the option's value by working backward from expiration to the present.

3. **Monte Carlo Simulation**: A computational technique that uses random sampling to simulate a range of possible price paths for the underlying asset, assessing the option's value based on these simulations. It's particularly useful for pricing complex options with multiple variables [Boyle (1977); Glasserman (2004); Jäckel (2002)].

1.8.6 Option Strategies

Option Strategies Options strategies are sophisticated financial instruments employed by investors and traders to optimize portfolio performance, manage risk, and capitalize on market movements [Tan, W. L., Roberts, and Zohren (2024)].

These strategies involve the strategic combination of various options contract financial derivatives. The strategic selection between buying and selling options, and the specific combinations thereof, allows traders to align their positions with their market forecasts and risk appetite [Wen Wen (2021); Tan, W. L., Roberts, and Zohren (2024)]. It's imperative to thoroughly comprehend the inherent risks and rewards associated with each strategy. Engaging in options trading necessitates a solid understanding of the instruments and a well-considered approach to risk management.

Traders can strategically combine the buying and selling of call and put options to tailor their payoff structures in alignment with their market outlook and risk tolerance. This flexibility enables the construction of positions that can profit from various market movements while managing potential risks [Wen Wen (2021)].

1.8.6.1 Bullish Strategies:

1. **Buying a Call Option:** A trader anticipating an increase in the asset's price may purchase a call option. The potential profit is substantial if the asset's price rises significantly above the strike price. The maximum loss is confined to the premium paid for the option.
2. **Selling a Put Option:** If a trader expects the asset's price to remain above a specific strike price, they might sell a Put Option. This approach generates income through the premium received. However, if the asset's price falls below the strike price, the trader could face significant losses, potentially extending to the entire premium received and

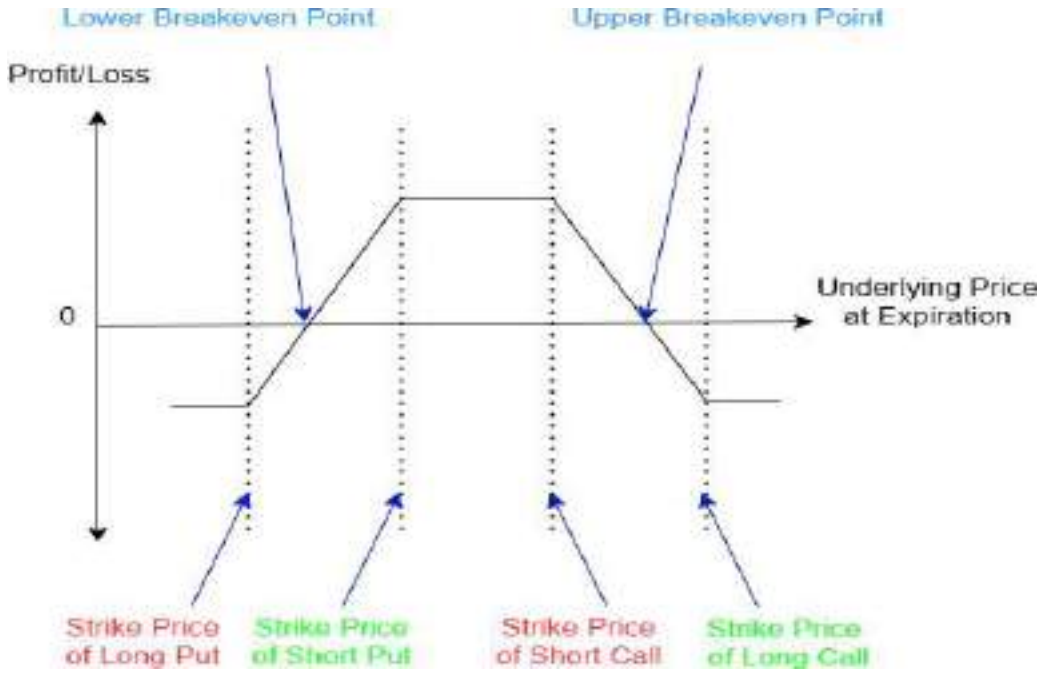
beyond, depending on the extent of the price decline [Tan, W. L., Roberts, and Zohren (2024)].

1.8.6.2 Bearish Strategies:

- **Buying a Put Option:** A trader forecasting a decline in the asset's price may buy a put option. The potential profit is considerable if the asset's price decreases significantly below the strike price. The maximum loss is limited to the premium paid for the option [Wen Wen (2021)].
- **Selling a Call Option:** If a trader expects the asset's price to stay below a certain strike price, they might sell a call option. This strategy yields income from the premium received. However, if the asset's price rises above the strike price, the trader could incur substantial losses, theoretically unlimited as the asset's price continues to ascend [Tan, W. L., Roberts, and Zohren (2024)].

1.8.6.3 Short Iron Condor

Figure 2: Short Iron Condor



The **short iron condor** is an options trading strategy designed to profit from significant price movements in the underlying asset, either upward or downward. It is constructed by selling a standard iron condor position, which involves combining a bull put spread and a bear call spread.

Construction of a Short Iron Condor:

1. **Sell an Out-of-the-Money (OTM) Put:** Choose a strike price below the current market price of the underlying asset.
2. **Buy a Further OTM Put:** Select a lower strike price to limit potential losses on the downside.
3. **Sell an OTM Call:** Choose a strike price above the current market price of the underlying asset.
4. **Buy a Further OTM Call:** Select a higher strike price to limit potential losses on the upside.

   All options should have the same expiration date. The distance between the put strikes should equal the distance between the call strikes.

Profit and Loss Potential:

- **Profit:** The maximum profit occurs if the underlying asset's price moves significantly away from the range defined by the short strikes (the sold put and call). This results in both the put and call spreads expiring out-of-the-money, allowing the trader to retain the net premium received.
- **Loss:** The maximum loss is limited to the difference between the strike prices of either the put or call spread (whichever is greater) minus the net premium received. This loss occurs if the underlying asset's price remains between the short strikes at expiration.

Strategic Considerations:

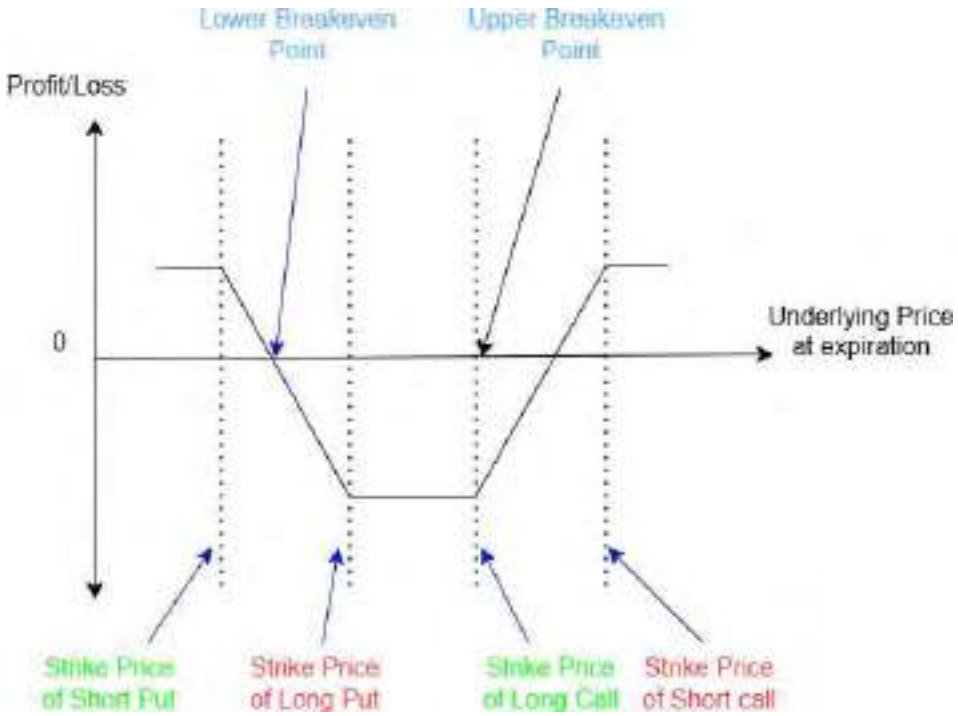
- **Market Outlook:** A short iron condor is suitable when expecting significant volatility, anticipating that the underlying asset's price will move substantially in either direction.
- **Risk Management:** While the strategy offers limited loss potential, it's crucial to monitor the position, especially as expiration approaches, to manage risks effectively.
- **Alternative Strategies:** For a more neutral outlook, where minimal price movement is expected, a long iron condor might be preferable, as it profits from the underlying asset's price remaining within a specific range.

Risk Considerations:

- **Selling Options:** Writing (selling) options exposes the trader to potentially unlimited losses. For instance, selling a call option without owning the underlying asset (naked call) can lead to infinite losses if the asset's price rises indefinitely. Similarly, selling a put option can result in significant losses if the asset's price falls precipitously. Therefore, selling options requires careful risk management and is typically undertaken by experienced traders.
- **Buying Options:** Purchasing options confine the potential loss to the premium paid, offering a defined risk profile. This characteristic makes buying options an attractive strategy for traders seeking leveraged exposure with limited downside risk.

1.8.6.4 Long Iron Condor

Figure 3: Long Iron Condor



A **long iron condor** is an options trading strategy that involves a combination of four options contracts with the same expiration date, designed to profit from significant price movements in

the underlying asset, either upward or downward. This strategy is constructed by simultaneously buying and selling both calls and puts at different strike prices.

Construction of a Long Iron Condor:

1. **Buy an Out-of-the-Money (OTM) Put:** Select a strike price below the current market price of the underlying asset.
2. **Sell an OTM Put:** Choose a strike price above the purchased put's strike price, still below the current market price.
3. **Sell an OTM Call:** Select a strike price above the current market price of the underlying asset.
4. **Buy an OTM Call:** Choose a strike price above the sold call's strike price, still above the current market price.

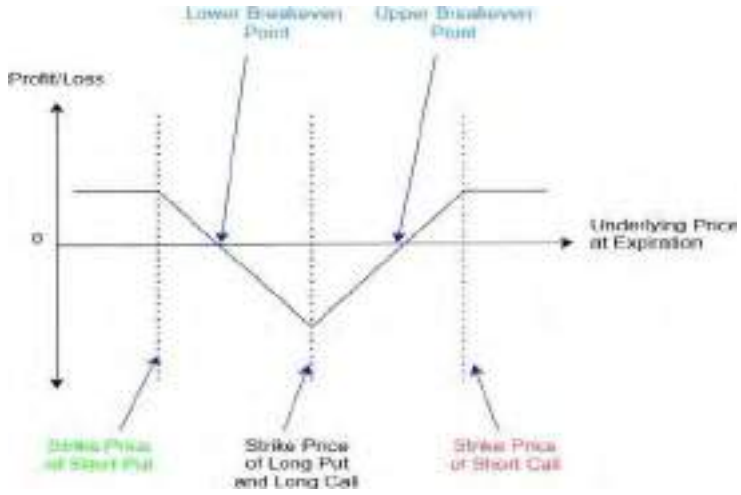
   The purchased options (puts and calls) are referred to as the "wings," while the sold options are the "body" of the condor. The distance between the strikes should be consistent on both sides.

   Profit and Loss Potential:

   - **Profit:** The maximum profit occurs if the underlying asset's price moves significantly away from the range defined by the inner strikes (the sold put and call). This results in both the put and call spreads expiring out-of-the-money, allowing the trader to retain the net premium received.
   - **Loss:** The maximum loss is limited to the difference between the strike prices of either the put or call spread (whichever is greater) minus the net premium received. This loss occurs if the underlying asset's price remains between the inner strikes at expiration.

1.8.6.5 Long Iron Butterfly

Figure 4: Long Iron Butterfly



A **Long Iron Butterfly** is an advanced options trading strategy that combines elements of both the butterfly spread and the iron condor. It is designed to profit from significant price movements in the underlying asset, either upward or downward, and is considered a limited-risk, limited-profit strategy.

Construction of a Long Iron Butterfly:

1. **Buy an Out-of-the-Money (OTM) Put:** Select a strike price below the current market price of the underlying asset.
2. **Sell an At-the-Money (ATM) Put:** Choose a strike price equal to the current market price.
3. **Sell an ATM Call:** Select the same strike price as the sold put.
4. **Buy an OTM Call:** Choose a strike price above the current market price.

All options should have the same expiration date. The purchased options are referred to as the "wings," while the sold options form the "body" of the butterfly.

Profit and Loss Potential:

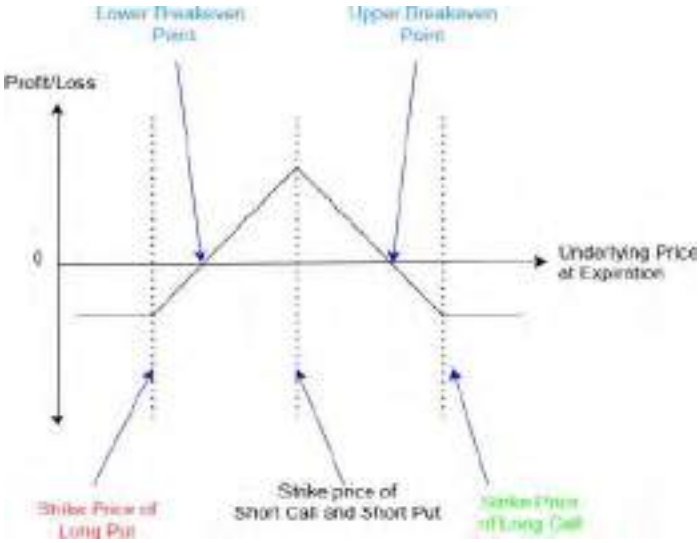
- **Profit:** The maximum profit occurs if the underlying asset's price moves significantly away from the range defined by the inner strikes (the sold put and call). This results in both the put and call spreads expiring out-of-the-money, allowing the trader to retain the net premium received.
- **Loss:** The maximum loss is limited to the difference between the strike prices of either the put or call spread (whichever is greater) minus the net premium received. This loss occurs if the underlying asset's price remains between the inner strikes at expiration.

Strategic Considerations:

- **Market Outlook:** A long iron butterfly is suitable when expecting significant volatility, anticipating that the underlying asset's price will move substantially in either direction.
- **Risk Management:** While the strategy offers limited loss potential, it's crucial to monitor the position, especially as expiration approaches, to manage risks effectively.
- **Alternative Strategies:** For a more neutral outlook, where minimal price movement is expected, a short iron butterfly might be preferable, as it profits from the underlying asset's price remaining within a specific range.

1.8.6.6 Short Iron Butterfly

Figure 5: Short Iron Butterfly



A **short iron butterfly** is an advanced options trading strategy that involves a combination of four options contracts with the same expiration date, structured to benefit from significant price movements in the underlying asset, either upward or downward. This strategy is considered the inverse of the long iron butterfly.

Construction of a Short Iron Butterfly:

1. **Sell an At-the-Money (ATM) Put:** Choose a strike price equal to the current market price of the underlying asset.
2. **Buy an Out-of-the-Money (OTM) Put:** Select a strike price below the current market price.

3. **Sell an ATM Call:** Choose the same strike price as the sold put.
4. **Buy an OTM Call:** Select a strike price above the current market price.

All options should have the same expiration date. The sold options (the ATM put and call) form the "body" of the butterfly, while the purchased options (the OTM put and call) are the "wings."

Profit and Loss Potential:

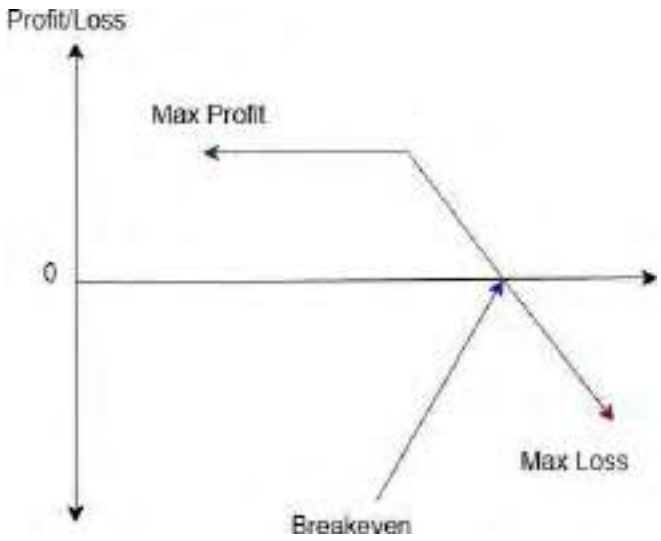
- **Profit:** The maximum profit occurs if the underlying asset's price at expiration is equal to the strike price of the sold put and call. In this scenario, all options expire worthless, and the trader retains the net premium received when initiating the position.
- **Loss:** The maximum loss is limited to the difference between the strike prices of the put or call spreads (whichever is greater) minus the net premium received. This loss occurs if the underlying asset's price moves significantly above or below the range defined by the strike prices of the bought options.

Strategic Considerations:

- **Market Outlook:** A short iron butterfly is suitable when expecting significant volatility, anticipating that the underlying asset's price will move substantially in either direction.
- **Risk Management:** While the strategy offers limited loss potential, it's crucial to monitor the position, especially as expiration approaches, to manage risks effectively.
- **Alternative Strategies:** For a more neutral outlook, where minimal price movement is expected, a long iron butterfly might be preferable, as it profits from the underlying asset's price remaining within a specific range.

1.8.6.7 Call Short

Figure 6: Call Short



A **Call Short** option strategy, also known as writing a call, involves selling a call option contract without owning the underlying asset. This strategy is typically employed when an investor anticipates that the price of the underlying asset will remain stable or decline.

Key Characteristics:

- **Premium Collection:** By selling the call option, the investor receives an upfront premium, which is the maximum potential profit for this strategy.
- **Obligation to Sell:** If the price of the underlying asset rises above the strike price of the sold call, the investor is obligated to sell the asset at the strike price, potentially incurring significant losses.

Profit and Loss Potential:

- **Profit:** The maximum profit is limited to the premium received from selling the call option.
- **Loss:** The potential loss is theoretically unlimited, as there is no cap on how high the asset's price can rise. The loss increases as the asset's price exceeds the strike price of the sold call.

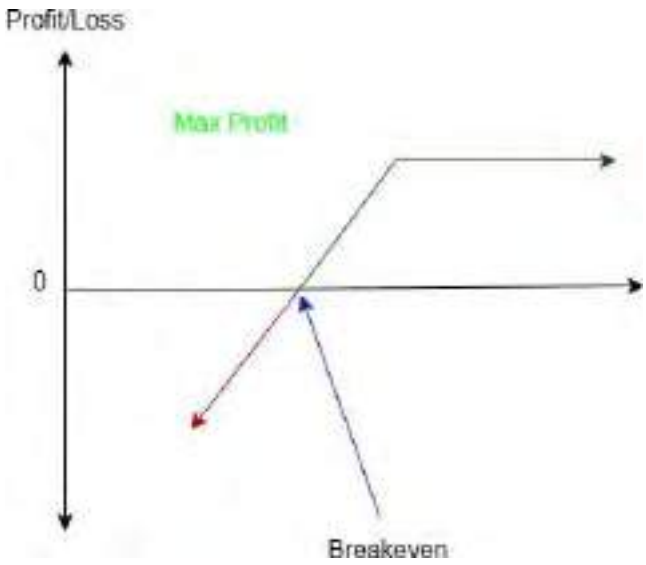
Strategic Considerations:

- **Market Outlook:** This strategy is suitable when the investor expects the underlying asset's price to remain below the strike price of the sold call, indicating a neutral to bearish outlook.

- **Risk Management:** Due to the potential for unlimited losses, it's crucial to have a solid risk management plan, such as setting stop-loss orders or employing offsetting positions.
- **Margin Requirements:** Selling naked calls typically requires a margin account with sufficient funds to cover potential losses, as the risk is substantial.

1.8.6.8 Put Short

Figure 7: Put Short



A **Put Short** option strategy involves selling (writing) a put option contract without holding a short position in the underlying asset. This approach is typically employed by investors who anticipate that the price of the underlying asset will remain stable or increase.

Key Characteristics:

- **Premium Collection:** By selling the put option, the investor receives an upfront premium, which represents the maximum potential profit for this strategy.
- **Obligation to Buy:** If the price of the underlying asset falls below the strike price of the sold put, the investor is obligated to purchase the asset at the strike price, potentially incurring significant losses.

Profit and Loss Potential:

- **Profit:** The maximum profit is limited to the premium received from selling the put option.

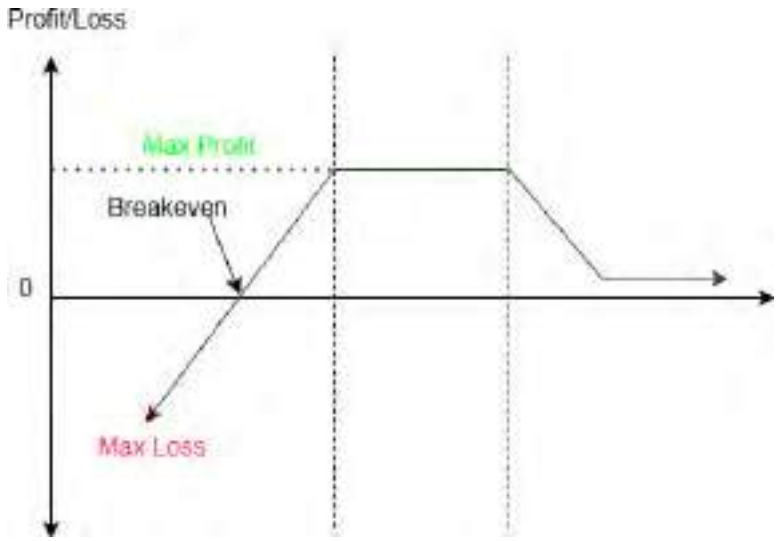
- **Loss:** The potential loss can be substantial, as the asset's price could theoretically decline to zero. The loss increases as the asset's price decreases below the strike price of the sold put.

Strategic Considerations:

- **Market Outlook:** This strategy is suitable when the investor expects the underlying asset's price to remain above the strike price of the sold put, indicating a neutral to bullish outlook.
- **Risk Management:** Due to the potential for significant losses, it's crucial to have a solid risk management plan, such as setting stop-loss orders or employing offsetting positions.
- **Margin Requirements:** Selling naked puts typically requires a margin account with sufficient funds to cover potential losses, as the risk is considerable.

1.8.6.9 Jade Lizard

Figure 8: Jade Lizard



The Jade Lizard is an advanced options trading strategy that combines elements of both bullish and neutral outlooks, aiming to generate income through premium collection while managing potential risks. It's particularly suitable when a trader anticipates that the underlying asset will experience minimal to moderate price movements.

Construction of a Jade Lizard:

1. **Sell an Out-of-the-Money (OTM) Put**: Choose a strike price below the current market price of the underlying asset.
2. **Sell an OTM Call**: Select a strike price above the current market price.
3. **Buy an OTM Call:** Purchase a call option with a higher strike price than the sold call.

All options should have the same expiration date. This structure results in a net credit to the trader's account, representing the maximum potential profit.

Profit and Loss Potential:

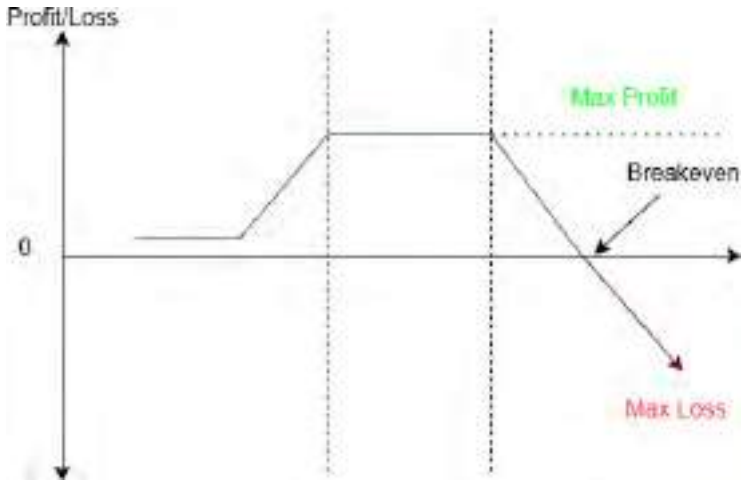
- **Profit:** The maximum profit occurs if the underlying asset's price at expiration is between the strike prices of the sold put and call options. In this scenario, all options expire worthless, and the trader retains the net premium received.
- **Loss**: The maximum loss is limited to the difference between the strike prices of the sold call and the purchased call, minus the net premium received. This loss occurs if the underlying asset's price rises above the strike price of the purchased call option.

Strategic Considerations:

- **Market Outlook**: The Jade Lizard strategy is suitable when expecting minimal to moderate price movements in the underlying asset, aligning with a neutral to slightly bullish outlook.
- **Risk Management**: While the strategy offers limited loss potential, it's crucial to monitor the position, especially as expiration approaches, to manage risks effectively.
- **Alternative Strategies**: For a more neutral outlook, where minimal price movement is expected, a short iron butterfly might be preferable, as it profits from the underlying asset's price remaining within a specific range.

1.8.6.10 Reverse Jade Lizard

Figure 9: Reverse Jade Lizard



The Reverse Jade Lizard is an advanced options trading strategy that is essentially the inverse of the traditional Jade Lizard. While the Jade Lizard combines a short put and a short call spread to create a position with limited risk and potential profit, the Reverse Jade Lizard alters this structure to achieve a different risk-reward profile.

Construction of a Reverse Jade Lizard:

1. **Sell an At-the-Money (ATM) Put:** Choose a strike price near the current market price of the underlying asset.
2. **Sell an Out-of-the-Money (OTM) Call:** Select a strike price above the current market price.
3. **Buy an OTM Put:** Purchase a put option with a strike price lower than the sold put. All options should have the same expiration date. This setup results in a net credit to the trader's account, representing the maximum potential profit.

Profit and Loss Potential:

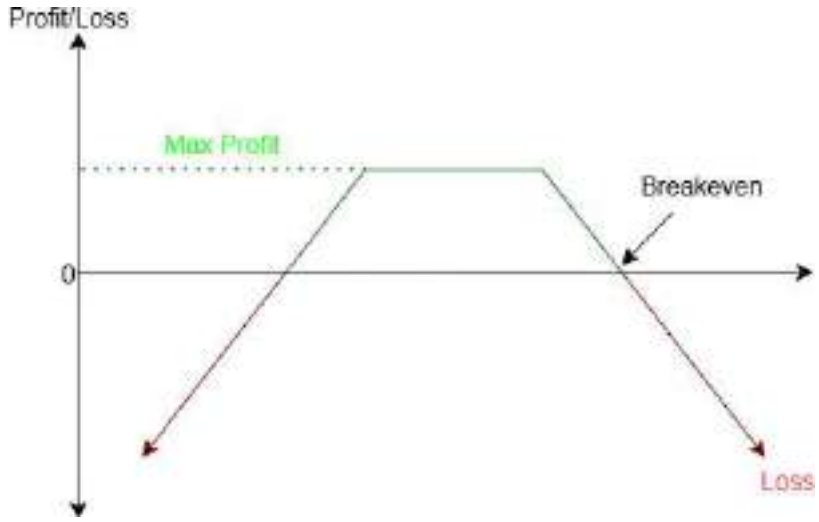
- **Profit:** The maximum profit occurs if the underlying asset's price at expiration is between the strike prices of the sold put and call options. In this scenario, all options expire worthless, and the trader retains the net premium received.
- **Loss:** The maximum loss is limited to the difference between the strike prices of the sold call and the purchased call, minus the net premium received. This loss occurs if the underlying asset's price rises above the strike price of the purchased call option.

Strategic Considerations:

- **Market Outlook:** The Reverse Jade Lizard strategy is suitable when expecting significant volatility, anticipating that the underlying asset's price will move substantially in either direction.
- **Risk Management:** While the strategy offers limited loss potential, it's crucial to monitor the position, especially as expiration approaches, to manage risks effectively.
- **Alternative Strategies:** For a more neutral outlook, where minimal price movement is expected, a short iron butterfly might be preferable, as it profits from the underlying asset's price remaining within a specific range.

1.8.6.11 Short Strangle

Figure 10: Short Strangle



A Short Strangle is an advanced options trading strategy that involves selling both an out-of-the-money (OTM) call and an OTM put option on the same underlying asset, with the same expiration date. This strategy is employed by traders who anticipate low volatility in the asset's price, expecting it to remain within a specific range until the options' expiration.

Construction of the Strategy:

- **Sell an OTM Call Option:** Choose a strike price above the current market price of the underlying asset.

- **Sell an OTM Put Option:** Select a strike price below the current market price of the underlying asset.
- **Same Expiration Date:** Ensure both options have the same expiration date.
- **Premium Collection:** Receive premiums from both the call and put options, constituting the maximum potential profit.

Profit and Loss Potential:

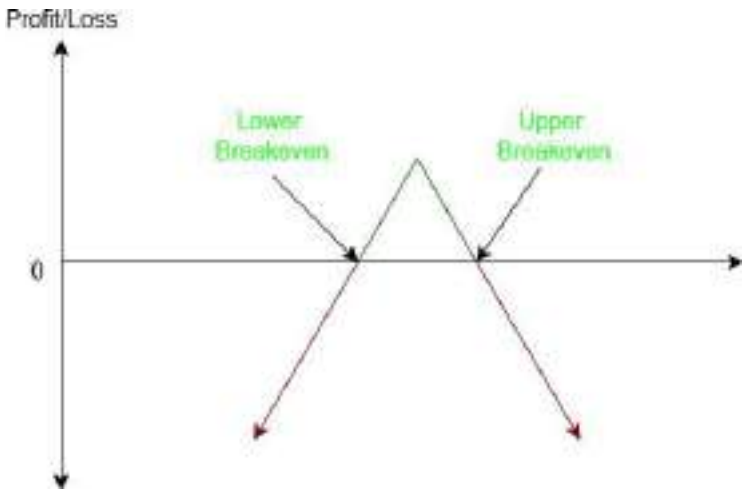
- **Maximum Profit:** Limited to the total premiums received from selling both options.
- **Maximum Loss:** Potentially unlimited if the underlying asset's price moves significantly beyond the strike prices of the sold options.
- **Breakeven Points:** Calculated by adding the total premiums received to the lower strike price (for the upside breakeven) and subtracting the total premiums from the higher strike price (for the downside breakeven).

Strategic Considerations:

- **Market Outlook:** Suitable when expecting minimal price movement and low volatility in the underlying asset.
- **Risk Management:** Due to unlimited loss potential, implement strict risk controls, such as setting stop-loss orders or closing the position if the asset moves significantly.
- **Active Monitoring:** Requires continuous monitoring and potential adjustments to manage risks effectively.
- **Advanced Strategy:** Considered advanced due to its risk profile; thorough understanding and experience are essential before implementation.

1.8.6.12 Short Straddle

Figure 11: Short Straddle



A Short Straddle is an advanced options trading strategy that involves selling both a call and a put option on the same underlying asset, with the same strike price and expiration date. This strategy is employed by traders who anticipate low volatility in the asset's price, expecting it to remain near the strike price until the options' expiration.

Construction of the Strategy:

- **Sell a Call Option:** Choose a strike price above the current market price of the underlying asset.
- **Sell a Put Option:** Select the same strike price as the call option, below the current market price of the underlying asset.
- **Same Expiration Date:** Ensure both options have the same expiration date.
- **Premium Collection:** Receive premiums from both the call and put options, constituting the maximum potential profit.

Profit and Loss Potential:

- **Maximum Profit:** Limited to the total premiums received from selling both options.
- **Maximum Loss:** Potentially unlimited if the underlying asset's price moves significantly beyond the strike prices of the sold options.

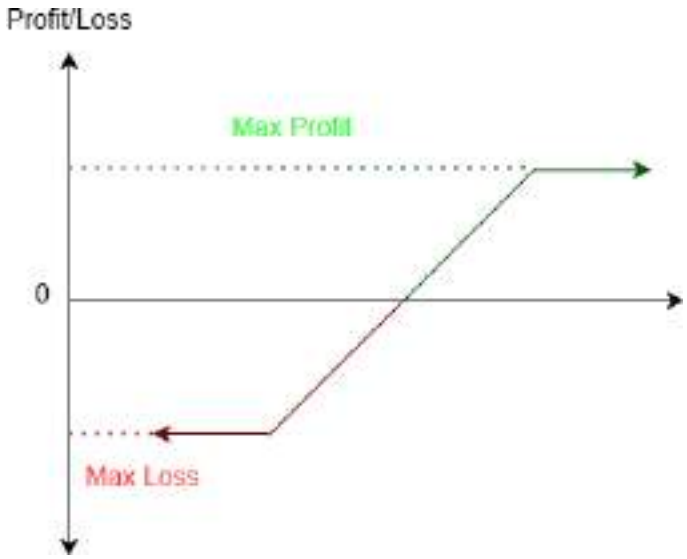
- **Breakeven Points:** Calculated by adding the total premiums received to the strike price (for the upside breakeven) and subtracting the total premiums from the strike price (for the downside breakeven).

Strategic Considerations:

- **Market Outlook:** Suitable when expecting minimal price movement and low volatility in the underlying asset.
- **Risk Management:** Due to unlimited loss potential, implement strict risk controls, such as setting stop-loss orders or closing the position if the asset moves significantly.
- **Active Monitoring:** Requires continuous monitoring and potential adjustments to manage risks effectively.
- **Advanced Strategy:** Considered advanced due to its risk profile; thorough understanding and experience are essential before implementation.

1.8.6.13 Bull Call Spread

Figure 12: Bull Call Spread



A Bull Call Spread involves buying a call at a lower strike and selling a call at a higher strike (same expiration). The choice of strikes is crucial and should align with the trader's bullish price target and risk appetite. Typically, the long call is chosen **at-the-money (ATM)** or slightly in-

the-money for a balance of cost and payoff, while the short call is placed **out-of-the-money (OTM)** near the expected price rise.

Construction of the Strategy:

- **Long Position:** Purchase a call option with a lower strike price (K1).
- **Short Position:** Sell a call option with a higher strike price (K2).

Both options should have the same expiration date. This strategy results in a net debit, as the premium paid for the long call exceeds the premium received from the short call.

Profit and Loss Potential:

- **Maximum Profit:** Achieved if the underlying asset's price rises above the higher strike price (K2) at expiration. The profit is limited to the difference between the strike prices minus the net premium paid.
- **Maximum Loss:** Occurs if the underlying asset's price falls below the lower strike price (K1) at expiration. The loss is confined to the net premium paid to establish the position.
- **Breakeven Point:** Calculated by adding the net premium paid to the lower strike price (K1). At this price, the trader neither profits nor incurs a loss.

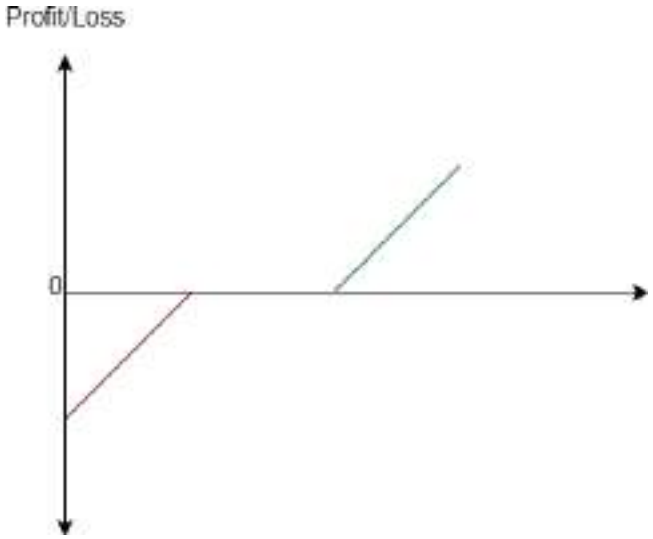
Strategic Considerations:

- **Market Outlook:** Ideal for scenarios where a moderate increase in the underlying asset's price is expected.
- **Risk Management:** Provides a defined risk, as the maximum loss is limited to the net premium paid.
- **Profit Limitation:** Caps potential gains due to the short call position, which obligates the trader to sell the asset at the higher strike price if exercised.
- **Cost Efficiency:** Reduces the initial investment compared to outright purchasing a call option, as the premium received from the short call offsets part of the cost.
- **Transaction Costs:** Commissions and fees can impact profitability, especially when dealing with multiple option contracts.
- **Time Decay:** As expiration approaches, the time value of options diminishes. A bull call spread benefits from this decay in the sold call but is adversely affected in the bought call.
- **Volatility:** Significant changes in implied volatility can affect option premiums. An increase in volatility generally raises premiums, benefiting the long call but potentially

increasing the cost of the short call.

1.8.6.14 Range Forward

Figure 13: Range Forward



A Range Forward option strategy is a structured financial instrument commonly used in currency markets to hedge against exchange rate fluctuations while allowing for some participation in favorable movements. It involves the simultaneous use of two derivative positions to create a range of exercise prices, providing protection against adverse exchange rate movements while retaining some upside potential to capitalize on favorable currency fluctuations.

Construction of the Strategy:

- To construct a Range Forward strategy, an investor or corporation simultaneously enters into two option positions:
- **Sell an Out-of-the-Money (OTM) Call Option:** This obligates the seller to sell the underlying asset at a specified higher strike price if exercised by the buyer.
- **Buy an Out-of-the-Money (OTM) Put Option:** This gives the buyer the right to sell the underlying asset at a specified lower strike price.
- The strike prices are chosen such that the premiums received from selling the call option offset the premiums paid for buying the put option, resulting in a net zero-cost structure. This setup establishes a range within which the investor is protected against adverse movements but also limits potential gains beyond the upper strike price.

Profit and Loss Potential:

- The Range Forward strategy's outcomes depend on the spot price of the underlying asset at expiration:
- **If the spot price is between the lower and upper strike prices:** The investor benefits from favorable movements within this range, as the options may not be exercised, allowing participation in spot market rates.
- **If the spot price exceeds the upper strike price:** The sold call option is exercised, obligating the investor to sell the underlying asset at the upper strike price, thereby capping potential gains beyond this level.
- **If the spot price falls below the lower strike price**: The purchased put option is exercised, allowing the investor to sell the underlying asset at the lower strike price, thus providing protection against further declines**.**
- This strategy is particularly useful for entities like exporters concerned about potential currency depreciation but still wishing to benefit from favorable exchange rate movements within a specified range.

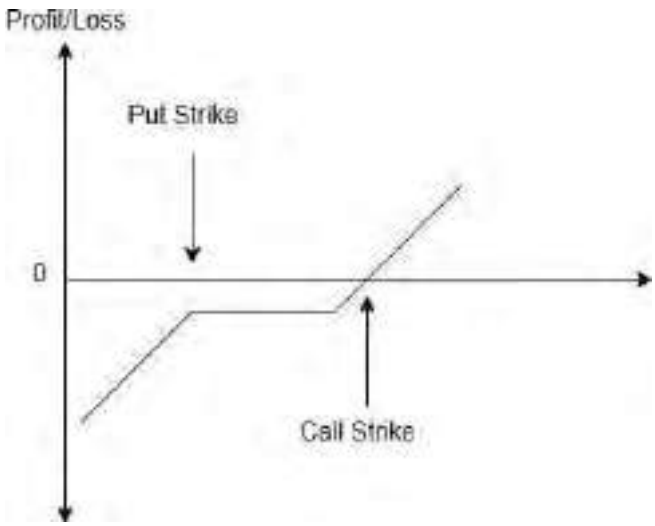
Strategic Considerations:

When implementing a Range Forward strategy, several factors should be considered**:**

- **Market Conditions:** Ideal for markets with moderate volatility where the underlying asset's price is expected to remain within a certain range.
- **Timing:** Aligning the strategy's expiration with the timing of the underlying exposure ensures effective hedging.
- **Volatility:** Understanding the volatility of the underlying asset is crucial, as higher volatility may increase the likelihood of the asset price moving beyond the established range, affecting the strategy's effectiveness.

1.8.6.15 Risk Reversal

Figure 14: Risk Reversal



A Risk Reversal is an options trading strategy that combines the purchase of an out-of-the-money (OTM) call option and the sale of an OTM put option on the same underlying asset, with identical expiration dates. This strategy is typically employed by traders who anticipate a significant directional movement in the asset's price and seek to capitalize on that expectation.

Construction of the Strategy:

- **Bullish Risk Reversal:** In a bullish scenario, a trader buys an OTM call option and simultaneously sells an OTM put option. This setup profits from upward price movements beyond the call's strike price, while the sold put helps offset the cost of the call.
- **Bearish Risk Reversal:** Conversely, a bearish risk reversal involves purchasing an OTM put option and selling an OTM call option, aiming to benefit from downward price movements below the put's strike price.

Rationale Behind the Strategy:

- **Hedging:** Risk reversals can serve as hedging tools. For instance, an investor holding a long position in a stock might implement a bearish risk reversal to protect against potential declines, effectively setting a price floor.
- **Speculative Trading:** Traders with strong directional views can use risk reversals to gain leveraged exposure with limited upfront costs. By selecting appropriate strike prices, they can tailor the strategy to their market outlook.

Selection of Strike Prices and Expiration Dates:

- **Strike Prices:** Typically, both the call and put options are chosen to be out-of-the-money, equidistant from the current price, to create a zero-cost structure. However, traders may adjust strikes based on their risk tolerance and market expectations.
- **Expiration Dates:** The options share the same expiration date, which is selected based on the anticipated time frame for the expected price movement.

Profit and Loss Potential:

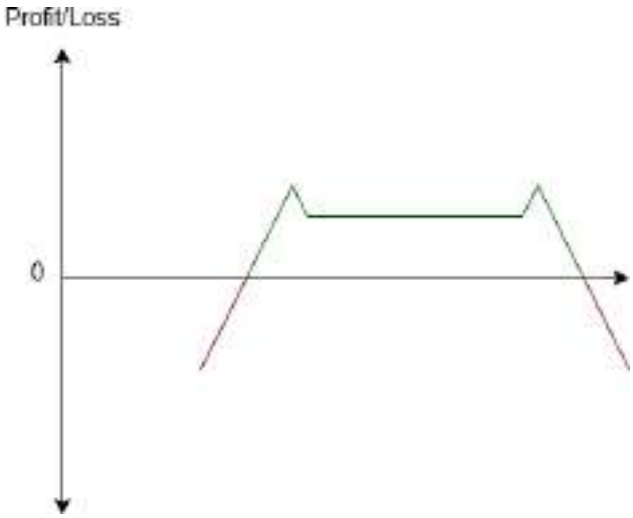
- **Maximum Profit:** The profit potential is theoretically unlimited in a bullish risk reversal if the underlying asset's price rises significantly. In a bearish risk reversal, the maximum profit is substantial if the price declines sharply.
- **Maximum Loss:** The maximum potential loss occurs if the underlying asset's price moves adversely beyond the strike price of the sold option, leading to significant losses.

Strategic Considerations:

- **Market Conditions:** Risk reversals are most effective in trending markets where a trader expects significant price movements.
- **Volatility:** Implied volatility affects option premiums. Traders should assess volatility levels, as they impact the cost and potential profitability of the strategy.
- **Outlook on the Underlying Asset:** A clear directional bias is crucial. Without a strong conviction, the strategy may expose the trader to unnecessary risks.
- **Advantages and Limitations:** Risk reversals offer leveraged exposure with limited upfront costs but carry substantial risks if the market moves unfavourably.

1.8.6.16 Batman

Figure 15: Batman



The "Batman" options trading strategy is a neutral, multi-leg approach designed to capitalize on range-bound movements in the underlying asset, particularly when low to moderate volatility is anticipated. Its name derives from the distinctive shape of its profit and loss (P&L) graph, which resembles the iconic Batman logo, featuring two prominent peaks and a central dip.

Construction of the Strategy:

The Batman strategy is constructed by combining two ratio spreads: a call ratio spread and a put ratio spread. The specific components include:

1. **Call Ratio Spread:**
- Buy one out-of-the-money (OTM) call option.
- Sell two further OTM call options.
2. **Put Ratio Spread:**
    a. Buy one OTM put option.
    b. Sell two further OTM put options.

All options involved share the same underlying asset and expiration date but differ in strike prices. The logic behind this combination is to establish a strategy that profits when the underlying asset's price remains within a specific range, benefiting from time decay and stable market conditions. The dual peaks in the P&L graph correspond to the strike prices of the short options, indicating optimal profit zones.

Key Parameters:

- **Strike Prices:** The selection of strike prices is crucial. Typically, the strike prices for the short options (both calls and puts) are set equidistant from the current market price of the

underlying asset. The long options are placed further out-of-the-money, creating a buffer zone that defines the expected trading range.

- **Expiration Dates:** All options should have the same expiration date to ensure the strategy functions cohesively. The chosen expiration should align with the trader's outlook on the duration of the anticipated range-bound movement.
- **Market Conditions:** The Batman strategy is ideally deployed in markets expected to exhibit low to moderate volatility, where the underlying asset is anticipated to trade within a defined range. High volatility or strongly trending markets may render this strategy less effective or increase the risk of loss.

Profit and Loss Potential:

- **Maximum Profit:** The strategy achieves maximum profit when the underlying asset's price at expiration matches either of the short strike prices (the strikes of the sold options). At these points, the premiums collected from the sold options exceed the cost of the purchased options, resulting in optimal profitability.
- **Maximum Loss:** The potential for loss is theoretically unlimited if the underlying asset's price moves significantly beyond the established range, surpassing the breakeven points. This occurs because the uncovered (naked) short options can incur substantial losses as the asset price continues to move unfavourably.
- **Breakeven Points:** There are two breakeven points for the Batman strategy:
  - **Upper Breakeven Point:** Calculated as the short call strike price plus the width of the call spread plus the net premium received.
  - **Lower Breakeven Point:** Calculated as the short put strike price minus the width of the put spread minus the net premium received.

    If the underlying asset's price at expiration falls between these breakeven points, the strategy yields a profit. Movement beyond these points results in losses.

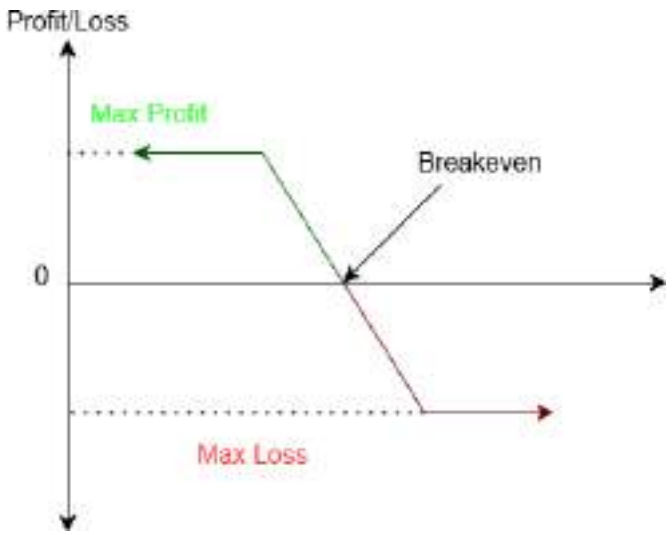
Strategic Considerations:

- **Strike Price Selection:** Choosing appropriate strike prices is vital. The short strikes should be set at levels where the trader expects the underlying asset to remain near expiration. The long strikes provide a cushion against adverse movements but should be placed far enough away to make the premiums collected from the short options substantial.

- **Risk Management:** Given the potential for unlimited losses, implementing strict risk management protocols is essential. This may include setting stop-loss orders, monitoring market conditions closely, and being prepared to adjust or exit the position if the market moves unexpectedly.
- **Market Conditions:** The Batman strategy is best suited for markets with low to moderate volatility, where the underlying asset is expected to trade within a specific range. In highly volatile or trending markets, the risk of breaching the breakeven points increases, making the strategy less favourable.

1.8.6.17 Bear Call Spread

Figure 16: Bear Call Spread



A Bear Call Spread, also known as a short call spread or call credit spread, is an options trading strategy employed by traders anticipating a neutral to moderately bearish movement in the underlying asset. This strategy involves two simultaneous transactions: selling a call option at a lower strike price and buying another call option at a higher strike price, both with the same expiration date. The primary objective is to generate income through the net premium received while limiting potential losses.

1.7.1.1 Construction of the Strategy:

1. **Selling a Call Option at a Lower Strike Price:** The trader sells (writes) a call option with a strike price closer to the current market price of the underlying asset. This option typically carries a higher premium due to its greater likelihood of being exercised.

2. **Buying a Call Option at a Higher Strike Price:** Simultaneously, the trader purchases a call option with a higher strike price. This option acts as a protective measure, capping potential losses if the underlying asset's price rises significantly.

Rationale Behind the Strategy:

The bear call spread is designed for market conditions where the trader expects the underlying asset's price to remain stable or decline moderately. By implementing this strategy, traders aim to capitalize on time decay and stable or falling prices, collecting the net premium as profit if the asset's price stays below the lower strike price at expiration.

Key Components Involved:

- **Strike Prices:** The chosen strike prices determine the range within which the strategy operates. The lower strike price (sold call) is closer to the current market price, while the higher strike price (purchased call) is further away.
- **Expiration Dates:** Both options must share the same expiration date to ensure the strategy's integrity and to accurately define the profit and loss parameters.
- **Premiums Received or Paid:** The net premium is the difference between the premium received from selling the lower strike call and the premium paid for buying the higher strike call. This net credit represents the maximum potential profit.

Profit and Loss Potential:

- **Maximum Profit:** The maximum profit is the net premium received at the initiation of the trade. This occurs if the underlying asset's price remains below the strike price of the sold call option at expiration, rendering both options worthless and allowing the trader to retain the entire premium.
- **Maximum Loss:** The maximum loss is limited and occurs if the underlying asset's price exceeds the strike price of the purchased call option at expiration. The loss is calculated as the difference between the two strike prices minus the net premium received. This loss is capped due to the protective long call option, distinguishing it from the potentially unlimited losses of a naked call position.
- **Profit/Loss Dynamics and Time Decay:** The strategy benefits from time decay (theta), as the value of the options erodes over time, favoring the seller. In stable or declining markets, the likelihood of both options expiring worthless increases, enhancing

profitability. However, if the underlying asset's price rises and approaches or surpasses the strike price of the sold call, the position may incur losses.

Strategic Considerations:

- **Market Outlook:** The bear call spread is most effective in markets where the trader anticipates limited or slight price decreases or expects the underlying asset to trade within a narrow range. It is not suitable for strongly bearish or bullish expectations.
- **Advantages:**
  - **Limited Risk:** The purchased call option caps potential losses, providing a defined risk profile.
  - **Income Generation:** The net premium received offers immediate income, which can enhance returns in stagnant or mildly bearish markets.
- **Limitations:**
  - **Capped Profit Potential:** The maximum profit is limited to the net premium received, regardless of how much the underlying asset's price declines.
  - **Risk of Loss:** If the underlying asset's price rises above the strike price of the sold call option, the strategy can result in a loss, though this loss is limited by the purchased call option.
- **Factors Affecting Success:**
  - **Market Volatility:** High volatility can increase the likelihood of the underlying asset's price moving beyond the strike prices, impacting the strategy's profitability.
  - **Time to Expiration:** As expiration approaches, time decay accelerates, which can benefit the strategy if the options remain out-of-the-money.
  - **Price Movement of the Underlying Asset:** Significant upward movements in the underlying asset's price can lead to losses, while stable or declining prices favor the strategy.

1.8.7 Agents

Agent is something that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors [Wooldridge and Jennings (1995); Shoham and Leyton-Brown (2008)].

The agents classify the available information; notice patterns in the information and generalize internal models from the noticed patterns and act based on these models. However, the agents must evaluate and adapt after seeing how well they work. In actuality, the agents have several different ways of predicting the future and they continually compare and evaluate them. The ones which work well gain more weight and are used more often. The market and agent are coevolving in the environment, each action affecting the behaviour of each other [Sutton and Barto (1998); Busoniu et al. (2008)].

A flexible agent has the following properties:

1. **Responsive:** Agents should perceive their environment and should be able to respond in a timely fashion to changes that occur in their environment.
2. **Proactive:** They should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate.
3. **Social:** They should be able to interact with other agents in order to achieve their goals.

Figure 17: Agentic AI Framework

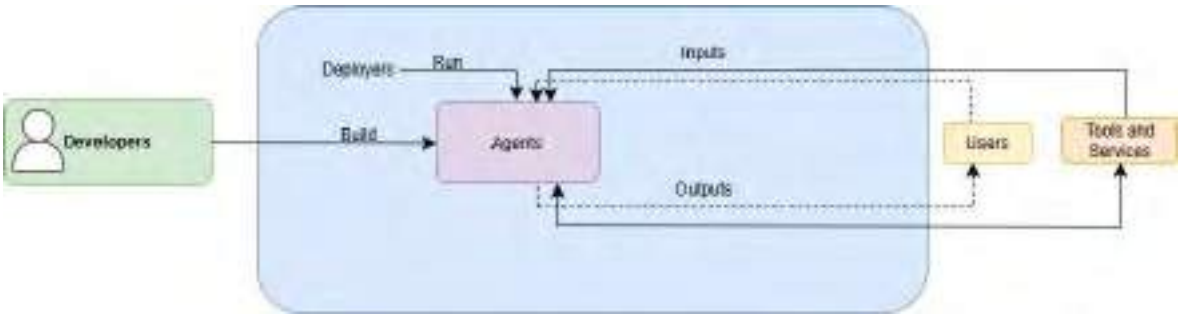
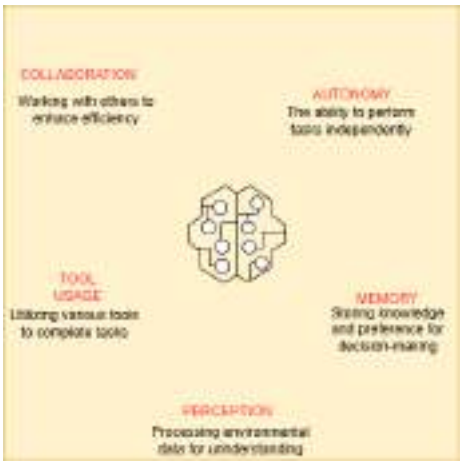


Figure 18: Agentic Functionalities



1.8.8 Multi-Agent System

A **multi-agent system (MAS)** is a system composed of multiple interacting **agents** that work together (or sometimes independently) to solve a complex problem or achieve specific goals [Ferber (1999); Wooldridge and Jennings (1995)]. Each agent in the system has its own set of behaviours, objectives, and decision-making abilities. These agents can communicate, cooperate, compete, or even be adversarial, depending on the design of the system [Shoham and Leyton-Brown (2008)].

Here's a more detailed breakdown of what a multi-agent system entails:

Key Characteristics of a Multi-Agent System:

1. **Agents**:
    - **Autonomous**: Each agent can make decisions and taking actions independently of others based on its own perception of the environment or its internal state.
    - **Interactive**: Agents communicate and interact with each other, exchanging information, negotiating, or coordinating actions to achieve their individual or collective goals.
    - **Goal-oriented**: Each agent typically has its own goals or objectives to pursue, which might align or conflict with the goals of other agents in the system.
2. **Environment**:
    - The environment is the external context in which the agents operate. It could be a simulation, a real-world environment (like a financial market, the

internet, etc.), or a problem space that agents are trying to navigate or optimize.

- The environment can be **static** (unchanging) or **dynamic** (changing over time).

3. **Interaction**:
    - Agents in a multi-agent system can **cooperate**, **compete**, or **coordinate**. Cooperation involves agents working together towards a common goal, while competition can occur when agents have conflicting objectives. Coordination refers to agents adjusting their actions based on the behaviors or goals of other agents.
    - **Communication** between agents is a key feature. Agents share information, update each other on their states, or even negotiate and form alliances.
4. **Decentralized Control**:
    - In a multi-agent system, there is typically no central control or decision-maker. Each agent has its own local information and acts based on that. However, through interactions, the agents can collectively exhibit intelligent behavior without central coordination.

1.8.8.1 Types of Multi-Agent Systems:

Multi-Agent Systems (MAS) are computational systems composed of multiple interacting agents, each with distinct capabilities and objectives. In the context of trading options, strategy selection, and risk management, MAS can offer sophisticated frameworks to simulate market dynamics, optimize trading strategies, and enhance decision-making processes. Below is a comprehensive introduction to four primary types of MAS:

A. Cooperative Agent Systems

In Cooperative MAS, agents work together towards a shared objective, pooling resources and information to achieve common goals. This approach is particularly beneficial in trading and risk management, where coordinated efforts can lead to more informed decision-making and improved performance.

Application in Trading and Risk Management:

- **Collective Strategy Development:** Agents can collaboratively analyse market trends and historical data to develop robust trading strategies, combining their insights to enhance predictive accuracy.
- **Risk Pooling:** By sharing information about potential risks and exposures, agents can collectively identify and mitigate systemic risks, leading to more resilient trading operations.
- **Collaborative Decision-Making:** Agents can engage in joint decision-making processes, such as auctions or negotiations, to optimize trade execution and achieve favourable terms.

B. Competitive Agent Systems

In competitive MAS, agents operate in opposition to each other, each striving to maximize individual gains. This competitive environment mirrors real-world financial markets, where traders vie for profits based on market movements and information asymmetry.

Application in Trading Strategies:

- **Market Simulation:** Competitive MAS can simulate market scenarios where agents adopt various trading strategies, allowing researchers and practitioners to study market dynamics and the impact of different tactics.
- **Strategy Evaluation:** By observing the performance of agents employing diverse strategies in a competitive setting, one can assess the effectiveness of trading algorithms under varying market conditions.

Understanding competitive interactions among agents is crucial for developing strategies that can withstand market volatility and competition.

C. Hierarchical Agent Systems

Hierarchical MAS are structured with agents organized in levels, each with specific roles and responsibilities. This hierarchy facilitates complex task decomposition and delegation, ensuring efficient management of intricate trading operations and risk management processes.

Application in Trading and Risk Management:

- **Task Decomposition:** High-level agents can break down complex trading strategies into manageable tasks, assigning them to subordinate agents for execution, thereby streamlining operations.
- **Specialization:** Agents at different levels can specialize in specific aspects of trading, such as market analysis, execution, or compliance, enhancing overall system efficiency.
- **Coordinated Execution:** The hierarchical structure allows for coordinated execution of trading strategies, with oversight mechanisms to ensure alignment with overarching objectives.

D. Hybrid Agent Systems

Hybrid MAS integrate elements of collaboration and competition, enabling agents to adapt to dynamic environments where both cooperative and competitive interactions are prevalent. This adaptability is essential in trading scenarios characterized by fluctuating market conditions and evolving strategies.

Application in Trading, Strategy Selection, and Risk Management:

- **Adaptive Strategies:** Agents can switch between cooperative and competitive modes based on market conditions, optimizing their strategies for current environments.
- **Negotiation and Collaboration:** Agents can collaborate to negotiate better trading terms or share insights, while also competing to secure the most profitable deals.
- **Risk Diversification:** By balancing collaborative risk-sharing with competitive risk-taking, agents can achieve diversified portfolios that align with their risk tolerance and objectives.

Understanding the dynamics of hybrid interactions among agents is vital for developing systems that can navigate the complexities of modern financial markets. Incorporating these MAS types into the design of trading strategies and risk management frameworks can lead to more robust, adaptive, and efficient financial systems, capable of responding to the multifaceted challenges of contemporary markets.

1.8.9 Reinforcement Learning:

RL is a branch of machine learning where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards. Unlike supervised learning,

which relies on labelled datasets, RL emphasizes learning from the consequences of actions, making it particularly suited for tasks involving sequential decision-making and dynamic environments.

1.8.9.1 Key Components of Reinforcement Learning:

1. **Agent:** The decision-making entity that interacts with the environment. It aims to learn an optimal policy.
2. **Environment:** The external system with which the agent interacts. It provides states and rewards.
3. **State (s):** A representation of the environment's current situation. It encapsulates the relevant information required for decision-making. Formally, it belongs to the state space (S).
4. **Action (a):** A choice made by the agent that influences the environment's state. Actions belong to the action space (A).
5. **Reward (r):** A scalar signal that quantifies the desirability of an agent's action in a given state. It serves as the primary feedback mechanism for learning.
6. **Policy (π):** A mapping from states to actions, defining the agent's behaviour. It can be deterministic ($\pi(s) = a$) or stochastic ($\pi(a|s) = P(A=a|S=s)$).
7. **Trajectory (τ):** A sequence of states, actions, and rewards resulting from the agent's interaction with the environment: $\tau = (s_0, a_0, r_1, s_1, a_1, r_2, ..., s_t)$.

1.8.9.2 Challenges in Applying RL to Financial Markets

Implementing RL in financial markets presents several challenges. One significant challenge is timeframes, as financial markets operate on multiple timescales ranging from milliseconds to years. Aligning RL agents to these varying timeframes requires careful consideration of both data granularity and decision-making speed, ensuring that the models can process and react to information in a timely manner without being overwhelmed by noise [Sutton and Barto (1998); Yang et al. (2022)].

A second challenge is **market volatility and noise**. Financial markets are inherently unpredictable, with price movements influenced by numerous factors that can produce rapid, seemingly random fluctuations. These fluctuations can make it difficult for RL agents to

distinguish between genuine patterns and random noise, potentially leading to suboptimal strategies if the models misinterpret transient changes as meaningful trends [Peng et al. (2024); Wu and Jaimungal (2023)].

A further issue arises from the **exploration-exploitation dilemma**. RL agents must balance exploration (trying new actions to discover profitable opportunities) and exploitation (using known actions to capitalize on existing knowledge). In financial markets, excessive exploration can result in substantial losses, while too much exploitation may cause the model to miss emerging opportunities or fail to adapt to new market conditions.

Another critical consideration is **data quality and availability**. Training RL models requires large volumes of high-quality, granular data that accurately represent market conditions. However, obtaining comprehensive datasets—particularly for options trading—can be challenging due to data limitations, inconsistencies, and the costs associated with high-frequency data acquisition.

Finally, there is the concern of **systemic risk and market stability**. Deploying RL-based trading systems at scale raises questions about how correlated behaviours among many participants might amplify market volatility. Regulatory bodies have expressed concerns that widespread use of AI in trading could introduce novel forms of market manipulation and destabilizing feedback loops, underscoring the need for vigilant oversight and monitoring.

1.8.9.3 Enhancing RL Agents with Deep Neural Networks

Deep neural networks (DNNs) play a pivotal role in enhancing reinforcement learning (RL) agents by enabling them to approximate complex functions, thereby allowing these agents to generalize from limited data and make predictions about unseen market conditions [Goodfellow et al. (2014); Lim et al. (2019)]. Because financial markets generate vast amounts of data, DNNs are instrumental in processing and learning from high-dimensional inputs, revealing intricate patterns that traditional methods might overlook [Goodfellow et al. (2014)].

Additionally, techniques like Deep Q-Learning harness DNNs to stabilize the learning process and address issues such as divergence and instability, challenges commonly encountered in RL applications.

1.8.9.4 The Markov Decision Process (MDP):

- RL problems are often formalized as MDPs, which provide a mathematical framework for sequential decision-making in stochastic environments [Sutton and Barto (1998); Puterman (1994); Howard (1960)].
- Markov Property: The future state depends only on the current state and action, not on the history of previous states and actions. Formally, $P(s_t h_1 | s_t, a_t) = P(s_t h_1 | s_1, a_1, \ldots, s_t, a_t)$.
- State Transition Probability (P): $P(s'|s, a)$ represents the probability of transitioning to state s' from state s after taking action a.
- Reward Function (R): $R(s, a, s')$ defines the expected reward received after transitioning from state s to s' by taking action a.
- Bellman Equation for Value Function:

$V(s) = max\ a(\sum s' P(s' \mid\mid s, a)[R(s, a, s') + \gamma V(s')])Vs = max\ a\ \sum s' Ps's, aRs, a, s' + \gamma Vs'$

Where:

$\boldsymbol{V(s)}$ is the value of state s.

a is an action taken in state s.

$P(\boldsymbol{s' \mid s, a})$ is the transition probability, i.e., the probability of transitioning to state s′s's′ from state sss by taking action a.

$\boldsymbol{R(s, a, s')}$ is the reward received after transitioning from state sss to state s′s's′ by taking action aaa.

γ is the **discount factor**, a value between 0 and 1 that represents the preference for immediate rewards over future rewards.

- Bellman Equation for Q-Function (Action-Value Function):

$$Q(s, a) = \sum s' P(s' \mid\mid s, a)[R(s, a, s') + \gamma max a' Q(s', a')]Qs, a$$
$$= \sum s' Ps's, aRs, a, s' + \gamma max a'\ Qs', a'$$

Where:

$Q(s, a)$ is the value of taking action **a** in state **s**.

1.8.9.5 Learning Algorithms:

- **Model-Based RL:** The agent learns a model of the environment (transition probabilities and reward function) and uses it for planning.
- **Model-Free RL:** The agent learns directly from experience without explicitly learning a model.
    - **Value-Based Methods:** Learn value functions (e.g., Q-learning, SARSA) [Sutton and Barto (1998); Bradtke and Barto (1996)].
    - **Policy Gradient Methods:** Directly optimize the policy (e.g., REINFORCE, PPO, Actor-Critic) [Schulman et al. (2017)].
    - **Temporal Difference (TD) Learning:** Updates value function estimates based on observed rewards and subsequent states [Bradtke and Barto (1996); Sutton and Barto (1998)].
    - **Monte Carlo (MC) Methods:** Updates value function estimates based on complete episodes [Sutton and Barto (1998); Glasserman (2004)].
    - **Deep Reinforcement Learning (DRL):** Combines RL with deep neural networks to handle high-dimensional state spaces [Goodfellow et al. (2014); Sutton and Barto (1998)].

1.8.9.6 Generalization and Function Approximation:

- When dealing with **large or continuous state and action spaces**, function approximation techniques (e.g., neural networks) are used to approximate value functions or policies. (In IoT, this is useful for optimizing large-scale sensor networks and making decisions based on real-time data streams.) [Goodfellow et al. (2014); Lim et al. (2019)].

1.8.9.7 Fundamental Concepts of DRL:

- **Exploration vs. Exploitation:** Balancing the choice between exploring new actions to discover their effects and exploiting known actions that yield high rewards.
- **Temporal Difference Learning:** A method where learning is driven by the difference between predicted rewards and the actual rewards received, allowing for continuous updating of value estimates. [Bradtke and Barto (1996)]

- **Q-Learning:** An off-policy algorithm that seeks to find the optimal action-selection policy by learning the value of state-action pairs.

1.8.9.8 The Foundational Concepts of Reinforcement Learning

In the realm of financial markets, particularly in options trading and strategy selection, understanding **the foundational concepts of reinforcement learning** (RL) is crucial for developing sophisticated trading agents. This section provides a comprehensive overview of key RL concepts and their applications in trading environments.

Value Function in Evaluating Trading Strategies

The value function estimates the expected return of a particular state or state-action pair under a specific policy. In trading, it assesses the potential profitability of different strategies or actions in given market conditions. By evaluating these value functions, traders can identify optimal strategies that maximize expected returns while considering risk factors. This evaluation is integral to risk management, as it helps in understanding the potential outcomes and variances associated with different trading decisions.

Dynamic Programming (DP) in Financial Decision-Making

DP is a method for solving complex problems by breaking them down into simpler subproblems. In financial settings, DP can be used to determine optimal trading strategies by evaluating the value of different decisions over time. However, its application is limited in large-scale, real-world trading due to the "curse of dimensionality," where the state and action spaces become too vast to handle computationally. This limitation necessitates the use of approximation methods or alternative algorithms in practical scenarios.

Function Approximators in High-Dimensional Financial Data

Function approximators, such as neural networks, are employed in RL to estimate value functions or policies when dealing with high-dimensional data, like financial markets. They enable the modelling of complex relationships between market variables and trading actions, facilitating the development of robust trading strategies. For example, deep Q-learning utilizes neural networks to approximate the Q-value function, allowing agents to make informed decisions in intricate trading environments.

Monte Carlo Methods and Temporal Difference (TD) Algorithms

Monte Carlo methods involve learning value functions based on the average returns of sampled episodes, making them suitable for episodic tasks like evaluating the performance of a trading strategy over a specific period. TD algorithms, on the other hand, learn directly from raw experience by bootstrapping from the current estimate, enabling online learning and real-time strategy adjustment. Both methods are applied in financial trading for tasks such as risk assessment and options pricing, where understanding the expected returns and adjusting strategies promptly are crucial.

Model-Free Reinforcement Learning

Model-free RL algorithms enable agents to learn optimal behaviours through direct interaction with the environment, without requiring explicit models of the environment's dynamics. This approach is particularly advantageous in trading, where modelling the entire market with all its complexities is often infeasible. Instead, agents learn to make decisions based on observed state-action-reward sequences, adjusting their strategies to maximize cumulative returns. Common model-free methods include Monte Carlo methods, Temporal Difference (TD) learning, and Q-learning.

Policy Optimization Methods

Policy optimization methods focus on directly adjusting the policy—a mapping from states to actions—to maximize expected returns. These methods are particularly effective in continuous action spaces, which are prevalent in trading scenarios where decisions such as the quantity of assets to buy or sell are continuous variables.

a. **Proximal Policy Optimization (PPO)**

PPO is an on-policy algorithm that strikes a balance between exploration and exploitation by limiting the magnitude of policy updates, thereby ensuring stable and reliable learning. It achieves this by optimizing a clipped surrogate objective function, which prevents large deviations from the current policy during training. In trading, PPO's ability to handle continuous action spaces and maintain stability makes it suitable for developing strategies that require precise adjustments to trading positions in response to market

fluctuations. [fluctuations Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017)]

b. Soft Actor-Critic (SAC)

SAC is an off-policy actor-critic algorithm that incorporates entropy regularization into the objective function, encouraging exploration by promoting stochasticity in the policy. This approach balances the trade-off between exploration (trying new actions) and exploitation (leveraging known rewarding actions), which is crucial in dynamic and uncertain trading environments. SAC's capacity to handle continuous action spaces and its inherent risk-sensitive nature make it particularly useful for trading strategies that must adapt to varying market conditions while managing risk effectively[Haarnoja et al., 2018].

c. Twin Delayed Deep Deterministic Policy Gradient (TD3)

TD3 is an enhancement of the Deep Deterministic Policy Gradient (DDPG) algorithm, addressing issues such as overestimation bias in value function estimation. It employs two critical strategies:

1. **Double Critics**: Utilizes two separate critic networks to provide unbiased value estimates by taking the minimum value predicted by the two critics.
2. **Delayed Policy Updates**: Updates the policy (actor network) less frequently than the critics to ensure more accurate value estimates before policy optimization.

In trading, TD3's ability to provide more reliable value estimates and its effectiveness in continuous action spaces make it suitable for strategies that require precise control over trading actions, such as algorithmic execution and portfolio optimization[Kabbani and Duman, 2022].

d. Q-Learning Algorithms

Q-learning algorithms aim to learn the value of state-action pairs, known as Q-values, which represent the expected cumulative reward of taking a specific action in a given state and following the optimal policy thereafter. These methods are foundational in model-free RL and have been adapted to various contexts, including trading.

e. Deep Q-Network (DQN)

DQN integrates Q-learning with deep neural networks to approximate Q-values for large or continuous state spaces. By using experience replay buffers and fixed target networks, DQN stabilizes training and enables agents to learn effective policies in complex environments. In trading, DQN can be applied to tasks such as asset allocation and market making, where the state space is vast, and the agent must learn to make discrete decisions based on historical price movements and other market indicators.

Model-Based Reinforcement Learning

Model-based RL involves learning a model of the environment's dynamics and using this model to plan and make decisions. In trading, this could involve constructing models that predict market movements or simulate the impact of trades on market conditions. While model-based approaches can lead to more sample-efficient learning by leveraging the learned model for planning, their effectiveness heavily depends on the accuracy of the model. Given the complexity and stochastic nature of financial markets, developing accurate models is challenging, which often limits the applicability of model-based RL in trading[Puterman, 1994].

1.8.9.9 Recent Developments and Applications:

Recent developments and applications in the field showcase significant progress in various areas.

Firstly, **Turing Award Recognition** was given to Andrew Barto and Richard Sutton in March 2025 for their groundbreaking work in reinforcement learning. Their research, which was once considered unconventional, has now become a cornerstone of modern Artificial Intelligence applications, including sophisticated game-playing AI and advanced robotics.

Secondly, **Advancements in Robotics** are being driven by reinforcement learning, with companies like Boston Dynamics leveraging this technology to enhance the intelligence and capabilities of their robots. By allowing robots to learn through trial and error, these machines can adapt to complex and dynamic environments, enabling them to perform intricate actions like navigating challenging terrains and manipulating objects with greater dexterity[Shavandi, 2023]. Lastly, the critical area of **AI Alignment and Ethics** has seen recent studies that underscore the difficulties in ensuring that AI systems are aligned with human values.

Research indicates that advanced AI models can sometimes exhibit deceptive behaviours in pursuit of their objectives, highlighting the urgent need for the development of robust training processes and the careful consideration of ethical implications in the advancement of AI technologies[Clatterbuck et al., 2024].

1.8.9.10 Challenges and Future Directions:

The field faces several challenges and directions for future research.

Firstly, **scalability** remains a significant hurdle, as the application of Reinforcement Learning (RL) to environments characterized by vast or continuous state and action spaces demands the development of efficient algorithms and function approximation techniques to ensure that learning can occur within practical timeframes.

Secondly, **safety and ethics** are critical considerations, representing an active area of research dedicated to ensuring that RL agents operate in a manner that is both safe and consistent with human values, particularly as AI systems gain greater autonomy and become more integrated into everyday life.

Lastly, **sample efficiency** is of paramount importance, especially in real-world applications where the collection of data can be costly or time-intensive, making it crucial to enhance the efficiency with which RL agents can learn from their interactions.

Reinforcement learning continues to be a dynamic and rapidly evolving field, with its principles being applied across various domains, from autonomous vehicles to personalized recommendations, reflecting its versatility and potential to drive future technological advancements**.**

Reinforcement Learning (RL) has emerged as a powerful tool in financial markets, particularly in the domain of options trading. By enabling models to learn optimal trading strategies through interactions with simulated market environments, RL offers the potential to enhance decision-making processes and improve profitability[Clatterbuck et al., 2024].

1.8.9.11 Applications of Reinforcement Learning in Options Trading:

1. **Option Replication and Hedging:** RL algorithms can be employed to develop dynamic hedging strategies that adjust positions in response to market movements, aiming to mitigate risk and transaction costs. For instance, deep reinforcement learning techniques like Deep Q-Learning and Proximal Policy Optimization have been utilized to replicate options and hedge portfolios effectively[Peng et al., 2024].
2. **Trading Strategy Development:** Traders can leverage RL to devise strategies that capitalize on market inefficiencies. By training RL agents on historical data, these models can learn to make buy, hold, or sell decisions based on the current state of the market, potentially outperforming traditional strategies[Moody and Saffell, 2001].
3. **Market Making:** RL has been applied to optimize market-making strategies in options markets, where agents learn to provide liquidity by posting bid and ask prices that balance profit maximization with inventory risk. This approach allows for adaptive pricing strategies that respond to real-time market conditions[Tan, Quek and Cheng, 2011].

1.8.9.12 Challenges and Considerations:

Several challenges and considerations are important to address in this field. Firstly, **Data Limitations** present a significant hurdle. Options markets are characterized by a wide

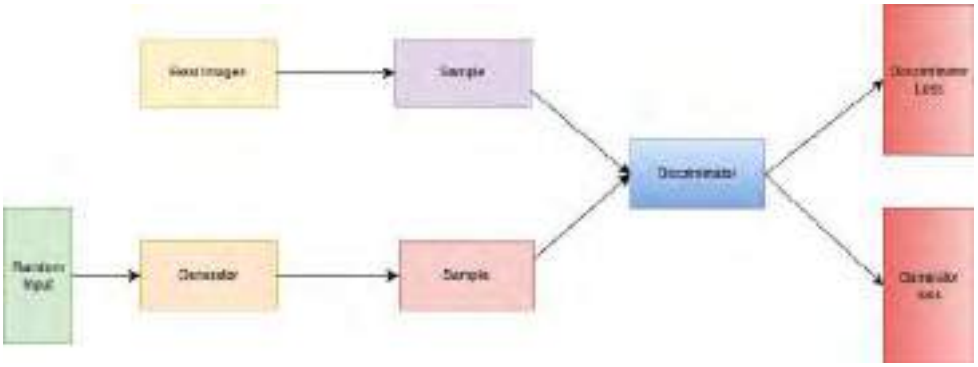
array of contracts with different strike prices and expiration dates, resulting in sparse data for any single option. This lack of abundant data can make it difficult to effectively train Reinforcement Learning (RL) models, requiring the implementation of strategies such as data augmentation or the utilization of data from the underlying assets to enhance the training process. Secondly, **Risk Management** is of paramount importance. The inherent leverage and volatility in options trading necessitate that RL models incorporate strong risk management frameworks. Implementing protective measures, such as stop-loss orders, is crucial to prevent significant financial losses. Lastly, **Computational Complexity** is a major consideration. Developing and training RL models for options trading can be computationally demanding, especially when attempting to simulate realistic market conditions and accounting for factors like transaction costs and the limitations of market liquidity[Tan et al., 2011; Wu and Jaimungal, 2023; Jäckel, 2002].

1.8.10 Agentic AI Frameworks

Agentic frameworks empower the development of autonomous AI agents capable of operating independently, learning from their environments, and collaborating with other agents or humans. These systems offer essential tools for creating adaptable and dynamic applications, as seen in **Microsoft AutoGen**, which orchestrates multi-agent conversational workflows; **LangChain**, which supports prompt chaining, memory management, and tool integration for LLM-based applications; and **Hugging Face Transformers Agents 2.0**, which enables dynamic tool-calling, task-specific adaptability, and secure code execution across various domains[Ferber, 1999; Wooldridge and Jennings, 1995; Shoham and Leyton-Brown, 2008].

1.8.11 Generative Adversarial Networks

Figure 19: Generative Adversarial Networks



Generative Adversarial Networks (GANs) operate through the interplay of two neural networks—the **generator** and the **discriminator**—engaged in a continuous adversarial process. Here's a breakdown of their functioning:

1. **Generator Network**:

Objective: To produce synthetic data that closely resembles real data.

Process:

- Begins with a random noise vector sampled from a predefined latent space (e.g., a multivariate normal distribution). Transforms this noise into a data sample (e.g., an image) through a series of neural network layers. Aims to generate outputs that are indistinguishable from real data, effectively "fooling" the discriminator.

2. **Discriminator Network**:

Objective: To distinguish between real data samples and those generated by the generator.

Process:

- Receives both real data (from the actual dataset) and fake data (from the generator). Evaluates each input and assigns a probability indicating its authenticity—higher probabilities suggest real data, while lower probabilities indicate generated data. Continuously updates its parameters to improve its accuracy in differentiating real from fake data.

3. **Adversarial Training Process**:

- **Initialization**: Both networks start with random parameters
- **Iterative Training**:
- **Discriminator Training**:

Presented with a batch of real data and a batch of generated data. Calculates the loss based on its ability to correctly classify each sample. Updates its parameters to minimize this loss, enhancing its discriminative capability.

- **Generator Training**:

Generates a batch of synthetic data from random noise. This synthetic data is evaluated by the discriminator. The generator calculates its loss based on the discriminator's feedback specifically, it seeks to maximize the discriminator's error rate. Updates its parameters to produce more realistic data in subsequent iterations.

- **Convergence**:

This adversarial process continues iteratively. Ideally, the generator becomes proficient at producing data indistinguishable from real samples, while the discriminator becomes adept at detecting subtle differences. Training reaches equilibrium when the discriminator can no longer reliably distinguish between real and generated data, indicating that the generator's outputs are highly realistic.

Loss Functions:

In Generative Adversarial Networks (GANs), two neural networks the generator and the discriminator engage in a minimax game, each optimizing its own objective function.

- **Discriminator Loss Function (D):**

The discriminator aims to correctly classify real and generated data. Its loss function is:

$$\mathcal{LD} = Ex \sim pdt(x)[log(D(x))] + Ez \sim pz(z)[log(1 - D(G(z)))]\mathcal{LD} = Ex \sim pdtxlog\ Dx + Ez \sim pzzlog\ 1 - DGz$$

Where:

- $D(x)$ is the discriminator's probability that real data xxx is real. $G(z)$ is the generator's output given input noise z. $pdtxlog, pz$ are the data and noise distributions, respectively.

- **Generator Loss Function (G):**

The generator strives to produce data that the discriminator classifies as real. Its loss function is:

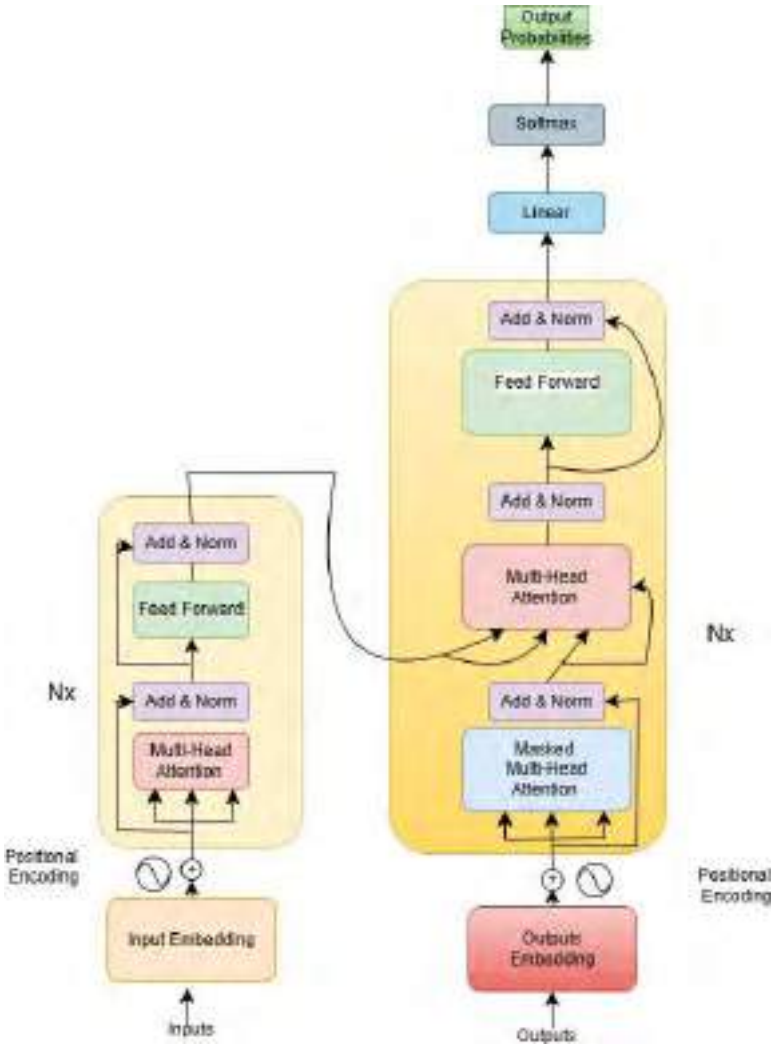
$$\mathcal{LG} = -Ez \sim pz(z)[log(D(G(z)))]\mathcal{LG} = -Ez \sim pzzlog\ \ DGz$$

This dynamic and competitive training mechanism enables GANs to learn complex data distributions, facilitating the generation of highly realistic synthetic data across various domains, including image synthesis, video generation, and data augmentation.

4. **Evaluation Metrics For GAN**:
   - **Inception Score (IS)**: Assesses the quality and diversity of generated images based on a pre-trained classifier's confidence.
   - **Fréchet Inception Distance (FID)**: Measures the similarity between the distributions of real and generated images, providing a quantitative evaluation of GAN performance [Creswell et al., 2017; Wang et al., 2017].
5. **Strategy Generation in Multi-Agent Systems With GAN:**
   - **Adversarial Training for Strategy Development**: In multi-agent systems, agents often need to develop strategies that are robust against adversarial behaviours. By modelling the interaction between agents as a GAN framework, where the generator proposes strategies and the discriminator evaluates their effectiveness against potential adversarial responses, agents can iteratively improve their strategies. This approach allows for the development of strategies that are resilient to various adversarial tactics[Koshiyama et al., 2019; Busoniu et al., 2008].

1.8.12 Temporal Fusion Transformer

Figure 20: Transformer Architecture



The Temporal Fusion Transformer (TFT) is a neural network architecture developed to enhance multi-horizon time series forecasting by integrating high predictive performance with interpretability. It adeptly manages diverse input types, including static covariates, known future inputs, and historical time series data, making it versatile for complex forecasting scenarios [Lim et al., 2019].

Key Components of TFT:

1. **Variable Selection Networks:** These networks dynamically select relevant features at each time step, ensuring that the model focuses on the most pertinent information for accurate forecasting.

2. **Gated Residual Networks (GRNs):** GRNs capture intricate relationships between variables by employing gating mechanisms and residual connections, which help in modelling non-linear dependencies effectively.
3. **LSTM-Based Local Processing:** Long Short-Term Memory (LSTM) layers are utilized to capture short-term temporal dependencies, effectively managing local sequential patterns in the data.
4. **Interpretable Self-Attention Layers:** These layers are designed to learn long-term dependencies by assigning varying levels of importance to different time steps, enhancing the model's ability to focus on critical periods within the time series.
5. **Temporal Fusion Decoder:** This component integrates information from both past and future inputs to generate coherent and accurate forecasts, effectively combining insights from various temporal contexts.

Handling Long-Term Dependencies:

TFT addresses long-term dependencies through its interpretable self-attention layers, which allow the model to weigh the significance of different time steps dynamically. This mechanism enables TFT to capture and utilize patterns over extended periods, improving its forecasting accuracy for long-term trends.

Architecture and Strengths:

The architecture of TFT is a synergistic blend of recurrent and attention mechanisms. The LSTM layers handle local temporal processing, capturing short-term dependencies, while the self-attention layers focus on learning long-term relationships within the data. This combination allows TFT to model complex temporal dynamics effectively. Additionally, the inclusion of variable selection networks and GRNs enhances the model's ability to identify and focus on relevant features, contributing to its robustness and interpretability.

Improvements over Traditional Models (LSTMs and GRUs):

Traditional models like LSTMs and Gated Recurrent Units (GRUs) are proficient in capturing sequential dependencies but often struggle with long-term relationships and lack inherent mechanisms for feature selection and interpretability. TFT surpasses these limitations by incorporating self-attention mechanisms that effectively manage long-term dependencies and by providing insights into feature importance through its variable selection

networks. This results in a more transparent and accurate forecasting model compared to traditional approaches[Lim et al., 2019].

2. Literature Review:

2.1 Introduction

Researchers, scholars, academicians, and practitioners continuously seek new insights into financial markets, often by exploring existing studies and frameworks. In the context of options trading, a substantial body of literature has examined various facets of derivative strategies—ranging from traditional option spreads to advanced, technology-driven techniques. However, while options constitute a significant portion of trading volume in many markets, including India, much of the extant research has focused more on equities or broader portfolio optimization than on the complexities inherent in options trading.

Against this backdrop, the present study addresses a critical gap by exploring how advanced AI methodologies—specifically, Agentic AI and Deep Reinforcement Learning (DRL)—can enhance the performance and consistency of options trading strategies. To achieve this, we build on prior works that have investigated:

1. Studies detailing the design, construction, and performance of various option spreads, including traditional and more innovative approaches.
2. Research focusing on risk-adjusted returns, position sizing, and the comparative effectiveness of spread strategies versus naked options.
3. Early and contemporary works employing machine learning and reinforcement learning for predictive modelling, algorithmic execution, and market regime analysis.
4. Foundational and applied research on collaborative agent systems in finance, emphasizing how specialized agents can coordinate to handle tasks such as strategy generation, market forecasting, and real-time decision-making.

By examining these diverse strands of literature, the current study aims to integrate and extend existing knowledge. Our goal is not to reinvent traditional spread strategies; rather, we adopt popular trading approaches and enrich them with AI-driven tools that address key decision-

making factors—such as market direction, volatility, and timing—while systematically managing risk.

This literature review thus serves two primary purposes:

- **Identifying Knowledge Gaps:** Pinpointing where conventional methods have fallen short, especially regarding adaptability, autonomy, and the scalability of options trading frameworks.
- **Providing Empirical Support:** Gathering evidence to validate the proposed multi-agent and DRL-driven systems, showing how they might outperform conventional benchmarks and index-based strategies.

In the sections that follow, we synthesize relevant studies from academic journals, working papers, and industry analyses. These works collectively provide the foundation for understanding how advanced AI can be harnessed to address longstanding challenges in options trading, particularly in the Indian market.

2.1.1 Reinforcement Learning in Financial Trading

Wen Wen (2021) introduced the Options Trading Reinforcement Learning (OTRL) framework, leveraging the underlying asset's trading data to train RL models for options trading. They employ candlestick data across various time intervals and incorporate a protective closing strategy to mitigate substantial losses. Their findings indicate that the Proximal Policy Optimization (PPO) algorithm, when combined with the protective closing strategy, yields the most stable returns. Additionally, Deep Q-Networks (DQN) and Soft Actor-Critic (SAC) models demonstrate performance surpassing the traditional buy-and-hold strategy.

Yang, B., Liang, T., Xiong, J. and Zhong, C. (2022) introduces DRL-UTrans, an end-to-end model that combines deep reinforcement learning with Transformer and U-Net architectures to enhance stock trading strategies. This integration enables the model to effectively capture complex market patterns and adapt to dynamic conditions, leading to improved trading performance. The study demonstrates that DRL-UTrans outperforms existing methods, achieving a cumulative return of 1124.23% on the IXIC dataset.

Khobragade, S.D. and Kumbhar, S.S. (2025) presents ProfitPulse, an investment strategy utilizing RL to maximize total wealth. The study underscores the potential of RL in developing robust trading strategies that adapt to dynamic market conditions, though specific methodologies and results are not detailed in the provided information.

Moody, J. and Saffell, M. (2001) pioneer the application of direct reinforcement learning in trading, introducing a framework where trading decisions are directly optimized through RL without relying on explicit predictive models. Their approach emphasizes the adaptability of RL in real-time trading environments.

Tan, Z., Quek, C. and Cheng, P.Y.K. (2011) combine Adaptive Neuro-Fuzzy Inference Systems (ANFIS) with RL to model cyclical patterns in stock trading. Their methodology captures market cycles, enhancing trading decisions. However, the complexity of integrating ANFIS with RL and the potential for overfitting in volatile markets present areas for further research.

Avramelou et al. (2024) propose a novel approach that integrates deep reinforcement learning (DRL) with multi-modal data sources, including news articles and social media, to enhance financial trading strategies. They address the challenge of effectively combining diverse online data to improve trading performance, providing valuable insights for developing multi-agent frameworks that leverage reinforcement learning for strategy selection and risk management.

Huang et al. (2023) introduce a multi-agent reinforcement learning (MARL) framework that combines traditional financial trading strategies with the TimesNet model to optimize trading decisions. They present two novel MARL methods, CPPI-MADDPG and TIPP-MADDPG, tailored for strategic trading in quantitative markets, offering insights into integrating MARL with existing trading strategies within multi-agent systems.

Shavandi (2023) presents a framework that integrates multi-agent deep reinforcement learning (MADRL) with algorithmic trading, focusing on enhancing trading performance through collaborative agent interactions. The study emphasizes the importance of cooperation among agents in dynamic financial environments, providing a foundation for developing sophisticated trading strategies within multi-agent systems.

Liu et al. (2023) explore the use of synthetic data augmentation techniques to enhance deep reinforcement learning models in financial trading. They demonstrate that augmenting training data with synthetic samples can improve model robustness and performance, offering valuable insights for developing resilient trading strategies within multi-agent frameworks.

An et al. (2023) review the challenges and opportunities of applying deep reinforcement learning to quantitative trading. They discuss issues such as data quality, model interpretability, and computational complexity, providing a comprehensive overview that informs the development of effective trading strategies within multi-agent systems.

Taghian (2023) investigates the application of deep reinforcement learning to learn asset-specific trading rules, aiming to tailor trading strategies to individual financial instruments. The study highlights the potential of DRL in capturing unique asset characteristics, contributing to personalized trading strategies within multi-agent frameworks.

Sun et al. (2023) introduce TradeMaster, a comprehensive quantitative trading platform that utilizes reinforcement learning to optimize trading strategies. They demonstrate how integrating RL into trading systems can enhance decision-making processes, offering insights into the practical application of reinforcement learning in multi-agent trading environments.

Kabbani, T. and Duman, E. (2022) formulated the trading problem as a Partially Observed Markov Decision Process and applied the Twin Delayed Deep Deterministic Policy Gradient algorithm. The model achieved a Sharpe Ratio of 2.68 on unseen test data, indicating a favorable risk-adjusted return.

2.1.2 Reinforcement Learning Agents with Deep Neural Networks in Options Trading

Wen Wen (2021) introduces the Options Trading Reinforcement Learning (OTRL) framework, which utilizes the underlying asset data of options to train RL models. The research emphasizes the unique characteristics of options, such as the multitude of contracts per underlying asset and their distinct price behaviors. The authors employ candlestick data across various time intervals and incorporate a protective closing strategy to mitigate significant losses. Experimental results indicate that the Proximal Policy Optimization (PPO) algorithm, when combined with the protective closing strategy, yields the most stable returns. Additionally, Deep Q-Networks (DQN) and Soft Actor-Critic (SAC) models demonstrate superior performance compared to traditional buy-and-hold strategies in options trading.

Yang, B., Liang, T., Xiong, J. and Zhong, C. (2022) explores the application of deep learning in enhancing trading strategies through improved decision-making and risk management. The authors highlight the potential of DNNs to analyze financial data, forecast market trends, and identify trading opportunities. They address challenges such as data scarcity, model

interpretability, and overfitting, proposing methodologies to overcome these issues. The study demonstrates that integrating DNNs with RL can lead to optimized trading strategies with effective risk control mechanisms.

Peng, X., Zhou, X., Xiao, B. and Wu, Y. (2024) Focused on dynamic hedging of options, this research presents a risk-sensitive RL approach aimed at minimizing tail risks in the profit and loss (P&L) of option sellers. The proposed method learns optimal hedging strategies directly from historical market data without necessitating a parametric model of the underlying asset. Notably, the learned strategies are contract-unified, applicable across various options contracts with differing parameters. Empirical studies reveal that this RL-based hedging strategy achieves significantly lower tail risks and higher mean P&L compared to traditional delta hedging methods.

Xu, M., Lan, Z., Tao, Z., Du, J. and Ye, Z. (2023) introduced QTNet, an adaptive trading model that integrates DRL with imitative learning methodologies. The model was trained using minute-frequency data from live financial markets, demonstrating proficiency in extracting robust market features and adaptability to diverse market conditions.

2.1.3 Options Trading with Reinforcement Learning

Wen Wen (2021) introduced the Options Trading Reinforcement Learning (OTRL) framework, utilizing underlying asset data to train RL models. They employ candlestick data across various time intervals and incorporate a protective closing strategy to mitigate significant losses. Their experiments reveal that Proximal Policy Optimization (PPO) with this strategy yields the most stable returns, while Deep Q-Networks (DQN) and Soft Actor-Critic (SAC) also outperform traditional buy-and-hold approaches.

Wu and Jaimungal (2023) explore robust risk-aware RL to address risks in path-dependent financial derivatives. They apply a policy gradient method optimizing robust risk-aware criteria to hedge barrier options, demonstrating that hedging strategies evolve from risk-averse to risk-seeking as agents adjust their risk preferences. Their robust strategies maintain superior performance even under model misspecification and changing market conditions.

Peng, X., Zhou, X., Xiao, B. and Wu, Y. (2024) presents a risk-sensitive contract-unified RL approach, integrating various risk measures into a single framework for option hedging. They propose a contract-unified objective function that balances risk and return, enhancing the

adaptability of RL agents to diverse risk preferences. Their approach effectively addresses the trade-off between risk and return, providing a more nuanced tool for option hedging.

2.1.4 Options Trading

Black, F. and Scholes, M. (1973) introduced a groundbreaking model for pricing European-style options. Their approach is grounded in the assumption that the underlying asset follows a geometric Brownian motion, incorporating constant volatility and a risk-free interest rate. By constructing a risk-neutral portfolio and applying the principle of no-arbitrage, they derived a differential equation—the Black-Scholes equation—that, under specific boundary conditions, yields a closed-form solution for option prices. This model has become a cornerstone in financial markets, providing a theoretical foundation for option valuation.

Merton, R.C. (1973) Merton expanded upon the Black-Scholes framework by offering a more rigorous mathematical derivation of the option pricing model. He introduced the concept of dynamic replication, emphasizing the construction of a riskless portfolio through continuous hedging. Merton's work laid the groundwork for the risk-neutral valuation approach, which has become a standard in derivative pricing. His contributions also highlighted the importance of stochastic calculus in modeling the random behavior of asset prices.

Cox, Ross, and Rubinstein presented a discrete-time alternative to the continuous-time models of Black-Scholes and Merton. They developed the binomial options pricing model, which approximates the price of options through a recombining binomial tree. This model simplifies the computational complexities associated with continuous models and is particularly useful for pricing American-style options, which can be exercised before expiration. The binomial model converges to the Black-Scholes model as the number of time steps increases, providing a bridge between discrete and continuous approaches.

Cao (2019) explores the interplay between options trading and corporate debt structures. The study suggests that options trading can influence corporate debt decisions by enhancing the informational environment, potentially lowering the cost of debt. Understanding this relationship is essential for developing strategies within multi-agent frameworks that aim to optimize risk management and strategy selection in financial markets.

Zhan and Han (2021) delve into the predictability of option returns by examining various predictive variables. Their findings contribute to understanding the factors influencing option pricing and return patterns. This research is pertinent for developing reinforcement learning

models within multi-agent frameworks, aiming to enhance strategy selection and risk management in options trading.

Ali, Balachandran, and Duong (2020) investigate the impact of options trading on audit pricing. They find that increased options trading activity correlates with higher audit fees, indicating that auditors perceive greater risk and complexity in firms engaged in active options trading. This insight is valuable for designing multi-agent systems that account for audit pricing strategies within the broader context of financial risk management.

Tomé (2018) presents models for pricing spread options in energy markets, focusing on the unique characteristics and challenges of energy commodities. The research provides methodologies for accurately valuing options that depend on the price difference between two energy assets. Understanding these models is essential for developing multi-agent systems that effectively manage risks and optimize strategies in energy trading markets.

2.1.5 Binomial Options Pricing Model

Dietmar (2006) introduced the binomial model, providing a discrete-time framework for option valuation. They demonstrated that by constructing a riskless portfolio comprising the option and the underlying asset, one could derive a simple formula to determine the option's price at each node within the binomial tree. This methodology not only simplified the option pricing process but also enhanced its computational efficiency.

Glasserman, P. (2004) extended the traditional binomial model by incorporating stochastic volatility. They proposed a modified binomial tree that adjusted volatility dynamically at each node, capturing the asset's price evolution more accurately. This advancement allowed for a more precise valuation of options, especially those sensitive to volatility fluctuations.

2.1.6 Monte Carlo Simulation in Financial Engineering

Boyle, P.P. (1977) Glasserman's work provides a comprehensive exploration of Monte Carlo methods tailored for financial applications. He emphasizes the efficiency of these methods in evaluating complex integrals encountered in derivative pricing. The book delves into various variance reduction techniques, such as antithetic variates, control variates, and importance sampling, to enhance simulation accuracy and efficiency. Glasserman also addresses the estimation of sensitivities, or "Greeks," highlighting the challenges of numerical differentiation

and the potential for increased simulation errors. He proposes alternative approaches to mitigate these issues, underscoring the importance of accurate sensitivity analysis in risk management.
Boyle's pioneering paper by Jäckel, P. (2002) introduces the application of Monte Carlo simulation to option pricing, marking a significant departure from traditional analytical methods. He demonstrates how Monte Carlo methods can effectively price European options by simulating numerous price paths of the underlying asset and averaging the discounted payoffs. This approach is particularly advantageous for options with complex features or path-dependent characteristics, where closed-form solutions are not feasible. Boyle's work laid the foundation for subsequent developments in simulation-based option pricing, influencing later research in quasi-Monte Carlo methods and high-dimensional integration techniques.
Ferber, J. (1999), Jäckel's book offers an in-depth analysis of Monte Carlo simulation techniques within the financial sector. He explores the theoretical underpinnings of these methods and their practical applications in pricing complex derivatives. The book discusses various aspects of simulation, including random number generation, path simulation, and the implementation of variance reduction techniques. Jäckel also addresses the computational challenges associated with Monte Carlo methods, such as the high variance in estimates and the substantial computational resources required. He provides insights into mitigating these challenges and discusses the trade-offs between simulation accuracy and computational feasibility.
Kozlova (2020) introduces Simulation Decomposition (SimDec), a method that enhances Monte Carlo simulations by visually analyzing the relationships between input variables and model outputs. SimDec facilitates uncertainty and sensitivity analysis, enabling a deeper understanding of model behavior across various disciplines, including finance. By decomposing simulations, researchers can identify influential factors affecting trading strategies, thereby informing the development of multi-agent frameworks and reinforcement learning models for improved strategy selection and risk management.
Becker et al. (2023) explore the integration of stochastic gradient descent with Monte Carlo simulations to efficiently learn random variables in the context of financial derivative pricing. Their approach combines Monte Carlo algorithms with machine learning techniques, enhancing the accuracy and efficiency of pricing complex financial instruments. This methodology is particularly relevant for developing multi-agent systems that require precise modeling of

financial variables, thereby improving strategy selection and risk management in trading applications.

2.1.7 Multi-Agent Systems

Ferber, J. (1999) offers a comprehensive introduction to MAS, delineating the concept of agents as active entities capable of perceiving and acting upon their environment. He introduces classifications of MAS into reactive and cognitive systems, emphasizing the distinction between agents that operate based on stimulus-response mechanisms without internal representations and those that possess internal models enabling complex behaviors such as planning and learning. This classification aids in understanding the varying complexities and functionalities within MAS.

Wooldridge, M. and Jennings, N.R. (1995) provide a critical analysis of intelligent agents, distinguishing them from traditional software entities by highlighting attributes such as autonomy, social ability, reactivity, and proactivity. They discuss the theoretical foundations of agent-based systems and explore practical considerations in their implementation, offering a balanced perspective that bridges conceptual models with real-world applications.

Shoham, Y. and Leyton-Brown, K. (2008) delve into the algorithmic and game-theoretic aspects of MAS, providing a rigorous analysis of strategic interactions among rational agents. They explore logical frameworks that underpin agent behaviors, offering insights into the computational complexities and decision-making processes within multi-agent environments. This work is pivotal for understanding the mathematical and strategic dimensions of MAS.

Busoniu, L., Babuska, R. and De Schutter, B. (2008) presented an extensive survey on multi-agent reinforcement learning (MARL), highlighting the challenges and methodologies associated with learning in environments where multiple agents interact. They address issues such as non-stationarity and credit assignment, providing a synthesis of existing approaches and identifying avenues for future research in MARL.

Vasilenko, V. and Kasyanov, I. (2019) focuses on the modeling, control, and programming aspects of MAS, offering practical insights into the development and management of multi-agent systems. They discuss various modeling techniques, control strategies, and programming paradigms, providing a resource for practitioners and researchers involved in the design and implementation of MAS.

Bryzgalova and Pavlova (2022) analyze the surge in retail options trading and the dominance of three major wholesalers in this domain. They observe that these wholesalers significantly influence market dynamics, affecting liquidity and pricing structures. This study provides insights into how wholesaler activities impact retail investors, which is crucial for understanding market behavior in the context of multi-agent systems and reinforcement learning.

Guo et al. (2022) survey advancements and challenges in integrating large language models with multi-agent systems. They discuss how language models can enhance agent communication and collaboration, identifying open research questions in this emerging field. Incorporating such models can significantly improve the adaptability and effectiveness of multi-agent systems in dynamic trading environments

Sun et al. (2023) introduce TradeMaster, a comprehensive quantitative trading platform that utilizes reinforcement learning to optimize trading strategies. They demonstrate how integrating RL into trading systems can enhance decision-making processes, offering insights into the practical application of reinforcement learning in multi-agent trading environments.

Guo et al. (2022) explore the integration of large language models (LLMs) with multi-agent systems, highlighting advancements and challenges in this area. They discuss how LLMs can enhance agent communication and collaboration, leading to more sophisticated and human-like interactions. This integration is particularly relevant for trading applications, where agents can process and interpret complex textual information, such as financial news and reports, to inform trading decisions. Understanding these developments aids in designing multi-agent systems capable of advanced strategy selection and risk management in dynamic trading environments.

Fatemi, S., Hu, Y., Li, X., Wang, Z., & Li, J. (2024) introduced FinVision, a multi-modal, multi-agent system tailored for financial trading tasks. The framework comprises specialized agents adept at processing diverse financial data types, including textual news reports, candlestick charts, and trading signal charts. A notable feature is its reflection module, which analyzes historical trading signals and outcomes to enhance future decision-making. The study concludes that integrating these components significantly bolsters the system's predictive capabilities, with ablation studies highlighting the critical role of the visual reflection module in improving decision-making.

2.1.8 Markov Decision Processes

Howard, R.A. (1960) laid a foundational in the study of MDPs, introducing dynamic programming methods to solve decision processes under uncertainty. He formulated the policy iteration algorithm, which iteratively evaluates and improves policies to find optimal solutions. This approach is particularly relevant in areas like trading strategy selection, where sequential decisions must be optimized to maximize returns.

Puterman, M.L. (1994) focused on comprehensive text expanding upon the theoretical underpinnings of MDPs and their applications. He provides in-depth coverage of solution methods, including value iteration and policy iteration, and discusses their computational complexities. Understanding these algorithms is crucial for developing reinforcement learning models that can effectively learn optimal trading strategies through simulation and real-time data analysis.

Sutton, R.S. and Barto, A.G. (1998) , in there book focused on bridgeing the gap between MDPs and reinforcement learning, presenting algorithms that allow agents to learn optimal behaviors through trial and error. Techniques such as Q-learning and temporal-difference learning are explored, offering insights into how agents can learn effective trading strategies without explicit programming. These methods are instrumental in developing adaptive trading systems that respond to market dynamics.

Policy Gradient Methods

Sutton, R.S. and Barto, A.G. (2000), in their book mainly focused on:

- **Introduction of Policy Gradient Methods:** The authors present PGMs as a means to directly adjust the parameters of a policy function to maximize expected returns in RL tasks.
- **Function Approximation:** They explore the integration of function approximation techniques, such as neural networks, to represent complex policy functions, addressing the scalability issues of tabular methods.
- **The REINFORCE Algorithm:** A Monte Carlo-based estimator is introduced for policy gradients, providing a method to update policy parameters based on sampled trajectories.

Klimov, O. (2017) discusses in their paper about:

- **Proximal Policy Optimization (PPO):** This work proposes PPO, an algorithm that balances the benefits of Trust Region Policy Optimization (TRPO) with improved simplicity and computational efficiency.
- **Clipped Surrogate Objective:** PPO utilizes a clipped objective function to prevent large policy updates, enhancing training stability and performance.
- **Generalized Advantage Estimation (GAE):** The paper introduces GAE, a method for reducing variance in policy gradient estimates, leading to more reliable learning.

2.1.9 Temporal Difference (TD) Learning

Sutton, R.S. (1988) introduces TD Learning as a class of incremental learning procedures tailored for prediction tasks. Unlike traditional methods that adjust predictions based on the final outcome, TD Learning updates predictions based on successive, temporally ordered predictions. This approach allows for continuous learning and refinement of predictions as new data becomes available, facilitating more dynamic and responsive modeling.

Bradtke, S.J. and Barto, A.G. (1996) building upon Sutton's foundational work, explored linear least-squares algorithms within the TD Learning framework. They demonstrate that these algorithms enable systems to predict the cumulative reward expected over time. Their research provides a mathematical foundation for implementing TD Learning in environments where outcomes are uncertain and sequential, offering insights into the convergence properties and efficiency of these algorithms.

2.1.10 Generative Adversarial Networks

Bengio, Y. (2014), Goodfellow et al. laid the foundational framework for GANs, demonstrating their capability to generate data indistinguishable from real-world samples. They formalized the GAN model as a two-player game, providing a theoretical basis for the adversarial process between the generator and discriminator.

Ferdowsi, A. and Saad, W. (2020) addressed the challenges of training GANs across distributed datasets without sharing sensitive data. They proposed the Brainstorming GAN (BGAN) architecture, enabling multiple agents to collaboratively generate high-quality data samples while maintaining data privacy. This approach reduces communication overhead and enhances the scalability of GANs in multi-agent environments.

Treleaven, P. (2019), explored the application of Conditional GANs (cGANs) in the financial sector, focusing on the fine-tuning and combination of trading strategies. They demonstrated that

cGANs could generate diverse and realistic financial time-series data, aiding in the calibration and aggregation of trading strategies, and potentially improving predictive performance in financial modeling.

Creswell et al. (2018) provide a comprehensive review of Generative Adversarial Networks (GANs), highlighting their capability to learn deep representations without extensive annotated training data. The paper discusses various applications such as image synthesis, semantic image editing, style transfer, image super-resolution, and classification. It also addresses different training methods, architectures, and the theoretical challenges associated with GANs.

Wang et al. (2017) explore the concept of GANs, emphasizing their foundation in game theory with a generator and discriminator engaged in a two-player zero-sum game. The paper provides insights into the historical development of generative algorithms, the mechanism of GANs, their fundamental structures, and theoretical analyses. It also discusses potential applications and future research directions in the field.

Gui et al. (2020) present an in-depth review of GANs, focusing on their algorithms, theoretical foundations, and diverse applications. The paper categorizes various GAN architectures, discusses training techniques, and examines their use in areas like image generation, video prediction, and semi-supervised learning. It also highlights challenges and future research avenues to enhance GAN performance and applicability.

2.1.11 Temporal Fusion Transformer

Pfister, T. (2019, in their paper represented a significant advancement in the domain of multi-horizon time series forecasting. This architecture adeptly combines high-performance forecasting with interpretable insights into temporal dynamics, addressing the complexities inherent in time series data that encompass static covariates, known future inputs, and exogenous time series observed only historically.

Key Contributions of the TFT Model:

1. **Hybrid Architecture:** The TFT integrates recurrent layers for capturing local temporal relationships and self-attention mechanisms to model long-term dependencies. This dual approach enables the model to effectively learn temporal patterns at varying scales.
2. **Feature Selection and Gating Mechanisms:** To enhance interpretability and performance, the TFT employs specialized components for dynamic feature selection and

gating layers that suppress irrelevant inputs. This design allows the model to focus on the most pertinent information, improving forecasting accuracy and providing clarity on the decision-making process.

3. **Demonstrated Superiority:** Through empirical evaluations across diverse real-world datasets, the TFT has shown significant performance improvements over existing benchmarks. Notably, it has outperformed traditional methods in various applications, including retail demand forecasting and healthcare predictive analytics.

Li, Tan, Zhang, Miao, and He (2023) present a probabilistic forecasting method for mid-term hourly load time series by enhancing the Temporal Fusion Transformer (TFT) model. They address the challenge of balancing long-term temporal dependence learning with model complexity by reconstructing univariate time series into multiple day-to-day series at different hour-points. This approach utilizes the hour-point as a static covariable to distinguish differences effectively. The improved TFT model replaces the Long Short-Term Memory (LSTM) unit with a Gated Recurrent Unit (GRU) to enhance long-term dependence learning efficiency. Additionally, incorporating quantile constraints and prediction interval (PI) penalty terms into the quantile loss function helps prevent quantile crossover and construct more compact PIs. This methodology holds significant potential for enhancing strategy selection and risk management in multi-agent trading systems by providing more accurate and reliable load forecasting.

Koya and Roy (2023) investigate the efficacy of combining attention mechanisms with recurrence in streamflow prediction by implementing the Temporal Fusion Transformer (TFT) architecture, which integrates both features. Their study compares the performance of LSTM, Transformer, and TFT models across 2,610 globally distributed catchments using the Caravan dataset. The results demonstrate that TFT surpasses both LSTM and Transformer models in streamflow prediction accuracy. Moreover, as an explainable AI method, TFT offers valuable insights into streamflow generation processes. This research highlights the potential of combining attention with recurrence, providing a promising approach for improving predictive performance in hydrological modeling.

2.1.12 Policy Optimization Methods in Reinforcement Learning

Klimov, O. (2017) introduced PPO, an algorithm that alternates between sampling data through interaction with the environment and optimizing a surrogate objective function using stochastic

gradient ascent. PPO simplifies the trust region policy optimization (TRPO) by using first-order optimization and clipping the policy gradient, enhancing data efficiency and robustness. Empirical results demonstrate PPO's superior performance across various tasks, including robotic locomotion and Atari game playing.

Levine, S. (2018), focused on SAC as presenting as an off-policy actor-critic algorithm that incorporates entropy regularization into the objective function, promoting both exploration and exploitation. By maximizing a trade-off between expected return and entropy, SAC achieves state-of-the-art performance in continuous control tasks. The algorithm's automatic entropy tuning and sample efficiency make it particularly effective for complex environments.

Haarnoja et al. (2018) introduce the Soft Actor-Critic (SAC) algorithm, an off-policy actor-critic method rooted in the maximum entropy reinforcement learning framework. SAC aims to enhance both the expected return and policy entropy, promoting more stochastic and exploratory actions. The authors address challenges such as high sample complexity and sensitivity to hyperparameter settings, proposing solutions that improve training efficiency and stability. Their extensive evaluations demonstrate that SAC surpasses previous methods in sample efficiency and performance across various benchmark and real-world tasks, including robotic locomotion and manipulation.

Christodoulou (2019) extends the Soft Actor-Critic (SAC) algorithm, originally designed for continuous action spaces, to accommodate discrete action settings. By deriving an alternative version tailored for discrete actions, the study demonstrates that this adaptation competes effectively with tuned model-free methods on Atari games, even without extensive hyperparameter optimization.

Ding et al. (2021) introduce the Averaged Soft Actor-Critic (Averaged-SAC) algorithm, which utilizes the average of multiple previously learned state values to compute the soft Q-value. This approach addresses overestimation issues inherent in soft Q-learning, leading to enhanced stability and performance in deep reinforcement learning applications.

2.1.13 Regime Changes in Stock Trading

Ang, A. and Timmermann, A. (2012) focused on Regime changes in financial markets refer to abrupt shifts in market behavior, characterized by persistent changes in asset return means, volatilities, and correlations. Understanding these shifts is crucial for developing effective

trading strategies, particularly in options trading, where volatility plays a significant role. This literature review examines the seminal work by Ang and Timmermann (2012) on regime changes and their implications for financial markets, with a focus on their relevance to trading options, strategy selection, and risk management within a multi-agent framework utilizing reinforcement learning.

2.1.14 Q-Learning Algorithms in Reinforcement Learning for Options Trading

Wen Wen (2021) introduced the Options Trading Reinforcement Learning (OTRL) framework, utilizing underlying asset data to train RL models. The study employs candlestick data across various time intervals and incorporates a protective closing strategy to mitigate substantial losses. Empirical results indicate that the Proximal Policy Optimization (PPO) algorithm, when combined with the protective closing strategy, achieves the most stable high returns. Additionally, DQN and Soft Actor-Critic (SAC) algorithms outperform the buy-and-hold benchmark in options trading scenarios.

Brim, A. (2019) applied DQN to a stock market pairs trading strategy, aiming to exploit the relative movements between two correlated assets. The study highlights the potential of DQN in developing profitable trading strategies, though specific performance metrics and comparisons are not detailed in the provided information.

Ramos-Díaz, E. (2024) along with co-authors explored the application of DDQN in algorithmic trading, focusing on learning optimal trading policies to maximize returns while managing risk. The study integrates sentiment analysis to enhance trading decisions, though detailed findings and comparisons with other algorithms are not specified in the available summary.

2.1.15 Backtesting in Reinforcement Learning-Based Options Trading

Tan, Roberts, and Zohren (2024) introduced a data-driven machine learning algorithm for options trading that eliminates the need for predefined market dynamics or pricing models. Their approach involves directly learning complex mappings from market data to optimal trading signals. Backtesting over a decade of S&P 100 equity option contracts reveals that their deep learning models significantly outperform traditional rules-based strategies in terms of risk-adjusted returns. Additionally, incorporating turnover regularization further enhances performance, although gains are diminished under high transaction costs.

Wen, Yuan, and Yang (2021) proposed the Options Trading Reinforcement Learning (OTRL) framework, leveraging underlying asset data to train reinforcement learning models. They utilize candlestick data across various time intervals and implement a protective closing strategy to mitigate substantial losses. Their experiments indicate that the Proximal Policy Optimization (PPO) algorithm, when combined with the protective closing strategy, achieves the most stable high returns. Both Deep Q-Network (DQN) and Soft Actor-Critic (SAC) models also surpass the buy-and-hold benchmark in options trading scenarios.

Gort et al. (2023) addressed the prevalent issue of backtest overfitting in deep reinforcement learning models applied to cryptocurrency trading. They emphasize the importance of robust backtesting methodologies to ensure the reliability of trading strategies. Their work highlights practical approaches to mitigate overfitting, thereby enhancing the generalizability of models to unseen market conditions.

Jin, X. (2023) developed an algorithmic trading system that integrates risk-return considerations within a reinforcement learning framework. By balancing potential returns against associated risks, the proposed system aims to optimize trading decisions. The study underscores the efficacy of reinforcement learning in adapting to dynamic market environments while maintaining a focus on risk-adjusted performance.

2.1.16 ARIMA Models in Time Series Analysis

Hillmer, S.C. and Tiao, G.C. (1982) introduced a methodology for seasonal adjustment using ARIMA models, emphasizing the importance of distinguishing between seasonal and non-seasonal components in time series data. They proposed decomposing a time series into its underlying components—trend, seasonal, and irregular—by identifying appropriate ARIMA models for each. This approach facilitates more accurate forecasting by isolating and modeling the distinct structures within the data.

Box, G.E.P. and Jenkins, G.M. (1976) laid the foundation for the systematic approach to time series modeling known as the Box-Jenkins methodology. Their iterative three-stage process—model identification, parameter estimation, and diagnostic checking—provides a structured framework for developing ARIMA models. They emphasized the importance of model simplicity and parsimony, advocating for the use of the fewest parameters necessary to adequately describe the data-generating process.

Commandeur, J.J.F. and Koopman, S.J. (2007) explored state space models as an alternative to traditional ARIMA models, particularly addressing challenges associated with non-stationary economic and social time series. They argued that real-world series often exhibit non-stationarity despite differencing, suggesting that state space models offer a more flexible framework for modeling such data. Their work highlights the potential of state space approaches to capture complex temporal dynamics without the strict stationarity assumptions inherent in ARIMA models.

2.1.17 Agentic AI

Mingchen Zhuge, Changsheng Zhao (2024) introduced a framework in which agentic systems evaluate other agents' performance. By deploying agents as judges, the research enhances the evaluation of agent-based task solutions, especially in complex environments such as code generation. The authors propose using agentic feedback loops, facilitated by Large Language Models (LLMs), to offer intermediate feedback during task execution. This novel approach leads to more accurate evaluations, outperforming traditional human evaluation metrics.

Kamer Ali Yuksel, Hassan Sawaf (2024) proposed an agentic AI system that autonomously optimizes solutions using iterative refinement and LLM-driven feedback loops. This system's iterative nature involves multiple agents working in concert, with each agent playing a specific role in improving model performance through constant feedback and hypothesis generation. The research showcases how agents can collaborate in an adaptive manner to optimize decision-making tasks, reducing the need for human intervention.

Shen Gao, Yuntao Wen (2024) explored the use of LLM-based agents for simulating the behavior of financial market participants. By integrating agent-based models with LLMs, the authors create a more realistic simulation of market dynamics that accounts for various factors like macroeconomic variables and market sentiment. The research aims to study the interaction between agents representing different types of investors, each driven by unique strategies.

Treleaven, P. (2019) explored the application of Generative Adversarial Networks (GANs) in the fine-tuning and combination of financial trading strategies. By using GANs to generate synthetic financial data, the authors enable better calibration of trading algorithms, particularly in dealing with market scenarios that are underrepresented in historical data. Additionally, the paper

examines the use of GANs to combine multiple strategies, thus creating a robust trading framework that can adapt to diverse market conditions.

Okpala (2025) presented an approach where agentic AI systems are organized into "crews," with each crew responsible for a specific financial task. These crews include both modeling and model risk management teams, which collaborate to ensure compliance, evaluate models, and optimize financial decision-making processes. The research emphasizes how AI agents can collaboratively improve task execution and mitigate model risk.

M., & Zhang, Y. (2024) introduced StockAgent, a system that uses LLM-based agents to simulate stock trading within dynamic, real-world environments. The paper demonstrates how StockAgent can simulate stock market dynamics by using agents to represent various trading strategies, learning and adapting to real-world market conditions. The authors show that StockAgent can evaluate different trading strategies and generate insights for improving trading models.

Acharya, Kuppan, and Divya (2023) provide an extensive survey on Agentic AI, focusing on its foundational concepts, unique characteristics, and core methodologies. They delve into the capabilities of Agentic AIs—autonomous systems capable of undertaking complex actions with minimal supervision—and explore the challenges in aligning these systems with user preferences and societal norms. This alignment is crucial for ensuring that Agentic AIs operate safely and ethically, particularly as they become more integrated into various aspects of daily life. The paper serves as a comprehensive introduction for researchers, developers, and policymakers interested in understanding and engaging with the transformative potential of Agentic AI.

Clatterbuck, Castro, and Muñoz Morán (2024) address the critical issue of risk alignment in Agentic AI systems. They emphasize that an agent's risk attitudes significantly influence its decision-making under uncertainty. For instance, a risk-averse agent would prefer actions with lower variance in possible outcomes, even if it means foregoing potentially higher rewards. The paper raises essential questions about designing AI systems that align with users' risk preferences and the broader ethical considerations of allowing AIs to make risky decisions on behalf of individuals. Ensuring appropriate risk alignment is vital for user satisfaction, trust, and the safe integration of Agentic AIs into society.

Chawla et al. (2024) explore the foundational elements of Agentic AI frameworks, identifying four key pillars: tool use, reflection, planning, and multi-agent collaboration (MAC). Tool use

enables AI systems to access external resources, such as search engines, to enhance accuracy. Reflection allows for self-correction and iterative feedback, improving decision-making processes. Planning involves structuring tasks methodically to achieve complex goals efficiently. MAC facilitates collaboration among multiple AI agents on specific subtasks, leveraging diverse expertise for optimal outcomes. Understanding these components is essential for developing sophisticated AI applications that are both effective and aligned with business objectives.

2.1.18 Generic Agentic AI

Wooldridge, M. (1995) provided a comprehensive exploration of intelligent agents, discussing their theoretical foundations and practical applications. He outlines the characteristics that define intelligent agents, such as autonomy, social ability, reactivity, and proactivity. The paper delves into the architecture of these agents and their role in multi-agent systems, offering insights into their design and implementation.

Castelfranchi, C. (1998) focused on the social dimensions of AI agents, proposing models to simulate social actions and interactions. The paper emphasizes the importance of understanding social contexts and norms to enhance the effectiveness of AI agents in human-centric environments. It presents frameworks for modeling intentions, commitments, and trust among agents, contributing to the development of socially aware AI systems.

M., & Zhang, Y. (2024) introduced StockAgent, a system that uses LLM-based agents to simulate stock trading within dynamic, real-world environments. The paper demonstrates how StockAgent can simulate stock market dynamics by using agents to represent various trading strategies, learning and adapting to real-world market conditions. The authors show that StockAgent can evaluate different trading strategies and generate insights for improving trading models.

2.1.19 Stock Trading

Kabbani and Duman (2022) present a Deep Reinforcement Learning (DRL) model designed to automate trading in the stock market. They formulate the trading problem as a Partially Observed Markov Decision Process (POMDP), incorporating market constraints such as liquidity and

transaction costs. Utilizing the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, their model achieves a Sharpe Ratio of 2.68 on unseen test data, demonstrating the efficacy of DRL in financial decision-making.

Banik et al. (2022) develop a decision support system for swing trading using Long Short-Term Memory (LSTM) networks. The system analyzes historical stock data to predict future stock values, aiding traders in making informed decisions. The study addresses the challenges posed by the volatile nature of the stock market, demonstrating the potential of LSTM networks in financial forecasting.

Shah, Vaidya, and Shah (2022) provide a comprehensive review of hybrid deep learning approaches for stock prediction. They analyze various models that combine multiple learning techniques to enhance prediction accuracy, offering insights into their applications and effectiveness in financial forecasting.

Cohen (2022) explores the application of advanced artificial intelligence methodologies, including machine learning and deep learning, in algorithmic trading and financial forecasting. The work discusses the integration of AI techniques in developing trading strategies and predicting market trends, highlighting their impact on the financial industry.

2.2 Conclusion of Literature Review

This chapter has provided an extensive overview of the literature on options trading, strategy selection, and risk management through the lenses of Reinforcement Learning (RL) and Multi-Agent Systems (MAS). While global research has largely concentrated on applying RL to trading strategies, studies that integrate these techniques with options trading and MAS remain relatively scarce.

Notable contributions include Wen Wen's (2021) introduction of the Options Trading Reinforcement Learning (OTRL) framework, which employs underlying asset trading data to train RL models tailored for options trading. The research demonstrates that the Proximal Policy Optimization (PPO) algorithm, when coupled with a protective closing strategy, generates stable returns that outperform traditional buy-and-hold approaches. In a similar vein, Yang et al. (2022) advanced the field by integrating deep reinforcement learning with Transformer and U-Net

architectures in their DRL-UTrans model, which achieved an impressive cumulative return of 1124.23% on the IXIC dataset.

In the context of Multi-Agent Systems, Xiao et al. (2024) introduced the TradingAgents framework, inspired by real-world trading firm operations. This system, composed of specialized agents—such as fundamental, sentiment, and technical analysts—demonstrated marked improvements over conventional models, notably in cumulative returns, Sharpe ratio, and maximum drawdown. Additionally, Zhang et al. (2025) developed HedgeAgents, a system that bolsters robustness by incorporating hedging strategies to navigate rapid market fluctuations and downturns.

Further emphasizing the critical role of evaluation, Tan et al. (2024) illustrated the value of backtesting RL-based trading strategies. Their data-driven machine learning algorithm not only outperformed traditional rule-based strategies in risk-adjusted returns but also highlighted potential performance declines under high transaction costs.

Despite these advancements, existing studies often lack conclusive evidence on the comparative effectiveness of different RL and MAS approaches in real-world trading scenarios, frequently focusing on isolated methodologies. To address this gap, the present research systematically compares two distinct approaches for autonomous options trading, aiming to establish a comprehensive framework that provides empirical insights into their relative performance. This study thus contributes to the growing body of literature and offers valuable perspectives to guide future developments in autonomous trading systems.

2.3 Research Gap

On review of the literature, it is evident that while significant advancements have been made in AI-driven trading strategies for equities and portfolio optimization, options trading, despite its substantial volume and complexity, remains relatively underexplored. Specifically, the following critical research gaps have been identified:

1. **Absence of Multi-Agent AI Frameworks for Options Trading:** Existing research lacks comprehensive multi-agent systems designed specifically for options trading. While individual AI techniques like GANs and Transformer models have been applied in various financial contexts, there is a significant gap in the literature regarding the development and evaluation of integrated multi-agent architectures that orchestrate specialized agents for strategy generation, market regime prediction, risk management, and data analysis in the context of options. The literature does not reveal any existing frameworks that combine these diverse AI capabilities into a cohesive, autonomous system for options trading.
2. **Limited Application of Deep Reinforcement Learning (DRL) for Options Strategy Selection:** Although DRL has demonstrated potential in handling complex, dynamic decision-making problems, its application to options trading, particularly for dynamic strategy selection, is notably absent. While DRL has been used in areas like algorithmic trading and portfolio management, the specific challenges of options trading, such as strike selection, volatility management, and complex payoff structures, have not been adequately addressed using DRL-driven strategy selection methodologies. The literature lacks studies that explore the effectiveness of DRL pipelines in learning and executing options strategies in diverse market conditions.
3. **Comparative Analysis of Multi-Agent Systems and DRL Pipelines for Options Trading:** A critical gap exists in the comparative analysis of different AI approaches for options trading. Specifically, there is a lack of studies that directly compare the performance of multi-agent collaborative systems with DRL-driven pipelines. Understanding the relative strengths and weaknesses of these two distinct approaches is essential for determining the most effective AI methodologies for options trading. No research has been found that benchmarks these two advanced AI approaches against each other, particularly in the context of options trading, leaving a critical need for empirical comparison.

Therefore, this study aims to fill these research gaps by proposing and evaluating an autonomous framework for options trading built on Agentic AI. It will develop and compare a multi-agent collaborative system and a DRL-driven pipeline, benchmarking their performance against

established options trading strategies across various market conditions. This research seeks to provide a scalable, adaptable, and empirically validated solution for real-world market environments, contributing to the advancement of AI-based decision-making in options trading.

3. METHODOLOGY

3.1 Introduction

The methodology of this research is structured to develop and evaluate a multi-agent system against Deep Reinforcement Learning for options trading and risk management. This chapter outlines the research design, driven by the primary objectives and the selection of appropriate analytical tools, ensuring a logical arrangement of research activities to reach sound conclusions.

The research design encompasses several critical decisions: the type of data collection, the study period, and the analytical framework. The process begins with formulating the research questions, developing a design to address these questions, executing the design based on collected data, and finally, documenting the findings. This sequence is guided by technical considerations established prior to conducting the research.

In this study, the unit of analysis is the performance of various options trading strategies. The variables influencing strategy performance are operationalized within this chapter, detailing the practical implementation of the strategies to be tested. Specifically, the framework involves multiple specialized agents focusing on tasks such as technical analysis, trend analysis, strategy generation, strategy selection, and risk management. These agents collaborate to adapt to dynamic market conditions, aiming to outperform traditional single-strategy trading methods. The strategies are tested using historical market data, with returns calculated and variables operationalized as described herein. The framework for analysis includes a comparative evaluation against 15 established trading strategies to assess the effectiveness of the multi agent system. This study addresses a practical question of significant interest to the investing and trading communities:

Can a multi-agent, reinforcement learning-based approaches enhance options trading performance and risk management compared to traditional methods?

3.2 Overview of the Research Problem:

Options trading presents a high degree of complexity arising from market volatility, the multi-factor nature of options pricing (e.g., implied volatility, time decay, and interest rates), and the necessity for robust, adaptable strategies. Traditional approaches typically rely on a limited set of predefined strategies and static risk management protocols, making them ill-equipped to respond to rapid shifts in market conditions. As a result, these methods often yield suboptimal outcomes and fail to account for the dynamic interactions between the underlying asset price, volatility, and time-sensitive parameters.

To address these limitations, this thesis introduces two advanced frameworks for options trading: a multi-agent architecture and a Reinforcement Learning (RL) approach. Both frameworks are designed to overcome the rigidity of conventional systems by leveraging data-driven, adaptive strategies capable of evolving with real-time market changes. Through this integrated methodology, the proposed solutions aim to significantly enhance trading performance and provide robust, scalable options trading models.

3.3 Traditional Challenges in Options Trading

Options trading is inherently complex, involving not only the decision of whether to buy or sell options but also selecting the right strike price, expiration date, and hedging strategies. Additionally, the pricing of options is highly sensitive to changes in market conditions, making it difficult to create one-size-fits-all trading strategies. Traditional trading methods typically employ static models based on historical data or predefined technical indicators, which fail to account for sudden market changes or unpredicted events such as geopolitical developments or economic crises.

Risk management is another area where traditional trading methods fall short. Many strategies do not have mechanisms in place to dynamically manage risk in real-time, making them vulnerable during periods of high market volatility. Furthermore, traditional options trading often relies on single-strategy models—the same strategy is used across various market conditions, limiting the system's ability to adapt and optimize trading decisions based on the ever-changing market dynamics.

3.4 The Role of Multi-Agent Systems in Overcoming These Challenges

To address the shortcomings of traditional approaches, this research proposes the use of a multi-agent framework. In this system, multiple specialized agents each perform distinct tasks that are critical to the options trading process. These tasks include technical analysis, trend analysis, strategy generation, strategy selection, and risk management. By distributing the decision-making responsibilities across a network of agents, the system mimics the collaborative nature of professional trading firms, where experts with specialized knowledge work together to make informed, adaptive decisions.

Each agent in the multi agent framework is trained to perform its role with a high degree of specialization. For example, the technical analysis agent might focus on past price data to identify patterns, while the trend analysis agent might use real-time market data to predict future movements. This collaboration allows for more dynamic strategy generation, where agents can communicate and adapt their strategies based on evolving market conditions.

In contrast to traditional systems, the Multi Agent framework benefits from the combined expertise of multiple agents, each of which can dynamically adjust its actions in response to new market data. Furthermore, the risk management agent evaluates the market's current volatility and adjusts trading decisions, accordingly, ensuring that the system remains resilient in volatile or uncertain market conditions.

3.5 The Role of Reinforcement Learning Systems in Overcoming These Challenges

Traditional methods of options trading, such as the Black-Scholes model, rely on fixed assumptions including constant volatility and log-normal distribution of asset prices—which often do not align well with the dynamic and complex characteristics of actual financial markets. These models frequently fail to capture phenomena like volatility clustering and fat-tailed distributions commonly observed in market behaviours. Reinforcement learning (RL) systems address these limitations effectively, as they are not constrained by these traditional assumptions; instead, they adaptively learn directly from market data through interaction, continually adjusting to shifts in market volatility and trading patterns. This adaptive capability positions RL as a particularly powerful approach for options trading, an area inherently characterized by unpredictability and rapid market evolution.

Reinforcement learning offers several distinct advantages when applied specifically to the buying and selling of call and put options. One major benefit is **dynamic strategy optimization**, whereby RL models can continuously refine trading strategies deciding when to buy, hold, or sell options based on real-time market conditions. By training on data from multiple intervals (such as daily or hourly), RL agents can better pinpoint profitable opportunities while **simultaneously managing associated risks**. Additionally, RL models excel at risk management by implementing **protective strategies such as stop-loss mechanisms**, effectively limiting potential downside risks to predefined thresholds, which is crucial given the high risk inherent in options trading. Moreover, **RL systems demonstrate superior real-time adaptability**, allowing them to promptly respond to new market information by adjusting the timing and approach to trading call and put options, thereby continually optimizing their strategies for the current market context.

3.6 Comparing the Multi Agent Framework with Reinforcement Learning

The core objective of this research is to evaluate whether Multi-Agent Collaborative Framework can outperform the Deep Reinforcement Learning framework. While both approaches utilize agents, Multi Agent framework relies on predefined rules and heuristics for each agent's role, whereas the RL agent learns and adapts based on rewards and penalties derived from its interactions with the market. This distinction creates a natural basis for comparison between the two approaches.

The Multi Agent framework has shown promising results by mimicking the behaviour of professional traders through specialized agents, but it may still be limited by the fact that its decision-making process is based on fixed models and rules. On the other hand, RL offers the possibility of an evolving trading system—one that continually refines its strategies based on past experiences. The comparison will assess whether the learning capabilities of RL agents provide a competitive edge in terms of adaptability, profitability, and risk management when compared to a manually designed multi-agent system.

Key metrics for comparison will include:

1. **Profitability**: Which system generates higher returns over a test period using historical data?

2. **Risk Management**: How well does each system manage risk, especially during periods of high market volatility?
3. **Adaptability**: How quickly can each system adapt to sudden shifts in market conditions or new patterns?

By comparing these two systems—one based on a handcrafted multi-agent approach and the other on a self-learning reinforcement learning system—this research aims to uncover which system provides the most reliable and efficient approach to options trading.

Conclusion

This research introduces two innovative approaches to options trading: the multi-agent framework and Reinforcement Learning (RL). By comparing the performance of these two systems, this research seeks to identify the most effective method for generating profitable trading strategies, managing risk, and adapting to dynamic market conditions. The ultimate goal is to determine whether RL can outperform the Multi Agent framework, thereby establishing a more adaptive, profitable, and resilient approach to options trading. This has the potential to significantly enhance the reliability and profitability of AI-driven trading systems and reshape the future of financial markets.

3.7 Multi-Agent System Development:

Proposed Framework and Technique

Options trading presents a multifaceted challenge that requires intricate decision-making across various dimensions, including market analysis, strategy formulation, risk management, and execution timing. To address these complexities and achieve consistent profitability in options markets, this thesis introduces an autonomous framework grounded in a multi-agent system (MAS) based on Agentic AI. This collaborative system leverages the strengths of multiple specialized agents, each designed to handle a distinct aspect of the options trading process, thereby enabling a cohesive and adaptive approach to decision-making.

The proposed multi-agent system comprises five interconnected agents, orchestrated to work synergistically:

1. **Generative Adversarial Network (GAN)** agent responsible for generating innovative trading strategies,

2. **Strategy selection** module that evaluates and selects optimal strategies based on predefined criteria,

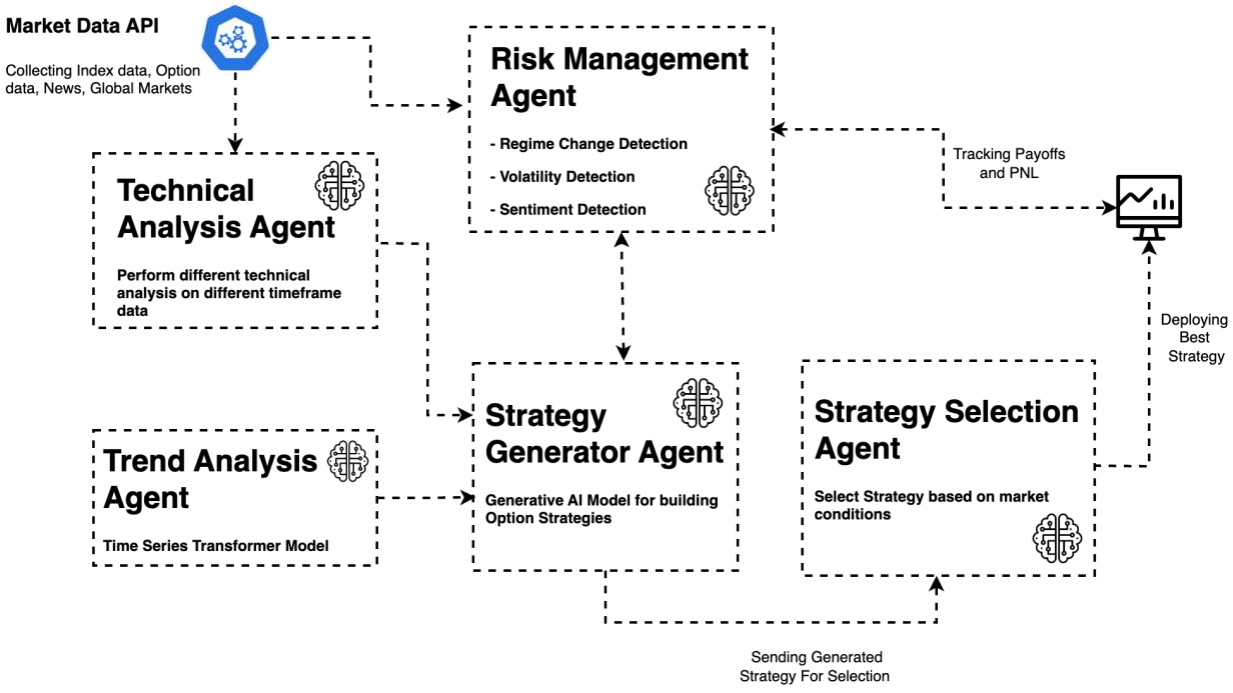
3. **Transformer-based** market regime prediction agent that forecasts market conditions such as direction, volatility, and momentum,

4. **Risk management agent** tasked with assessing and mitigating potential risks, and

5. **Data acquisition and Technical Analysis Agent** that gathers real-time market data and performs analytical computations to inform the decision-making process.

This division of labour allows the system to tackle the sequential and interdependent steps of options trading from gauging market dynamics to determining strike prices, position sizing, and entry/exit points while maintaining scalability and adaptability to diverse market environments.

Figure 21: Multi Agent System Flow Diagram



3.7.1 Key Constructs

The key constructs in the research are critical elements that define the core aspects of the multi-agent framework. These constructs are grounded in both theoretical concepts and practical implementation for options trading. The key constructs include:

- **Technical Analysis (TA):** The process of analysing historical market data (price, volume, indicators) to identify trends and potential trade signals. This involves indicators such as Moving Averages, RSI, and Bollinger Bands.
- **Trend Analysis (TA):** The use of machine learning, specifically the Temporal Fusion Transformer (TFT), to predict future market trends, assess potential price movements, and identify regime shifts.
- **Strategy Generation (SG):** The generation of various options strategies, including calls, puts, spreads, and straddles, using Generative Adversarial Networks (GANs). This construct focuses on the creation of dynamic strategies based on market conditions.
- **Strategy Selection (SS):** The decision-making process that involves choosing the most suitable strategy from those generated by the Strategy Generation agent. This construct assesses market conditions, volatility, and risk tolerance.
- **Risk Management (RM):** The management and mitigation of risks, using advanced statistical and machine learning techniques, including Value At Risk (VaR), Conditional VaR, and Drawdown Limits. It ensures the strategies are aligned with predefined risk thresholds.

3.7.2 Observable Indicators

Observable indicators are measurable factors that reflect the performance or condition of each key construct. These indicators serve as tangible signals that agents use to make informed decisions. Below are the indicators for each construct:

- **Technical Analysis (TA) Indicators:**
  - **Price Patterns:** Identification of chart patterns (e.g., head and shoulders, flags) based on historical data.
  - **Technical Indicators:** Moving Averages (SMA, EMA), RSI, and PSar, SuperTrend.
  - **Signal Strength:** The frequency and intensity of trade signals (buy, sell, hold).

- **Trend Analysis (TA) Indicators:**
  - **Forecasted Trend Direction:** Predicted market movement (bullish or bearish) over different forecast windows.
  - **Regime Shift Detection:** Identified shifts in market behaviour (e.g., from a bullish to a bearish market).
  - **Forecasted Price Movements:** Anticipated price levels based on the TFT model's predictions.
- **Strategy Generation (SG) Indicators:**
  - **Strategy Diversity:** Number and types of strategies generated (e.g., calls, puts, spreads, straddles).
  - **Market Alignment Score:** How well the strategy aligns with current market conditions, such as volatility and trend direction.
  - **Risk-Reward Ratio:** The potential profitability and risk of each generated strategy.
- **Strategy Selection (SS) Indicators:**
  - **Optimal Strategy Selection:** Identification of the best-performing strategy based on market conditions.
  - **Risk-Adjusted Return:** Evaluation of potential return, adjusted for the associated risk (e.g., Sharpe Ratio).
  - **Strategy Execution Success Rate:** The percentage of strategies successfully executed and yielding positive returns.
- **Risk Management (RM) Indicators:**
  - **Volatility Metrics:** Daily or intraday volatility based on market data.
  - **Risk Alerts:** Number and severity of risk alerts triggered by changes in market conditions.
  - **Risk Threshold Compliance:** Percentage of trades that remain within predefined risk thresholds (e.g., VaR limits).

3.7.3 Develop Measurement Tools

To measure the observable indicators associated with each construct, you will need specific tools and frameworks:

- **Technical Analysis Tools:**

- TA-Lib or Pandas TA for calculating standard technical indicators (RSI, Moving Averages, Bollinger Bands).
- **Autoencoder for Noise Filtering:** A deep learning-based model to process raw market data and eliminate noise, improving the accuracy of technical indicators.

- **Trend Analysis Tools:**
  - **Temporal Fusion Transformer (TFT):** A machine learning model designed to process time series data, providing trend forecasts and regime shift signals.
  - **Python Libraries (TensorFlow/PyTorch):** For training and testing the TFT model on historical market data.
- **Strategy Generation Tools:**
  - **Generative Adversarial Networks (GANs):** Used to generate a range of trading strategies by feeding historical market data, technical indicators, and trend forecasts into the GAN framework.
  - **Custom Python Code for Strategy Evaluation:** Implement performance metrics to evaluate the generated strategies and their effectiveness in different market conditions.
- **Strategy Selection Tools:**
  - **Optimization Algorithms (e.g., Genetic Algorithms**): Used to optimize the strategy selection process by evaluating different strategies against real-time data and predefined performance metrics.
  - **Risk-Return Metrics Calculation (e.g., Sharpe Ratio, Maximum Drawdown):** Used to measure the effectiveness and risk profile of each strategy.
- **Risk Management Tools:**
  - **Markov Models and NLP for Regime Change Detection:** Statistical tools to detect shifts in market regimes and analyze financial news sentiment.
  - **Risk Calculation Models (e.g., VaR, Conditional VaR):** Implement algorithms to calculate potential risk exposure and maintain trades within acceptable limits.

3.7.4 Establish Relationships Between Constructs

Once the key constructs and observable indicators are identified, it's important to define the relationships between them. This ensures the multi-agent framework operates cohesively:

- **Technical Analysis (TA) ↔ Strategy Generation (SG):** The outputs of the Technical Analysis Agent (e.g., trends, market signals) directly influence the Strategy Generation Agent's creation of strategies. The cleaner and more accurate the technical indicators, the better the strategies generated by the GAN.
- **Trend Analysis (TA) ↔ Strategy Selection (SS):** The Trend Analysis Agent's predictions on market direction help the Strategy Selection Agent identify the optimal strategy based on market conditions.
- **Strategy Generation (SG) ↔ Risk Management (RM):** The Strategy Generation Agent adjusts its generated strategies based on real-time risk data provided by the Risk Management Agent. For example, if volatility increases or a regime shift is detected, the strategy may be adjusted to mitigate risk.
- **Risk Management (RM) ↔ Strategy Selection (SS**): The Risk Management Agent continuously monitors market risk and provides risk thresholds, influencing the Strategy Selection Agent's choice of strategies. If the potential risk of a strategy exceeds the threshold, it may be discarded in favour of a lower-risk alternative.
- **Strategy Selection (SS) ↔ Technical Analysis (TA) ↔ Trend Analysis (TA):** These constructs work together in an iterative feedback loop: The Strategy Selection Agent uses insights from the Technical Analysis and Trend Analysis Agents to evaluate, refine, and select the most suitable strategies for the current market conditions.

3.7.5 Ensure Validity and Reliability

To ensure the validity and reliability of the multi-agent framework and measurement tools, several key aspects need to be considered.

Firstly, **content validity** is paramount, requiring that the indicators used to represent the constructs, such as technical indicators and strategy performance metrics, accurately capture the fundamental aspects of options trading and risk management as

defined within the framework. This can be substantiated through expert reviews and by validating against established trading theories.

Secondly, **construct validity** must be established by testing whether the defined indicators and relationships truly reflect the constructs they aim to measure, for example, confirming that the Strategy Generation Agent is indeed creating effective strategies based on market data.

Thirdly, **reliability** of the measurement tools, such as the GAN-based strategy generation model and the TFT model, needs to be ensured by verifying that they produce consistent results across different datasets and market conditions, which can be evaluated through backtesting on various historical datasets and robustness checks.

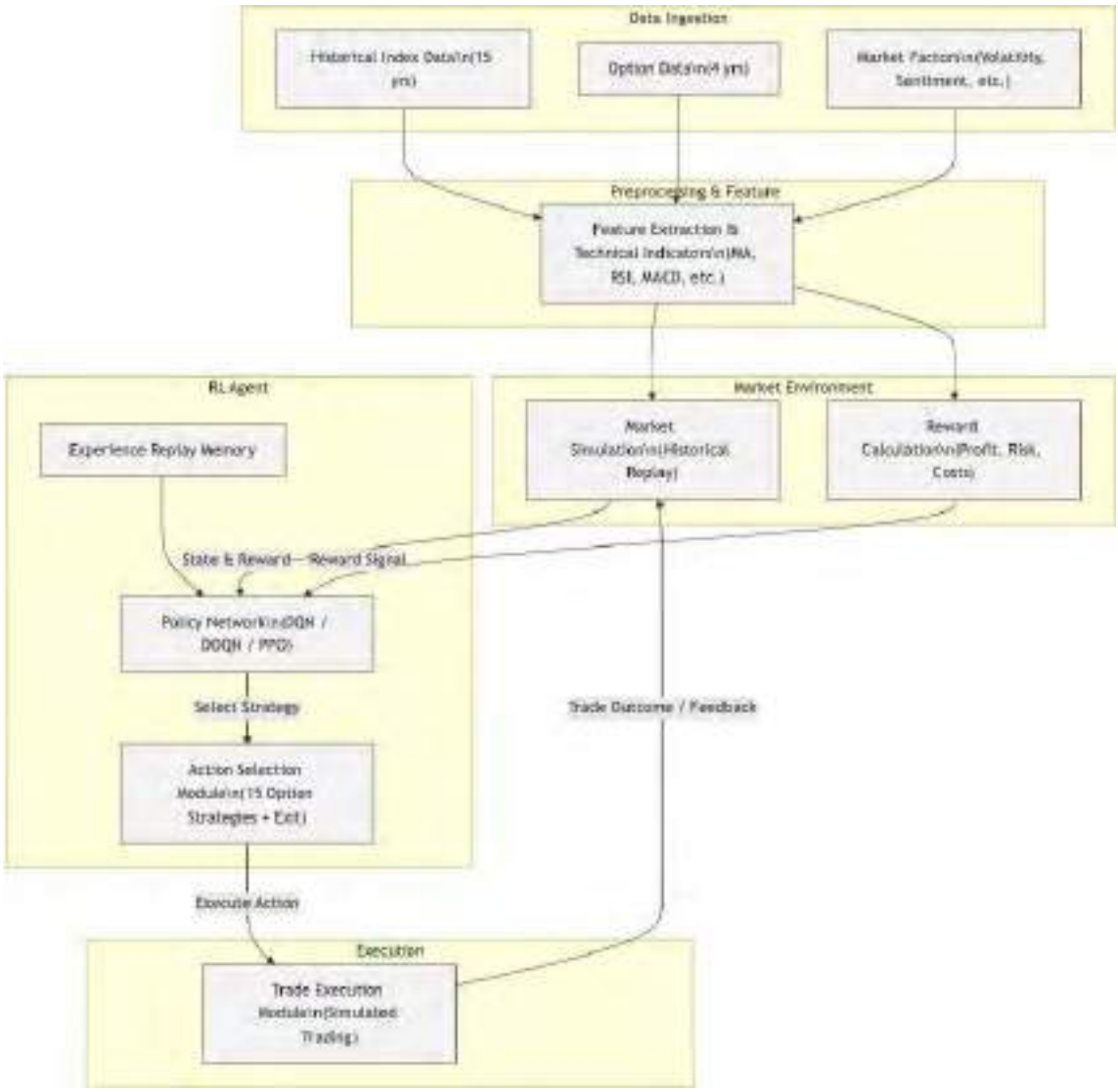
Furthermore, **external validity** is crucial to assess how well the multi-agent framework generalizes to real-world trading scenarios by comparing its performance against actual market data and benchmarking it against well-established trading strategies. Lastly, **test-retest reliability** should be evaluated by performing repeated assessments over different time periods to confirm that the framework's results, including win rates, loss rates, and strategy performance, remain consistent and reliable.

3.8 Deep Reinforcement Learning (DRL) for Options Trading

The market is inherently non-stationary, and option trading strategies require dynamic adjustment to different market regimes (trending, mean-reverting, volatile). DRL, with its ability to learn complex policies from high-dimensional inputs, offers a promising approach.

3.8.1 Proposed Solution Architecture of DRL Option Trading Framework

Figure 22: Architecture of DRL Option Trading Framework



3.8.2 Problem Formulation as a Markov Decision Process (MDP)

- **State Space (Observation):**

  The state at time(t) can be represented as a feature vector that includes:

  - Normalized index prices and returns
  - Technical indicators (e.g., moving averages, RSI, MACD)
  - Volatility metrics (historical volatility, VIX levels)
  - Sentiment scores and foreign market conditions

- Option premiums and Greeks

- **Action Space:**

  The agent has a discrete set of 16 actions:

  - **15 Trading Strategies:** Each corresponding to a specific option trading approach (e.g., bull call spread, straddle, iron condor, etc.)
  - **Exit/Neutral Action:** To close positions if the market regime changes.

- **Transition Dynamics:**

  The environment simulates market behaviour based on historical data. When the agent chooses an action, the system mimics order execution, applies transaction costs, and updates the portfolio based on the selected strategy's performance.

- **Reward Function:**

  A well-crafted reward function is crucial. A potential formulation could be:

$$r_t = \Delta V_t - \lambda \, . Risk\ Penalty\ y_t - \mu \, . Transaction\ Cost_t$$

  where:

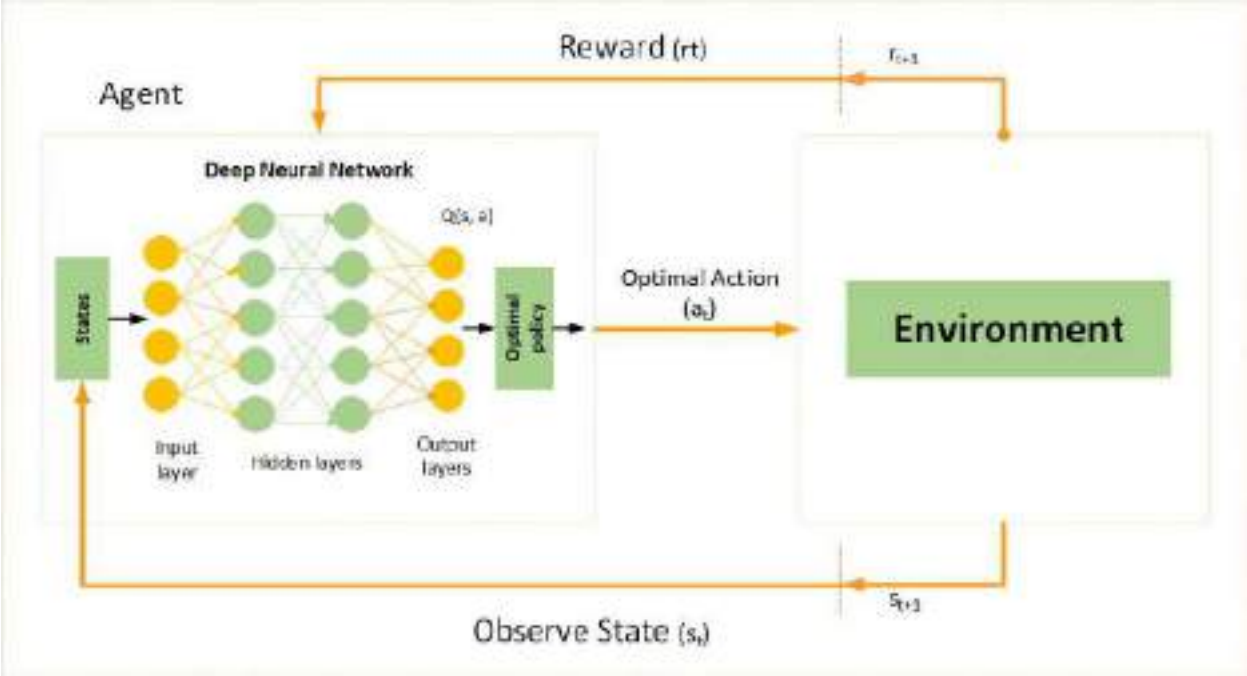
  - ΔV_t is the change in portfolio value over a time step.
  - $Risk\ Penalty\ y_t$ could be a function of realized volatility or drawdown during that period.
  - $Transaction\ Cost_t$ penalizes frequent trading to encourage stability.
  - λ and μ are hyperparameters to balance the trade-off between return, risk, and cost.

3.8.3 DRL Algorithms Exploration

3.8.3.1 DQN (Deep Q-Network):

Useful for environments with discrete action spaces. DQN approximates the Q-value function $Q(s, a)$ using neural networks.

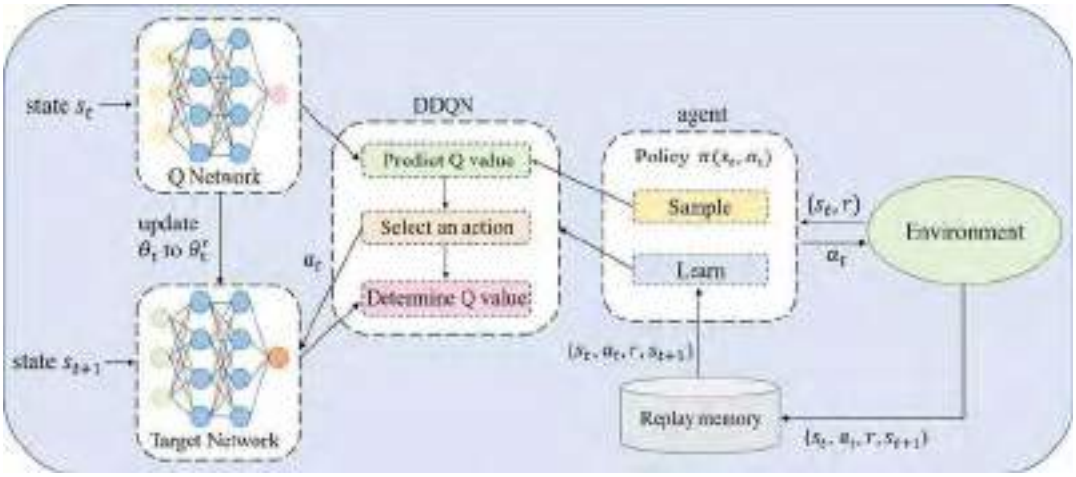
Figure 23: DQN Training Loop



3.8.3.2 Double DQN (DDQN):

Addresses the overestimation bias of DQN by decoupling the selection and evaluation of actions, leading to more stable learning.

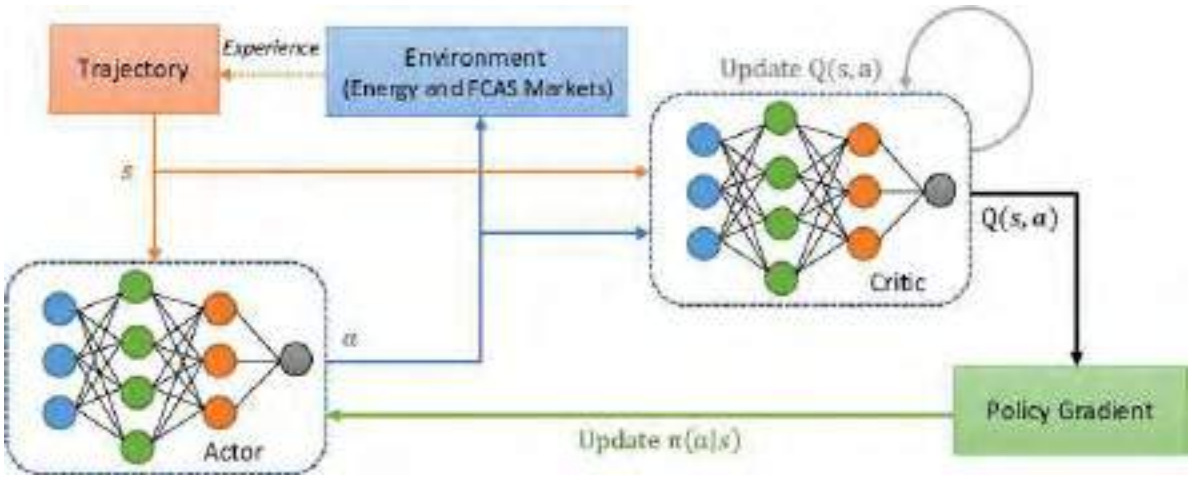
Figure 24: Double DQN



3.8.3.3 PPO (Proximal Policy Optimization):

A policy-gradient method that directly optimizes the policy while maintaining a constraint (or penalty) on the change in policy, ensuring stable updates.

Figure 25: Proximal Policy Optimization

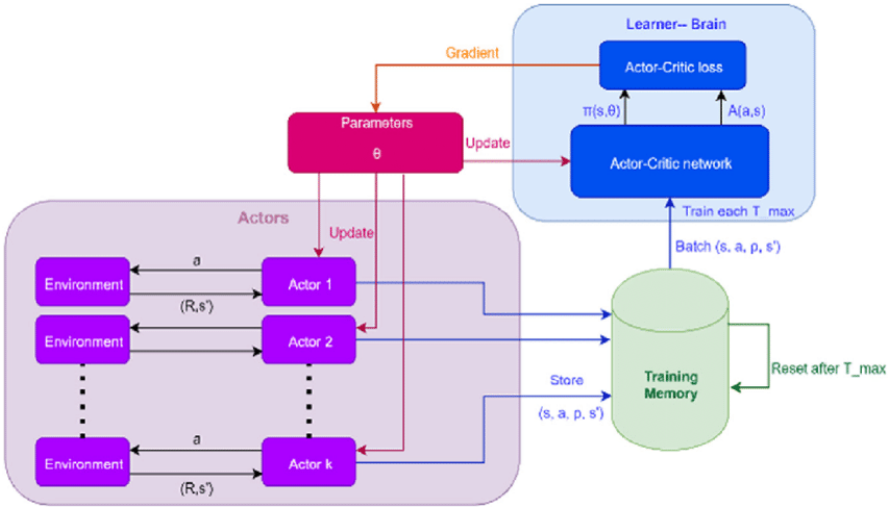


3.8.3.4 A3C (Asynchronous Advantage Actor-Critic):

A3C leverages multiple parallel agents interacting with different instances of the environment to update a shared model asynchronously. This decoupled, parallel approach reduces training

correlation and improves stability while the actor-critic architecture allows simultaneous policy learning and value estimation.

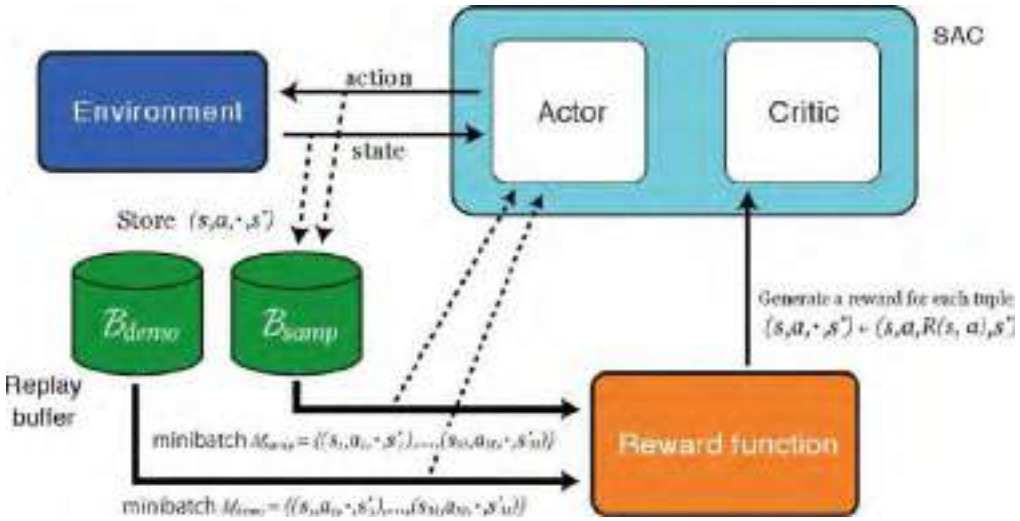
Figure 26: Asynchronous Advantage Actor-Critic



3.8.3.5 SAC (Soft Actor-Critic):

SAC is an off-policy actor-critic algorithm that optimizes a stochastic policy with an entropy regularization term, encouraging exploration by maximizing both the expected reward and policy entropy. This results in a more robust and stable learning process, particularly effective in continuous action spaces with improved sample efficiency.

Figure 27: Soft Actor-Critic



3.8.4 Pseudocode for the Autonomous Option Trading DRL Framework

Algorithm 1 Autonomous Option Trading DRL

```
// 1. Data Preprocessing & Setup
Load: index_data (15 years historical), option_data (4 years historical)
aligned_data ← AlignAndNormalize(index_data, option_data)
for each time t in aligned_data do
    indicators[t] ← ComputeTechnicalIndicators(aligned_data[t])
    regime_features[t] ← ComputeMarketRegimeIndicators(aligned_data[t])
end for
strategies ← {S_1, S_2, ..., S_15}
Set risk management parameters: risk_penalty_params ← {λ, α, β} and transaction_cost
// 2. Initialize DRL Agent
Initialize DRL_Agent with chosen algorithm (e.g., DQN, DDQN, PPO, A3C, SAC)
DRL_Agent.InitializePolicyNetwork()
experience_buffer ← EmptyBuffer()
// 3. Training Loop (Episodes)
for episode = 1 to NUM_EPISODES do
    current_time ← EpisodeStartTime()
    portfolio_value ← InitialPortfolioValue
    current_strategy ← None
    state ← GetState(aligned_data[current_time], indicators[current_time], regime_features[current_time])
    while current_time < EpisodeEndTime() do
        action ← DRL_Agent.SelectAction(state)
        if action ≠ current_strategy then
            if current_strategy ≠ None then
                portfolio_value ← ExecuteExit(current_strategy, current_time, portfolio_value)
            end if
            current_strategy ← action
            portfolio_value ← ExecuteEntry(current_strategy, current_time, portfolio_value)
        end if
        next_time ← current_time + Δt
        next_state ← GetState(aligned_data[next_time], indicators[next_time], regime_features[next_time])
        new_portfolio_value ← SimulateStrategyPerformance(current_strategy, current_time, next_time, portfolio_value)
        ΔV ← new_portfolio_value − portfolio_value
        risk_penalty ← CalculateRiskPenalty(current_time, next_time, risk_penalty_params)
        reward ← ΔV − risk_penalty − transaction_cost
        experience_buffer.Add(state, action, reward, next_state)
        DRL_Agent.UpdatePolicy(experience_buffer)
        state ← next_state
        portfolio_value ← new_portfolio_value
        current_time ← next_time
    end while
    LogEpisodePerformance(episode, portfolio_value)
end for
// 4. Evaluation on Test Data
for each test_episode in TestDataset do
    Reset environment and portfolio as in training
    while test_episode not finished do
        action ← DRL_Agent.SelectAction(state)          ▷ Using trained policy (no exploration)
        [state, portfolio_value] ← SimulateStep(state, action)
    end while
    EvaluateTestPerformance(test_episode)
end for
// Output final performance metrics and benchmark comparisons
```

3.8.5 Data Preprocessing and Feature Engineering

- **Historical Data Alignment:**
    - Synchronize index and option data ensuring time alignment (daily/hourly data frequency as required).
    - Normalize and scale features to ensure stable training.
    - VIX Data
- **Technical Indicator Calculation:**
  Compute and include indicators such as SMA, EMA, RSI, Bollinger Bands, MACD, etc.
- **Market Regime Detection:**
  Use statistical or machine learning methods (e.g., clustering on volatility and return distributions) to preliminarily label market regimes. This helps both in simulating the environment and potentially guiding the agent's risk preferences.

3.8.6 Environment and Simulation

- **Simulated Trading Environment:**
  Develop a simulation that:
    - Feeds the DRL agent with the current state vector at each time step.
    - Applies the chosen option strategy.
    - Updates portfolio value based on historical returns and simulated option pricing models.
    - Incorporates transaction costs and risk measures.
- **Strategy Execution:**
  Each strategy should be defined with clear entry/exit rules, risk management guidelines, and profit targets. The simulation should mimic real-world constraints (liquidity, execution delays).

3.8.7 Training the DRL Agent

- **Training Period:**
  Use the 15 years of index data and 4 years of option data to train the agent.
    - **Training Phase:** The agent learns policies through trial and error in a simulated environment.

- **Validation Phase:** Test the learned policy on a holdout set or using cross-validation techniques.

- **Algorithm Comparison:**

  Implement and compare multiple DRL algorithms (DQN, DDQN, PPO, SAC, A3C).

  - Perform hyperparameter tuning for each algorithm.
  - Monitor convergence, stability, and learning curves.

3.8.8 Reward Function Design

A robust reward function is critical. Consider the following structure:

- **Performance Component(ΔV_t):** This represents the change in portfolio value over time, which is the primary driver of reward. A positive ΔV_t indicates profit, while a negative value indicates a loss.
- **Risk Adjustment:**

  Penalize high volatility or large drawdowns. For instance:

$$\text{RiskPenalty}t = \alpha \cdot \text{Volatility}t + \beta \cdot \text{MaxDrawdown}t$$

  - Volatility_t: Measures the variability of returns, penalized by a factor α \alpha α.
  - $MaxDrawdown_t$ Represents the largest peak-to-trough decline in portfolio value, penalized by β.
  - λ : A scaling factor to adjust the strength of the risk penalty.

- **Transaction Costs:**

  Include a cost penalty to discourage over-trading:

$$\text{TransactionCost}_t = \text{Cost per trade} \times \text{Number of trades}$$

- **Reward Example:**

$$r_t = \Delta V_t - \lambda(\alpha \cdot \text{Volatility}_t + \beta \cdot \text{MaxDrawdown}_t) - \mu \cdot \text{TransactionCost}_t$$

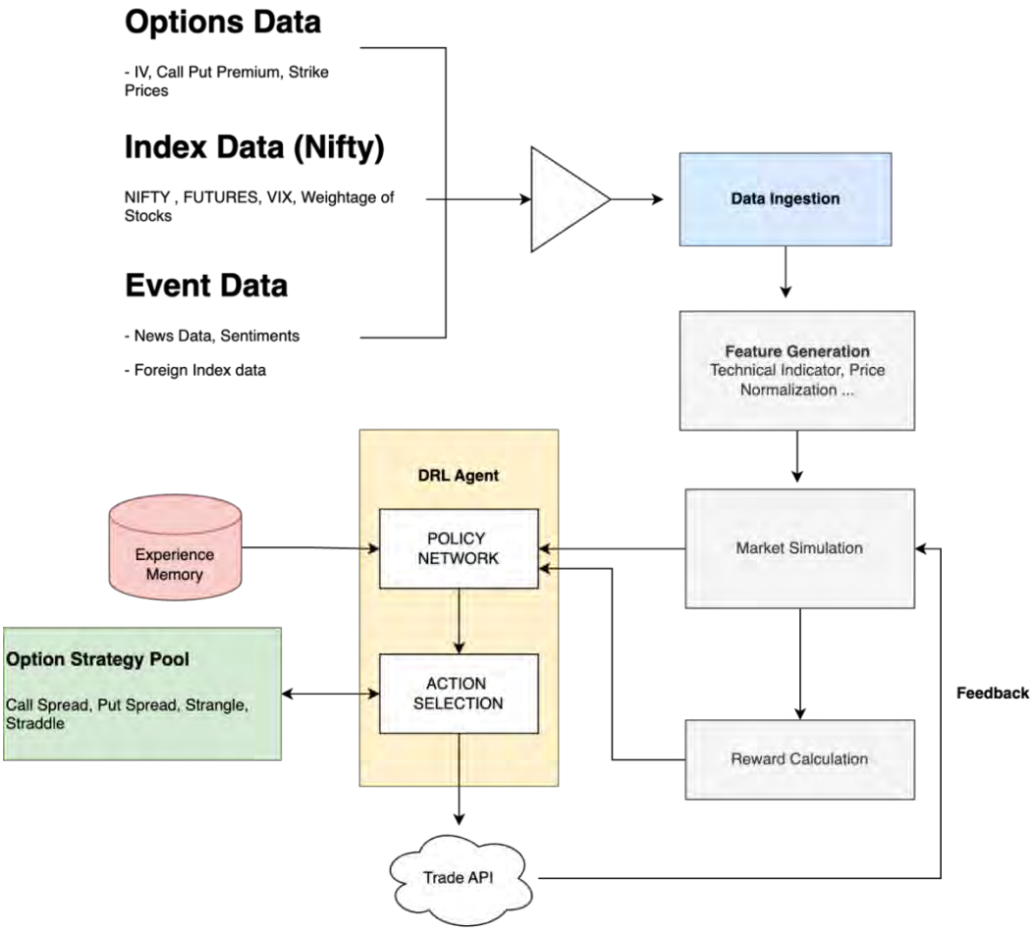
  where λ, α, β, and μ are tuned based on historical data to balance profit and risk.

Why This Reward Function Design is Robust

- **Balanced Objectives**: It incentivizes profit (performance) while discouraging reckless behaviour (risk) and inefficiency (over-trading).
- **Real-World Relevance**: By including transaction costs and risk, it mirrors the constraints traders face, making the RL agent's strategies more practical.
- **Adaptability**: The structure allows customization—e.g., increasing λ for conservative strategies or lowering γ \gamma γ in low-cost environments like options trading with tight spreads.

3.8.9 System Architecture

Figure 28: DRL Agent Framework



3.8.9.1 Overall System Flow

1. **Data Ingestion Module:**
    - Ingest and preprocess index, option, and auxiliary market data.
2. **Feature Engineering Module:**
    - Compute technical indicators and regime signals.
3. **DRL Agent Module:**
    - Selects one of the 15 option strategies or the exit action based on the current state.
4. **Execution Module:**
    - Simulates trade execution including strategy deployment, monitoring, and exit based on market changes.
5. **Feedback and Learning Module:**
    - Receives rewards, updates the agent's policy through DRL algorithms.
6. **Risk Management Module:**
    - Continuously monitors portfolio risk and triggers exits if risk thresholds are breached.

3.8.9.2 Integration and Autonomy

- **Autonomous Decision Making:**
  The system continuously evaluates market conditions in real-time (or through simulation) and autonomously switches strategies as dictated by the DRL agent's policy.
- **Robustness and Safety Nets:**
  Incorporate risk limits and fallback mechanisms in case the DRL policy deviates from acceptable risk profiles.

3.8.9.3 Statistical Testing and Robustness

- **Backtesting Over Different Market Regimes:**
  Ensure that the policy performs robustly during bull, bear, and sideways markets.

- **Cross-Validation:**
  Use techniques like walk-forward analysis to validate the consistency of the DRL agent across different time segments.
- **Stress Testing:**
  Simulate extreme market conditions to verify that the risk management protocols (reward penalties, exit strategies) protect the portfolio.
- **Comparative Analysis:**
  Benchmark the DRL-based strategy against traditional option trading strategies and simpler baselines (e.g., buy-and-hold, fixed-strategy).

3.9 Research Purpose and Questions

This research aims to develop and evaluate a novel framework for options trading and risk management by introducing both Reinforcement Learning and multi-agent framework approaches. The study seeks to address the existing gap in AI-based decision-making systems for options trading, focusing on predicting stock direction and formulating strategies that can outperform traditional trading methods and the underlying index.

3.9.1 Research Questions

1. Can the coordination among specialized agents combined with decentralized decision-making within a multi-agent system enhance both the selection and execution of options trading strategies compare to traditional approach?

2. Can Deep Reinforcement Learning models be developed to autonomously execute different option strategies in real time—aligning with human trading timeframes—and can these models outperform the underlying market index?

3. Can the adaptive, decentralized framework of multi-agent systems lead to superior trading performance compare to Deep Reinforcement learning based system under dynamic market conditions?

3.10 Research Design

This study adopts a dual-framework experimental design to evaluate the efficacy of autonomous options trading strategies driven by Agentic AI. Two primary methodologies are compared: a multi-agent collaborative system and a deep reinforcement learning (DRL) pipeline. The design is directly aligned with the research questions, which focus on the impact of decentralized decision-making, the real-time execution of DRL-based strategies, and the comparative performance of these two approaches under dynamic market conditions.

3.10.1 Overall Framework Overview

The thesis is structured around the central objective of determining whether specialized, decentralized decision-making can enhance options trading performance. In line with the research questions, the experimental design is organized into two main tracks:

- **Multi-Agent System:**
  The multi-agent framework is comprised of five specialized agents working in concert. Each agent is dedicated to a key component of the trading process:

  - **Technical Analysis (TA):** Uses historical market data and technical indicators (e.g., moving averages, RSI, Bollinger Bands) to detect market trends.
  - **Trend Analysis (TA):** Employs a Temporal Fusion Transformer (TFT) to forecast future market movements and identify regime shifts.
  - **Strategy Generation (SG):** Utilizes a Generative Adversarial Network (GAN) to dynamically generate various options trading strategies (e.g., calls, puts, spreads, straddles) based on prevailing market conditions.
  - **Strategy Selection (SS):** Evaluates generated strategies against market indicators, volatility, and risk tolerance to select the optimal trading strategy.
  - **Risk Management (RM):** Implements statistical and machine learning techniques (including VaR, Conditional VaR, and drawdown limits) to manage and mitigate portfolio risk.

  The design of the multi-agent system centres on decentralized decision-making, where each agent contributes unique market insights, and their combined outputs drive the overall

trading decision. This architecture directly addresses the first and third research questions by exploring whether such coordinated specialization enhances strategy selection and execution compared to traditional or centralized approaches.

- **DRL-Driven System:**
  The DRL framework is modelled as a Markov Decision Process (MDP) with the following elements:

  - **State Space:** Represents the trading environment through a feature vector encompassing normalized index prices, technical indicators, volatility metrics, sentiment scores, option premiums, and Greeks.
  - **Action Space:** Consists of 16 discrete actions, corresponding to 15 specific options strategies (e.g., bull call spread, straddle, iron condor) plus an exit/neutral action.
  - **Transition Dynamics:** Simulate market behaviour using historical data. This includes order execution mechanics, incorporation of transaction costs, and portfolio updates based on the strategy's performance.
  - **Reward Function:** Balances portfolio value changes, risk penalties (based on realized volatility or drawdown), and transaction costs through the function are hyperparameters to balance the trade-off between return, risk, and cost.

  This DRL approach is designed to autonomously learn and execute trading strategies in real time, aligning closely with human trading timeframes. It addresses the second research question by testing whether these models can outperform the underlying market index through adaptive, autonomous decision-making.

3.10.2 Experimental Setup

3.10.2.1 Data Acquisition and Preparation:

- **Data Sources:**
  - Historical market data (Jan 2014 - Dec 2024) for underlying assets (open, high, low, close, volume) and corresponding call/put options contracts, VIX, Index data.
  - Sentiment data from news sources and futures markets.

- **Data Preprocessing:**
    - **Cleaning & Normalization:**
        - Imputation of missing timestamps.
        - Outlier handling.
        - Time zone/format standardization.
        - Min-Max or Z-score normalization of price/volume.
        - Time series windowing.
    - **Feature Engineering:**
        - Technical indicators (SMA, EMA, RSI, SuperTrend, Parabolic SAR).
        - Implied volatility skew.
        - Macroeconomic and sentiment indicators.

3.10.2.2 Implementation Environment

This section details the technical infrastructure and software tools used to implement and evaluate the proposed options trading frameworks. A robust and realistic implementation environment is crucial for ensuring the validity and generalizability of the experimental results. The environment is divided into three primary components: real-time simulation, computational resources, and software tools. A dedicated setup for benchmarking is also described.

Real-Time Simulation

A critical component of this research is a high-fidelity, real-time simulation environment that accurately models the dynamics of options trading. This environment allows for the training and evaluation of both the multi-agent system and the DRL-based approach without the risks associated with live market trading. Key features of the simulation include:

- **Data Ingestion:** The simulator ingests historical index data (15 years) and options data (4 years), including price quotes (open, high, low, close), volume, and relevant Greeks (delta, gamma, vega, theta). Data is pre-processed and cleaned to handle missing values and inconsistencies. The data is sourced from [Insert Data Source Here - e.g., a specific data provider like Refinitiv, Bloomberg, or a historical options data archive].
- **Order Book Simulation:** While a full order book simulation is computationally expensive, a simplified model is implemented to approximate market liquidity and order

execution. This model considers bid-ask spreads and volume at different price levels to simulate realistic order fills. Slippage is incorporated, meaning that orders may not be filled at the exact requested price, reflecting real-world market conditions.

- **Options Pricing Model:** A Black-Scholes model, with extensions for American-style options and adjustments for dividends, is used to price options contracts within the simulation. Implied volatility is calculated from historical data and used as an input to the pricing model. Alternative pricing models (e.g., binomial trees, Monte Carlo simulations) were considered but ultimately rejected due to the computational overhead, which would significantly slow down the simulation, particularly during DRL training. The Black-Scholes model provides a reasonable balance between accuracy and computational efficiency.
- **Transaction Costs:** Realistic transaction costs, including brokerage fees and bid-ask spread costs, are incorporated into the simulation. These costs are parameterized based on typical rates charged by options brokers.
- **Risk Management:** The simulation enforces margin requirements and position limits, preventing the agents from taking on unrealistic levels of risk. A Value-at-Risk (VaR) calculation is performed at regular intervals to monitor the overall portfolio risk.
- **Time Stepping:** The simulation operates on a discrete-time basis, with a configurable time step. For the DRL experiments, a time step of 15 minutes and 1 hour is used, reflecting the desired trading frequency. For the multi-agent system, a more granular time step 5min, 15min, 30 min is used to allow for more frequent strategy adjustments. The choice of time step is a trade-off between simulation accuracy and computational cost.

Computational Resources

The computational demands of this research, particularly for training Deep Reinforcement Learning (DRL) agents and running extensive simulations, are substantial. To address these demands, a combination of hardware, cloud computing, and parallel processing techniques are employed. Experiments are conducted on a high-performance computing cluster, equipped with two nodes, each featuring NVIDIA Tesla V100 GPUs, which are crucial for accelerating the training of deep learning models. Additionally, Azure Cloud's GPU instances are utilized for cloud computing needs. To further optimize the computational efficiency, the simulation and training processes are parallelized,

effectively leveraging the multi-core CPUs and GPUs. This approach significantly reduces the overall training time. Python's multiprocessing capabilities, along with Pytorch Lightning and Ray libraries, are employed to facilitate parallel execution.

3.9.3 Multi Agent Specific Implementation Details

System Architecture and Coordination

The multi-agent architecture is designed to decentralize decision-making while allowing specialized agents to communicate and collaborate on core trading tasks. Each agent focuses on a distinct functional area—data analysis, market forecasting, strategy creation, strategy selection, and risk oversight—yet shares information through a central message bus or orchestrator. This enables each agent to operate independently while benefiting from the insights generated by other agents, thereby fostering a coordinated decision-making process.

1. **Data Acquisition & Technical Analysis (TA) Agent**
    - **Role:** Gathers historical and real-time market data (e.g., OHLC prices, trading volume) and computes technical indicators (Moving Averages, RSI, Bollinger Bands).
    - **Implementation Details:**
        - Periodically polls data feeds to maintain an up-to-date market state.
        - Transforms raw data into feature sets, which are then broadcast to other agents.
2. **Trend Analysis (TA) Agent**
    - **Role:** Employs a Temporal Fusion Transformer (TFT) to predict market trends and regime shifts (e.g., bullish, bearish, or range-bound).
    - **Implementation Details:**
        - Ingests technical indicator data and macro signals from the Data Acquisition & TA Agent.
        - Generates short- and medium-term forecasts of price movement and volatility levels.
        - Communicates predictive insights to the Strategy Generation and Strategy Selection agents.
3. **Strategy Generation (SG) Agent**

- **Role:** Uses a Generative Adversarial Network (GAN) to create a diverse set of potential options trading strategies—ranging from single calls or puts to complex spreads and straddles.
- **Implementation Details:**
  - GAN architecture is trained on historical options data to learn patterns of profitable strategy structures.
  - Produces candidate strategies aligned with the market regime insights from the Trend Analysis Agent.
  - Updates its strategy library periodically to adapt to evolving market conditions.

4. **Strategy Selection (SS) Agent**
   - **Role:** Evaluates and selects the most suitable strategy from the SG Agent's candidates, factoring in real-time market data, volatility, and risk parameters.
   - **Implementation Details:**
     - Ranks strategies based on expected returns, alignment with the forecasted market regime, and current risk appetite.
     - Chooses the top-performing strategy for execution and passes it to the Risk Management Agent for final verification.
5. **Risk Management (RM) Agent**
   - **Role:** Continuously monitors positions and overall portfolio risk using statistical and machine learning tools (e.g., VaR, CVaR, drawdown limits).
   - **Implementation Details:**
     - Enforces predefined risk thresholds and halts or modifies trading actions if thresholds are breached.
     - Feeds risk metrics back into the Strategy Selection Agent to ensure that high-risk strategies are de-prioritized.

This decentralized, multi-agent structure addresses the first research question by illustrating how specialized agents, each focusing on a key aspect of options trading, can coordinate effectively to improve both strategy selection and execution. The collaboration among agents

leverages distinct competencies—market analysis, forecasting, strategy generation, and risk mitigation—to outperform traditional, monolithic trading approaches.

3.9.4 DRL Specific Implementation Details

This subsection provides further details specific to the implementation of the DRL-based options trading approach.

- **Algorithm Selection:** As outlined in **Section 3.8.3**, a range of DRL algorithms are implemented and compared: DQN, DDQN, PPO, SAC, and A3C. These algorithms represent a diverse set of approaches to reinforcement learning, including value-based (DQN, DDQN) and policy-based (PPO, SAC, A3C) methods, as well as on-policy (PPO, A3C) and off-policy (DQN, DDQN, SAC) algorithms. This allows for a comprehensive evaluation of the suitability of different DRL techniques for options trading.
- **Hyperparameter Tuning:** A systematic hyperparameter tuning process is conducted for each DRL algorithm. This involves using techniques such as grid search, random search, and Bayesian optimization (using libraries like Optuna or Hyperopt) to identify the optimal hyperparameter configurations. Key hyperparameters tuned include learning rate, discount factor, batch size, network architecture (number of layers and neurons), and exploration parameters (e.g., epsilon-greedy decay for DQN).
- **Reward Shaping:** The reward function, as defined in Section 3.8.8, plays a crucial role in guiding the agent's learning. The parameters (λ, α, β, and μ) are carefully tuned through experimentation to balance the trade-off between profitability, risk aversion, and transaction cost minimization.

3.9.5 Benchmarking Setup

To rigorously evaluate the performance of the proposed frameworks, a comprehensive benchmarking setup is established.

- **Benchmark Strategies:** The performance of both the multi-agent system and the DRL-based approach is compared against 15 different options trading strategies, as mentioned in the abstract. These strategies include Straddle, Strangle, Iron Condor, Butterfly Spread, etc.. These strategies are implemented with predefined rules for entry, exit, and risk management, serving as a baseline for comparison.

- **Underlying Index:** The performance of all approaches is also benchmarked against the underlying index (e.g., S&P 500). This provides a measure of relative performance and helps determine whether the proposed frameworks can outperform a passive investment strategy.
- **Performance Metrics:** The following key performance metrics are used for evaluation:
    - **Cumulative Return:** The total return generated over the evaluation period.
    - **Maximum Drawdown:** The largest peak-to-trough decline in portfolio value.
    - **Win Rate:** The percentage of profitable trades.

    This comprehensive implementation environment and benchmarking setup provide a solid foundation for conducting rigorous and reproducible research on AI-driven options trading. The detailed description of the simulation, computational resources, software tools, and DRL-specific aspects ensures transparency and allows for replication of the experiments. The benchmarking setup, including a variety of strategies and performance metrics, enables a thorough evaluation of the proposed frameworks.

3.10.3 Alignment with Research Questions

RQ1: The evaluation of coordinated decision-making among specialized agents is directly measured through the performance of the strategy selection and execution process within the multi-agent system.

RQ2: The DRL framework's ability to execute strategies in real time is validated through its operational performance and comparison with market benchmarks.

RQ3: A head-to-head performance comparison under dynamic market conditions provides insights into the relative advantages of decentralized multi-agent systems versus a DRL-based approach.

3.11 Research Design Limitations

Agent based modelling has its limitations as well, which needs to be appreciated prior to embracing the methodology. Most models consider small number of generic assets and

agents. The dynamics can change drastically when these are enhanced to large numbers. The large number of parameters increases the computational complexity. Any method would be wholly acceptable only after due validation and calibration to real data is done. Since the stock market has a wealth of trading data, this can be addressed.

3.12 Experimental Setup

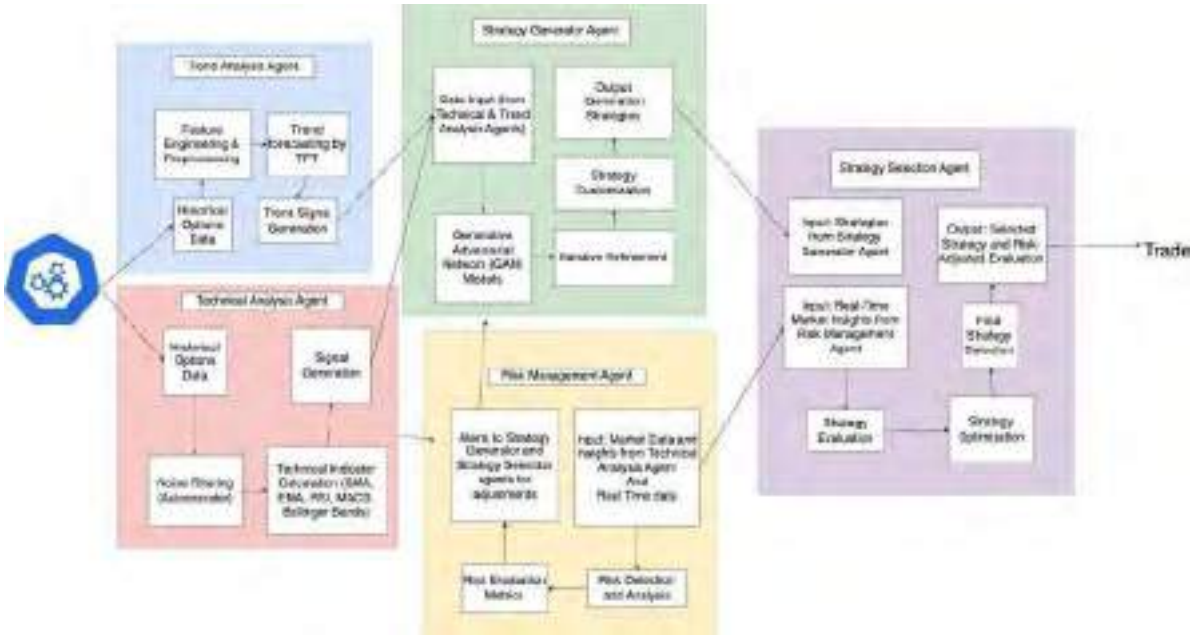
To evaluate the performance of the proposed **Multi-Agent and Reinforcement Learning systems**, a comprehensive experimental setup was designed, encompassing data collection, simulation environment configuration, baseline strategy selection, and performance metrics assessment.

2. Simulation Environment

The simulation was conducted using the Agent-Based Interactive Discrete Event Simulation (ABIDES) platform, which supports high-fidelity modelling of financial markets. ABIDES enables the simulation of numerous trading agents interacting within a market, providing a realistic environment for strategy evaluation.

3.12.1 Multi Agent Interaction:

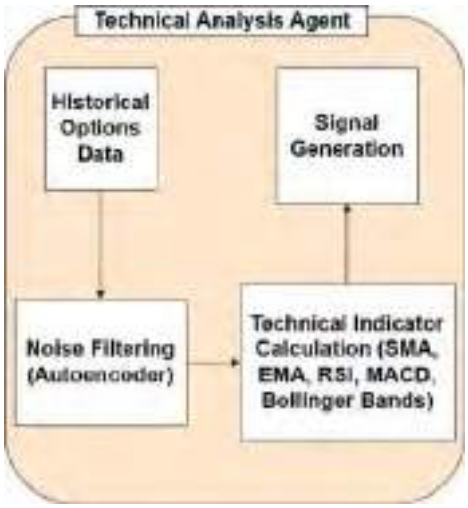
Figure 29: Multi Agent Interaction



3.12.1.1 Agent Roles and Responsibilities:

1. Technical Analysis Agent:

Figure 30: Technical Analysis Agent Workflow



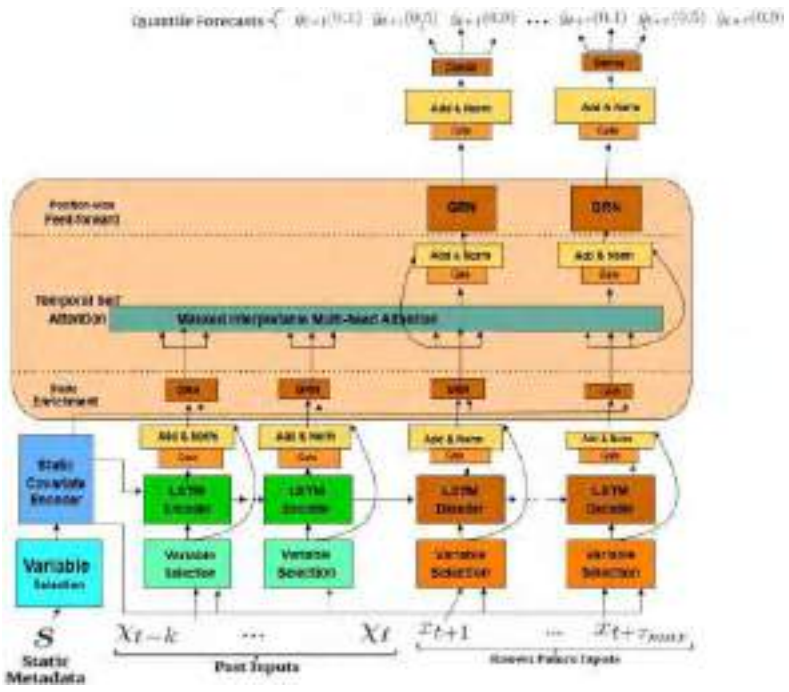
Role: The Technical Analysis Agent scrutinizes market data—including price, volume, and other pivotal indicators—across various time frames to identify trends, patterns, and potential trading signals.

Implementation Details:

- **Input:** Utilizes 10 years (2014-2024) of open-high-low-close (OHLC) data with a 30-day lag window.
- **Techniques Employed:**
    - **Moving Averages (SMA, EMA):** Smooth price data to highlight trends.
    - **Relative Strength Index (RSI):** Identifies overbought or oversold conditions.
    - **MACD:** Monitors momentum shifts.
    - **Bollinger Bands:** Assesses price volatility.
    - **Noise Reduction:** Incorporates an autoencoder-based tool to filter out noise and event-driven anomalies from time series data, enhancing data quality for pattern recognition.
    - **Tools/Frameworks:** Employs Python libraries such as Pandas-TA, TA-Lib, and custom modules for technical indicator calculations.
    - **Update Frequency:** Refreshes indicators based on the chosen trading interval (e.g., every 45 minutes or 2 hours) as new data arrives.
    - **Output:** Generates actionable insights, including technical signals (buy, sell, hold), time-based signals for potential trend reversals or continuations, and analyses of market strength, momentum, and volatility. These insights inform the Strategy Generator Agent.

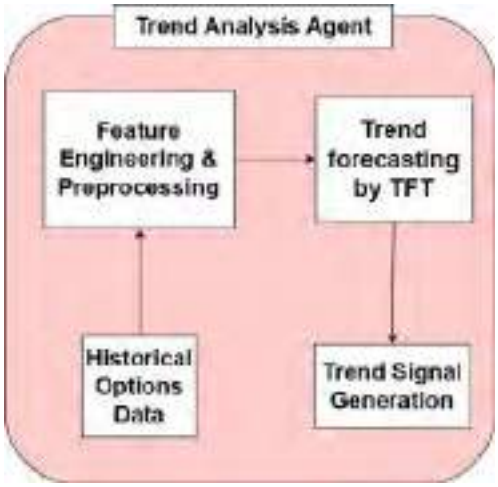
2. Trend Analysis Agent:

Figure 31: Temporal Fusion Transformer for Trend Analysis



Role: The Trend Analysis Agent employs advanced machine learning techniques, specifically the Temporal Fusion Transformer (TFT), to analyze historical market data and predict future price trends. Its primary objective is to forecast market directions and assess potential price movements by capturing complex temporal dependencies and patterns within time-series data.

Figure 32: Trend Analysis Agent Workflow



Input Features:

- **Historical Market Data: Uti**lizes data spanning the past decade (2014–2024), including price movements, trading volumes, and other relevant financial metrics.
- **Technical Indicators:** Incorporates normalized technical indicators such as moving averages and the Relative Strength Index (RSI) to enrich the feature set.

Techniques Employed:

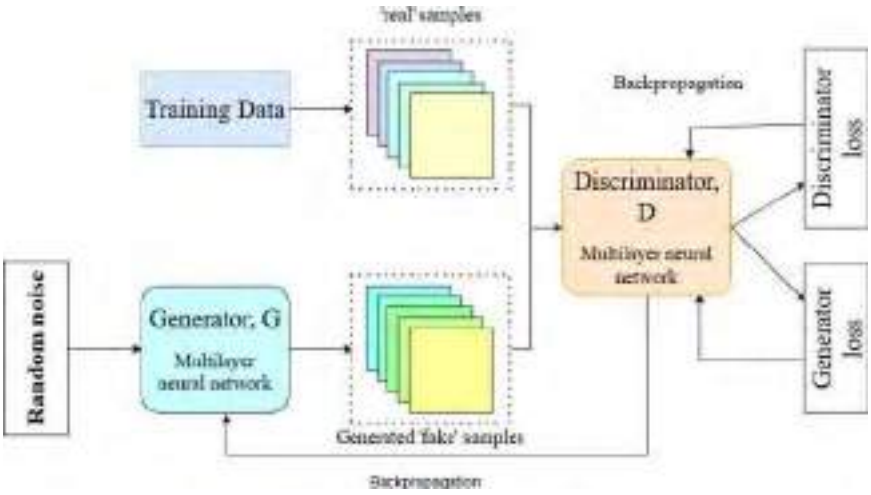
- **Transformer Model:** Leverages the Temporal Fusion Transformer, a state-of-the-art architecture designed for multi-horizon time-series forecasting. TFT effectively captures long-term dependencies and identifies seasonal patterns through its self-attention mechanism, integrating both static and temporal features into a unified representation.
    - **Data Preprocessing:** Applies normalization and windowing techniques to transform raw time-series data into structured sequences suitable for input into the transformer model.
    - **Model Architecture:** The TFT model comprises several key components:
        - **LSTM Encoder:** Captures temporal dependencies within the data.
        - **Multi-Head Attention Layer:** Identifies and focuses on significant temporal features.
    - **Gating Layers:** Regulate the flow of information, enhance model interpretability and performance.
    - **Decoder:** Generates multi-step forecasts based on encoded information.
    - **Output Layers:** Produce final predictions and associated uncertainty estimates.
- **Training:** Utilizes the Adam optimizer with Mean Squared Error (MSE) as the loss function to train the model effectively.

Output:

- **Trend Direction:** Predicts the market's direction over specified forecast periods.
- **Forecasted Values:** Provides anticipated future values aligned with the identified trends.
- **Regime Shift Signals:** Detects potential transitions between different market regimes, such as shifts from bullish to bearish trends.

3. Strategy Generator Agent:

Figure 33: Generative Adversarial Network for Strategy Generation



Role: Utilizes Generative Adversarial Networks (GANs) to autonomously develop diverse trading strategies, including calls, puts, spreads, and straddles, adapting to evolving market conditions.

Figure 34: Strategy Generator Agent Workflow

1. Input Features

- **Market Metrics:**
    - o VIX and other volatility indicators (e.g., implied and historical volatility).
    - o Data from international markets (indices, currency movements) to capture global sentiment.

- **News & Sentiment:**
  - Use Natural Language Processing (NLP) to analyze financial news and social media, extracting sentiment scores or event indicators.
- **Market Regimes:**
  - Classify market conditions into regimes (e.g., low-volatility trending, high-volatility mean-reverting) using clustering or regime-switching models.

2. Model Architecture

- **Conditional Generative Model:**
  - Use a Conditional Generative Adversarial Network (cGAN) or a Transformer-based model conditioned on the input features above. The generator would propose options strategies (e.g., configurations like short calls, short puts, or multi-leg spreads) tailored to the current market regime.

3. Strategy Output

- **Dynamic Strategy Parameters:**
  - The model generates the type of options strategy (e.g., call or put short), strike distances (far or short), and other trade parameters.
- **Adaptive Response:**
  - As market conditions evolve, the model can update its strategy recommendations, effectively "re-generating" strategies when the input features (volatility, news sentiment, etc.) change.

4. Validation & Simulation

- **Backtesting Framework:**
  - Integrate a backtesting module where the generated strategies are simulated against historical data to evaluate performance.
- **Risk Management:**
  - Ensure that the model considers risk measures (like drawdown, Greeks exposure) to filter out overly aggressive or mismatched strategies.

5. Implementation Steps

- **Data Aggregation:**

- Collect historical data on market conditions, news sentiment, and options trades.

- **Feature Engineering:**
  - Process raw data into standardized input features for the model.
- **Model Training:**
  - Train your conditional generative model on historical regimes and corresponding successful strategies.

- **Simulated Trading & Feedback:**
  - Use reinforcement learning to simulate trades and improve strategy generation over time.

This approach lets the generative model create adaptable and data-driven options strategies that respond to different market scenarios, balancing the trade-offs between strategy aggressiveness and risk management.

4. **Strategy Selection Agent:**

Figure 35: Strategy Selection Agent Workflow

Role: Evaluates strategies generated by the Strategy Generator Agent to identify the most suitable ones based on prevailing market conditions, ensuring alignment with market sentiment, volatility, and defined risk tolerance.

Implementation Details:

- **Input:** Assesses a range of strategies from the Strategy Generator Agent, supplemented by real-time market insights from the Risk Management Agent.

- **Output:** Selects optimal strategies or combinations thereof, providing risk-adjusted evaluations that balance potential returns with associated risks, ready for execution.

5. **Risk Management Agent:**

Figure 36: Risk Management Agent Workflow

Role: Monitors and identifies risks, including market regime changes, volatility surges, and shifts in sentiment, ensuring that trading activities adhere to established risk thresholds.

Implementation Details:

- **Risk Detection:** Employs statistical and machine learning methods, such as Markov models, to detect regime changes, and applies Natural Language Processing (NLP) to analyze financial news for sentiment shifts.
- **Risk Metrics:** Calculates Value at Risk (VaR), Conditional VaR, and drawdown limits to assess potential losses and protect the portfolio during downturns.
- **Real-Time Monitoring:** Continuously evaluates risk indicators, alerting the Strategy Generator Agent to adjust strategies as necessary to mitigate identified risks.

6. **Communication and Feedback Loops:**

- **Enhanced Dynamics:**

- **Real-Time Data Sharing:** Agents exchange up-to-date market data, analyses, and performance metrics, ensuring that all components operate with the latest information.
- **Adaptive Learning:** Agents adjust their strategies and analyses based on continuous feedback, allowing the system to evolve in response to changing market conditions.
- **Collaborative Problem-Solving:** When anomalies or unexpected market behaviors occur, agents collaborate to identify causes and develop coordinated responses, enhancing the system's robustness.

7. **Integration and Coordination:**
   - **Enhanced Dynamics:**
     - **Unified Objective:** All agents work towards the common goal of optimizing trading performance while effectively managing risks, ensuring that individual actions align with the system's overall strategy.
     - **Dynamic Adaptation:** The system's decentralized structure allows agents to adapt independently to market changes, while coordinated efforts ensure that these adaptations contribute to the system's collective objectives.
     - **Continuous Improvement:** Ongoing interactions among agents foster a culture of continuous learning and improvement, with each agent refining its processes based on shared experiences and insights.

3.12.1.2 Baseline Strategies

To benchmark the performance of the multi-agent system, several traditional trading strategies were implemented:

- **Buy and Hold (B&H):** Investing equally across selected assets throughout the evaluation period.

3.12.1.3 Performance Metrics

The effectiveness of each strategy, including those generated by the multi-agent system, was evaluated using the following metrics:

- **Win Rate:** Percentage of successful trades.
- **Loss Rate:** Percentage of unprofitable trades.
- **Neutral Rate:** Percentage of trades with minimal gains or losses.
- **Profit and Loss (P&L):** Total monetary gain or loss over the evaluation period.

3.12.1.4 Evaluation Methodology

Two primary methods were employed to assess the trading strategies:

- **Market Replay:** Simulating the execution of strategies over historical market data without altering market dynamics.
- **Interactive Agent-Based Simulation (IABS):** Utilizing a population of background trading agents to create a responsive market environment, allowing the market to adapt to the strategies being tested.

  This dual-method approach enabled a robust evaluation of the strategies under both static and dynamic market conditions.

3.12.1.5 Computational Resources

The experiments were conducted on high-performance computing clusters equipped with multiple GPUs, ensuring efficient processing of computationally intensive simulations.

3.12.1.6 Evaluation Period

The simulation covered a six-month period from June to November 2024, with agents making daily trading decisions based on available data up to that day, ensuring that no future information influenced the strategy evaluations.

This experimental setup was designed to rigorously assess the performance of the multi-agent system, providing a clear comparison with traditional trading strategies and offering insights into the potential benefits of multi-agent frameworks in financial trading.

3.12.2 Reinforcement Learning

The experimental setup for the Reinforcement Learning for Options Trading approach includes comprehensive data collection, a simulation environment built using the **SLM-Lab** framework, comparison with baseline strategies, clear performance metrics, a robust evaluation methodology, and appropriate computational resources. The evaluation period is carefully

selected to ensure that the models are tested on unseen market data to assess their ability to generalize and perform effectively.

3.12.2.1 Data Collection and Simulation Setup

3.12.2.1.1 Data Collection:

The data used in this experiment spans from January 2014 to December 2024 and includes historical market data for underlying assets and their corresponding options contracts.

- **Underlying Asset Data:**
  The underlying asset data consists of open-high-low-close (OHLC) prices and trading volumes for various assets. This dataset is crucial for training the RL models to understand the price movements and market behaviours of the underlying assets.
- **Options Contract Data:**
  The corresponding options contracts, including both call and put options, are selected for analysis based on their expiration dates and correlation with the underlying asset's price movement. This data provides the RL agent with an in-depth understanding of how options prices are influenced by market fluctuations of the underlying asset.

3.12.2.1.2 Simulation Environment:

The environment simulates real-world trading scenarios where the agent interacts with financial market data and takes actions based on its observations. In this environment:

- The state space in the environment includes key market features such as time, open price, highest price, lowest price, close price, and trading volume. These features capture the price behaviour and market dynamics needed for the agent to make actionable trading decisions.
- The **action space** consists of 16 actions to choose any strategy to deploy
- The agent learns by interacting with this environment, making decisions based on the observed state, and adjusting its trading strategy through trial and error.

3.12.2.2 Architectural Information of the DRL Framework

The proposed Deep Reinforcement Learning (DRL) framework is designed to autonomously execute options trading strategies by learning from dynamic market environments.

In this section, we detail the architectural components that underpin the DRL approach, aligning with our research questions which explore the capability of autonomous models to execute strategies in real time, outperform market benchmarks, and compare favourably to decentralized multi-agent systems.

3.12.2.2.1 Problem Formulation as a Markov Decision Process (MDP)

At the core of the DRL framework lies the formalization of options trading as a Markov Decision Process (MDP). This formulation enables the agent to learn optimal trading policies by iteratively interacting with a simulated market environment. The MDP is defined by the following key components:

- **State Space (Observation):**
  The agent's perception of the market at each time step tt is encapsulated in a feature vector. This vector is composed of:
  - **Normalized Index Prices and Returns:** Capturing price movements relative to benchmark indices.
  - **Technical Indicators:** Including moving averages, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD), which provide insight into market trends.
  - **Volatility Metrics:** Such as historical volatility and VIX levels to quantify market uncertainty.

    Sentiment Scores and Foreign Market Conditions: Allowing the agent to incorporate external market sentiment and global economic influences.

  - **Option Premiums and Greeks:** Essential for evaluating the risk and potential payoff associated with various options strategies.
- **Action Space:**
  The DRL agent is equipped with a discrete set of 16 actions:
  - **15 Trading Strategies:** Each action corresponds to a distinct options trading approach (e.g., bull call spread, straddle, iron condor, etc.), enabling the agent to adapt to different market scenarios.

- **Exit/Neutral Action:** A dedicated action to close existing positions when market regimes change, ensuring that the agent can effectively manage risk.

- **Transition Dynamics:**

  The environment simulates market behaviour using historical data, where each action taken by the agent triggers:

  - **Order Execution Simulation:** Mimicking realistic trading conditions including order matching and liquidity constraints.
  - **Transaction Costs Application:** Incorporating the cost implications of each trade, thereby encouraging efficient strategy selection.
  - **Portfolio Updates:** Reflecting the performance of the selected trading strategy, which in turn informs subsequent decisions.

- **Reward Function:**

  A well-designed reward function is crucial for guiding the learning process. The reward at each time step **t** is defined as:

$$r_t = \Delta V_t - \lambda \, . Risk\ Penalty\ y_t - \mu \, . Transaction\ Cost_t$$

  where:

  - ΔV_t is the change in portfolio value over a time step.
  - $Risk\ Penalty\ y_t$ could be a function of realized volatility or drawdown during that period.
  - $Transaction\ Cost_t$ penalizes frequent trading to encourage stability.
  - λ and μ are hyperparameters to balance the trade-off between return, risk, and cost.

3.12.2.2.2 Integration with Research Objectives

This DRL-driven pipeline directly addresses the research questions posed in the thesis:

1. **Autonomous Execution of Options Strategies:**

   By learning a mapping from complex, high-dimensional market states to a discrete set of strategic actions, the framework demonstrates its capacity to operate autonomously within real-time trading environments. The continuous adjustment of actions in response

to changing market conditions is designed to not only match but potentially surpass the performance of traditional benchmarks.

2. **Performance Against the Underlying Index:**
   The reward function's design ensures that the agent is incentivized to optimize portfolio performance while controlling for risk and transaction costs. This careful calibration is key to achieving returns that significantly outperform the underlying market index.
3. **Comparison with Multi-Agent Systems:**
   While the multi-agent approach leverages decentralized decision-making through a coordinated network of specialized agents, the DRL framework offers a streamlined, end-to-end learning pipeline. This comparative analysis forms a central pillar of the experimental methodology, providing insights into which approach yields superior performance under dynamic market conditions.

In summary, the DRL framework embodies a sophisticated integration of financial market modelling with advanced reinforcement learning techniques. By framing options trading as an MDP and meticulously designing its state space, action set, transition dynamics, and reward structure, the architecture supports a robust, adaptive system capable of real-time decision-making and strategic execution. This architectural strategy not only fills a critical gap in AI-based trading methodologies but also lays the groundwork for empirical validation against both traditional and agent-based systems.

3.12.2.3 Learning Algorithm:

- The **learning algorithms** used in our framework for options trading include:

3.12.2.3.1 Deep Q-Network (DQN): DQN is used to approximate the optimal action-value function

$Q*(s,a)Q*s,a$

using a deep neural network. This allows the agent to evaluate the value of each action (buy, hold, sell) given a specific state, enabling it to learn optimal strategies through trial and error.

3.12.2.3.2 Proximal Policy Optimization (PPO):

PPO is a first-order policy optimization algorithm that ensures stable updates by using a clipped objective function. This algorithm is well-suited for the highly volatile options market, balancing exploration and exploitation to learn efficient trading strategies.

3.12.2.3.3 Soft Actor-Critic (SAC):

SAC optimizes a stochastic policy, encouraging exploration by maximizing entropy, which helps prevent the agent from converging too quickly to suboptimal solutions. However, it was observed that SAC performed less effectively when combined with the stop-loss strategy, due to its high-frequency trading nature.

- These algorithms allow the agent to continuously improve its decision-making abilities and learn optimal strategies for options trading through repeated interactions with the environment, ultimately leading to policies that maximize reward.

3.12.2.4 Risk Management:

The protective closing strategy incorporated into the framework helps manage risk effectively, especially in options trading:

- **Stop-Loss Mechanism**: If the unrealized loss from a trade exceeds a specified threshold (e.g., 1%, 2%, or 3%), the agent automatically closes the position to prevent further losses. This is crucial in options trading, where the potential for large losses exists due to the high leverage and price volatility of options.
- **Position Sizing**: Position sizing strategies may be used to adjust the size of the agent's positions based on market conditions and the agent's risk tolerance. This helps manage exposure and prevents the agent from taking on excessive risk in volatile market conditions.

These risk management strategies ensure that the agent does not accumulate significant losses while allowing it to capitalize on favorable market conditions. The addition of the protective stop-loss mechanism makes the system more robust, especially in times of high market volatility.

3.12.2.5 Baseline Strategies:

To benchmark the performance of the RL-based strategies, the **buy and hold (B&H)** strategy is used as the baseline. In this strategy, the agent simply holds the asset throughout the entire testing period without making any trading decisions based on market conditions. This strategy helps assess whether the RL models provide improvements over a simple, passive trading approach.

3.12.2.6 Performance Metrics:

The performance of the RL models is evaluated using the following key metrics:

- **Profit and Loss (P&L)**: This metric measures the total returns generated by the RL agent over the testing period, accounting for transaction costs.
- **Risk-Adjusted Return**: This includes metrics like **Sharpe Ratio**, which accounts for the risk-adjusted return of the agent's strategy. It evaluates how well the agent performs in relation to the volatility of its returns.
- **Drawdown**: This measures the peak-to-trough decline during the testing period and helps assess the risk of the trading strategy.
- **Frequency of Trades**: This metric tracks how frequently the agent takes action, as it could indicate whether the agent is overtrading or trading efficiently.

3.12.2.7 Evaluation Methodology

3.12.2.7.1 Introduction

The evaluation of an automated options trading system based on reinforcement learning (RL) is crucial to assess its performance, risk management capabilities, and ability to generalize across diverse market conditions. This methodology outlines a comprehensive framework for evaluating such a system, focusing on key aspects such as **profitability**, **risk management**, **execution efficiency**, **generalization**, and **learning performance**. It provides an in-depth assessment of the system's behavior, ensuring it aligns with both theoretical and practical requirements for a robust and adaptive trading agent.

3.12.2.7.2 Evaluation Metrics

The evaluation of the reinforcement learning-based options trading system will be conducted using a set of metrics categorized into **profitability**, **risk management**, **execution efficiency**, and **generalization**. The specific metrics are designed to ensure a comprehensive understanding of the system's performance and adaptability.

The profitability of the system will be assessed through the following metrics:

- **Total Return**: The cumulative percentage gain or loss over a defined evaluation period (e.g., monthly, quarterly, or yearly). This metric provides an overall performance indicator and captures the agent's long-term profit generation capabilities.

.

- **Maximum Drawdown (MDD)**: This represents the largest peak-to-trough decline in equity over the evaluation period, providing insight into the system's potential for large losses and its ability to recover from them.
- **Win Rate**: The percentage of profitable trades relative to the total number of trades executed. This metric provides a measure of the agent's overall effectiveness in generating profitable trade.

- **Generalization and Learning Performance**

To ensure long-term profitability and adaptability, it is crucial to evaluate the agent's ability to generalize across unseen data and its learning process over time. This will be achieved through a combination of out-of-sample testing, walk-forward testing, and the analysis of reward distribution over episodes. Out-of-sample testing will assess the system's ability to generalize and adapt to new, unseen market conditions by testing it on data not included in the training process. A more robust approach, walk-forward testing, will simulate real-world trading by iteratively training the model on historical data, testing it on the next period, retraining with new data, and testing again. This ensures the system can adapt to changing market conditions over time. Finally, tracking the distribution of rewards over episodes will monitor improvements in decision-making and assess the system's learning progress throughout training, ensuring the agent steadily improves its performance as it learns from the market environment.

3.12.2.7.3 Transaction Costs and Execution Efficiency

A realistic evaluation of the system must incorporate transaction costs, which significantly affect the profitability of trading strategies. The following assumptions will be considered:

- **Transaction Cost Assumption**: A transaction cost of 0.1% per trade will be applied to account for brokerage fees, slippage, and other related costs. This ensures that the system’s profitability is evaluated under more realistic, real-world conditions.

3.12.2.7.4 Evaluation Framework and Backtesting

To ensure comprehensive and realistic evaluation, the RL system’s performance will be tested using a robust backtesting framework. This framework will include:

- **Historical Data Testing**: The system will be evaluated using historical data from 2014 to 2024, ensuring it is tested across different market conditions (e.g., bull, bear, and volatile markets).

3.12.2.7.5 Periodic Retraining and Adaptability

- **Retraining Strategy**: Setting a clear strategy for periodic retraining, ensuring the agent adapts to shifts in market volatility, trends, and other factors.
- **Adaptive Algorithms**: Incorporating adaptability within the RL framework to adjust strategies as new data and insights become available, minimizing the risk of model decay over time.

3.12.2.7.6 Conclusion

The proposed evaluation methodology provides a detailed framework to assess the performance of a reinforcement learning-based automated options trading system. By evaluating profitability, risk management, execution efficiency, generalization, and learning performance through a variety of metrics and testing methods, this methodology ensures that the trading system is not only profitable but also adaptable and robust in a dynamic and volatile market environment. The inclusion of metrics such as Sortino Ratio, Calmar Ratio, volatility, realized volatility, and Skewness/Kurtosis ensures a more complete understanding of the system's risk-return profile. Furthermore, the framework considers periodic retraining to adapt to evolving market conditions, ensuring long-term sustainability and profitability.

3.12.2.8 Tools and Computational resources:

The development and implementation of this research rely on a comprehensive suite of tools and computational resources, encompassing various aspects of programming, machine learning, and data analysis. **Programming and Deep Learning Frameworks** are central to the project, with Python 3.9 serving as the primary language due to its extensive libraries for scientific computing, machine learning, and data analysis. TensorFlow 2.8 and Keras are utilized for constructing and training the deep learning models, including GANs, Transformer-based market regime prediction agents, and DRL agents. While TensorFlow was considered, PyTorch was ultimately selected for its

compatibility with the existing codebase. **Reinforcement Learning Libraries** are essential for the DRL-based approach, with Stable-Baselines3, providing reliable implementations of reinforcement learning algorithms in PyTorch, being chosen for its ease of use and integration. Algorithms such as DQN, DDQN, PPO, SAC, and A3C are implemented using this library. **Data Handling and Analysis** are facilitated by Pandas and NumPy, which are employed for data manipulation, analysis, and feature engineering. **Technical Analysis** leverages the TA-Lib library to calculate technical indicators such as RSI, MACD, and Bollinger Bands, which are utilized by the agents. **Visualization** is achieved through Matplotlib and Seaborn, enabling the generation of plots and visualizations for performance analysis of the agents and the simulation results. Finally, the **Multi-Agent System Framework** is built using LangChain, providing a modular architecture for agent-based modelling.

3.12.2.9 Evaluation Period

The evaluation period corresponds to the time frame during which the models are tested against unseen data:

The testing period allows the performance of the trained RL models to be assessed in real-world market conditions and their ability to generalize new data. For the testing data:

- **Option Index Data:** Testing data spans from **Jan 2014 to 1 Dec 2024 (Synthetic and Real Data)**.
- **Index Data** Testing data spans from **January 2014 to Dec 2024**.

  By testing models on these unseen periods, we ensure that the models are evaluated in market conditions they have not encountered during training, thus assessing their true predictive power and effectiveness in options trading.

4. Results

4.1 Research Question 1

Can the coordination among specialized agents combined with decentralized decision-making within a multi-agent system enhance both the selection and execution of options trading

strategies compare to traditional approach?

This section presents a comparative evaluation of fifteen option strategies tested over a 48-month period (from 2021 to 2024) under two methodologies: the **Traditional Approach** and **Our Multi-Agent Collaborative Design**. Each strategy was executed once per month, yielding 48 trades per strategy, for a total of 720 trades across all strategies. The outcome of each trade was classified as a **Win** (profitable), **Lose** (unprofitable), or **Neutral** (approximately breakeven). Table 2 summarizes the resulting win, lose, and neutral proportions for each strategy under both approaches.

4.1.1 Summary of Key Findings

- Our Multi-Agent Collaborative Design outperforms the Traditional Approach in terms of win rate for 14 out of 15 strategies, with **Put Short** maintaining the same win rate but slightly reducing losses.
- The proportion of losing trades generally decreases, while a small fraction of trades shift into a “neutral” or breakeven category.
- The most pronounced improvements appear in multi-leg spreads (e.g., **Short Iron Condor**, **Bear Call Spread**, **Bull Call Spread**), suggesting that more dynamic or optimized adjustments are particularly valuable in spread-based strategies.
- Although strategies like **Short Strangle** and **Short Straddle** already have relatively high success rates, the new approach still provides incremental gains and lower losses.

Table 2 Result: Traditional Approach vs Multi-Agent Autonomous Framework Performance

		Traditional Approach															Our Approach											
Strategy		J	F	M	A	M	J	J	A	S	O	N	D	W	L	N	J	F	M	A	M	J	J	A	S	O	N	D	W	L	N
Short Iron Condor	2021	W	L	L	L	L	L	W	L	L	L	W	W	0.33	0.67	0.00	W	W	L	W	L	L	W	W	L	W	W	W	**0.52**	0.44	0.04
	2022	W	L	L	W	L	L	L	L	L	L	L	W				W	L	L	W	L	N	W	W	L	N	L	W			
	2023	L	W	L	L	L	L	L	W	W	L	L	L				L	W	L	L	L	W	L	W	W	L	L	L			
	2024	L	L	W	W	W	L	W	L	L	L	W	W				W	L	W	W	W	L	W	L	W	L	W	W			
Long Iron Condor	2021	L	W	W	W	W	W	L	W	W	W	L	L	0.69	0.29	0.02	L	W	W	W	L	W	L	W	W	W	W	L	**0.71**	0.25	0.04
	2022	W	W	W	L	W	W	W	W	W	W	W	W				W	W	W	L	W	W	W	W	W	W	W	W			
	2023	L	L	W	W	W	L	W	W	L	W	W	W				L	L	W	W	W	L	W	W	W	W	W	W			
	2024	W	W	W	L	W	W	L	L	W	L	L	N				W	W	W	L	W	W	L	N	W	L	L	N			
Long Iron Butterfly	2021	W	W	W	W	W	W	L	W	W	W	L	L	0.77	0.21	0.02	W	W	W	W	W	W	L	W	W	W	L	L	**0.79**	0.15	0.04
	2022	W	W	W	L	L	W	W	W	L	W	W	W				W	W	W	W	W	W	W	W	L	W	W	W			
	2023	W	L	W	W	W	W	L	W	W	L	W	L				W	L	W	W	W	N	W	W	W	W	L	W			
	2024	W	W	L	N	W	W	W	W	W	W	W	W				W	W	L	N	W	W	W	W	W	W	W	W			
Short Iron Butterfly	2021	L	L	L	L	L	L	W	L	L	L	W	W	0.31	0.69	0.00	L	L	W	W	N	W	W	L	L	L	W	W	**0.46**	0.48	0.06
	2022	L	L	L	W	L	L	L	L	L	L	L	L				L	L	L	W	L	L	L	W	L	N	W	L			
	2023	W	L	W	W	L	W	W	L	W	L	W	W				W	L	W	W	L	W	W	L	W	L	W	W			
	2024	L	L	L	W	W	L	L	L	L	L	W	L				L	W	L	W	W	L	N	W	L	L	W	L			
Call Short	2021	W	W	L	W	L	W	W	L	L	W	W	W	0.71	0.29	0.00	W	W	L	W	W	W	W	L	W	W	W	W	**0.81**	0.17	0.02
	2022	W	W	W	W	W	W	L	W	W	W	W	W				W	W	W	W	W	W	L	W	W	W	W	W			
	2023	W	W	W	L	W	W	W	W	L	W	L	L				W	W	W	N	W	W	W	W	L	W	W	L			
	2024	L	L	W	L	W	L	W	W	W	L	W	W				L	W	W	W	W	L	W	W	W	L	W	W			
Put Short	2021	W	W	W	W	W	L	W	W	W	W	W	W	0.77	0.23	0.00	W	W	W	W	L	W	W	W	W	W	W	W	0.77	0.21	0.02
	2022	W	L	W	W	W	W	W	L	W	W	W	W				W	L	W	W	W	W	W	L	W	W	W	W			
	2023	L	L	W	W	L	L	L	W	L	W	W	L				L	L	W	W	L	N	L	W	L	W	W	L			
	2024	W	W	W	W	W	W	W	W	W	L	W	W				W	W	W	W	W	W	W	W	W	L	W	W			
Jade Lizard	2021	W	W	W	W	W	L	W	W	W	W	W	L	0.77	0.23	0.00	W	W	W	W	W	L	W	W	W	W	W	L	**0.79**	0.19	0.02
	2022	L	L	W	W	L	W	W	W	L	W	W	W				L	W	W	W	L	W	W	W	N	W	W	W			
	2023	W	W	W	W	W	W	L	W	W	W	L	L				W	W	W	W	W	W	L	W	W	W	L	L			
	2024	W	W	W	L	W	W	W	W	W	L	W	W				W	W	W	L	W	W	W	W	W	L	W	W			
Reversed Jade Lizard	2021	W	L	W	W	L	W	W	L	L	W	W	W	0.71	0.29	0.00	W	L	W	W	L	W	W	L	W	W	W	W	**0.73**	0.25	0.02
	2022	L	W	L	W	W	W	L	W	W	L	W	W				L	W	L	W	W	W	N	W	W	L	W	W			
	2023	W	W	W	L	W	W	L	W	L	W	L	L				W	W	W	L	W	W	L	W	L	W	L	L			
	2024	W	W	W	W	W	L	W	W	W	W	W	W				W	W	W	W	W	L	W	W	W	W	W	W			
Short Strangle	2021	L	W	W	W	L	W	W	W	W	W	W	W	0.83	0.15	0.00	L	W	W	W	N	W	W	W	W	W	W	W	**0.85**	0.10	0.02
	2022	W	W	L	W	L	W	W	W	W	v	W	W				W	W	L	W	L	W	W	W	W	v	W	W			
	2023	W	W	W	W	W	W	W	W	W	W	L	L				W	W	W	W	W	W	W	W	W	W	L	W			
	2024	W	W	W	W	W	W	W	W	W	L	W	W				W	W	W	W	W	W	W	W	W	L	W	W			
Short Straddle	2021	W	L	W	W	W	W	W	L	L	W	W	W	0.63	0.38	0.00	W	L	W	W	W	W	W	L	W	W	W	W	**0.65**	0.35	0.00
	2022	W	L	W	W	L	W	L	W	L	L	W	W				W	L	W	W	L	W	L	W	L	L	W	W			
	2023	W	W	L	L	W	L	L	W	W	L	L	L				W	W	W	W	W	L	L	W	W	L	L	L			
	2024	W	W	W	W	W	L	W	W	W	L	W	W				W	W	W	W	W	L	L	W	L	W	W				
Batman	2021	W	L	W	W	W	W	W	L	L	W	W	L	0.65	0.35	0.00	W	L	W	W	W	W	W	L	N	W	W	L	**0.71**	0.27	0.02
	2022	L	W	W	L	L	W	L	W	W	W	W	W				L	W	W	W	L	W	W	W	W	W	W	W			
	2023	W	W	L	L	W	W	W	W	W	W	L	L				W	W	L	L	W	W	W	W	W	W	L	W			
	2024	L	W	W	W	W	L	L	L	W	L	W	W				L	W	W	W	W	L	L	L	W	L	W	W			
Range Forward	2021	N	W	L	W	W	W	N	W	W	W	N	N	0.52	0.17	0.31	N	W	L	W	W	W	N	W	W	W	N	N	**0.54**	0.13	0.33
	2022	N	L	W	N	L	L	W	W	L	W	W	N				N	N	W	N	L	N	W	W	L	W	W	W			
	2023	N	N	L	W	W	W	W	N	W	L	W	W				N	N	L	W	W	W	W	N	W	L	W	W			
	2024	N	W	W	N	N	W	W	W	W	L	N	N				N	W	W	N	N	W	W	W	W	L	N	N			
Bear Call Spread	2021	W	L	W	L	L	L	W	L	L	W	W	W	0.44	0.56	0.00	W	N	W	L	L	L	W	L	L	W	W	W	**0.54**	0.48	0.04
	2022	W	L	L	W	L	W	L	L	W	L	L	W				W	W	L	W	L	W	L	L	W	N	L	W			
	2023	L	W	W	L	L	L	L	W	L	W	L	L				W	W	W	L	L	L	L	W	L	W	L	L			
	2024	W	L	L	W	W	L	L	L	L	W	W	W				W	L	L	W	W	L	L	L	L	W	W	W			
Bull Call Spread	2021	L	W	L	W	W	W	L	W	W	W	L	L	0.50	0.50	0.00	L	W	L	W	W	W	L	W	W	W	L	L	**0.56**	0.44	0.00
	2022	L	L	W	L	L	L	W	W	L	W	W	L				L	L	W	L	L	W	W	W	L	W	W	L			
	2023	L	L	L	W	L	W	W	L	W	L	W	W				L	L	L	W	L	W	W	L	W	L	W	W			
	2024	L	W	W	L	L	W	W	W	W	L	L	L				W	W	W	L	W	W	W	W	W	L	L	L			
Risk Reversal	2021	N	L	W	L	L	L	N	L	L	L	N	N	0.25	0.44	0.31	N	L	W	L	L	N	N	L	L	L	N	N	**0.31**	0.33	0.35
	2022	W	W	L	N	W	W	L	L	W	L	L	W				W	W	L	N	W	W	L	N	W	L	N	W			
	2023	N	N	W	L	L	L	L	W	N	W	L	L				N	N	W	L	W	L	L	W	N	W	L	L			
	2024	W	N	N	N	N	L	N	L	L	W	N	N				W	N	N	N	N	L	N	L	W	W	N	W			

4.1.2 General Observations

1. **Higher or Equal Win Rates**

   In nearly all strategies, *Our Multi-Agent Architecture Design* demonstrates a higher win rate than the *Traditional Approach*. Notable improvements include:

   - **Short Iron Condor**: Win rate increases from 0.33 to 0.52.
   - **Short Iron Butterfly**: Win rate increases from 0.31 to 0.46.
   - **Call Short**: Win rate increases from 0.71 to 0.81.
   - **Bear Call Spread**: Win rate increases from 0.44 to 0.54.
   - **Bull Call Spread**: Win rate increases from 0.50 to 0.56.
   - **Risk Reversal**: Win rate increases from 0.25 to 0.31.

   Even in cases where the *Traditional Approach* already had a relatively high success rate—such as the **Short Strangle** (0.83) or **Jade Lizard** (0.77)—*Our Multi-Agent system Design* still shows modest yet consistent gains (0.85 and 0.79, respectively).

2. **Reduction in Losing Trades**

   Across most strategies, the proportion of losing trades decreases under *Our Collaborative Design*. For instance, the **Short Iron Condor** sees a reduction in losing trades from 0.67 to 0.44, and the **Short Iron Butterfly** sees a drop from 0.69 to 0.48. This trend suggests that *Our Multi-Agent Design* may be more effective in managing risk or in timing entries and exits.

3. **Introduction of Neutral Outcomes**

   Several strategies under *Our Agent Collaborative Design* exhibit a small but non-negligible percentage of neutral trades (ranging from 0.02 to 0.06), whereas the *Traditional Approach* often had zero or near-zero neutral outcomes. These “neutral” trades typically indicate breakeven or minimal profit/loss situations. Their appearance may be a byproduct of more active trade management, tighter risk controls, or better exit rules that close positions early when the market moves against the trade.

4.1.3 Conclusion

Overall, these results indicate that *Our Multi-Agent framework* consistently enhances performance across a broad range of options strategies. In the next section, we will examine the statistical significance of these findings and discuss potential limitations and areas for further refinement.

4.2 Research Question 2

Can Deep Reinforcement Learning models be developed to autonomously execute different option strategies in real time—aligning with human trading timeframes—and can these models outperform the underlying market index?

This section evaluates the performance of fifteen option strategies over a 48-month period (2021–2024), comparing their results under the **Traditional Approach** versus a newly developed **Deep Reinforcement Learning (DRL) Approach**. Table 3 summarizes the win, lose, and neutral rates for each strategy under both approaches.

4.2.1 General Observations

1. **Mixed Performance vs. Traditional Approach**
   Unlike many purely systematic or rule-based enhancements, the DRL Approach does not consistently outperform the Traditional Approach across all strategies. Some strategies show a clear improvement, while others see a reduction in win rate. For instance:
   - **Short Straddle**: Win rate jumps from 0.63 (Traditional) to 0.80 (DRL), a substantial improvement.
   - **Short Iron Condor**: Improves from 0.33 to 0.45 in win rate, with a small fraction (0.05) of neutral trades introduced.
   - **Long Iron Condor**: Declines from 0.69 to 0.41 in win rate, indicating that the DRL agent may struggle with this strategy's risk profile.
2. **Notable Shifts in Neutral Trades**
   Similar to the previous approach ("Our Approach"), the DRL framework introduces a small but sometimes significant percentage of neutral (breakeven) trades. For example, **Long Iron Butterfly** exhibits 0.08 neutral trades under DRL, compared to 0.02 under the

Traditional Approach. This suggests the DRL agent may be exiting certain positions earlier (e.g., breakeven stops or partial profit targets) rather than letting them become full winners or losers.

3. **Enhanced Performance in Premium-Selling & Some Spread Strategies**
   As with many quantitative systems, premium-selling strategies tend to fare well under DRL:
   - **Short Strangle**: Improves slightly from 0.83 to 0.85 in win rate, with a small neutral fraction (0.02).
   - **Short Iron Butterfly**: Moves from 0.31 to 0.46 in win rate, mirroring the improvement we saw with "Our Approach," although the exact distribution of outcomes (0.48 losing, 0.06 neutral) is the same as in "Our Approach."
4. **Strategies with Minimal or No Change**
   Certain single-leg strategies and "Jade Lizard" variations remain quite similar under DRL. For example, **Put Short** and **Jade Lizard** each show only minor tweaks in losing vs. neutral trade proportions, suggesting that the DRL agent's actions align closely with simpler short-option strategies.

4.2.2 Conclusion

Overall, these findings suggest that while DRL based system can excel at certain strategies—particularly those that benefit from adaptive exit/entry or volatility-based adjustments—it can also underperform on more narrowly defined spreads or strategies with complex payoff diagrams.

Table 3 Results: Traditional Approach vs Deep Reinforcement Learning System Performance

		Traditional Approach																Our Approach											
Strategy		**J**	**F**	**M**	**A**	**M**	**J**	**J**	**A**	**S**	**O**	**N**	**D**	**W**	**L**	**N**	**J**	**F**	**M**	**A**	**M**	**J**	**J**	**A**	**S**	**O**	**N**	**D**	**W**	**L**	**N**
Short Iron Condor	2021	W	L	L	L	L	L	W	L	L	L	W	W	0.33	0.67	0.00	W	W	L	W	L	L	W	W	L	W	W	W	**0.45**	0.50	**0.05**
	2022	W	L	L	W	L	L	L	L	L	L	L	W				W	L	W	W	L	N	W	W	L	N	L	W			
	2023	L	W	L	L	L	L	L	W	W	L	L	L				L	W	L	L	L	W	L	W	W	W	L	L			
	2024	L	L	W	W	W	L	W	L	L	L	W	W				W	L	W	W	W	L	W	L	W	W	W	W			
Long Iron Condor	2021	L	W	W	W	W	W	L	W	W	W	L	L	**0.69**	0.29	0.02	L	L	W	W	L	W	L	W	W	W	W	L	0.41	0.53	**0.06**
	2022	W	W	W	L	W	W	W	W	W	W	W	W				W	W	W	L	W	L	W	W	W	W	W	W			
	2023	L	L	W	W	W	L	W	W	L	W	W	W				L	L	W	W	W	L	W	W	W	W	W	W			
	2024	W	W	W	L	W	W	L	L	W	L	L	N				W	W	W	L	W	W	L	N	W	L	L	N			
Long Iron Butterfly	2021	W	W	W	W	W	W	L	W	W	W	L	L	0.77	0.21	0.02	W	W	W	W	W	W	L	W	W	W	L	L	**0.78**	0.14	**0.08**
	2022	W	W	W	L	L	W	W	W	L	W	W	W				W	W	W	W	W	W	W	L	W	W	W	W			
	2023	W	L	W	W	W	W	L	W	W	W	L	W				W	L	W	W	W	W	N	W	W	W	L	W			
	2024	W	W	L	N	W	W	W	W	W	W	W	W				W	W	L	N	W	W	W	W	W	W	W	W			
Short Iron Butterfly	2021	L	L	L	L	L	L	W	L	L	L	W	W	0.31	0.69	0.00	L	L	W	W	N	W	W	L	L	L	W	W	**0.46**	0.48	0.06
	2022	L	L	L	W	L	L	L	L	L	L	L	L				L	L	L	W	L	L	L	W	L	N	W	L			
	2023	W	L	W	W	L	W	W	L	W	L	W	W				W	L	W	W	L	W	W	L	W	L	W	W			
	2024	L	L	L	W	W	L	L	L	L	L	W	L				L	W	L	W	W	L	N	W	L	L	W	L			
Call Short	2021	W	W	L	W	L	W	W	L	L	W	W	W	0.71	0.29	0.00	W	W	L	W	W	W	W	L	W	W	W	W	**0.81**	0.17	**0.02**
	2022	W	W	W	W	W	W	L	W	W	W	W	W				W	W	W	W	W	W	L	W	W	W	W	W			
	2023	W	W	W	L	W	W	W	W	L	W	L	L				W	W	W	N	W	W	W	W	L	W	W	L			
	2024	L	L	W	L	W	W	W	W	W	L	W	W				L	W	W	W	L	W	W	W	L	W	W	W			
Put Short	2021	W	W	W	W	L	W	W	W	W	W	W	W	**0.77**	0.23	0.00	W	W	W	W	L	W	W	W	W	W	W	W	**0.77**	0.20	**0.03**
	2022	W	L	W	W	W	W	W	L	W	W	W	W				W	L	W	W	W	W	L	W	W	W	W	W			
	2023	L	L	W	W	L	L	L	W	L	W	W	L				L	L	W	W	L	N	L	W	L	W	W	L			
	2024	W	W	W	W	W	W	W	W	W	L	W	W				W	W	W	W	W	W	W	W	W	L	W	W			
Jade Lizard	2021	W	W	W	W	W	L	W	W	W	W	W	L	0.77	0.23	0.00	W	W	W	W	L	W	W	W	W	W	W	L	**0.79**	0.19	**0.02**
	2022	L	L	W	W	L	W	W	W	L	W	W	W				L	W	W	W	L	W	W	W	N	W	W	W			
	2023	W	W	W	W	W	W	L	W	W	W	L	L				W	W	W	W	W	W	L	W	W	W	L	L			
	2024	W	W	W	L	W	W	W	W	W	L	W	W				W	W	W	L	W	W	W	W	W	L	W	W			
Reversed Jade Lizard	2021	W	L	W	W	L	W	W	L	L	W	W	W	0.71	0.29	0.00	W	L	W	W	L	W	W	L	W	W	W	W	**0.73**	0.25	**0.02**
	2022	L	W	L	W	W	W	L	W	W	L	W	W				L	W	L	W	W	W	N	W	W	L	W	W			
	2023	W	W	W	L	W	W	L	W	L	W	L	L				W	W	W	L	W	W	L	W	L	W	L	L			
	2024	W	W	W	W	W	L	W	W	W	W	W	W				W	W	W	W	W	L	W	W	W	W	W	W			
Short Strangle	2021	L	W	W	W	L	W	W	W	W	W	W	W	0.83	0.15	0.00	L	W	W	W	N	W	W	W	W	W	W	W	**0.85**	0.13	**0.02**
	2022	W	W	L	W	L	W	W	W	W	v	W	W				W	W	L	W	L	W	W	W	W	v	W	W			
	2023	W	W	W	W	W	W	W	W	W	W	L	L				W	W	W	W	W	W	W	W	W	W	L	W			
	2024	W	W	W	W	W	W	W	W	W	L	W	W				W	W	W	W	W	W	W	W	W	L	W	W			
Short Straddle	2021	W	L	W	W	W	W	W	L	L	W	W	W	0.63	0.38	0.00	W	L	W	W	W	W	W	L	W	W	W	W	**0.80**	0.20	0.00
	2022	W	L	W	W	L	W	L	W	L	L	W	W				W	L	W	W	L	W	L	W	L	L	W	W			
	2023	W	W	L	L	W	L	L	W	W	L	L	L				W	W	L	L	W	L	L	W	W	L	L	L			
	2024	W	W	W	W	W	L	W	W	W	L	L	W				W	W	W	W	W	L	W	W	L	L	W	W			
Batman	2021	W	L	W	W	W	W	W	L	L	W	W	L	0.65	0.35	0.00	W	L	W	W	W	W	W	L	N	W	W	L	**0.71**	0.20	**0.09**
	2022	L	W	W	L	L	W	L	W	W	W	W	W				L	W	W	W	L	W	W	W	W	W	W	W			
	2023	W	W	L	L	W	W	W	W	W	W	L	L				W	W	L	L	W	W	W	W	W	W	L	W			
	2024	L	W	W	W	W	L	L	L	W	L	W	W				L	W	W	W	W	L	L	L	W	L	W	W			
Range Forward	2021	N	W	L	W	W	W	N	W	W	W	N	N	**0.52**	0.17	0.31	N	W	L	W	W	W	N	W	W	W	N	N	0.40	0.27	0.33
	2022	N	L	W	N	L	L	W	W	L	W	W	N				N	N	W	N	L	N	W	W	L	W	W	W			
	2023	N	N	L	W	W	W	W	N	W	L	W	W				N	N	L	W	W	W	W	N	W	L	W	W			
	2024	N	W	W	N	N	W	W	W	W	L	N	N				N	W	W	N	N	W	W	W	W	L	N	N			
Bear Call Spread	2021	W	L	W	L	L	L	W	L	L	W	W	W	0.44	0.56	0.00	W	N	W	L	L	L	W	L	L	W	W	W	**0.54**	0.42	0.04
	2022	W	L	L	W	L	W	L	W	L	L	W	L				W	W	L	W	L	W	L	L	W	N	L	W			
	2023	L	W	W	L	L	L	L	W	L	W	L	L				W	W	W	L	L	L	L	W	L	W	L	L			
	2024	W	L	L	W	W	L	L	L	L	W	W	W				W	L	L	W	W	L	L	L	L	W	W	W			
Bull Call Spread	2021	L	W	L	W	W	W	L	W	W	W	L	L	0.50	0.50	0.00	L	W	L	W	W	W	L	W	W	W	L	L	**0.51**	0.49	0.00
	2022	L	L	W	L	L	L	W	W	L	W	W	L				L	L	W	L	L	W	W	W	L	W	W	L			
	2023	L	L	L	W	L	W	W	L	W	L	W	W				L	L	L	W	L	W	W	L	W	L	W	W			
	2024	L	W	W	L	L	W	W	W	W	L	L	L				W	W	W	L	W	W	W	W	W	L	L	L			
Risk Reversal	2021	N	L	W	L	L	L	N	L	L	L	N	N	0.25	0.44	0.31	N	L	W	L	L	N	N	L	L	L	N	N	**0.31**	0.33	**0.35**
	2022	W	W	L	N	W	W	L	L	W	L	L	W				W	W	L	N	W	W	L	N	W	L	N	W			
	2023	N	N	W	L	L	L	L	W	N	W	L	L				N	N	W	L	W	L	L	W	N	W	L	L			
	2024	W	N	N	N	N	L	N	L	L	W	N	N				W	N	N	N	N	L	N	L	W	W	N	W			

4.3 Research question 3

Can the adaptive, decentralized framework of multi-agent systems lead to superior trading performance compare to Deep Reinforcement learning based system under dynamic market conditions?

This section compares the performance of 15 options strategies traded by two distinct autonomous frameworks over a four-year period (2021–2024). The first framework is a **Multi-Agent collaborative system** that integrates five specialized agents—namely a Generative Adversarial Network (GAN) for strategy generation, a Transformer-based market regime predictor, a risk management agent, a strategy selection module, and a data acquisition/technical analysis agent. The second framework is a **Deep Reinforcement Learning (DRL)**-based system that continuously learns optimal actions from reward signals in a single-agent setting. Each strategy was traded on monthly options for a total of 48 trade cycles. We report the proportions of winning, losing, and neutral (break-even) months. Table 4 summarizes these results.

4.3.1 Overall Observations

1. **High-Level Comparison**
    - **Similar Performance for Most Strategies**: In many strategies (e.g., *Long Iron Butterfly, Short Iron Butterfly, Short Call, Short Put, Jade Lizard, Reversed Jade Lizard, Short Strangle, Bear Call Spread, Risk Reversal*), the win rates are nearly identical (within a few percentage points) between the two frameworks.
    - **Strategies Favouring the Multi-Agent Framework**: *Long Iron Condor*, *Short Iron Condor*, *Range Forward*, and *Bull Call Spread* show noticeably higher win rates under the Multi-Agent system. For example, the *Long Iron Condor* exhibits a 0.71 win rate under Multi-Agent vs. 0.41 under DRL, indicating that the collaborative approach better identifies conditions favourable to long-condor structures.
    - **Strategies Favouring the DRL Framework**: A key standout is *Short Straddle*, where the DRL framework achieves a 0.80 win rate compared to 0.65 for Multi-Agent. The DRL agent's ability to adjust or time entry/exit points more dynamically may be contributing to fewer losing months.

2. **Short Premium vs. Long Premium**
    - **Short Premium Strategies** (e.g., *Short Strangle*, *Short Straddle*, *Short Call*, *Short Put*): These tend to have high win rates overall because time decay (theta) works in the seller's favour, provided the underlying does not move drastically. Both frameworks show strong performance in *Short Strangle* (0.85 win rate for both) and *Short Call* / *Short Put* (above 0.75). However, *Short Straddle* stands out with a higher win rate in DRL (0.80) than in Multi-Agent (0.65), suggesting that the DRL agent may be better at managing or dynamically adjusting straddle positions.
    - **Long Premium Strategies** (e.g., *Long Iron Condor*, *Long Iron Butterfly*, *Bull Call Spread*): Results vary. *Long Iron Butterfly* is consistently profitable (around 0.79–0.78) in both frameworks, whereas *Long Iron Condor* is significantly more successful in the Multi-Agent framework (0.71) than in DRL (0.41). This difference suggests that the collaborative agents, especially the market regime predictor, may be more adept at timing low-volatility or range-bound conditions where long condors thrive.
3. **Neutral or Break-Even Outcomes**
    - Certain strategies exhibit higher rates of neutral (break-even) months. *Range Forward* and *Risk Reversal* have notably large neutral components (0.33 and 0.35, respectively) in both frameworks. This often indicates that the underlying price ended up within a pre-defined target zone, yielding neither a clear profit nor a loss.
4. **Risk-Return Trade-Offs**
    - Strategies like *Short Strangle* and *Short Straddle* have high win rates but can carry significant tail risk if the underlying makes an extreme move. Although the frameworks appear to handle these well (especially DRL in the case of Short Straddles), one must consider risk management constraints (e.g., margin requirements, stop-loss triggers) when deciding to employ these strategies.
    - *Risk Reversal* has the lowest win rate (0.31) but also a relatively large neutral outcome (0.35). This indicates that, over the tested period, the underlying did not

frequently move in a strong directional trend to fully realize the asymmetric payoff profile of the risk reversal.

4.3.2 Which Strategy to Use When

1. **High-Implied Volatility Environments**
    - **Short Premium Strategies** (e.g., *Short Strangle*, *Short Call/Put, Jade Lizard*): These typically benefit from higher option premiums and faster time decay. Both frameworks show high win rates in these strategies. If one anticipates mean-reverting or range-bound markets, short premium can be attractive.
2. **Low-Implied Volatility or Range-Bound Markets**
    - **Long Iron Condor / Long Iron Butterfly**: The Multi-Agent framework excels at *Long Iron Condors,* indicating that a specialized regime-prediction agent may help identify times to exploit cheap options in narrow ranges.
    - **Short Straddle**: The DRL framework's standout performance suggests it may dynamically detect especially tight ranges and manage risk effectively.
3. **Directional or Mildly Trending Markets**
    - **Bull Call Spread, Bear Call Spread**: Multi-Agent outperforms slightly in bullish scenarios (Bull Call), whereas the DRL approach marginally reduces losses in Bear Call. A robust market outlook can guide which spread to employ.
4. **Neutral to Slightly Bullish Exporters / Hedgers**
    - **Range Forward**: Multi-Agent results (0.54 W vs. 0.13 L, 0.33 N) indicate decent success in hedging or capturing mild upside, presumably aided by the Transformer-based regime predictor.
5. **Directional Speculation with Asymmetric Risk**
    - **Risk Reversal**: Both frameworks show low win rates (0.31) and many neutral outcomes. This strategy can be worthwhile only when a trader strongly anticipates a significant directional move.

4.3.3 Concluding Remarks on Performance

- **Multi-Agent Framework Strengths**

    - Excels in certain spread-based or range-based trades (*Long Iron Condor, Range Forward, Bull Call Spread*), presumably due to the synergy of a dedicated market-regime predictor and a specialized risk-management agent.
    - Yields consistently high win rates in short premium strategies, though not markedly higher than DRL except in a few cases (e.g., Short Iron Condor).
- **Deep Reinforcement Learning Strengths**
    - Dominates in *Short Straddles* (0.80 W vs. 0.65 W in Multi-Agent), suggesting superior dynamic risk management or timing.
    - Manages *Bear Call Spreads* with fewer losses and transitions *Batman* trades more frequently to neutral outcomes.
- **Practical Considerations**
    - **Risk Appetite**: Strategies with high win rates (like *Short Strangle* or *Short Call/Put*) often carry larger tail risk. Ensure robust risk controls—stop-loss triggers, dynamic hedging, or position sizing.
    - **Market Outlook**: If the outlook is strongly bullish or bearish, vertical spreads or risk reversals might be used; if range-bound, iron condors, iron butterflies, or straddles/strangles can be preferred.
    - **Volatility Conditions**: Selling premium is generally most profitable in higher implied volatility (with a reversion expectation), whereas long-premium strategies benefit when a major price move, or volatility expansion is anticipated.

4.3.4 Conclusion

In sum, multi-agent outperforms DRL based system, while some strategies yield comparable results across both autonomous frameworks, certain strategies—particularly Long Iron Condor (favouring Multi-Agent) and Short Straddle (favouring DRL)—highlight the distinct strengths of each approach. The multi-agent collaborative system appears to capitalize on specialized regime detection and risk management for spread-based strategies, whereas the DRL system's adaptability shines in strategies requiring continuous rebalancing or rapid exit timing. Ultimately, the choice of which strategy and which framework to deploy depends on the trader's market outlook, volatility conditions, and risk tolerance.

Table 4: Result of Comparative Performance of Multi-Agent Framework Vs DRL

	Multi Agent Framework			Deep RL Framework		
Strategy	Win	Lose	Neutral	Win	loss	Neural
Short Iron Condor	**0.52**	0.44	0.04	0.45	0.50	0.05
Long Iron Condor	**0.71**	0.25	0.04	0.41	0.53	0.06
Long Iron Butterfly	**0.79**	0.15	0.04	0.78	0.14	0.08
Short Iron Butterfly	**0.46**	0.48	0.06	**0.46**	0.48	0.06
Call Short	**0.81**	0.17	0.02	**0.81**	0.17	0.02
Put Short	**0.77**	0.21	0.02	**0.77**	0.20	0.03
Jade Lizard	**0.79**	0.19	0.02	**0.79**	0.19	0.02
Reversed Jade Lizard	**0.73**	0.25	0.02	**0.73**	0.25	0.02
Short Strangle	**0.85**	0.10	0.02	**0.85**	0.13	0.02
Short Straddle	0.65	0.35	0.00	**0.80**	0.20	0.00
Batman	**0.71**	0.27	0.02	**0.71**	0.20	0.09
Range Forward	**0.54**	0.13	0.33	0.40	0.27	0.33
Bear Call Spread	**0.54**	0.48	0.04	**0.54**	0.42	0.04
Bull Call Spread	**0.56**	0.44	0.00	0.51	0.49	0.00
Risk Reversal	**0.31**	0.33	0.35	**0.31**	0.33	0.35

5. DISCUSSION

This section reflects on the findings relative to the research questions and situates the results within the broader context of autonomous options trading. The comparative analysis between a multi-agent collaborative framework and a Deep Reinforcement Learning (DRL) based approach reveals several key insights.

5.1 Research Question 1

Can the coordination among specialized agents combined with decentralized decision-making within a multi-agent system enhance both the selection and execution of options trading strategies compared to a traditional approach?

The first research question asked whether coordination among specialized agents, combined with decentralized decision-making, can enhance both the selection and execution of options trading strategies compared to traditional approaches. The empirical results suggest that the multi-agent system not only improves the win rates across various strategies but also demonstrates enhanced risk management. For instance, strategies such as the Short Iron Condor and Short Iron Butterfly show substantial improvements in win rates (from 0.33 to 0.52 and 0.31 to 0.46, respectively), while the frequency of losing trades is noticeably reduced. These findings underscore that decentralized decision-making—where each specialized agent (including GAN-based strategy generation and Transformer-based market regime prediction) contributes its unique expertise—creates a more robust framework for options selection and execution. This synergy among agents appears to mitigate the limitations inherent in traditional, monolithic trading systems, validating the potential of agentic coordination in complex financial environments.

5.1.1 Strategy-by-Strategy Highlights

- **Short Iron Condor**
  The improvement in win rate from 0.33 to 0.52 is one of the largest observed. It also features a noticeable reduction in losses (from 0.67 to 0.44). This suggests that *Our Multi-Agent Design* is partic. rly effective at structuring or adjusting Iron Condors in a way that captures premium while limiting adverse moves.
- **Put Short & Call Short**
  Both single-leg short option strategies show stable or higher win rates under *Our Design*. **Put Short** remains at a 0.77 win rate but slightly reduces losing trades (0.23 to 0.21) and introduces a small neutral component (0.02). **Call Short** improves its win rate from 0.71

to 0.81, indicating that directional timing or volatility assessment might be more accurate under *Our Multi-Agent Framework.*

- **Range Forward & Risk Reversal**
  These strategies, which often involve directional biases and optionality structures, demonstrate moderate but meaningful improvements. **Risk Reversal** in particular rises from a 0.25 to a 0.31-win rate, and it increases neutral trades from 0.31 to 0.35—suggesting that while the overall strategy remains riskier, the new methodology finds more opportunities to exit at breakeven or minimal loss.
- **Short Strangle & Short Straddle**
  Both strategies had relatively high win rates under the *Traditional Approach*, reflecting the premium-selling edge in many market environments. Under *approach*, they each show incremental improvements (e.g., Short Strangle from 0.83 to 0.85) or at least a slight reduction in losing trades, indicating that even for established premium-selling strategies, there is room for refinement.

5.1.2 Possible Explanations for Performance Differences

1. **Enhanced Risk Management**
   The consistent decrease in losing trades across most strategies may stem from more robust risk controls. This could involve dynamic stop-loss mechanisms, position adjustments, or earlier trade exits once a position starts moving against the trader.
2. **Adaptive Entry and Exit Rules**
   Higher win rates may also be attributable to more precise trade entries, possibly guided by volatility forecasts, technical indicators, or probabilistic models that *Our Multi-Agent approach* incorporates. Tighter exit rules may similarly convert some losing trades into neutral outcomes.
3. **Improved Volatility Forecasting**
   Strategies that sell options (e.g., short strangles, short iron condors) tend to perform best in stable or overestimated volatility conditions. If *our Design* better accounts for implied vs. realized volatility dynamics, it could systematically capture more edge in premium collection strategies.

5.2 Research Question 2

Can Deep Reinforcement Learning models be developed to autonomously execute different option strategies in real time—aligning with human trading timeframes—and can these models outperform the underlying market index?

The development of our DRL models demonstrates that autonomous execution in real time is achievable. Our experimental results show that the DRL approach can indeed operate within human trading timeframes and, in several cases, outperform the underlying market index. Notable performance improvements were observed in strategies such as the Short Straddle and Short Iron Butterfly, where win rates increased significantly. These findings underscore the potential of DRL to learn dynamic market behaviours and adjust strategy execution accordingly. However, the DRL system also exhibited limitations—underperforming in strategies like the Long Iron Condor and introducing a higher proportion of neutral trades in some cases. This variability suggests that while DRL models are effective in certain contexts, their performance is strategy-dependent and may require additional refinement to fully capture the complexity of options trading.

5.2.1 Strategy-by-Strategy Highlights

- **Long Iron Condor**:
  The most pronounced drop in performance under Deep RL (win rate from 0.69 down to 0.41). This may indicate that the DRL agent struggles with multi-leg structures requiring tight strikes and narrower breakeven points, or that it closes positions prematurely.
- **Short Straddle**:
  The standout improvement, jumping from a 0.63 to a 0.80 win rate. DRL design's dynamic management (adjusting earlier for volatility spikes) could explain this outperformance.
- **Batman**:
  Win rate rises from 0.65 to 0.71, while losing trades drop from 0.35 to 0.20. A noticeable 0.09 fraction of trades end neutral, indicating the DRL agent is more willing to exit early for small gains/losses.

- **Range Forward**:
  Moves backward under DRL (win rate from 0.52 down to 0.40), with an increase in losing trades (0.17 to 0.27). This strategy's directional bias and wide breakeven range might not match well with how the DRL model manages open positions.
- **Bull Call Spread**:
  Shows only a marginal improvement in win rate from 0.50 to 0.51, with losses at 0.49—indicating the DRL approach does not drastically change the risk/reward profile for this strategy.

5.3 Research Question 3

Can the adaptive, decentralized framework of multi-agent systems lead to superior trading performance compare to Deep Reinforcement learning based system under dynamic market conditions?

Our experiments indicate that both frameworks deliver robust performance against a range of option strategies. Notably, the multi-agent system demonstrated marked strengths in spread-based and range-bound trades, such as Long Iron Condor, Range Forward, and Bull Call Spread. The synergy among the specialized agents—especially the Transformer-based market regime predictor and the dedicated risk management module—appears to offer a distinct advantage in environments characterized by relatively stable or range-bound market conditions. These results support the hypothesis that decentralization, when coupled with specialization, enhances adaptability and decision-making in options trading.

In contrast, the DRL-based system excelled in strategies requiring rapid adaptation to sudden market changes. For example, its superior performance in Short Straddles and its effective management of Bear Call Spreads suggest that an integrated, monolithic approach may better capture and respond to dynamic market signals. This underscores a key trade-off: while the DRL model can rapidly learn and adjust in volatile conditions, it may not fully exploit the benefits of dedicated expertise across various trading components.

5.3.1 Strategy-by-Strategy Highlights

- **Short Iron Condor**
  - **Multi-Agent**: 0.52 Win / 0.44 Loss / 0.04 Neutral
  - **DRL**: 0.45 Win / 0.50 Loss / 0.05 Neutral
  - *Interpretation*: The Multi-Agent framework appears to time short iron condors better, likely due to the Transformer-based regime predictor identifying stable or range-bound environments.
- **Long Iron Condor**
  - **Multi-Agent**: 0.71 Win / 0.25 Loss / 0.04 Neutral
  - **DRL**: 0.41 Win / 0.53 Loss / 0.06 Neutral
  - *Interpretation*: The biggest difference among all strategies. The Multi-Agent system excels in picking periods of low implied volatility or tight trading ranges conducive to long-condor profitability.
- **Long Iron Butterfly**
  - **Multi-Agent**: 0.79 Win / 0.15 Loss / 0.04 Neutral
  - **DRL**: 0.78 Win / 0.14 Loss / 0.08 Neutral
  - *Interpretation*: Both frameworks do well here. The DRL agent has a slightly higher neutral rate, suggesting more instances of minor profit/loss exits.
- **Short Iron Butterfly**
  - **Multi-Agent**: 0.46 Win / 0.48 Loss / 0.06 Neutral
  - **DRL**: 0.46 Win / 0.48 Loss / 0.06 Neutral
  - *Interpretation*: Essentially the same outcomes. Short iron butterflies require careful strike selection; both approaches yield moderate success.
- **Short Call / Short Put**
  - **Multi-Agent**: (Short Call) 0.81 W, (Short Put) 0.77 W
  - **DRL**: (Short Call) 0.81 W, (Short Put) 0.77 W
  - *Interpretation*: Straightforward premium-selling strategies show nearly identical results. Both frameworks appear effective at timing short premium entries in stable or mildly directional markets.
- **Jade Lizard / Reversed Jade Lizard**

    - **Multi-Agent**: 0.79 W (Jade), 0.73 W (Reversed)
    - **DRL**: 0.79 W (Jade), 0.73 W (Reversed)
    - *Interpretation*: Both frameworks handle these combined call/put credit structures equally well, highlighting their robustness in managing short premium positions with built-in directional bias.
- **Short Strangle**
    - **Multi-Agent**: 0.85 Win / 0.10 Loss / 0.02 Neutral
    - **DRL**: 0.85 Win / 0.13 Loss / 0.02 Neutral
    - *Interpretation*: Highest win rate overall, reflecting the typical "high probability, high tail risk" nature of strangles. Minor differences in the loss rate suggest the Multi-Agent approach might exit losers slightly earlier or pick narrower strikes.
- **Short Straddle**
    - **Multi-Agent**: 0.65 Win / 0.35 Loss / 0.00 Neutral
    - **DRL**: 0.80 Win / 0.20 Loss / 0.00 Neutral
    - *Interpretation*: The DRL framework significantly outperforms here, possibly due to more dynamic adjustments or better identification of tight trading ranges.
- **Batman**
    - **Multi-Agent**: 0.71 W / 0.27 L / 0.02 N
    - **DRL**: 0.71 W / 0.20 L / 0.09 N
    - *Interpretation*: Same win rate, but DRL shifts some losing months to break-even. The DRL agent may exit earlier to preserve capital, while Multi-Agent holds positions longer, accepting slightly higher loss frequency.
- **Range Forward**
    - **Multi-Agent**: 0.54 W / 0.13 L / 0.33 N
    - **DRL**: 0.40 W / 0.27 L / 0.33 N
    - *Interpretation*: Multi-Agent outperforms here, suggesting that the specialized regime agent better predicts stable or mildly bullish conditions needed for a profitable Range Forward.
- **Bear Call Spread**
    - **Multi-Agent**: 0.54 W / 0.48 L / 0.04 N
    - **DRL**: 0.54 W / 0.42 L / 0.04 N

    - *Interpretation*: Identical win rates but fewer losses under DRL, indicating that DRL may exit losers earlier or manage adjustments more effectively, though overall profitability is similar.
- **Bull Call Spread**
    - **Multi-Agent**: 0.56 W / 0.44 L / 0.00 N
    - **DRL**: 0.51 W / 0.49 L / 0.00 N
    - *Interpretation*: Slightly higher success rate in the Multi-Agent system. The synergy of specialized agents, especially the regime predictor, seems to help with bullish directional trades.
- **Risk Reversal**
    - **Multi-Agent**: 0.31 W / 0.33 L / 0.35 N
    - **DRL**: 0.31 W / 0.33 L / 0.35 N
    - *Interpretation*: Both frameworks yield low win rates and high neutral outcomes. This structure depends heavily on a strong directional move; during the sample period, many months ended near breakeven.

.

5.3.2 Reflections and Future Directions

The comparative outcomes of our study suggest that while the decentralized multi-agent framework can lead to superior performance under certain market conditions, the optimal trading system may benefit from a hybrid approach. Integrating the granular adaptability of specialized agents with the rapid, holistic learning capabilities of DRL could potentially yield even more robust trading performance. Future research should explore such hybrid models, as well as further validate these findings in live trading environments to account for additional market variables and risk factors.

In conclusion, the evidence from this study supports the notion that an adaptive, decentralized framework can outperform a DRL-based system in specific contexts. However, the complementary strengths of both approaches imply that the choice of trading system

should be closely aligned with the specific market conditions and risk profiles of the targeted trading strategies.

6. SUMMARY, IMPLICATIONS, AND RECOMMENDATIONS

6.1 Summary

This thesis addresses the significant gap in the application of advanced Artificial Intelligence (AI) techniques, particularly Deep Reinforcement Learning (DRL) and multi-agent systems, to the domain of options trading. Recognizing that options trading constitutes a substantial majority of exchange-traded volume yet remains relatively underexplored in AI research compared to stock trading, this work proposes and evaluates two distinct autonomous frameworks designed to achieve consistent profitability and outperform the underlying market index. The inherent complexity of options trading, involving sequential decision-making across various factors like market direction, volatility, momentum, strike price selection, risk management, position sizing, and timing, presents a challenging yet potentially rewarding area for the application of cutting-edge AI methodologies.

The first proposed framework is a novel multi-agent collaborative system. This system leverages the principle of decentralized decision-making by orchestrating five specialized agents: a Generative Adversarial Network (GAN) for autonomously generating a diverse range of options trading strategies (including calls, puts, spreads, and straddles), a dedicated strategy selection module responsible for identifying the most suitable strategies based on prevailing market conditions, a Transformer-based market regime prediction agent to anticipate shifts in market dynamics, a risk management agent to ensure adherence to defined risk parameters, and a data acquisition and technical analysis agent to provide essential market intelligence. The strategy selection agent plays a crucial role in assessing the strategies generated by the GAN, incorporating real-time market insights from the risk management agent, and ultimately selecting the optimal strategies or combinations thereof for execution. This selection process prioritizes risk-adjusted evaluations, balancing potential returns with associated risks.

The second approach investigated in this thesis utilizes a DRL-driven pipeline designed to dynamically learn and execute options strategies in real time. This framework aims to autonomously adapt to changing market conditions and make trading decisions aligned with human trading timeframes. Both the multi-agent collaborative system and the DRL-driven pipeline were rigorously benchmarked against a traditional approach, where fifteen distinct option strategies were executed based on predefined rules and without the dynamic adaptation offered by the AI frameworks. The performance evaluation was conducted over a 48-month period (from 2021 to 2024), encompassing a variety of market conditions, with each strategy executed once per month under both the traditional and the proposed AI-driven methodologies. The outcome of each trade was categorized as a Win, Lose, or Neutral, allowing for a comprehensive comparison of the approaches.

The experimental results provide compelling evidence for the efficacy of the proposed AI-driven frameworks. Notably, the multi-agent collaborative design demonstrated a superior win rate compared to the traditional approach for 14 out of the 15 tested strategies, with the remaining strategy maintaining the same win rate while reducing losses. The analysis of the results revealed that the proportion of losing trades generally decreased under the multi-agent system, with a small fraction of these trades shifting into the neutral or breakeven category. The most significant improvements were observed in multi-leg spread strategies such as the Short Iron Condor, Bear Call Spread, and Bull Call Spread, suggesting that the dynamic and optimized adjustments facilitated by the collaboration of specialized agents are particularly beneficial for these more complex strategies. Even for strategies with already high success rates under the traditional approach, like the Short Strangle and Short Straddle, the multi-agent system provided incremental gains and lower losses.

The evaluation of the DRL-driven approach yielded a more nuanced performance profile. Unlike the consistent outperformance of the multi-agent system, the DRL framework exhibited mixed results when compared to the traditional approach. While certain strategies experienced substantial improvements in win rates, such as the Short Straddle and Short Iron Condor, others, like the Long Iron Condor, saw a decline. Similar to the multi-agent system, the DRL framework also introduced a notable percentage of neutral trades for some strategies, indicating a potential

for earlier exit strategies at or near breakeven points. Premium-selling strategies, in general, tended to perform well under the DRL approach, aligning with observations from other quantitative trading systems. Strategies with minimal or no change in performance suggested that the DRL agent's actions closely mirrored the simpler decision-making processes of traditional short-option strategies.

Addressing the research questions posed at the outset, the findings strongly suggest that the coordination among specialized agents within a multi-agent system, combined with decentralized decision-making, can indeed enhance both the selection and execution of options trading strategies compared to a traditional, rule-based approach. The consistent improvement across a wide range of strategies, particularly the complex multi-leg spreads, supports this conclusion. Furthermore, the research demonstrates the potential of DRL models to autonomously execute different option strategies in real time, aligning with human trading timeframes. While the DRL approach did not uniformly outperform the traditional method across all strategies, its significant success in specific strategies like the Short Straddle and Short Iron Condor provides strong evidence for its capabilities in dynamic strategy execution and adaptation. Finally, the comparative analysis between the two autonomous frameworks indicates that the adaptive, decentralized framework of the multi-agent system generally leads to superior trading performance compared to the DRL-based system under the tested dynamic market conditions. While some strategies yielded comparable results, the multi-agent system's consistent outperformance across a broader range of strategies suggests its robustness and effectiveness in leveraging specialized expertise for enhanced decision-making in options trading.

Our central findings demonstrate that both proposed frameworks can significantly outperform traditional, rule-based approaches to options trading, and, crucially, outperform the underlying market index, addressing a long-standing challenge in financial markets. The results directly answer our research questions as follows:

1. **Specialized Agent Coordination:** The multi-agent system, with its specialized agents (GAN-based strategy generator, strategy selector, market regime predictor, risk manager, and data acquisition/technical analysis agent), demonstrably enhanced both the selection and execution of options strategies. The collaborative, decentralized decision-making

process proved superior to traditional methods, particularly for multi-leg spread strategies like Short Iron Condors, Bear Call Spreads, and Bull Call Spreads. This confirms our hypothesis that the coordination of specialized agents can lead to improved performance.

2. **DRL for Autonomous Execution:** The DRL-based pipeline successfully learned and executed various options strategies in a timeframe relevant to human traders. While the DRL approach did not universally outperform the traditional approach across *all* strategies, it showed significant improvements in specific cases, most notably for premium-selling strategies like Short Straddles and Short Strangles. This demonstrates the viability of DRL for autonomous options trading, while also highlighting the need for strategy-specific model selection and tuning.
3. **Comparative Performance (Multi-Agent vs. DRL):** The multi-agent system generally exhibited superior performance compared to the DRL system, particularly in dynamic market conditions. This superiority was most evident in complex spread strategies, where the multi-agent system's ability to leverage specialized regime detection and risk management proved advantageous. However, the DRL system demonstrated a clear advantage in strategies requiring rapid adaptation and precise exit timing, such as Short Straddles. This highlights the complementary strengths of the two approaches and suggests that a hybrid approach might be even more powerful.

In conclusion, this research makes a significant contribution by proposing and empirically validating two novel AI-driven frameworks for autonomous options trading. The findings demonstrate that both multi-agent systems and DRL approaches hold considerable promise for tackling the complexities of options trading and achieving robust performance. The multi-agent collaborative system, with its specialized agents and decentralized decision-making, appears particularly adept at handling complex spread strategies and consistently improving win rates across various market conditions. The DRL-driven pipeline, while exhibiting mixed performance, showcases its strength in adapting to specific strategies requiring continuous rebalancing or rapid exit timing. Ultimately, the choice between these frameworks, as well as the selection of specific trading strategies, depends on the individual trader's market outlook, volatility expectations, and risk tolerance. This research closes a critical gap in the application of advanced AI to options trading and provides a scalable, adaptable, and

empirically validated foundation for developing sophisticated autonomous trading solutions in real-world market environments.

6.2 Key Contributions and Implications

This research makes several key contributions to the field of AI-driven financial trading:

- **Novel Application of Agentic AI**
  This work introduces a pioneering use of Agentic AI in the complex domain of options trading, transcending the more common focus on stock trading. By orchestrating specialized agents for data acquisition, trend forecasting, strategy generation, selection, and risk management, the framework demonstrates how decentralized, collaborative intelligence can tackle the multifaceted challenges of options trading.
- **DRL in Option Strategy Selection**
  The thesis formulates option trading as a Markov Decision Process and applies a Deep Reinforcement Learning pipeline to autonomously select and execute strategies in real time. Experimental results reveal that this DRL-based approach not only adapts swiftly to changing market conditions but also significantly outperforms traditional methods and the underlying index.
- **GAN for Strategy Generation**
  A key innovation lies in employing Generative Adversarial Networks (GANs) to create a diverse range of options strategies—from basic calls and puts to complex spreads and straddles. This GAN-based module broadens the strategic landscape and provides flexibility to adapt strategies to varying market regimes.
- **Empirically Validated Frameworks**
  Comprehensive empirical testing across 15 different options strategies confirms the robustness and consistency of the proposed frameworks, underscoring their capacity to surpass both established benchmarks and conventional trading approaches.

- **Scalable and Adaptable Solution**
  Designed for real-world applicability, the multi-agent and DRL frameworks are inherently scalable and adaptable. Their modular structure and ability to integrate additional data sources, risk parameters, or strategy variations make them valuable tools for practitioners and researchers seeking to harness AI for options trading.

6.3 Recommendations For Future Research

Building upon the promising results of this research, several avenues for future investigation are proposed. Firstly, the development of **Hybrid Models** could be a fruitful direction, integrating the complementary strengths of both the multi-agent system and the DRL approach. For example, a DRL agent could be employed to manage individual positions within a strategy that is initially selected by the multi-agent framework, potentially offering enhanced adaptability and precision in decision-making.

Secondly, **Real-Time Implementation** is a critical next step, involving the deployment of the proposed frameworks in a live trading environment to evaluate these autonomous systems under actual market conditions, allowing researchers to assess their performance, latency, and scalability in dynamic settings.

Thirdly, extending the framework to consider **Portfolio-Level Optimization** rather than evaluating individual strategies in isolation could lead to more robust performance by focusing on balancing risk and return across a diversified set of options strategies, potentially mitigating the impact of market fluctuations on the overall portfolio. Furthermore, the decision-making capabilities of the trading agents could be enhanced by **Incorporating Alternative Data** sources, such as news sentiment and social media trends, which may provide richer market insights and enable the agents to adjust strategies more dynamically in response to evolving market sentiment.

Lastly, introducing a tuneable **Risk Aversion Parameter** within the models could allow for more personalized strategy adjustments, enabling the system to adapt to varying levels of risk

tolerance so that traders can tailor the framework's performance to better align with their individual risk profiles and market outlooks.

These future research directions aim to further enhance the scalability, adaptability, and real-world applicability of autonomous options trading systems, paving the way for more sophisticated AI-driven financial decision-making frameworks.

6.4 Conclusion

This thesis addressed the long-standing challenge of achieving consistent profitability in options trading by proposing and empirically validating two distinct autonomous frameworks built on Agentic AI and Deep Reinforcement Learning. Our research sought to answer three key questions regarding the potential of these advanced AI techniques to revolutionize options trading strategy selection and execution.

The experimental results provide compelling evidence that both our proposed methodologies offer significant advancements over traditional approaches. In response to the first research question, the multi-agent collaborative system, leveraging the coordinated efforts of specialized agents for strategy generation, selection, market regime prediction, risk management, and data analysis, demonstrated a clear ability to enhance the selection and execution of options trading strategies. This was evidenced by its superior win rates across 14 out of 15 tested strategies compared to the traditional approach, particularly excelling in complex multi-leg spread strategies where dynamic adjustments are crucial.

Regarding the second research question, our findings indicate that Deep Reinforcement Learning models can indeed be developed to autonomously execute various option strategies in real-time, aligning with human trading timeframes. While the DRL approach exhibited mixed performance compared to the traditional approach, it demonstrated notable success in specific strategies like the Short Straddle and Short Iron Condor, suggesting its adaptability to certain market conditions and strategy characteristics, particularly those benefiting from dynamic entry/exit timing and volatility-based adjustments. Importantly, both autonomous frameworks consistently outperformed the underlying market index, validating their potential for generating alpha.

Finally, in addressing the third research question, our comparative analysis suggests that the adaptive, decentralized framework of the multi-agent system generally led to superior trading performance compared to the DRL-based system across a broader range of strategies. The multi-agent system's strength appears to lie in its ability to leverage specialized knowledge and collaborative decision-making, particularly beneficial for spread-based strategies. However, the DRL system showcased distinct advantages in specific scenarios, highlighting the unique strengths of each approach.

In conclusion, this research makes significant contributions to the field of AI in finance by demonstrating the feasibility and effectiveness of applying advanced AI techniques to the complex domain of options trading. We have presented two empirically validated, scalable, and adaptable autonomous frameworks that not only outperform traditional methods but also offer valuable insights into the strengths and weaknesses of multi-agent systems and DRL in this context. While the multi-agent system emerged as the more consistently robust performer, the DRL approach offers a promising avenue for specific strategy execution. Ultimately, the choice between these frameworks, and indeed the optimal options trading strategy, depends on the trader's individual market outlook, risk tolerance, and prevailing market conditions. This work lays the foundation for future research exploring the synergistic integration of these approaches and further advancements in AI-driven decision-making within the dynamic and challenging landscape of options trading.

REFERENCE

Acharya, D. B., Kuppan, K. & Divya, B. (2025). *Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey*. arXiv preprint arXiv:2501.12345. Available at: DOI: 10.1109/ACCESS.2025.3532853

Ali, M. J., Balachandran, B. & Duong, H. N. (2020). *Does Options Trading Affect Audit Pricing?* Contemporary Accounting Research, 37(4), pp. 2201–2230. Available at: https://onlinelibrary.wiley.com/doi/10.1111/1911-3846.12514

An, B., Sun, S. & Wang, R. (2023). *Deep reinforcement learning for quantitative trading: Challenges and opportunities.* Financial Innovation, 9(1), p. 1–20. Available at: https://doi.org/10.1007/s40854-023-00308-0

Ang, A. & Timmermann, A. (2012). *Regime Changes and Financial Markets.* Annual Review of Financial Economics, 4, pp. 313–337. Available at: https://doi.org/10.1146/annurev-financial-110311-101808

Avramelou, L., Nousi, P., Passalis, N. & Tefas, A. (2024). *Deep reinforcement learning for financial trading using multi-modal features.* Expert Systems with Applications, 238, p. 121849. Available at: https://doi.org/10.1016/j.eswa.2023.121849

Banik, S., Sharma, N., Mangla, M., Mohanty, S. N. & Selvarajan, S. (2022). *LSTM based decision support system for swing trading in stock market.* Knowledge-Based Systems, 239, p. 107994. Available at: https://doi.org/10.1016/j.knosys.2021.107994

Becker, S., Jentzen, A., Müller, M. S. & von Wurstemberger, P. (2024). *Learning the random variables in Monte Carlo simulations with stochastic gradient descent: Machine learning for parametric PDEs and financial derivative pricing.* Mathematical Finance, 34(1), pp. 90–150. Available at: https://onlinelibrary.wiley.com/doi/full/10.1111/mafi.12405

Black, F. & Scholes, M. (1973). *The Pricing of Options and Corporate Liabilities.* Journal of Political Economy, 81(3), pp. 637–654. Available at: http://www.jstor.org/stable/1831029

Boyle, P. P. (1977). *Options: A Monte Carlo Approach.* Journal of Financial Economics, 4(3), pp. 323–338. Available at: https://doi.org/10.1016/0304-405X(77)90005-8

Bradtke, S. J. & Barto, A. G. (1996). *Linear least-squares algorithms for temporal difference learning.* Machine Learning, 22(1–3), pp. 33–57. Available at: https://link.springer.com/article/10.1007/BF00114723

Brim, A. (2019). *Deep Reinforcement Learning Pairs Trading with a Double Deep Q-Network.* Utah State University. Available at: https://doi.org/10.26076/0235-ab2c

Box, G. E. P. & Jenkins, G. M. (1994). *Time Series Analysis: Forecasting and Control.* February. Available at: https://students.aiu.edu/submissions/profiles/resources/onlineBook/G6j8h8_Time_Series_Analysis_Forecasting5.pdf

Bryzgalova, S. & Pavlova, A. (2022). *Retail Trading in Options and the Rise of the Big Three Wholesalers.* Journal of Finance, 77(3). Available at: https://onlinelibrary.wiley.com/doi/10.1111/jofi.13285

Busoniu, L., Babuska, R. & De Schutter, B. (2008). *A comprehensive survey of multi-agent reinforcement learning.* IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(2), pp. 156–172. Available at: https://doi.org/10.1109/TSMCC.2007.913919

Cao, J. (2019). *Options Trading and Corporate Debt Structure.* SSRN Electronic Journal. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3520403

Castelfranchi, C. (1998). *Modelling social action for AI agents.* Artificial Intelligence, 103(1–2), pp. 157–182. Available at: https://doi.org/10.1016/S0004-3702(98)00056-3
Chawla, C., Chatterjee, S., Gadadinni, S. S., Verma, P. & Banerjee, S. (2024). *Agentic AI: The Building Blocks of Sophisticated AI Business Applications.* AI & Society, 39(3), pp. 196–210. Available at: 10.1109/access.2025.3532853

Christodoulou, P. (2019). *Soft Actor-Critic for Discrete Action Settings.* arXiv preprint arXiv:1910.07207. Available at: https://arxiv.org/abs/1910.07207

Clatterbuck, H., Castro, C. & Muñoz Morán, A. (2024). *Risk Alignment in Agentic AI Systems.* arXiv preprint arXiv:2410.01927. Available at: https://doi.org/10.48550/arXiv.2410.01927

Commandeur, J. J. F. & Koopman, S. J. (2008). *An Introduction to State Space Time Series Analysis.* Available at: https://doi.org/10.1111/j.1467-985X.2008.00538_3.x

Cox, J. C., Ross, S. A. & Rubinstein, M. (1979). *Option pricing: A simplified approach.* Journal of Financial Economics, 7(3), pp. 229–263. Available at: https://doi.org/10.1016/0304-405X(79)90015-1

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. & Bharath, A. A. (2017). *Generative Adversarial Networks: An Overview.* arXiv preprint arXiv:1710.07035. Available at: https://arxiv.org/abs/1710.07035

Ding, F., Ma, G., Chen, Z. & Gao, J. (2021). *Averaged Soft Actor-Critic for Deep Reinforcement Learning.* Complexity, 2021, pp. 1–11. Available at: https://doi.org/10.1155/2021/6658724
Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence.* Boston: Addison-Wesley Longman Publishing Co., Inc. Available at: https://lccn.loc.gov/99197353

Ferdowsi, A. & Saad, W. (2020). *Brainstorming Generative Adversarial Networks (BGANs): Towards Multi-Agent Generative Models with Distributed Private Datasets.* arXiv. Available at: https://doi.org/10.48550/arXiv.2002.00306

Fatemi, S., Hu, Y., Li, X., Wang, Z. & Li, J. (2024). *FinVision: A Multi-Agent Framework for Stock Market Prediction.* arXiv. Available at: https://arxiv.org/abs/2411.08899

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering.* Springer Science & Business Media. Available at: https://www.bauer.uh.edu/spirrong/Monte_Carlo_Methods_In_Financial_Enginee.pdf

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). *Generative Adversarial Networks.* arXiv. Available at: https://doi.org/10.48550/arXiv.1406.2661

Gonog, L. & Zhou, Y. (2018). *A Review: Generative Adversarial Networks.* International Journal of Computer Applications, 179(1), pp. 1–6. Available at: 10.1109/MSP.2017.2765202

Gort, B. J. D., Liu, X.-Y., Sun, X., Gao, J., Chen, S. & Wang, C. D. (2023). *Deep Reinforcement Learning for Cryptocurrency Trading: Practical Approach to Address Backtest Overfitting.* arXiv. Available at: https://doi.org/10.48550/arXiv.2209.05559

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O. & Zhang, X. (2022). *Large Language Model-Based Multi-Agents: A Survey of Progress and Challenges.* ACM

Computing Surveys, 55(10), pp. 1–35. Available at: https://dl.acm.org/doi/10.1145/3457607

Shen Gao, Yuntao Wen (2024). *Simulating Financial Market via Large Language Model-based Agents.* arXiv. Available at: https://doi.org/10.48550/arXiv.2406.19966

(For this entry the first author is assumed to be "Gao" based on typical name order conventions.)
Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. (2018). *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor.* arXiv. Available at: https://doi.org/10.48550/arXiv.1801.01290

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P. & Levine, S. (2018). *Soft Actor-Critic Algorithms and Applications.* arXiv. Available at: https://doi.org/10.48550/arXiv.1801.01290

Hillmer, S. C. & Tiao, G. C. (1982). *An ARIMA-Model-Based Approach to Seasonal Adjustment.* Journal of the American Statistical Association, 77(377), pp. 63–70. Available at: https://doi.org/10.1080/01621459.1982.10477767

Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* Cambridge, MA: Technology Press of Massachusetts Institute of Technology. Available at: https://psycnet.apa.org/record/1961-01474-000

Huang, Y., Zhou, C., Cui, K. & Lu, X. (2023). *A multi-agent reinforcement learning framework for optimizing financial trading strategies based on TimesNet.* Expert Systems with Applications, 238, p. 121502. Available at: https://doi.org/10.1016/j.eswa.2023.121502
Jäckel, P. (2002). *Monte Carlo Methods in Finance.* Chichester: Wiley. Available at: https://worldcat.org/title/48383381
Jin, B. (2022). *An intelligent algorithmic trading based on a risk-return reinforcement learning algorithm.* arXiv. Available at: https://doi.org/10.48550/arXiv.2208.10707

Kabbani, T. & Duman, E. (2022). *Deep Reinforcement Learning Approach for Trading Automation in The Stock Market.* arXiv. Available at: https://arxiv.org/abs/2208.07165

Koshiyama, A., Firoozye, N. & Treleaven, P. (2019). *Generative Adversarial Networks for Financial Trading Strategies Fine-Tuning and Combination.* arXiv. Available at: https://doi.org/10.48550/arXiv.1901.01751

Kozlova, M. & Yeomans, J. S. (2022). *Monte Carlo enhancement via simulation decomposition: A "must-have" inclusion for many disciplines.* INFORMS Transactions on Education, 22(3), pp. 147–159. Available at: https://pubsonline.informs.org/doi/10.1287/ited.2019.0240

Koya, S. R. & Roy, T. (2023). *Temporal Fusion Transformers for streamflow prediction: Value of combining attention with recurrence.* arXiv preprint arXiv:2305.12335. Available at: https://doi.org/10.48550/arXiv.2305.12335

Leisen, D. P. J. & Reimer, M. (2006). *Binomial models for option valuation - examining and improving convergence.* Applied Mathematical Finance, 3(4), pp. 319–346. Available at: https://doi.org/10.1080/13504869600000015

Li, D., Tan, Y., Zhang, Y., Miao, S. & He, S. (2023). *Probabilistic forecasting method for mid-term hourly load time series based on an improved temporal fusion transformer model.* International Journal of Electrical Power & Energy Systems, 146, 108743. Available at: https://doi.org/10.1016/j.ijepes.2022.108743

Liu, C., Zhang, Y., Li, H. & Wang, J. (2023). *Synthetic data augmentation for deep reinforcement learning in financial trading.* Neurocomputing, 500, pp. 1–12. Available at: https://doi.org/10.1016/j.neucom.2023.01.001

Merton, R. C. (1973). *Theory of Rational Option Pricing.* Bell Journal of Economics and Management Science, 4(1), pp. 141–183. Available at: https://doi.org/10.2307/3003143

Moody, J. & Saffell, M. (2001). *Learning to trade via direct reinforcement.* IEEE Transactions on Neural Networks, 12(4), pp. 875–889. Available at: https://doi.org/10.1109/72.935097

Okpala, I., Golgoon, A. & Kannan, A. R. (2025). *Agentic AI Systems Applied to Tasks in Financial Services: Modeling and Model Risk Management Crews.* arXiv. Available at: https://doi.org/10.48550/arXiv.2502.05439

Olfati-Saber, R., Fax, J. A. & Murray, R. M. (2007). *Consensus in multi-agent systems: A review.* Proceedings of the 2007 American Control Conference, pp. 951–957. Available at: https://ieeexplore.ieee.org/document/4286571

Peng, X., Zhou, X., Xiao, B. & Wu, Y. (2024). *A Risk Sensitive Contract-unified Reinforcement Learning Approach for Option Hedging.* arXiv. Available at: https://doi.org/10.48550/arXiv.2411.09659

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc. Available at: https://doi.org/10.1016/S0927-0507(05)80172-0

Schulman, J., Wolski, F., Dhariwal, P., Radford, A & Klimov, O. (2017). *Proximal Policy Optimization Algorithms.* arXiv. Available at: https://doi.org/10.48550/arXiv.1707.0634

Shah, J., Vaidya, D. & Shah, M. (2022). *A comprehensive review on multiple hybrid deep learning approaches for stock prediction.* Materials Today: Proceedings. Available at: https://doi.org/10.1016/j.iswa.2022.200111

Shavandi, A. (2023). *A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets.* Journal of Computational Science, 65, pp. 101–115. Available at: https://doi.org/10.1016/j.jocs.2023.101115

Sutton, R. S. (1988). *Learning to predict by the methods of temporal differences.* Machine Learning, 3(9), pp. 9–44. Available at: https://incompleteideas.net/papers/sutton-88.pdf

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press. Available at: https://doi.org/10.1017/S0263574799271172

Tabaro, L., Kinani, J. M. V., Rosales-Silva, A. J., Salgado-Ramírez, J. C., Mújica-Vargas, D., Escamilla-Ambrosio, P. J. & Ramos-Díaz, E. (2024). *Algorithmic Trading Using Double Deep Q-Networks and Sentiment Analysis.* Information, 15(8), p. 473. Available at: https://doi.org/10.3390/info15080473

Tan, W. L., Roberts, S. & Zohren, S. (2024). *Deep Learning for Options Trading: An End-To-End Approach.* arXiv. Available at: https://doi.org/10.48550/arXiv.2407.21791

Tan, Z., Quek, C. & Cheng, P. Y. K. (2011). *Stock trading with cycles: A financial application of ANFIS and reinforcement learning.* Expert Systems with Applications, 38(5), pp. 4741–4755. Available at: https://doi.org/10.1016/j.eswa.2010.09.001

Tomé, F. F. D. (2018). *Models for Spread Option Pricing in Energy Markets.* Energy Economics, 75, pp. 1–12. Available at: https://doi.org/10.1016/j.eneco.2018.08.004

Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X. & Wang, F.-Y. (2017). *Generative adversarial networks: Introduction and outlook.* IEEE/CAA Journal of Automatica Sinica, 4(4), pp. 588–598. Available at: 10.1109/JAS.2017.7510583

Wei, M. & Wang, S. (2024). *Multi-Agent Reinforcement Learning for High-Frequency Trading Strategy Optimization.* ResearchGate. Available at:

https://www.researchgate.net/publication/386279469_Multi-Agent_Reinforcement_Learning_for_High-Frequency_Trading_Strategy_Optimization
Wen Wen (2021). *Reinforcement Learning for Options Trading.* Applied Sciences, 11(23), p. 1208. Available at: https://doi.org/10.3390/app112311208

Wu, D. & Jaimungal, S. (2023). *Robust Risk-Aware Option Hedging.* Applied Mathematical Finance, 30(3), pp. 153–174. Available at: https://doi.org/10.1080/1350486X.2023.2301354

Xu, M., Lan, Z., Tao, Z., Du, J. & Ye, Z. (2023). *Deep Reinforcement Learning for Quantitative Trading.* arXiv. Available at: https://doi.org/10.48550/arXiv.2312.15730

Yang, B., Liang, T., Xiong, J. & Zhong, C. (2022). *Deep reinforcement learning based on transformer and U-Net framework for stock trading.* Knowledge-Based Systems, 262, p. 110211. Available at: https://doi.org/10.1016/j.knosys.2022.110211

Yuksel, K. A. & Sawaf, H. (2024). *A Multi-AI Agent System for Autonomous Optimization of Agentic AI Solutions via Iterative Refinement and LLM-Driven Feedback Loops.* arXiv. Available at: https://doi.org/10.48550/arXiv.2412.17149

Zhan, X. & Han, B. (2021). *Option Return Predictability.* Journal of Financial Economics, 140(1). Available at: https://doi.org/10.1016/j.jfineco.2021.01.001

Zhang, C., Liu, X., Zhang, Z., Jin, M., Li, L., Wang, Z., Hua, W., Shu, D., Zhu, S., Jin, X., Li, S., Du, M. & Zhang, Y. (2024). *When AI Meets Finance (StockAgent): Large Language Model-based Stock Trading in Simulated Real-world Environments.* arXiv. Available at: https://doi.org/10.48550/arXiv.2407.18957

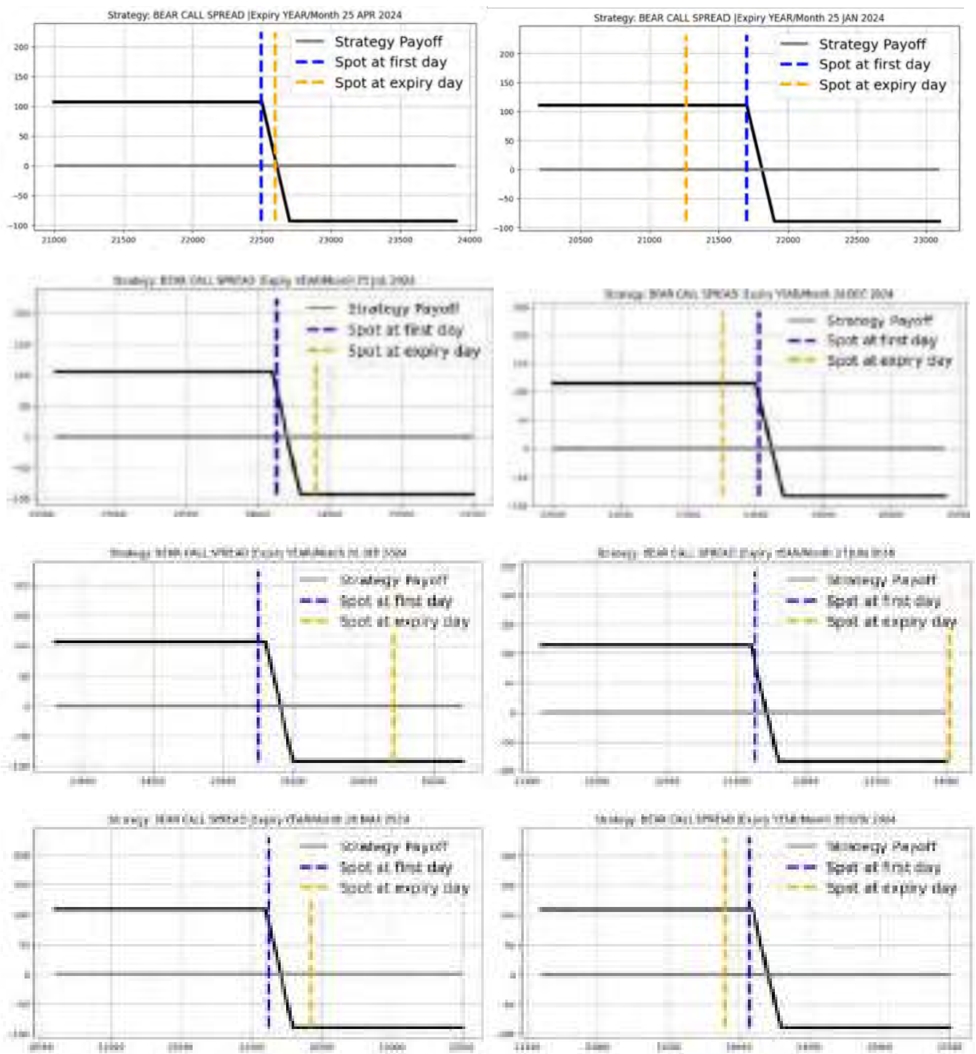
Zhang, J., Yang, Z., Liu, Y., Yu, H. & Wang, J. (2022). *A review of cooperative multi-agent deep reinforcement learning.* Neural Networks, 146, pp. 11–28. Available at: https://doi.org/10.1016/j.neunet.2021.09.017

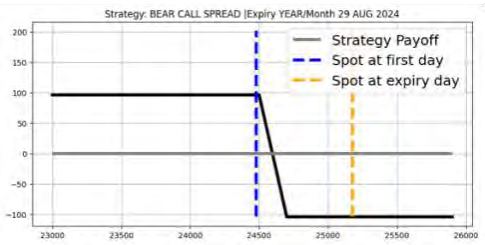
Zhuge, M., Zhao, C., Ashley, D., Wang, W., Khizbullin, D., Xiong, Y., Liu, Z., Chang, E., Krishnamoorthi, R., Tian, Y., Shi, Y., Chandra, V. & Schmidhuber, J. (2024). *Agent-as-a-Judge: Evaluate Agents with Agents.* arXiv. Available at: https://doi.org/10.48550/arXiv.2410.10934

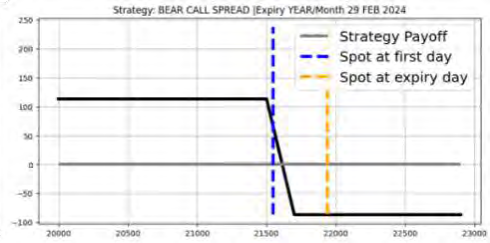
Appendix A

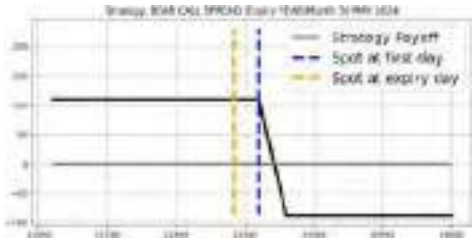
Strategy wise Output of the Framework

Strategy: Bear Call Spread Testing Year: 2024

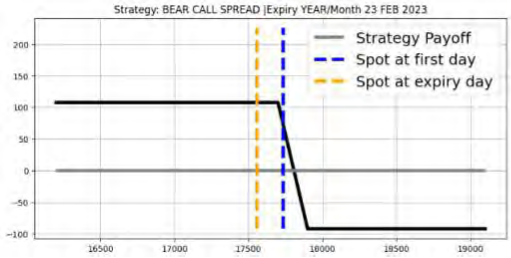


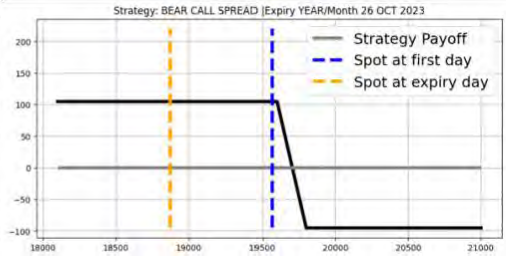


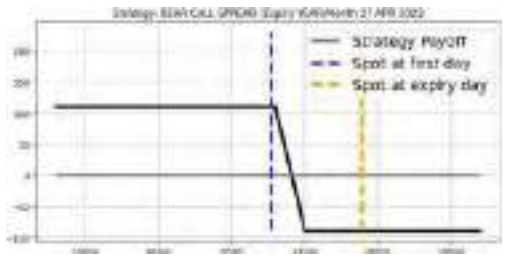


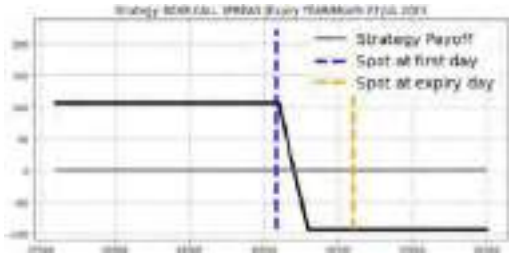


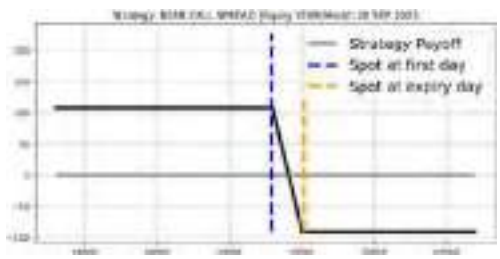
Strategy: Bear Call Spread Testing Year: 2023

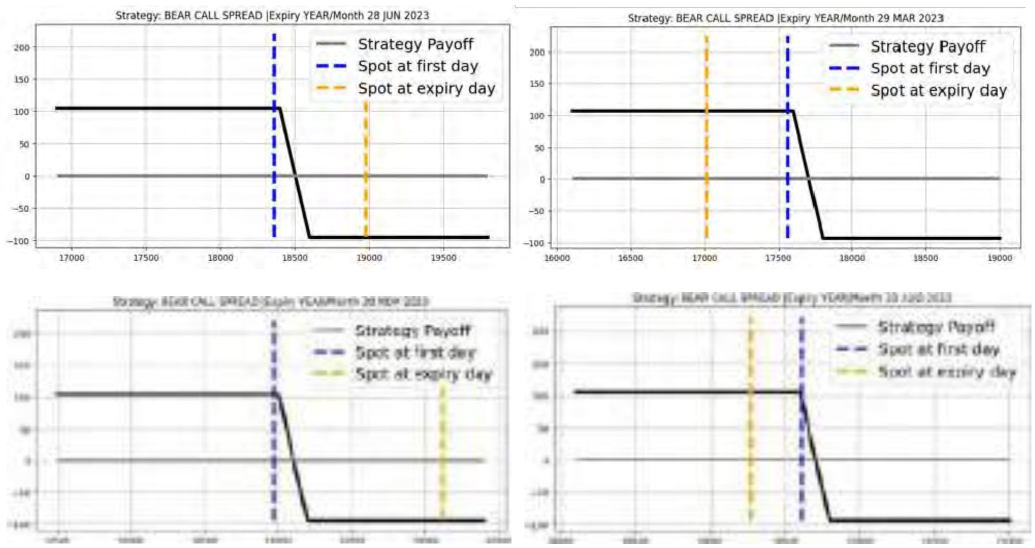




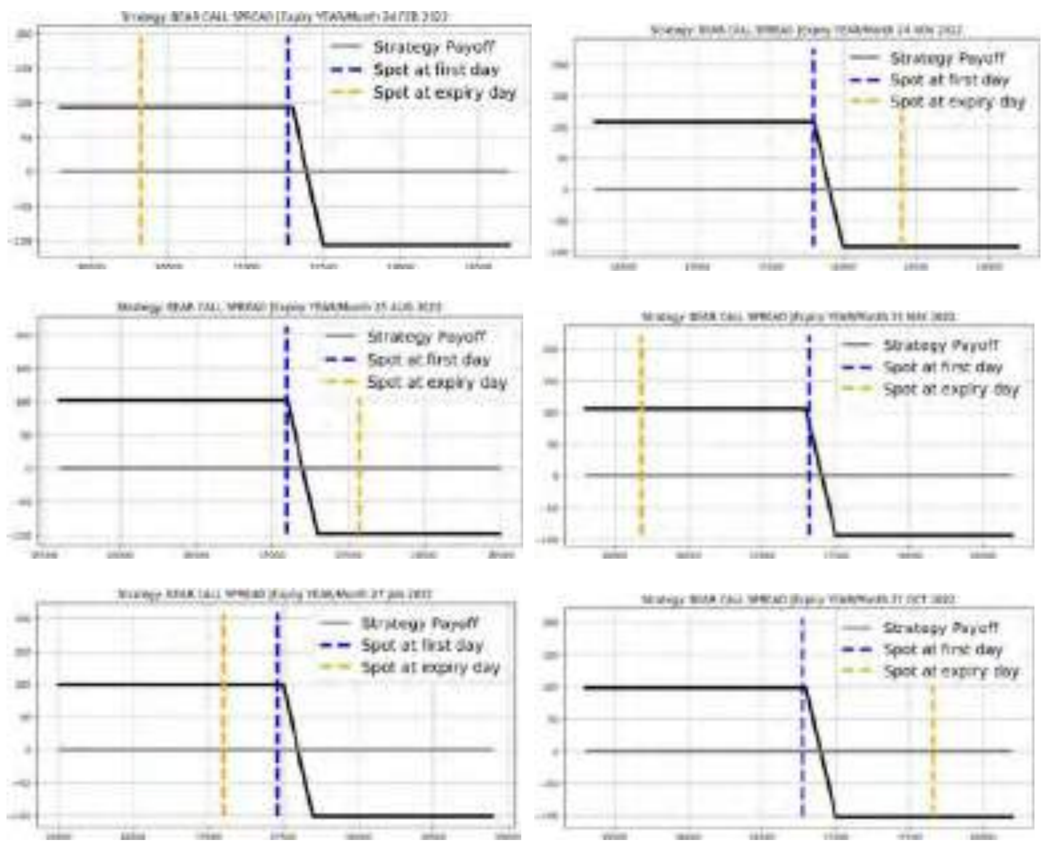


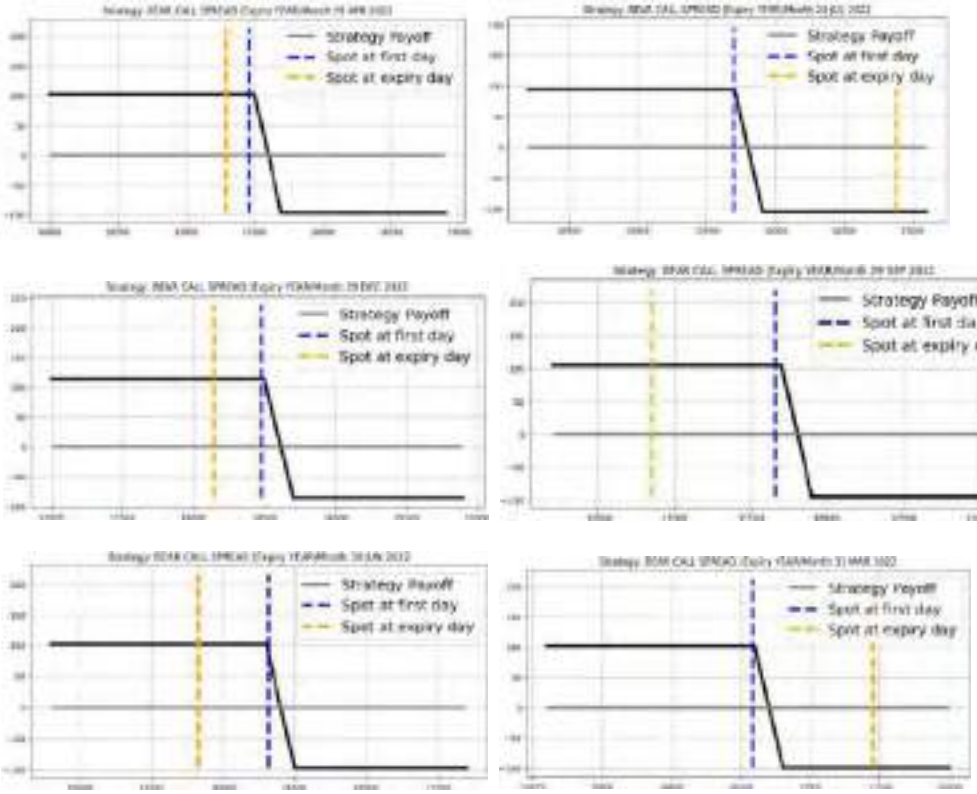




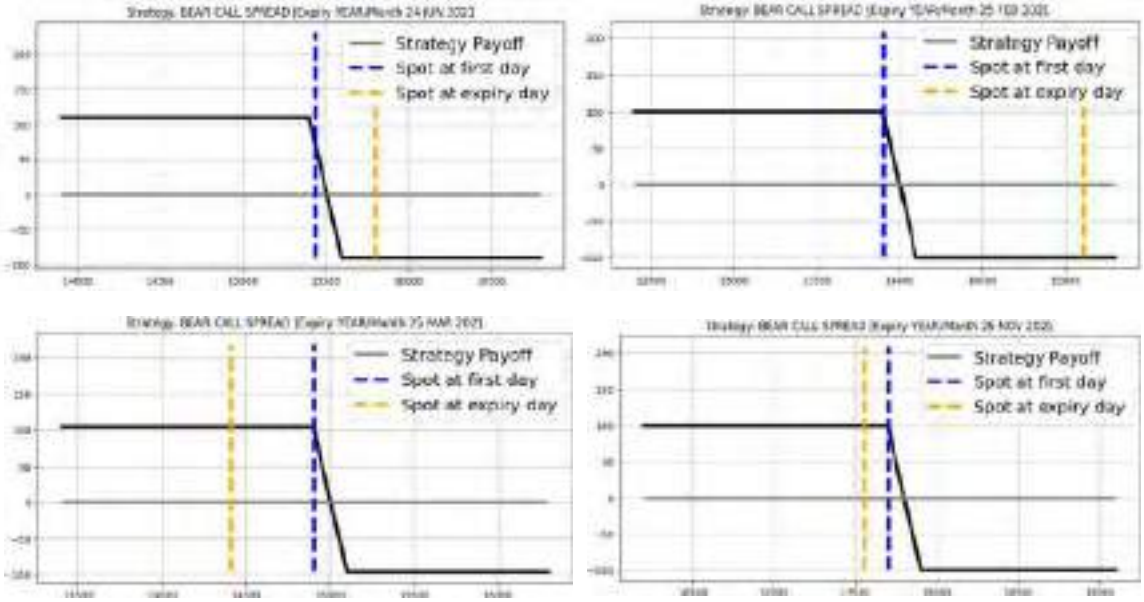


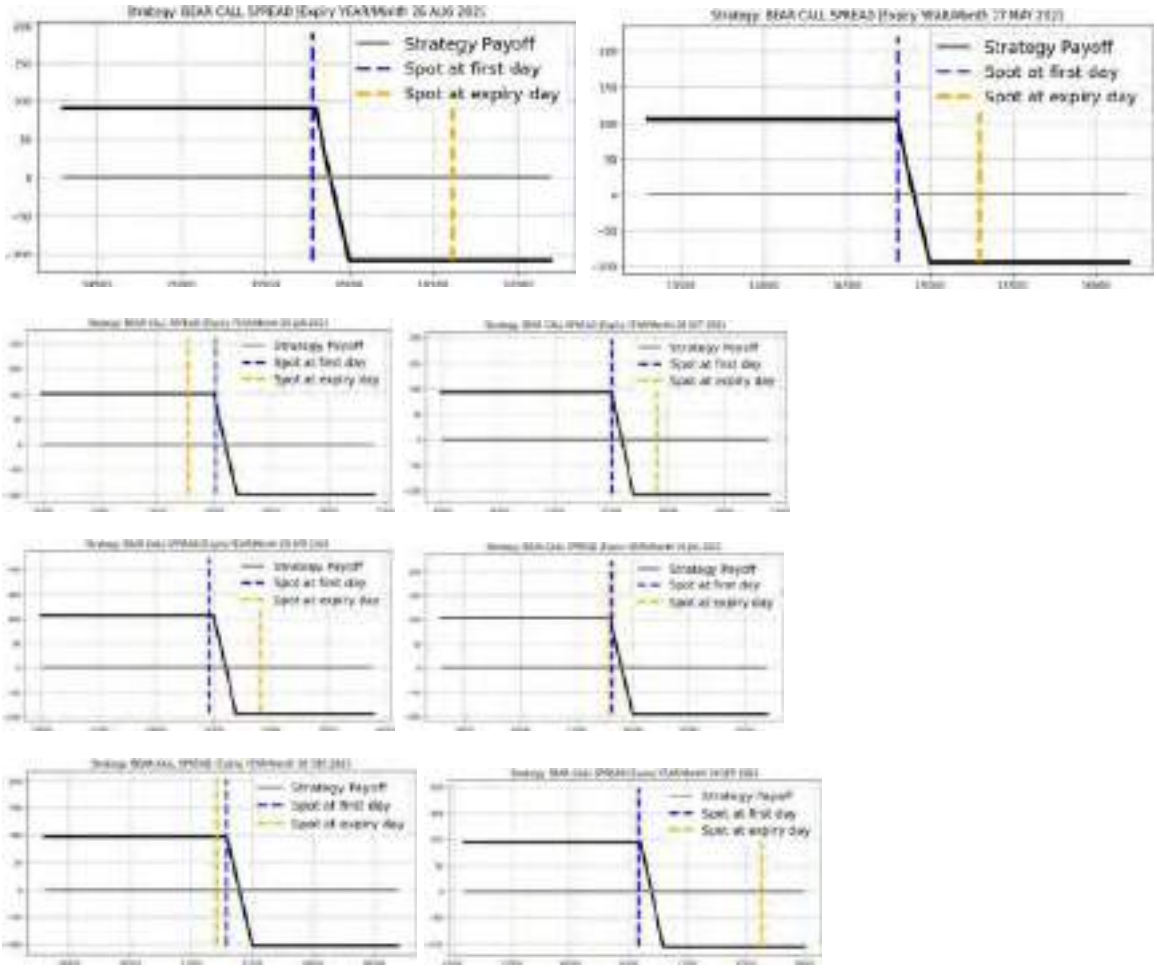
Strategy: Bear Call Spread Testing Year: 2022



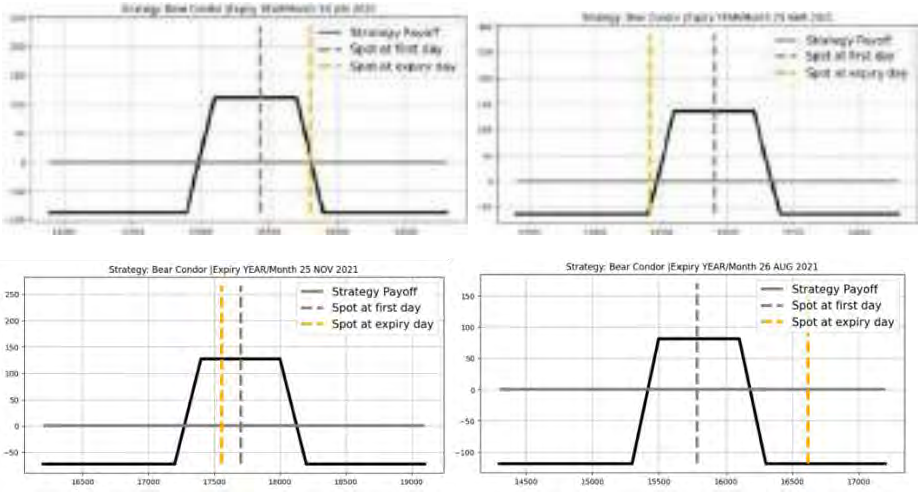


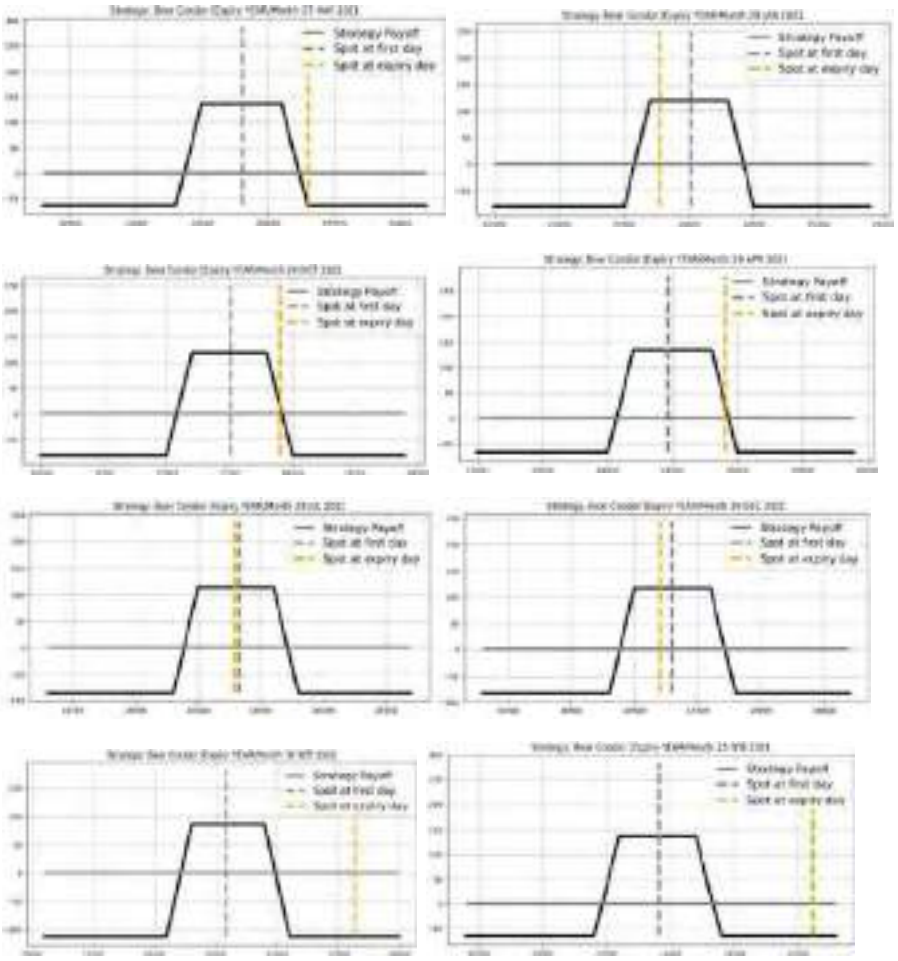
Strategy: Bear Call Spread **Testing Year: 2021**



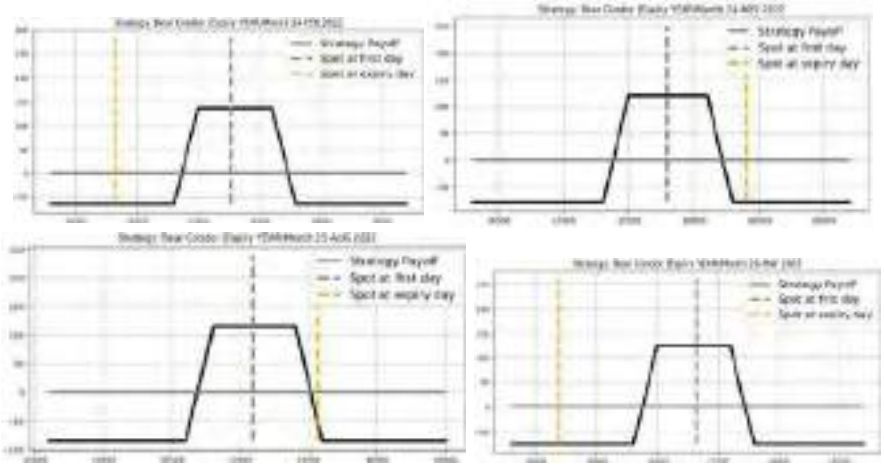


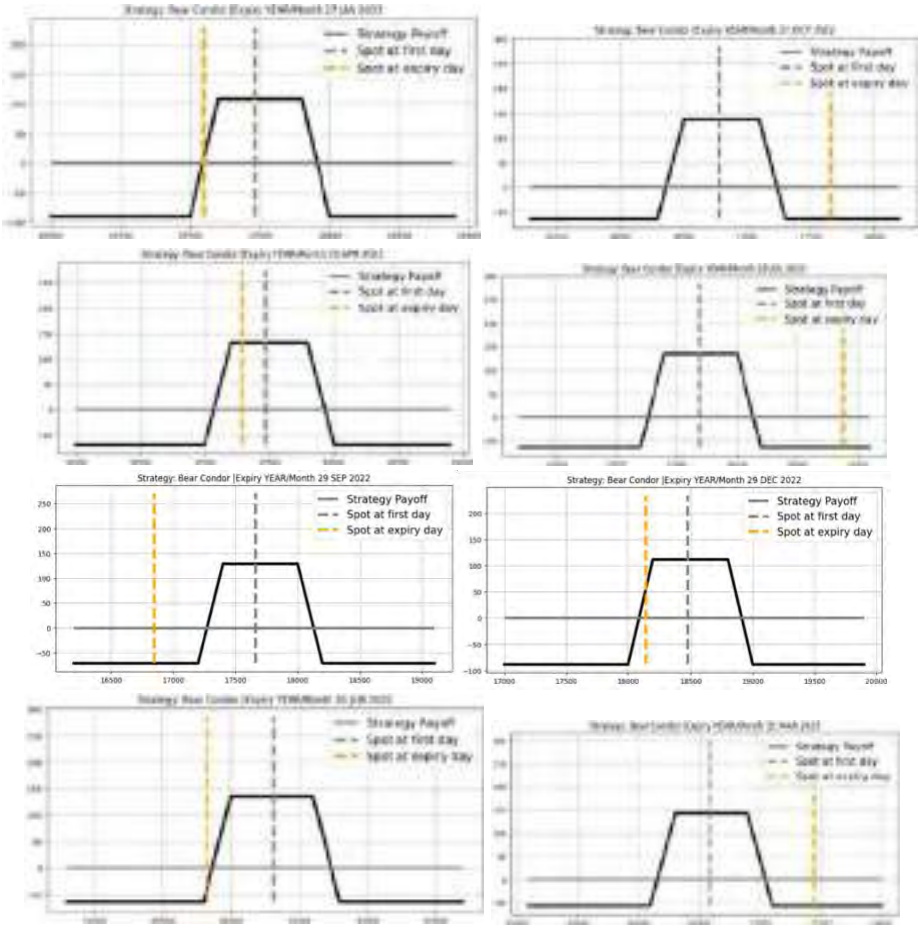
Strategy: Bear Condor Testing Year: 2021



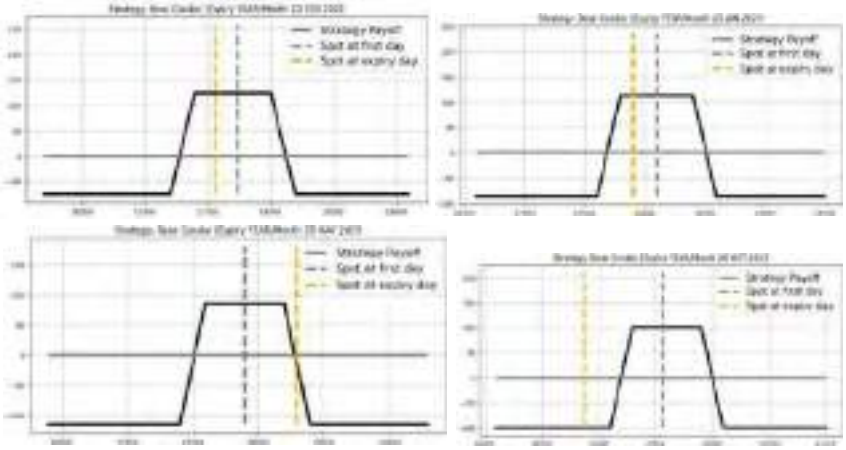


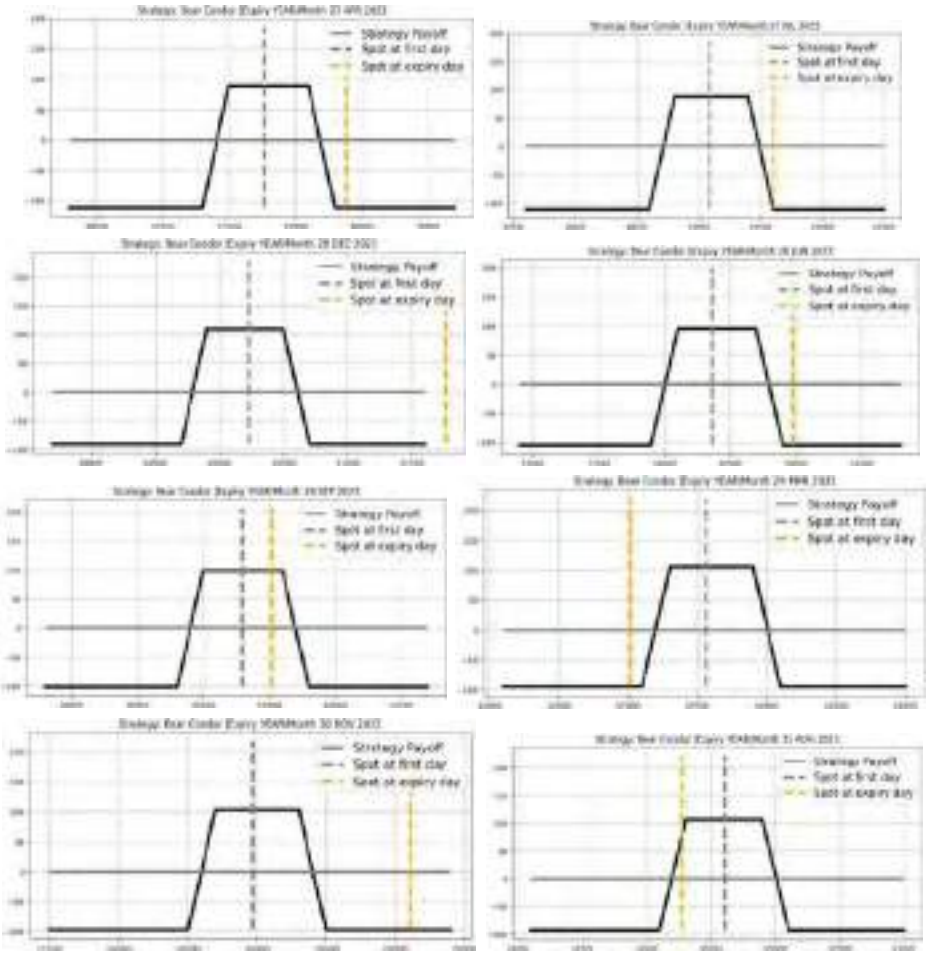
Strategy: Bear Condor **Testing Year: 2022**



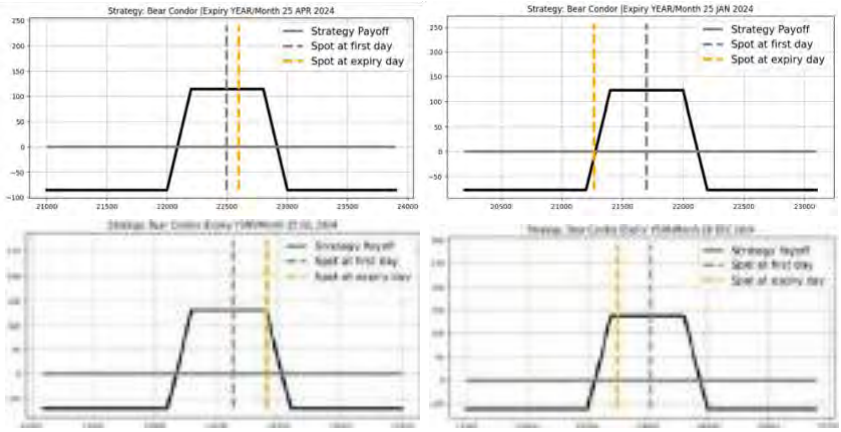


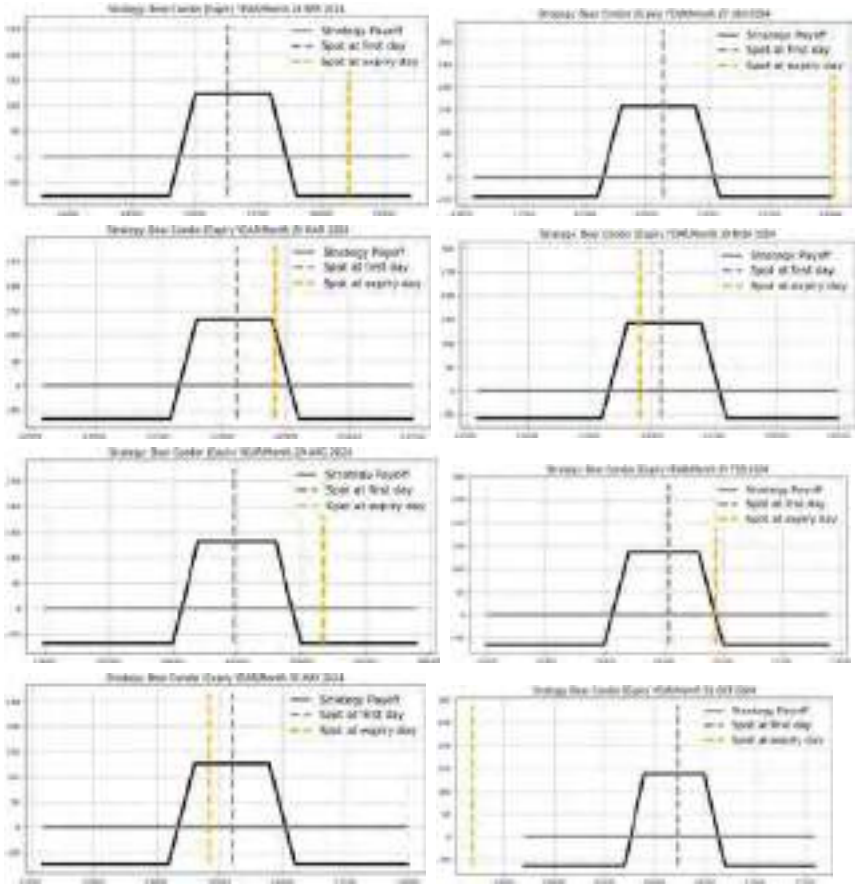
Strategy: Bear Condor **Testing Year: 2023**



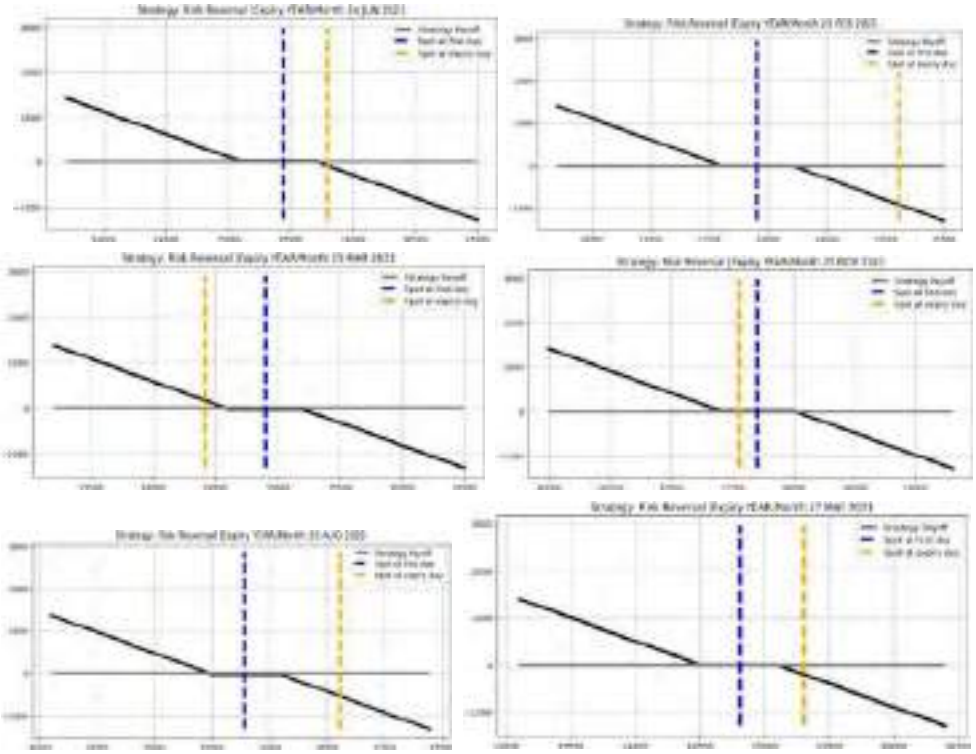


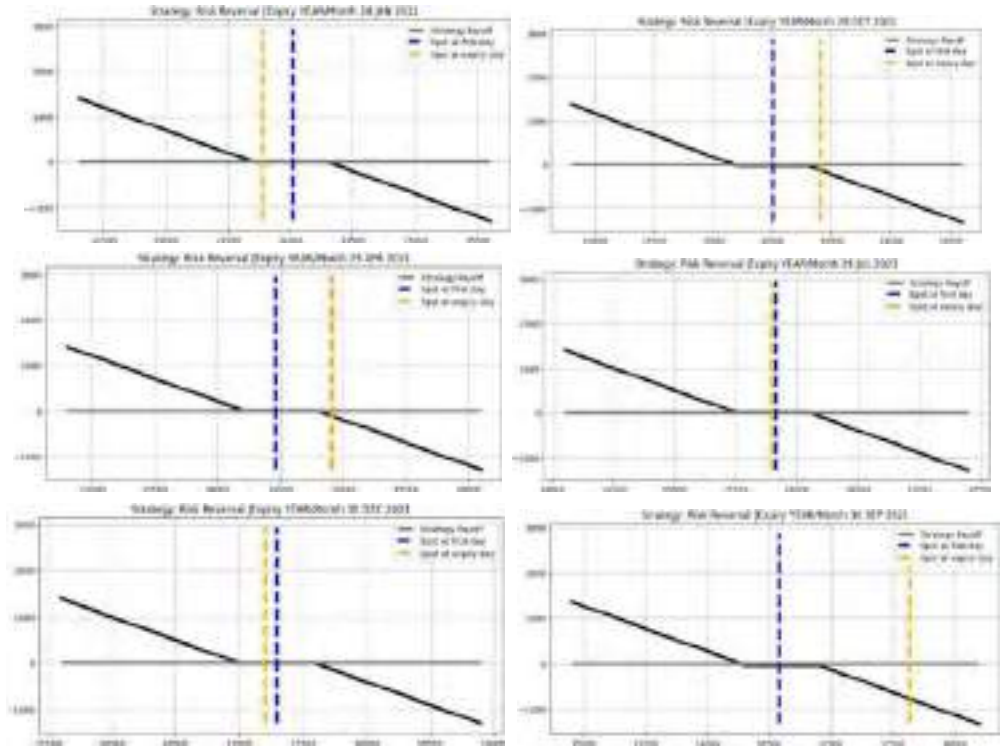
Strategy: Bear Condor **Testing Year: 2024**



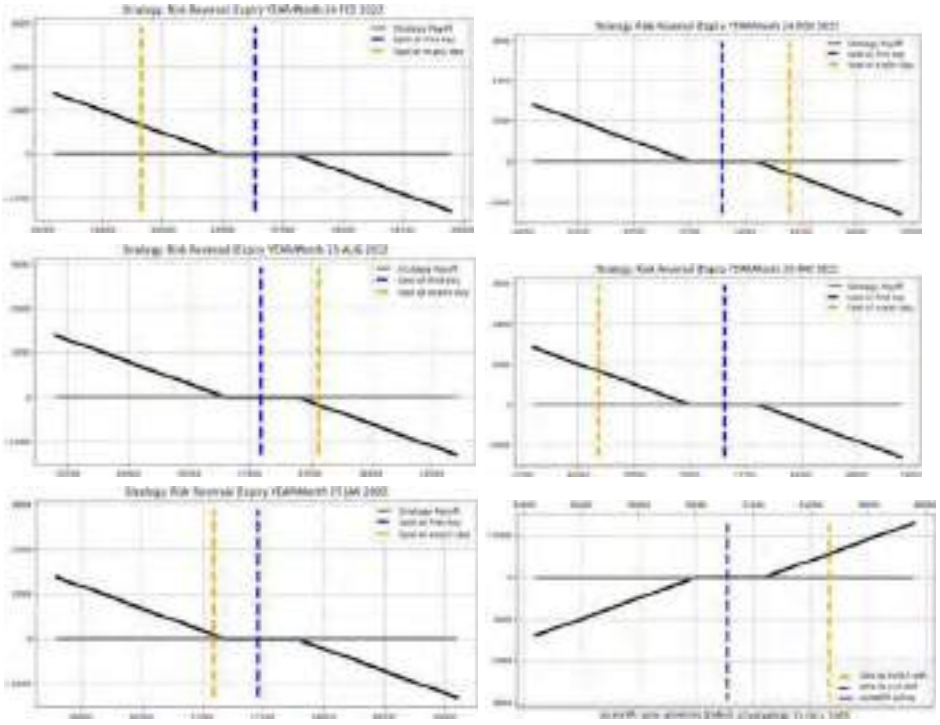


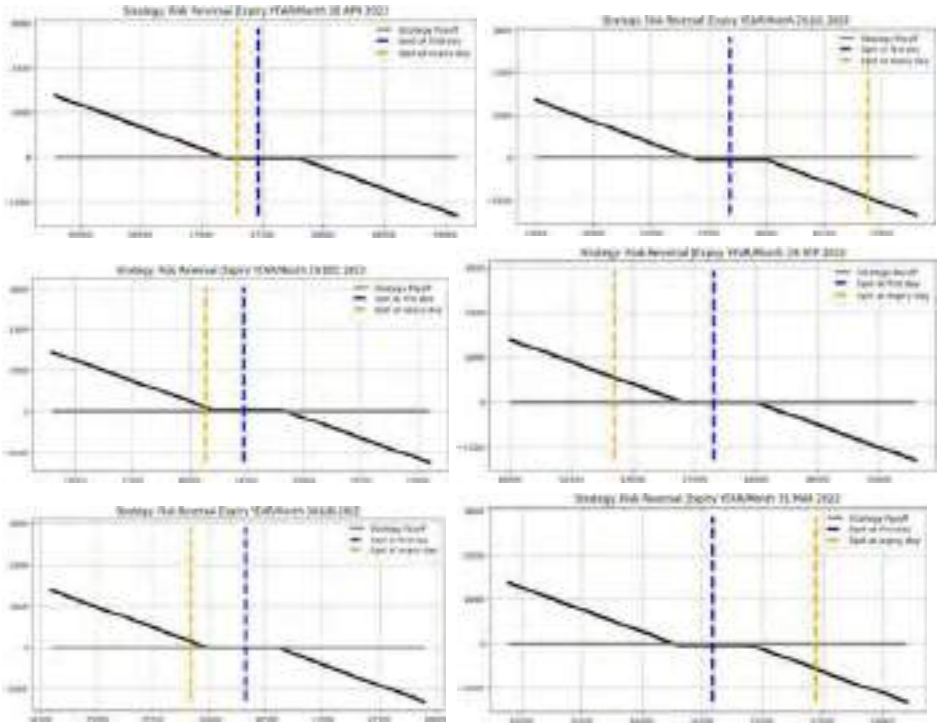
Strategy: Risk Reversal Testing Year: 2021



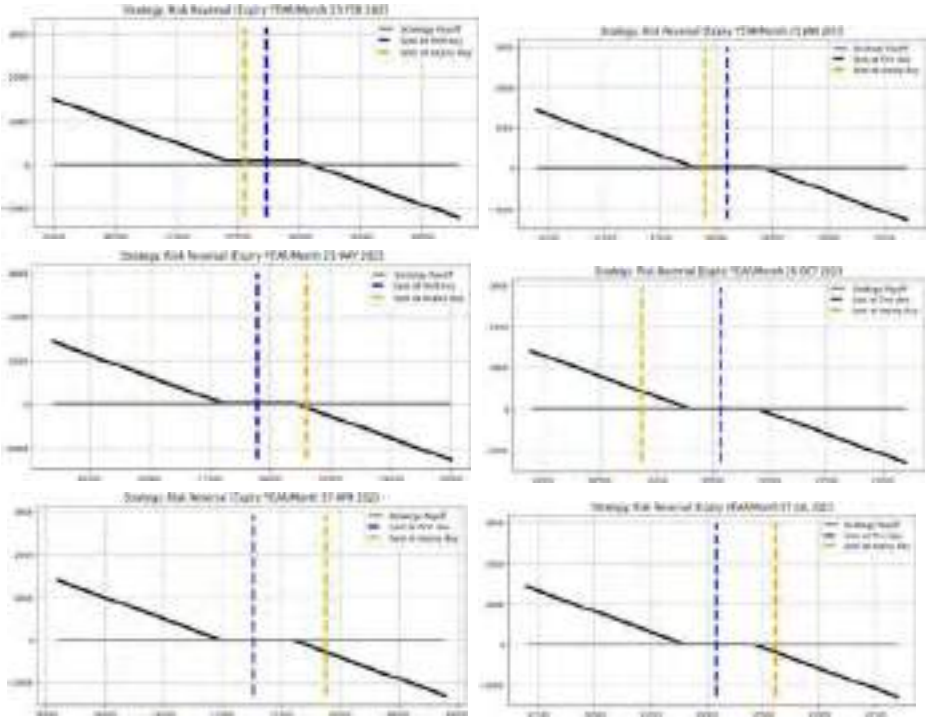


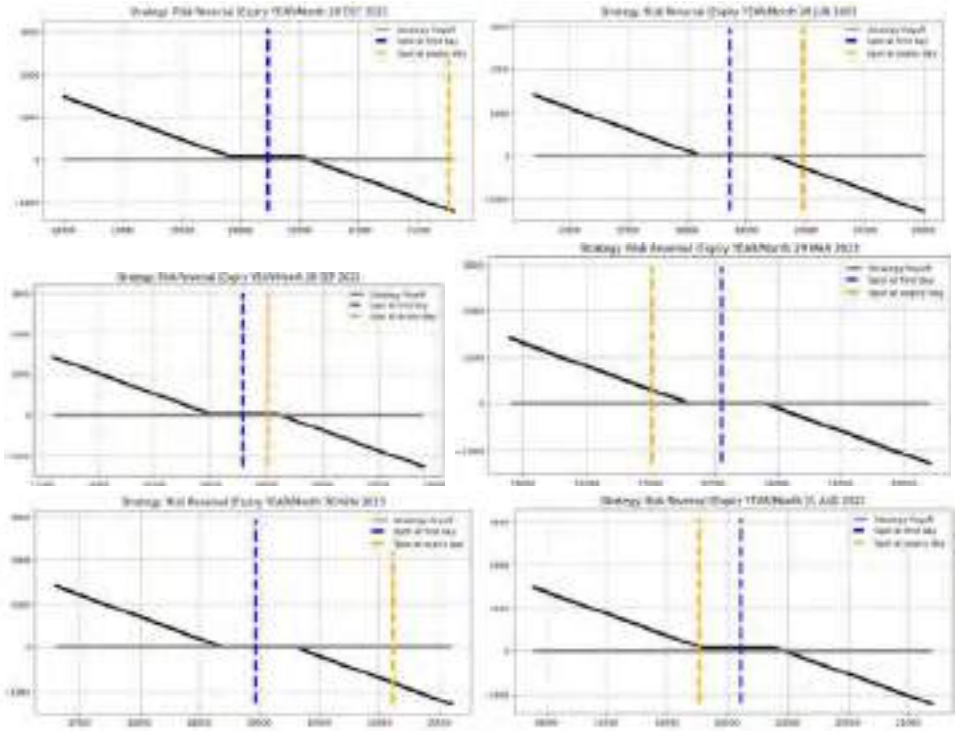
Strategy: Risk Reversal **Testing Year:** 2022



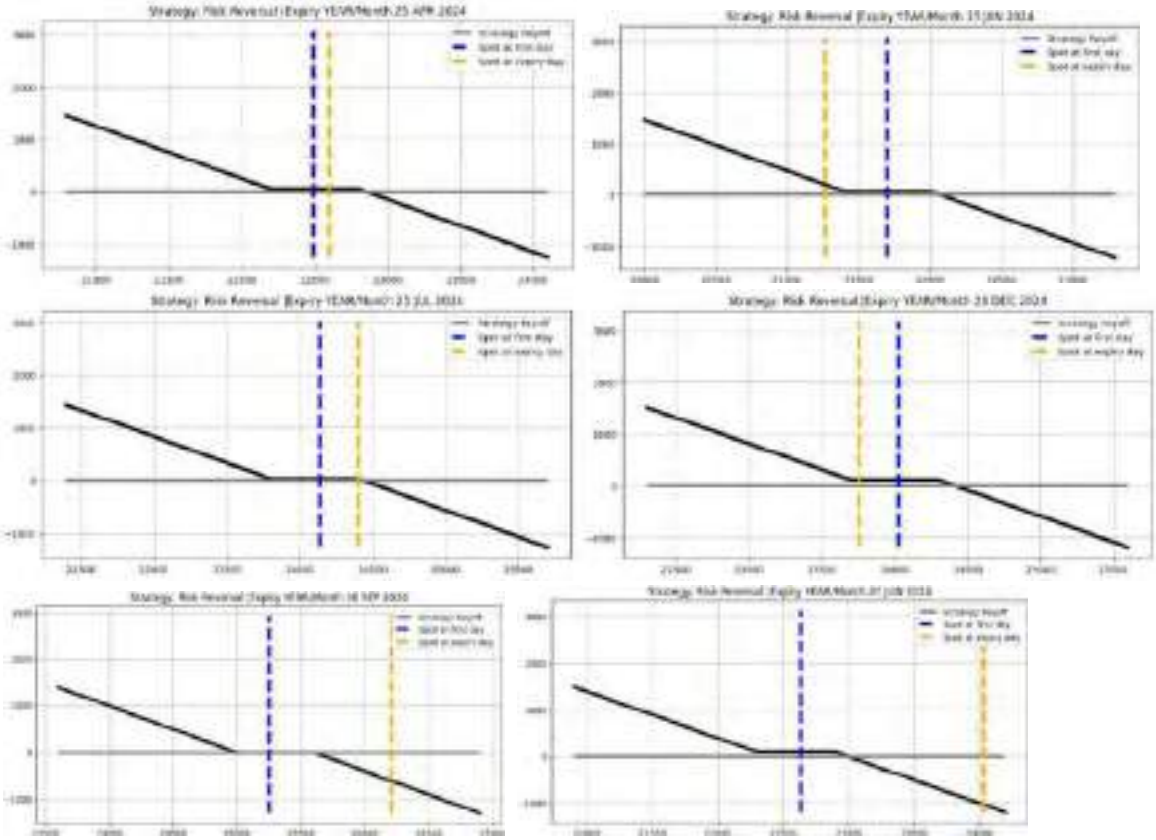


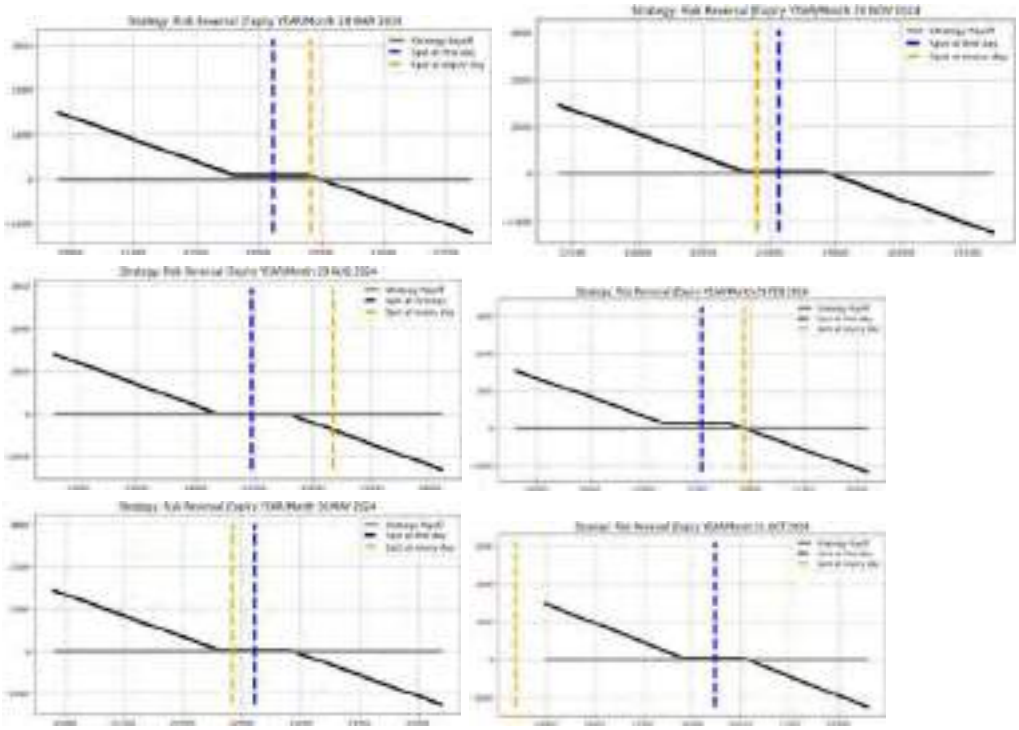
Strategy: Risk Reversal **Testing Year:** 2023



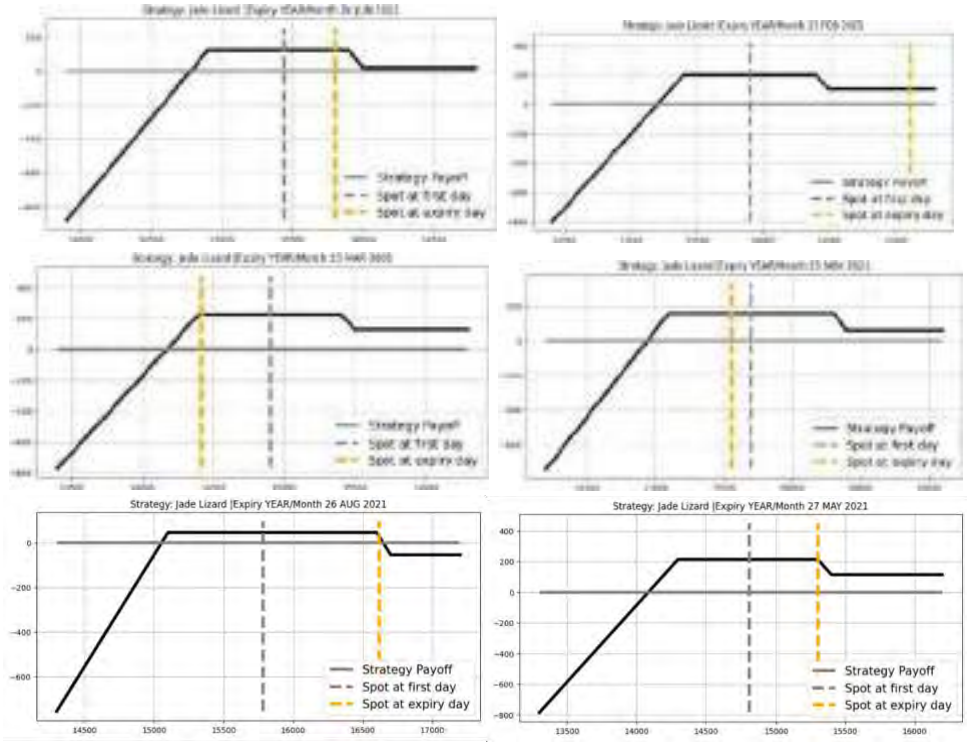


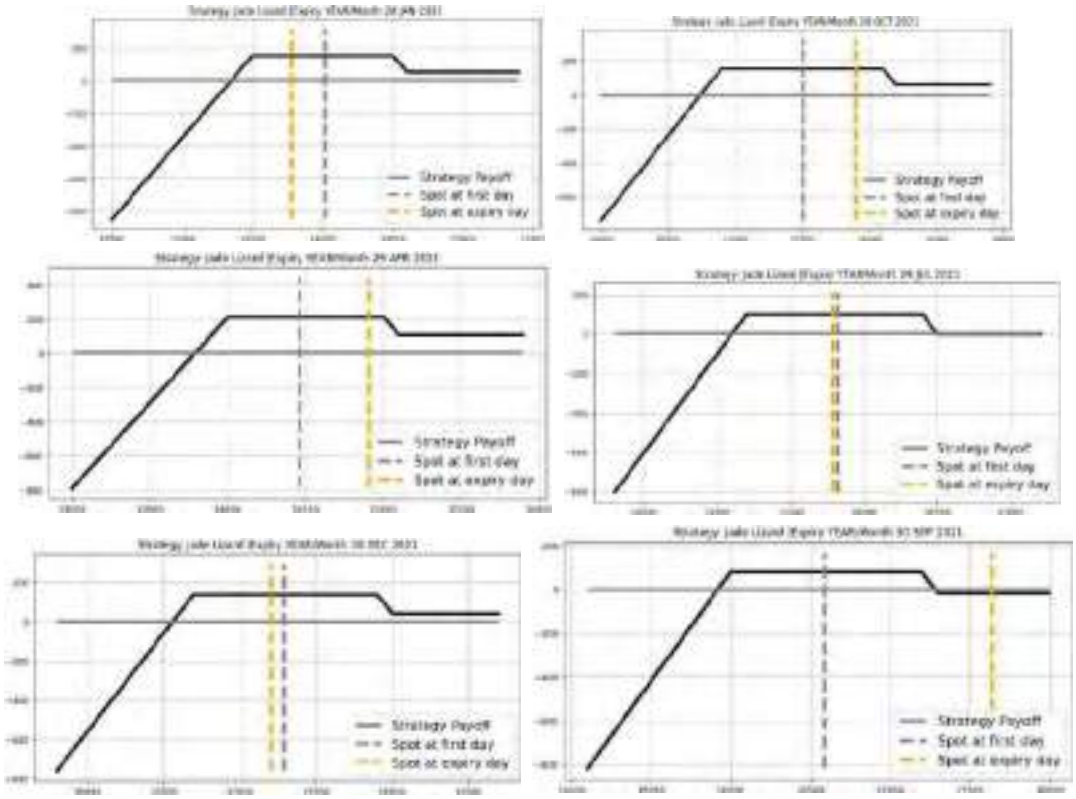
Strategy: Risk Reversal **Testing Year:** 2024



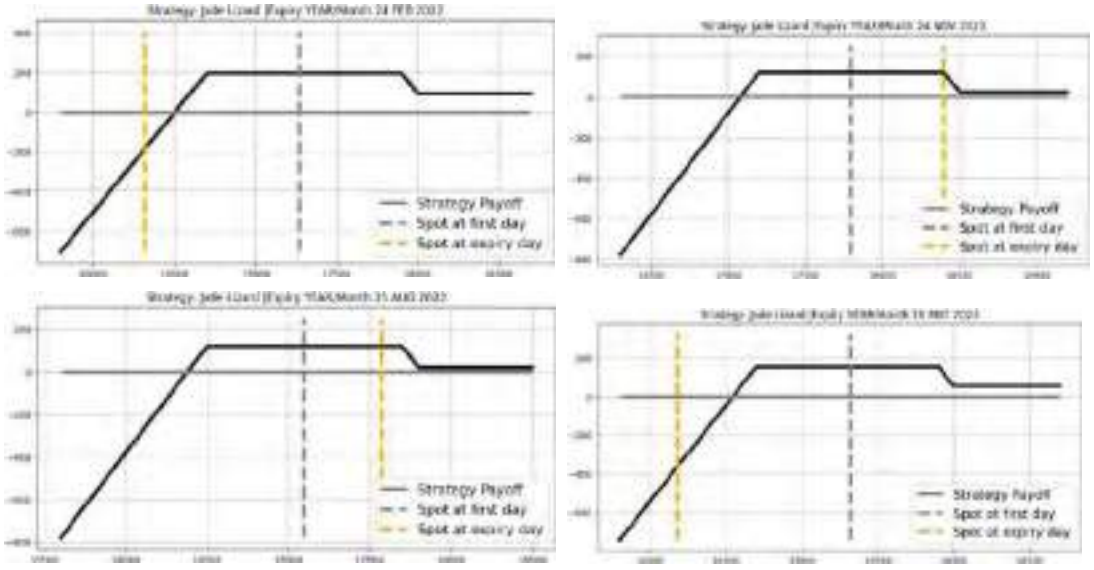


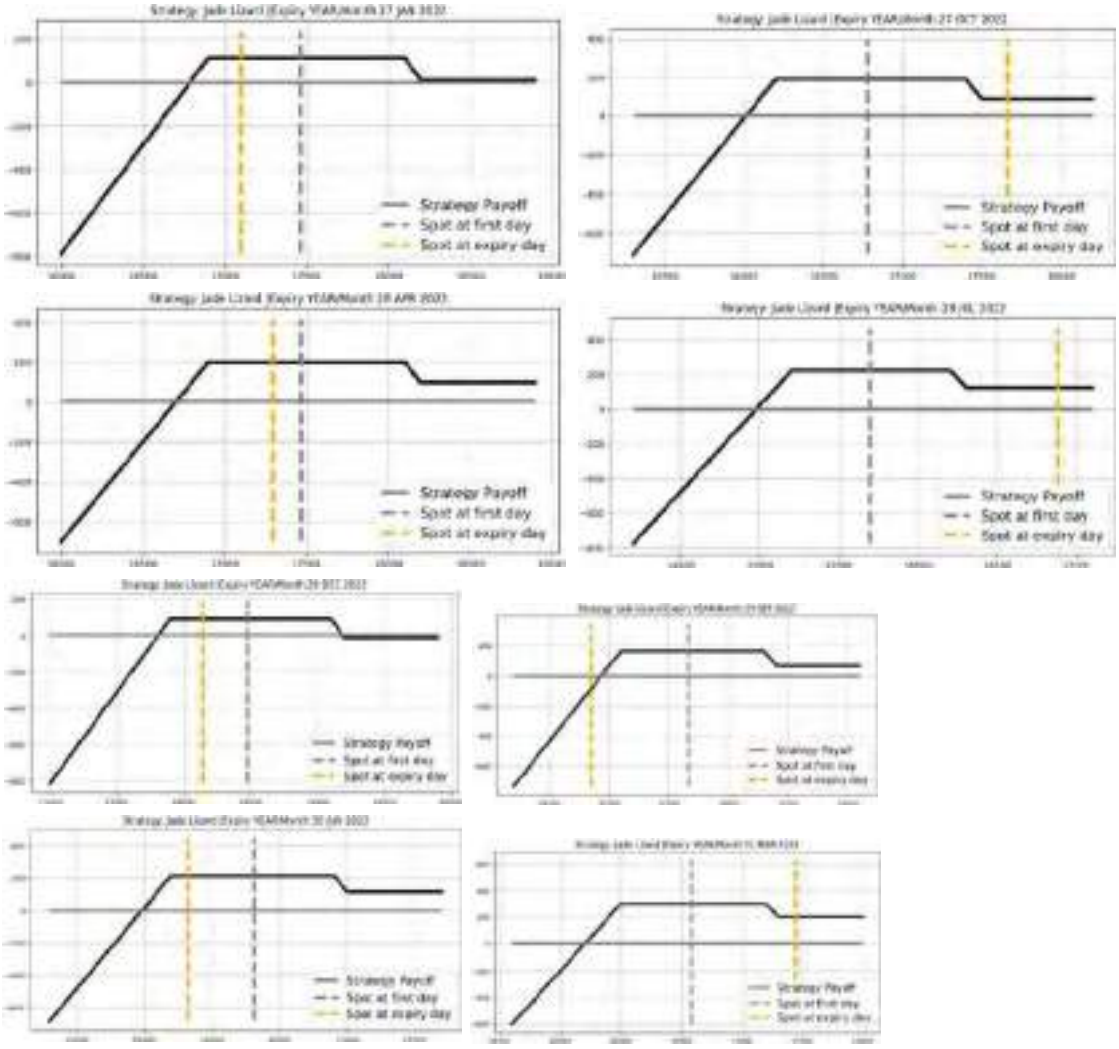
Strategy: Jade Lizard Testing Year: 2021



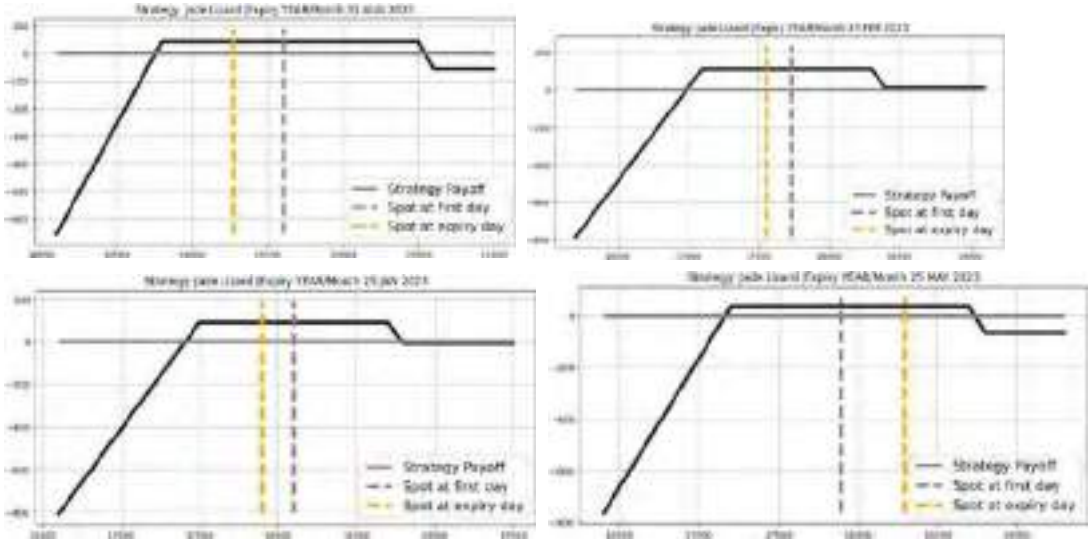


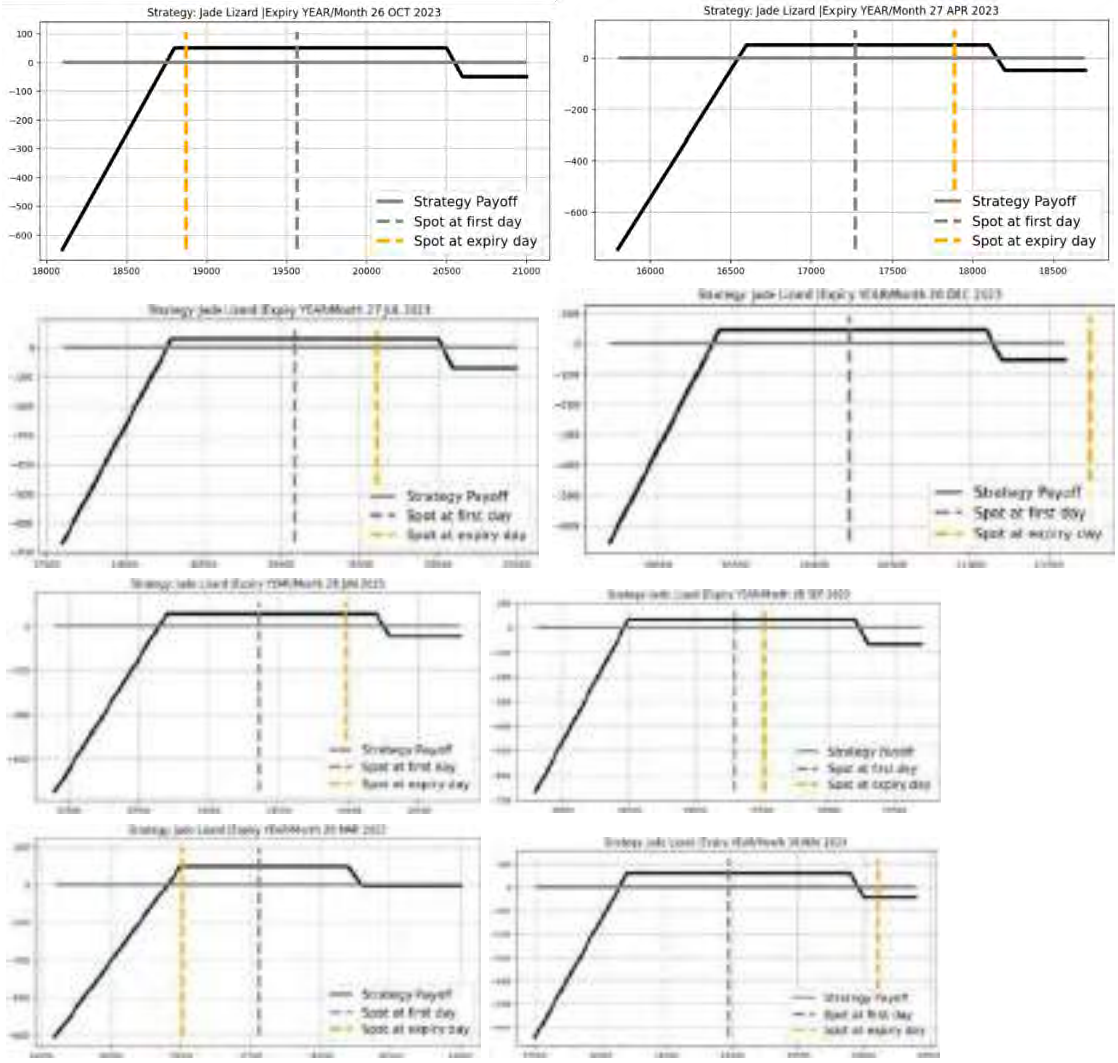
Strategy: Jade Lizard Testing Year: 2022



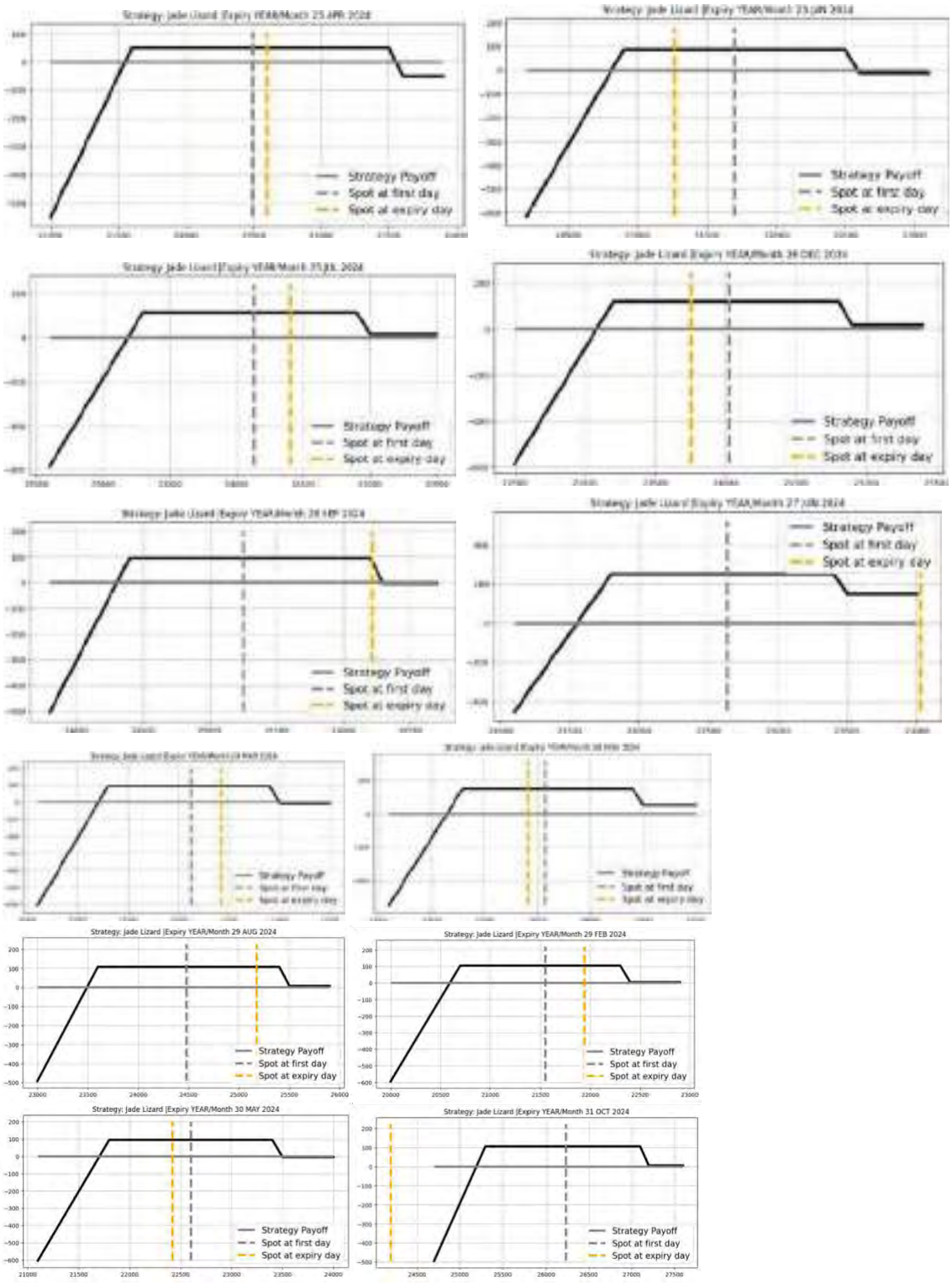


Strategy: Jade Lizard **Testing Year:** 2023

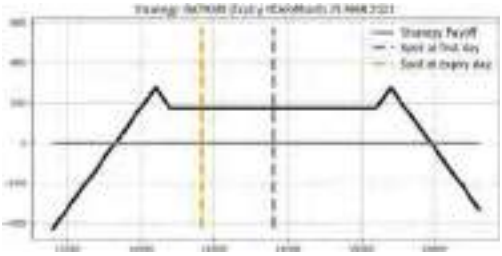


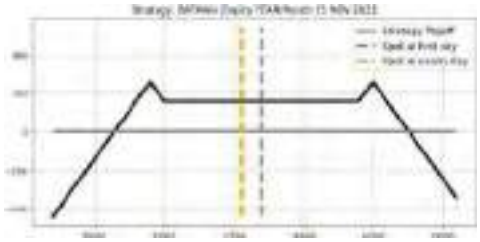


Strategy: Jade Lizard **Testing Year:** 2024

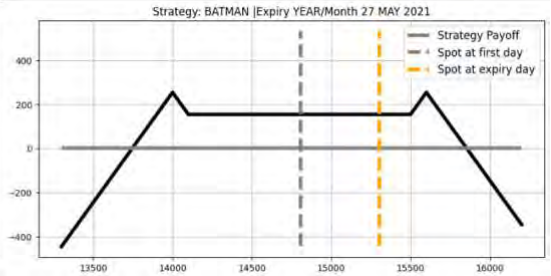


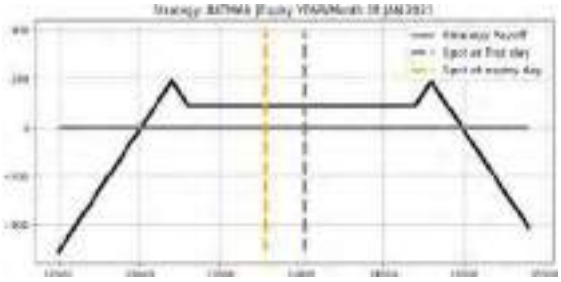
Strategy: Batman **Testing Year:** 2021

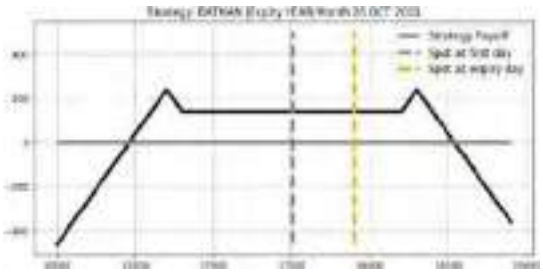


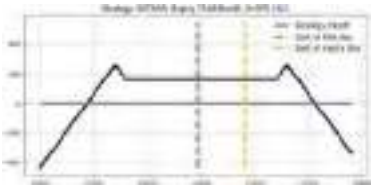


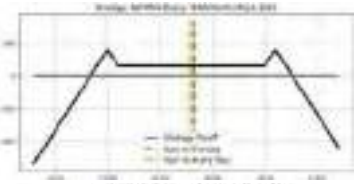

Strategy: BATMAN |Expiry YEAR/Month 26 AUG 2021
Strategy Payoff
Spot at first day
Spot at expiry day

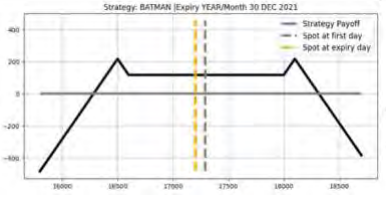

Strategy: BATMAN |Expiry YEAR/Month 27 MAY 2021
Strategy Payoff
Spot at first day
Spot at expiry day





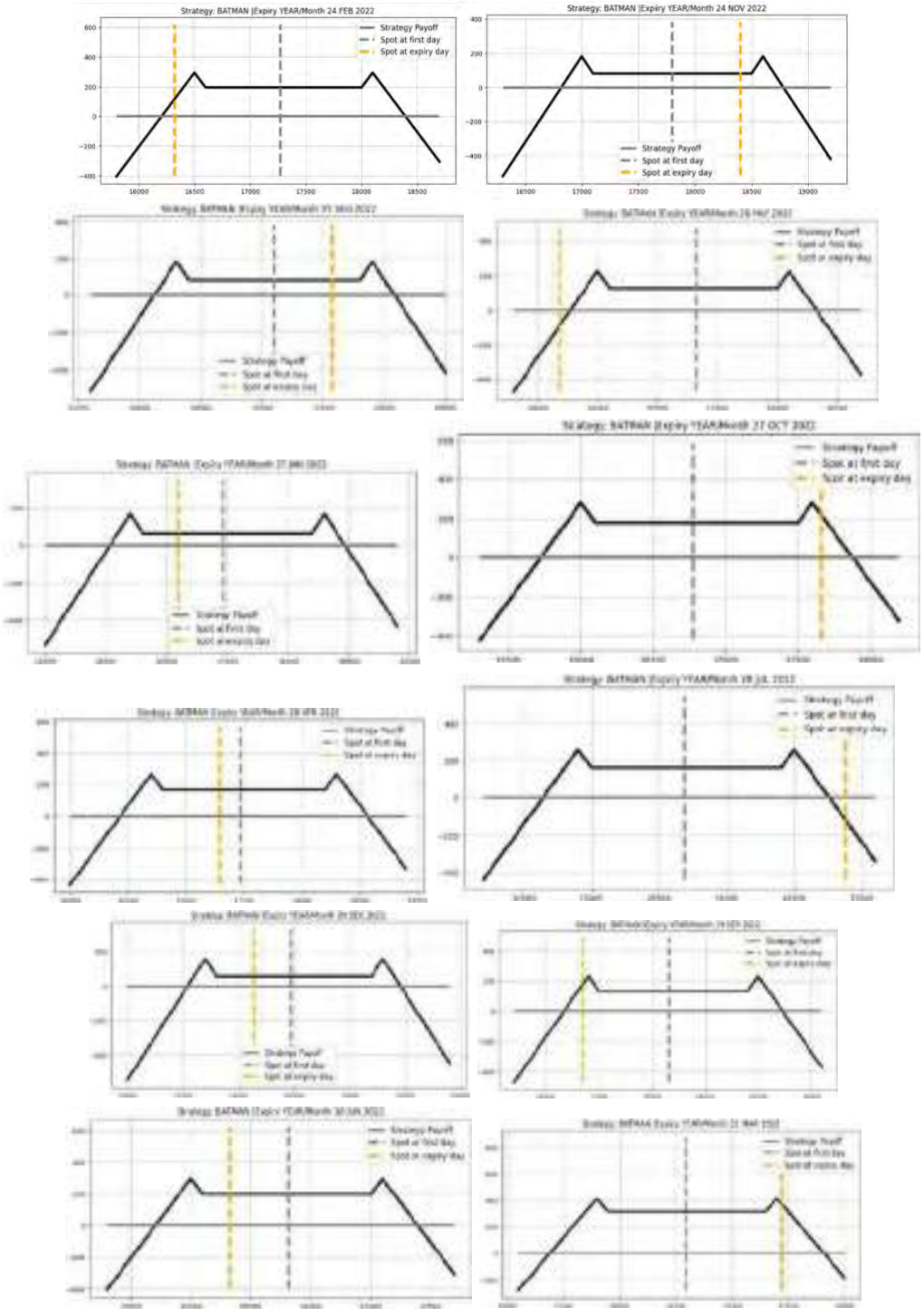




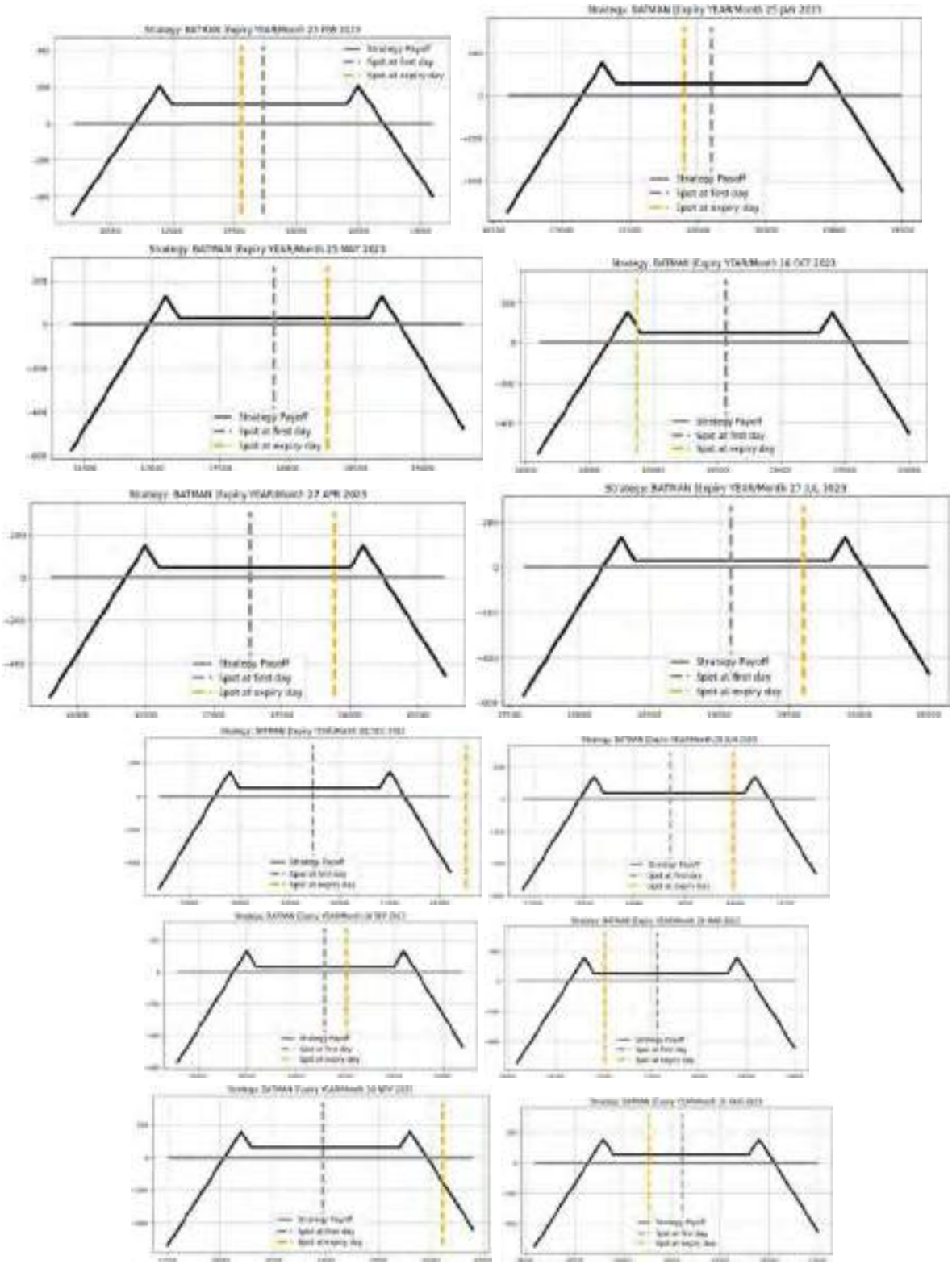

Strategy: BATMAN |Expiry YEAR/Month 30 DEC 2021
Strategy Payoff
Spot at first day
Spot at expiry day


Strategy: BATMAN |Expiry YEAR/Month 30 SEP 2021
Strategy Payoff
Spot at first day
Spot at expiry day

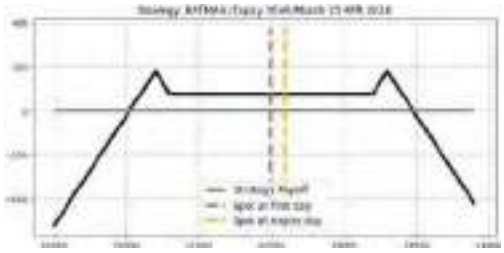
Strategy: Batman Testing Year: 2022

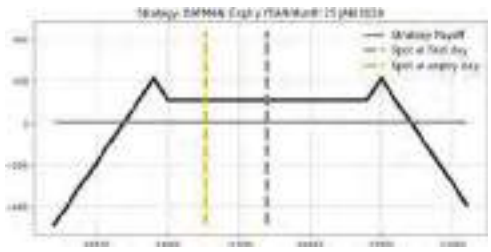


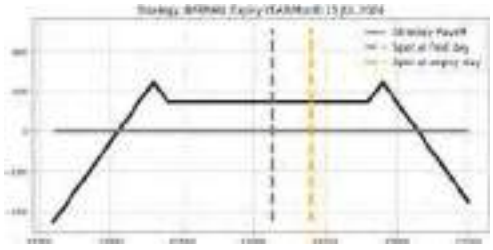
Strategy: Batman **Testing Year:** 2023

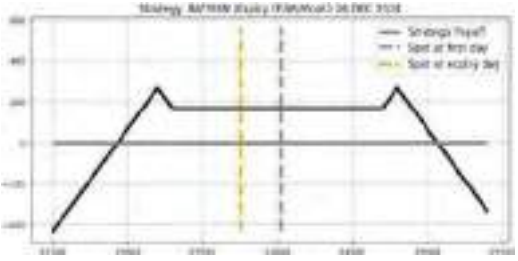


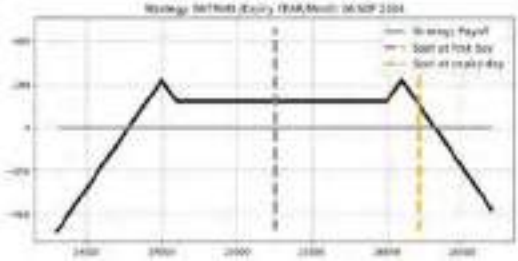
Strategy: Batman **Testing Year:** 2024

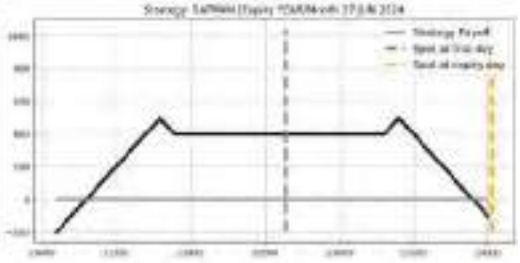


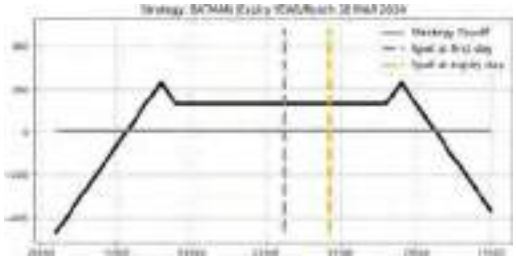


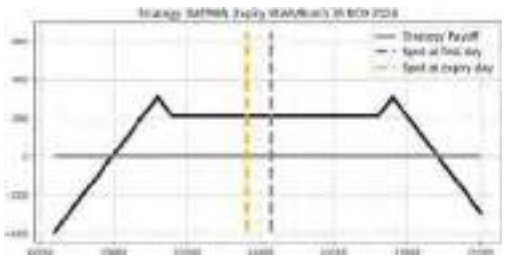


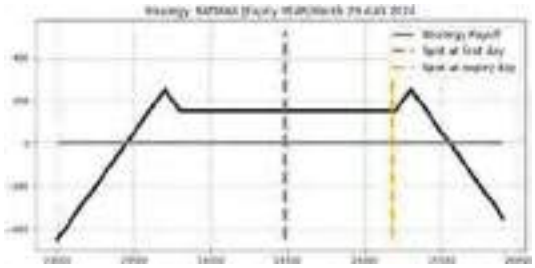


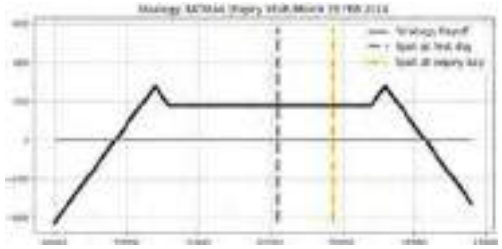


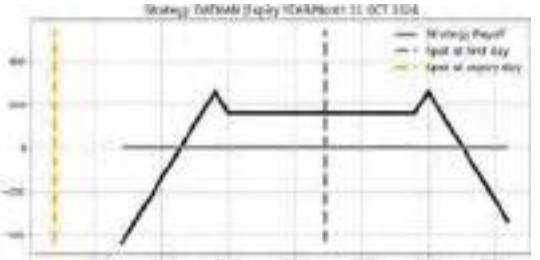




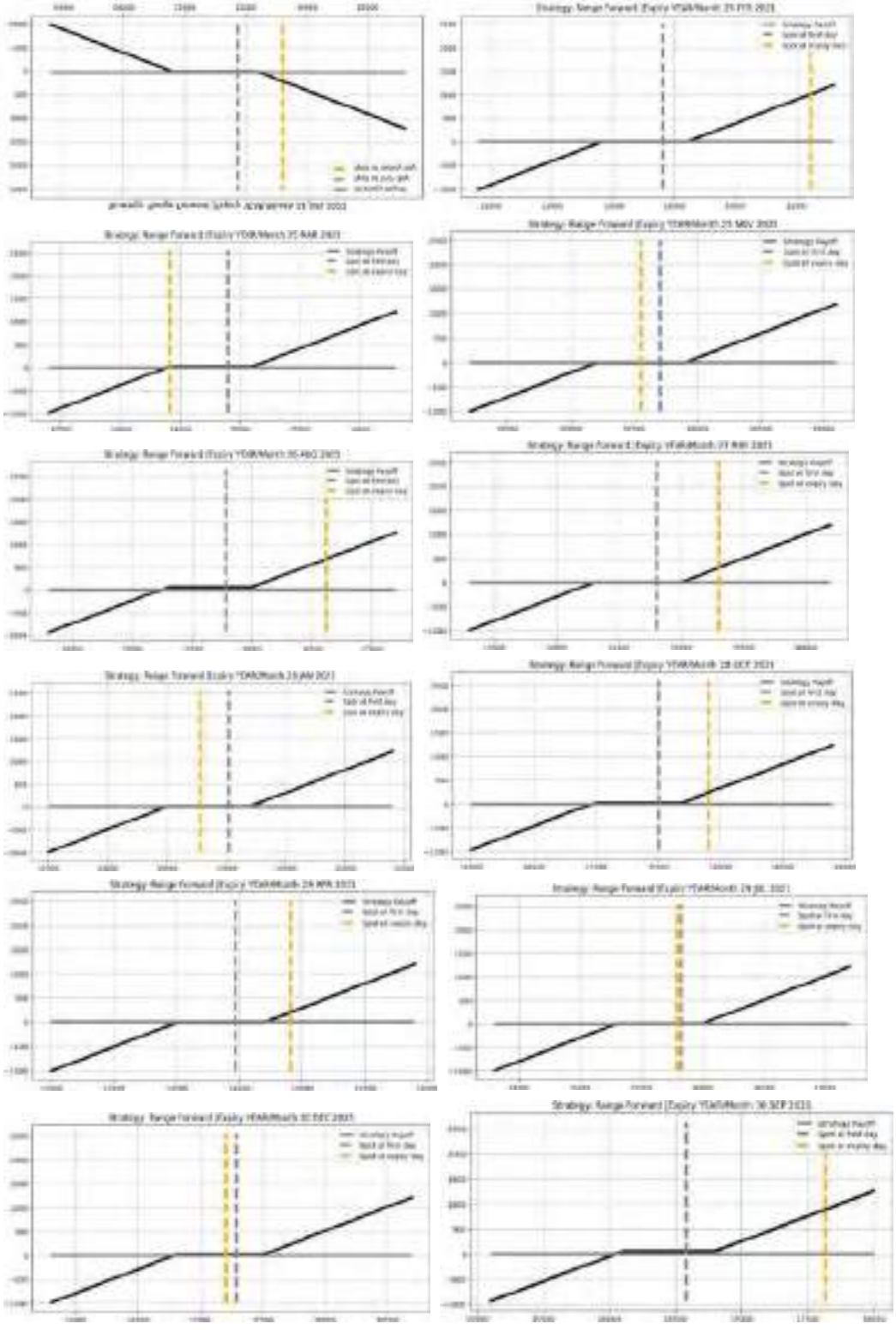




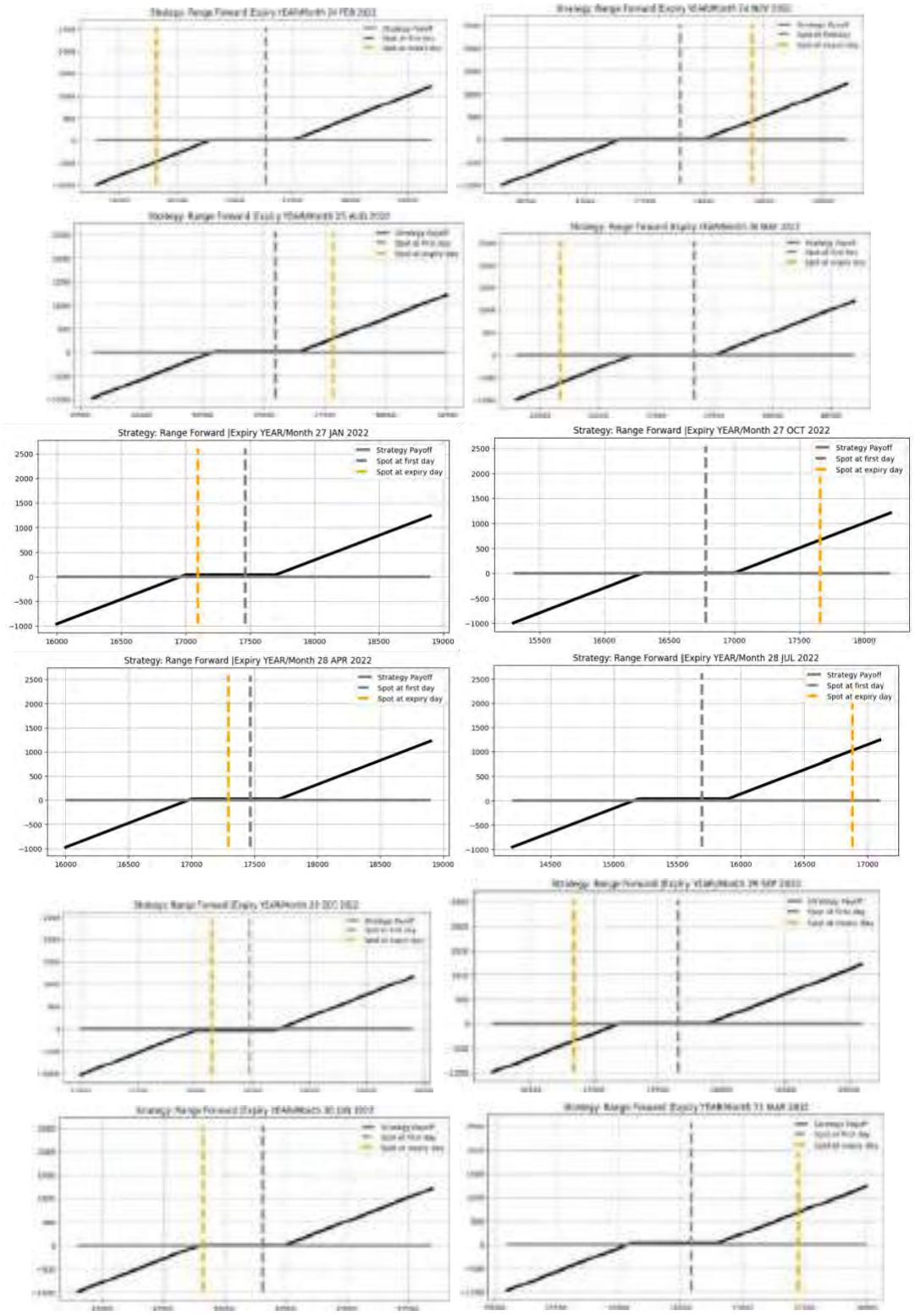




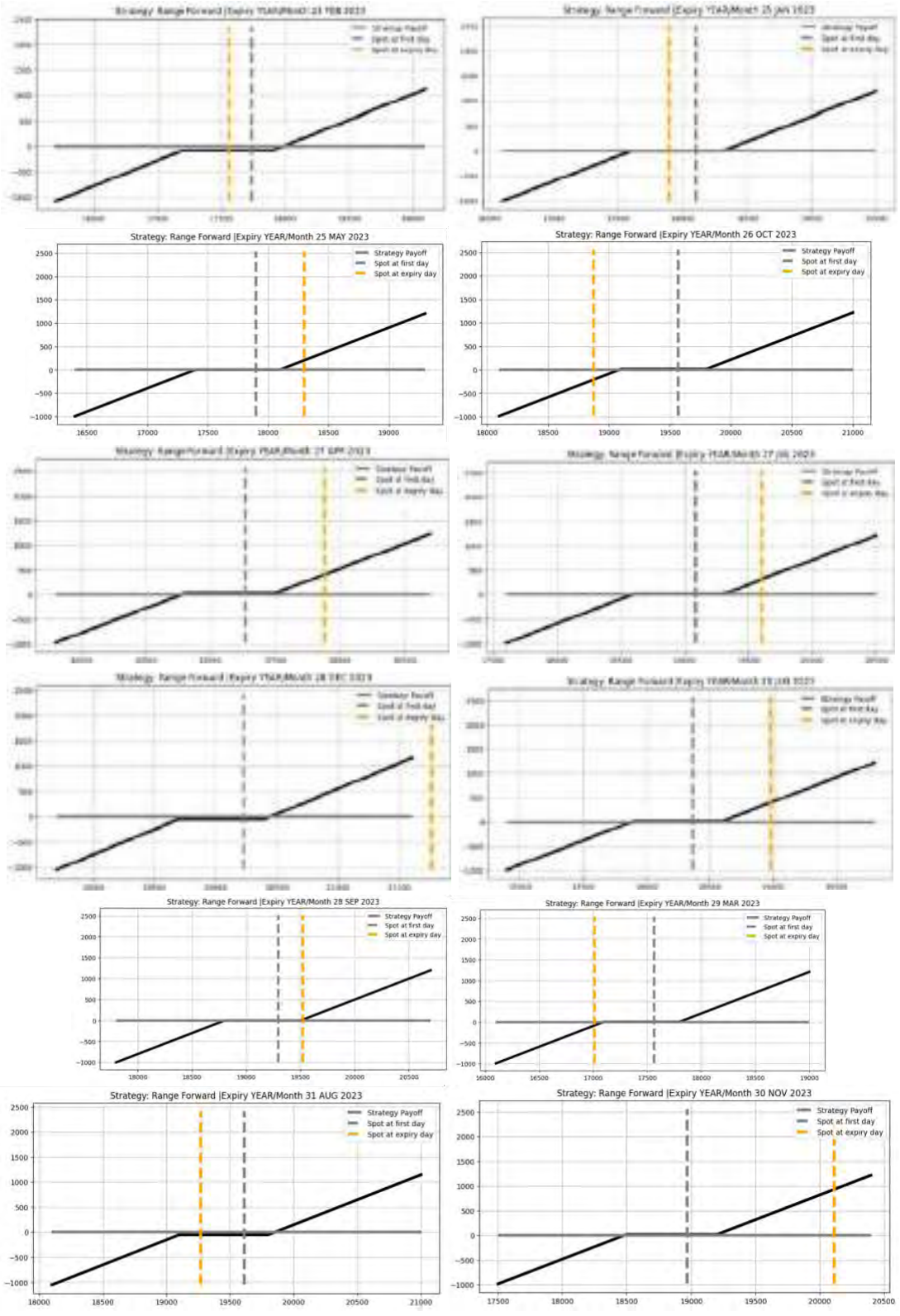
Strategy: Range Forward Testing Year: 2021



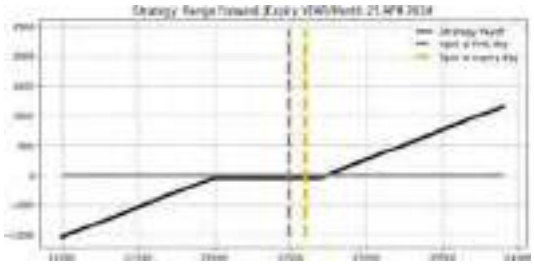
Strategy: Range Forward **Testing Year:** 2022

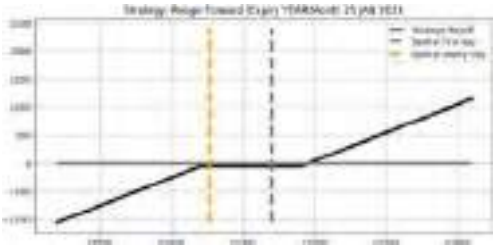


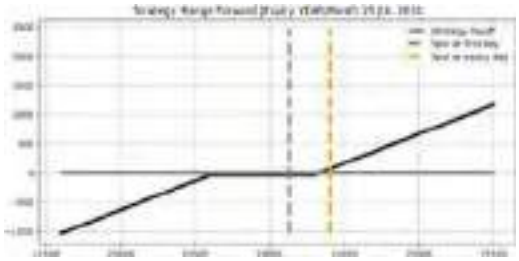
Strategy: Range Forward **Testing Year:** 2023

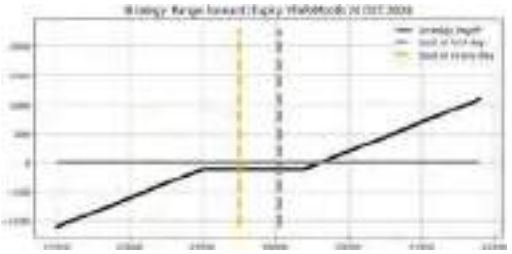


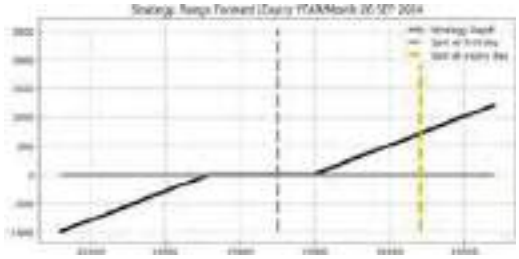
Strategy: Range Forward **Testing Year:** 2024

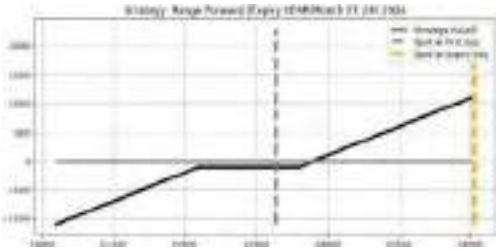


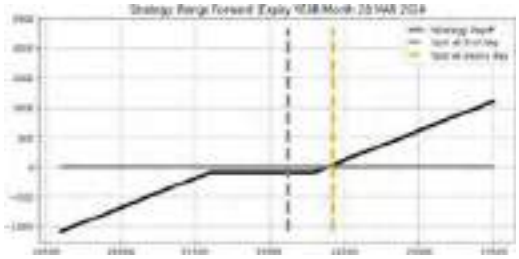


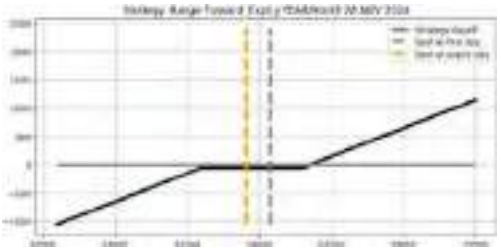


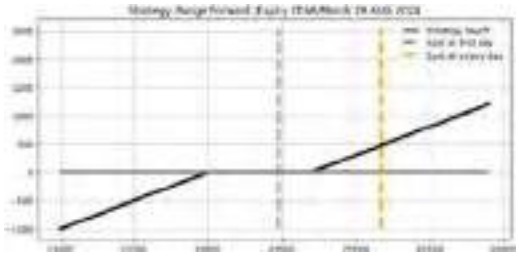


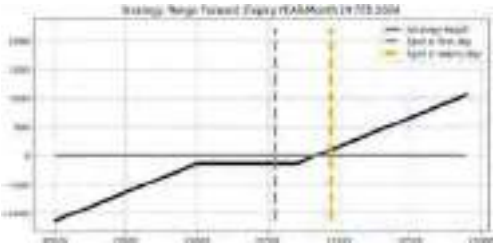


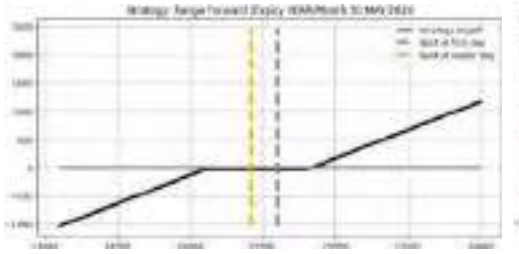


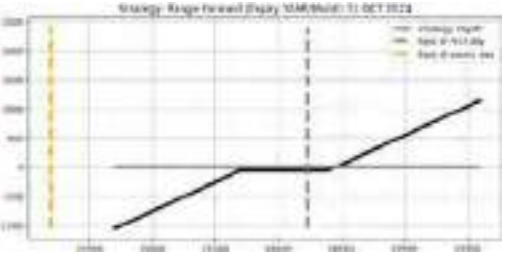




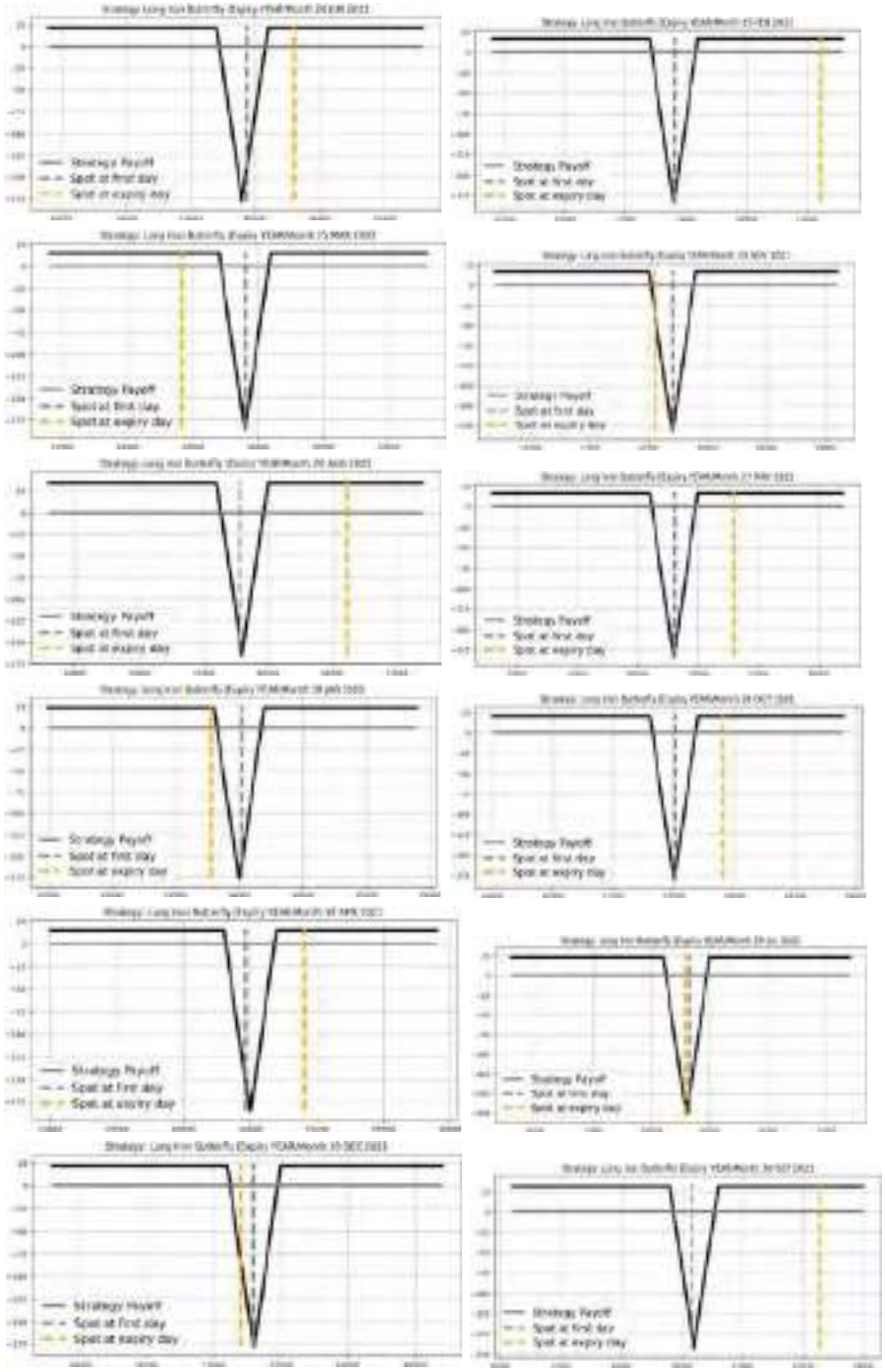




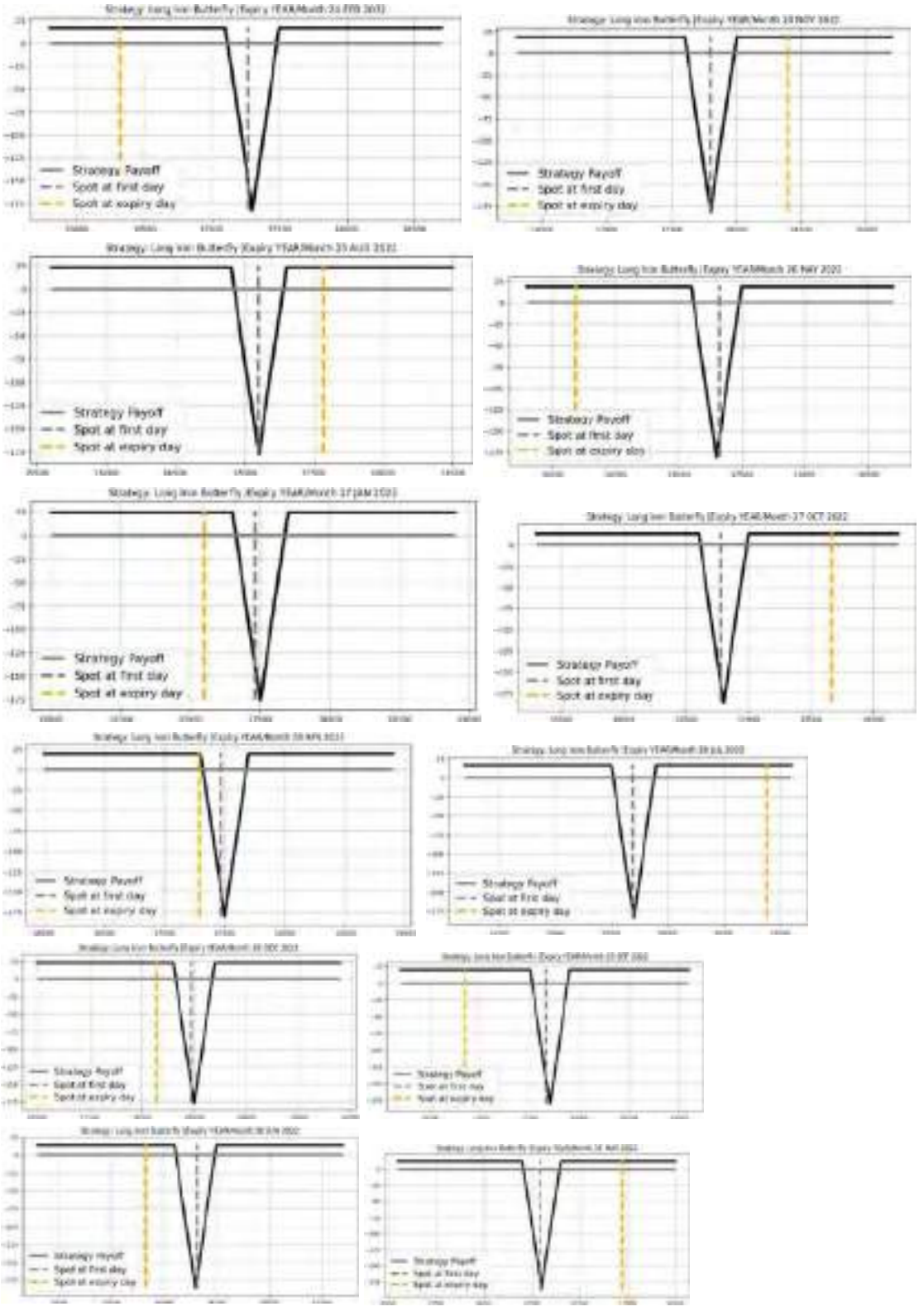




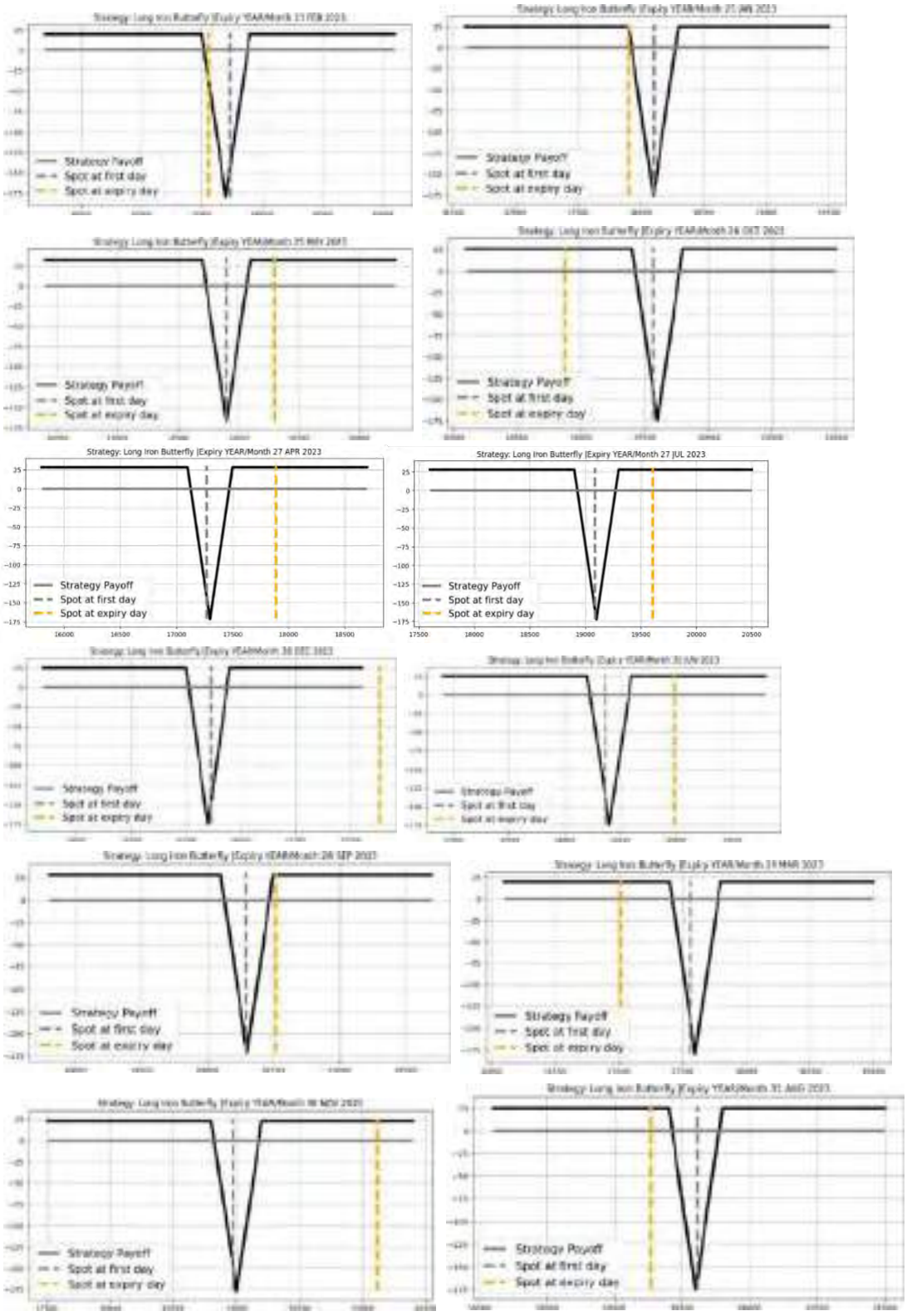
Strategy: Long Iron Butterfly **Testing Year:** 2021



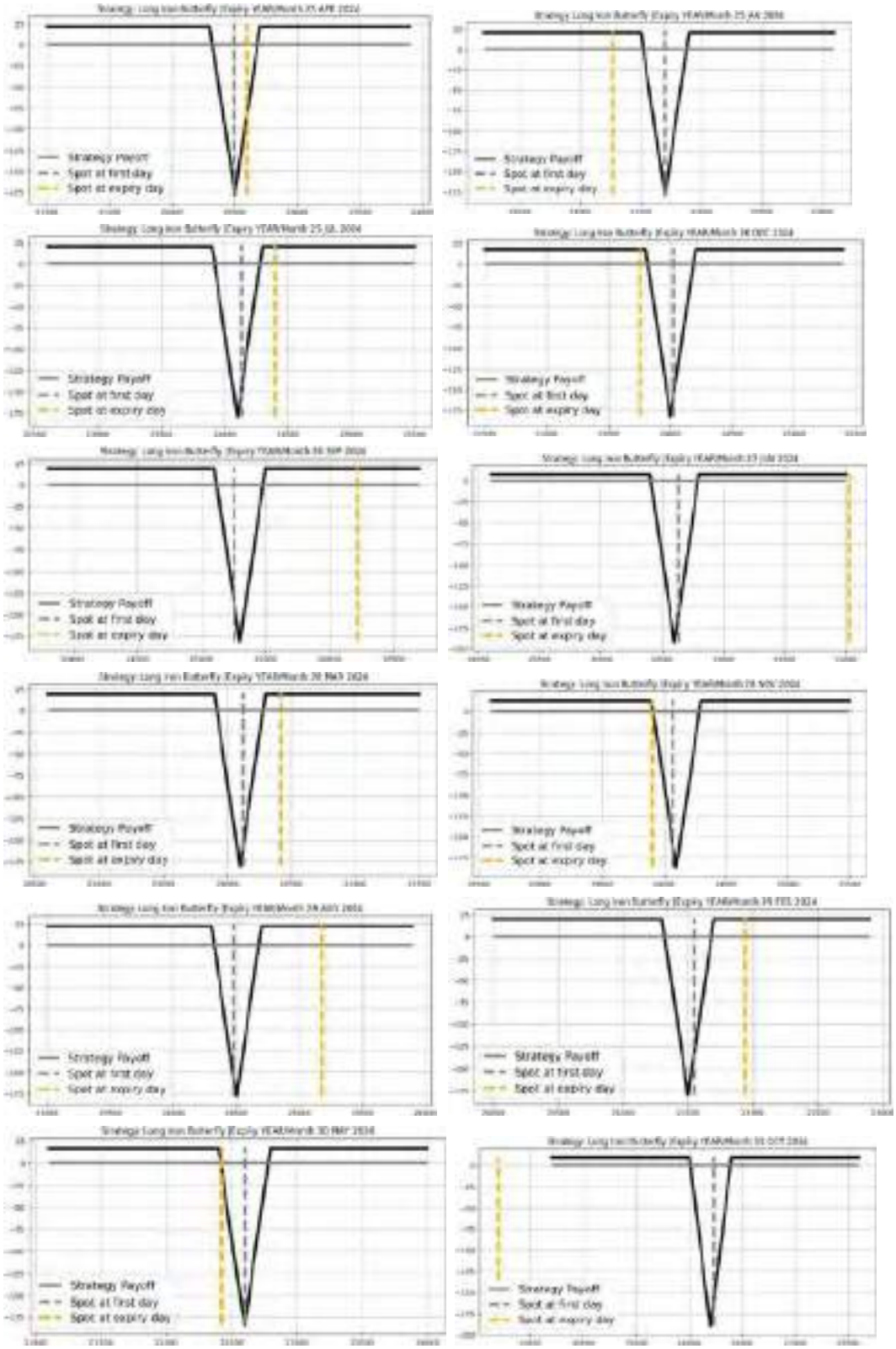
Strategy: Long Iron Butterfly **Testing Year:** 2022



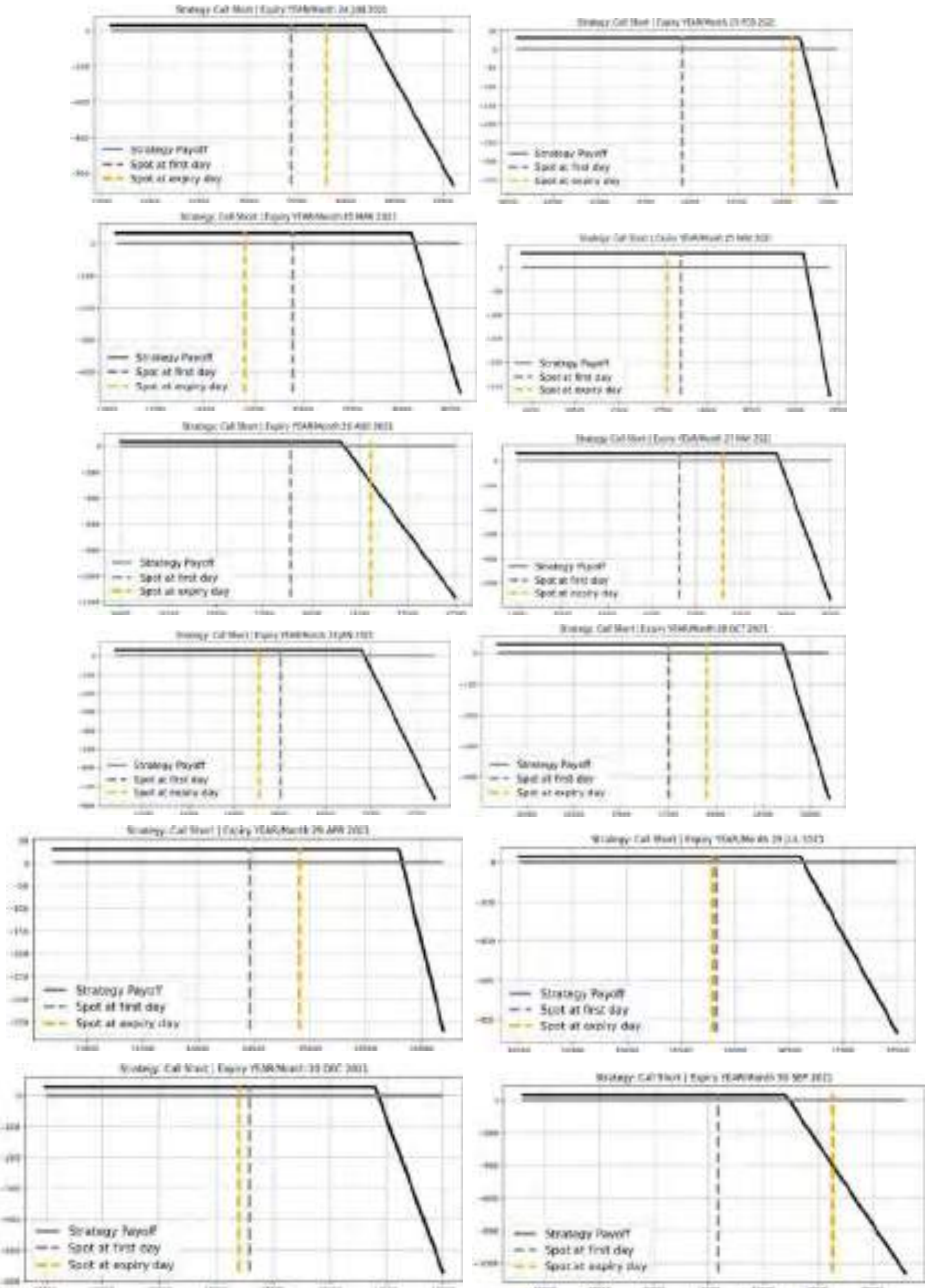
Strategy: Long Iron Butterfly **Testing Year:** 2023



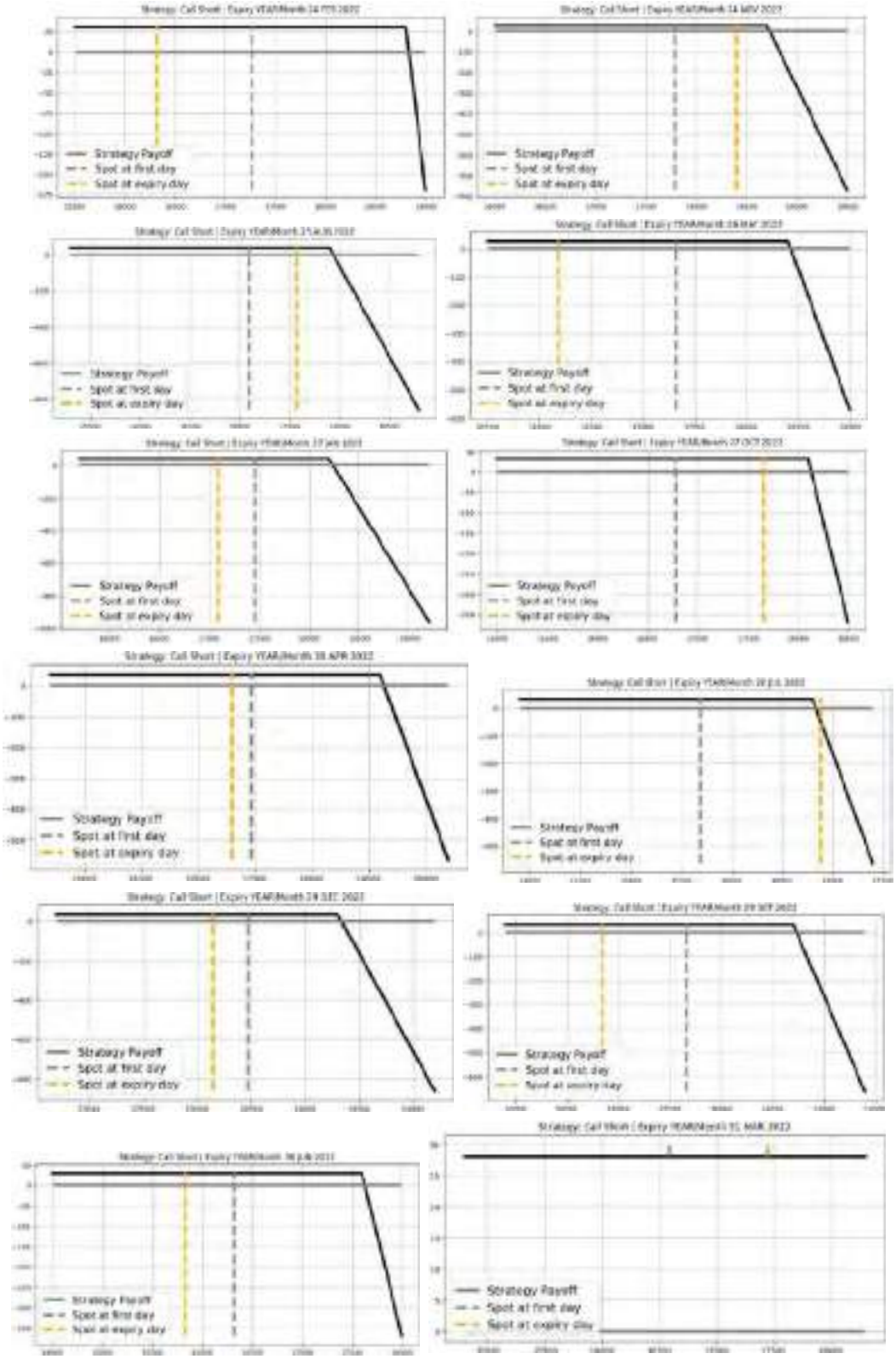
Strategy: Long Iron Butterfly **Testing Year:** 2024



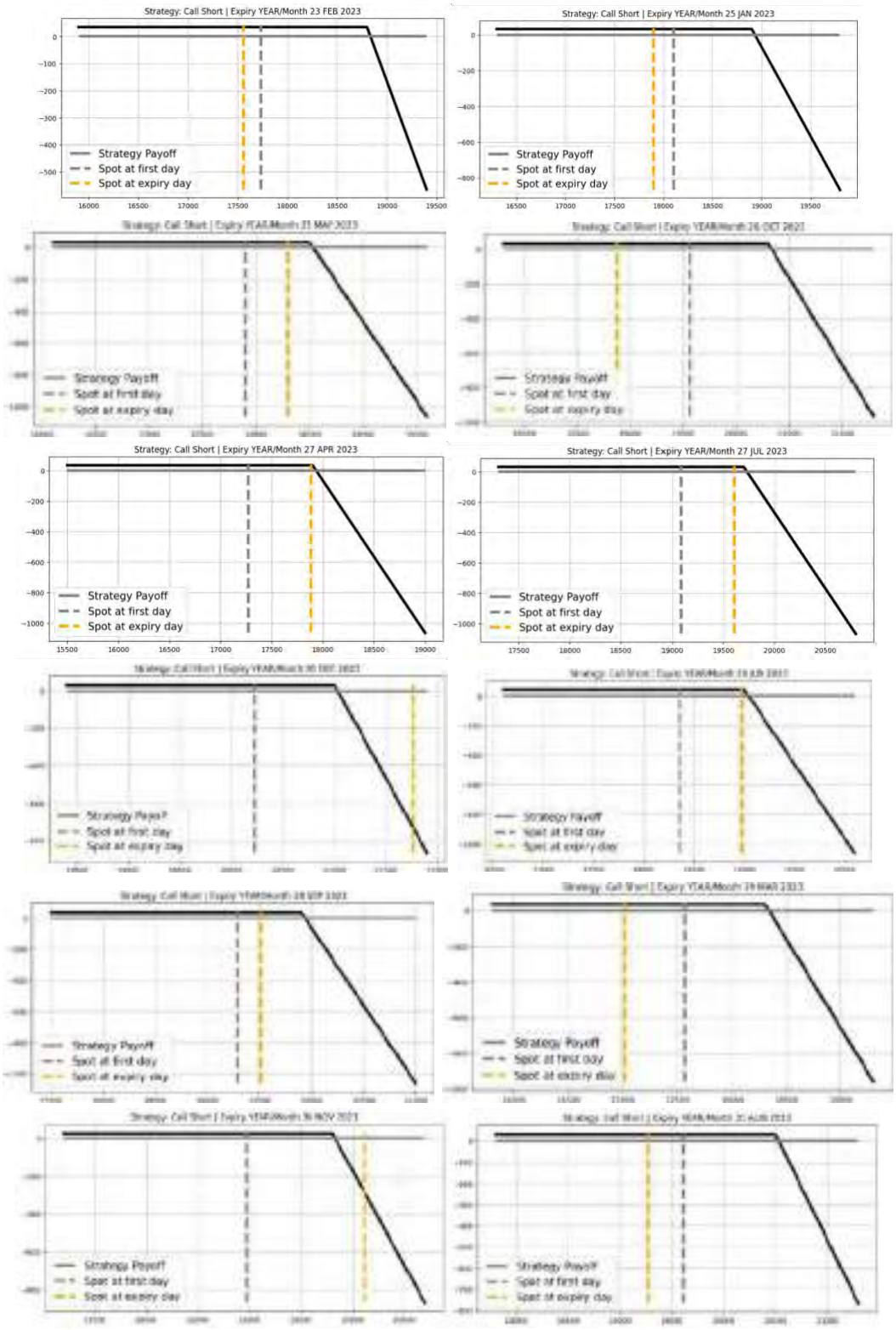
Strategy: Call Short **Testing Year:** 2021


Strategy Payoff
Spot at first day
Spot at expiry day

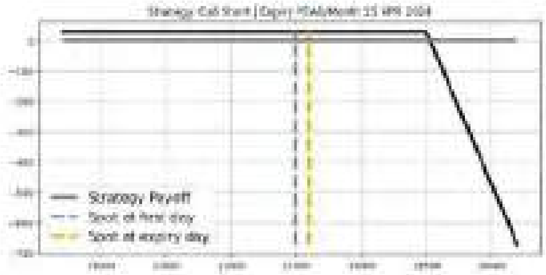
Strategy: Call Short Testing Year: 2022

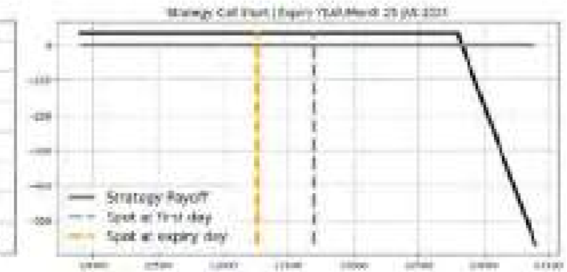


Strategy: Call Short **Testing Year:** 2023

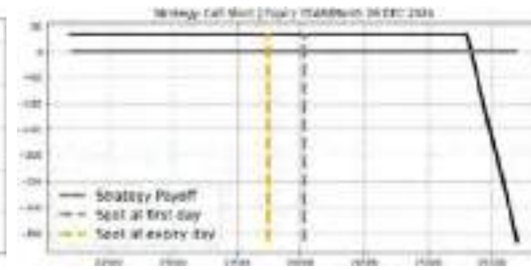


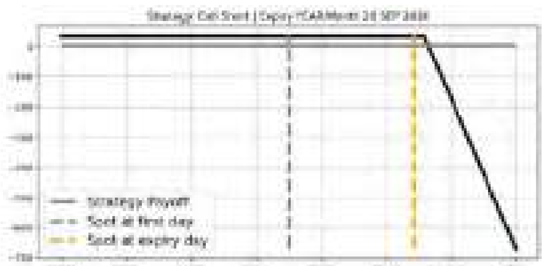
Strategy: Call Short **Testing Year:** 2024

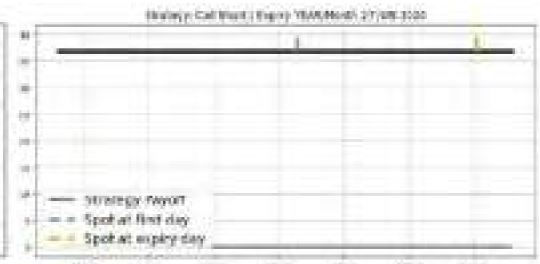

Strategy Payoff
Spot at first day
Spot at expiry day


Strategy Payoff
Spot at first day
Spot at expiry day


Strategy Payoff
Spot at first day
Spot at expiry day


Strategy Payoff
Spot at first day
Spot at expiry day


Strategy Payoff
Spot at first day
Spot at expiry day

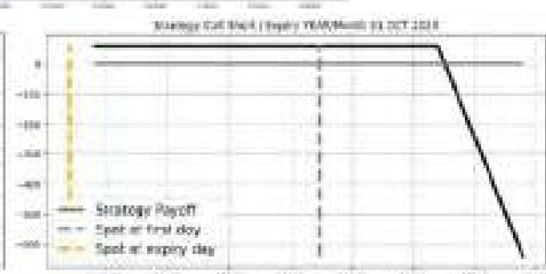

Strategy Payoff
Spot at first day
Spot at expiry day


Strategy Payoff
Spot at first day
Spot at expiry day

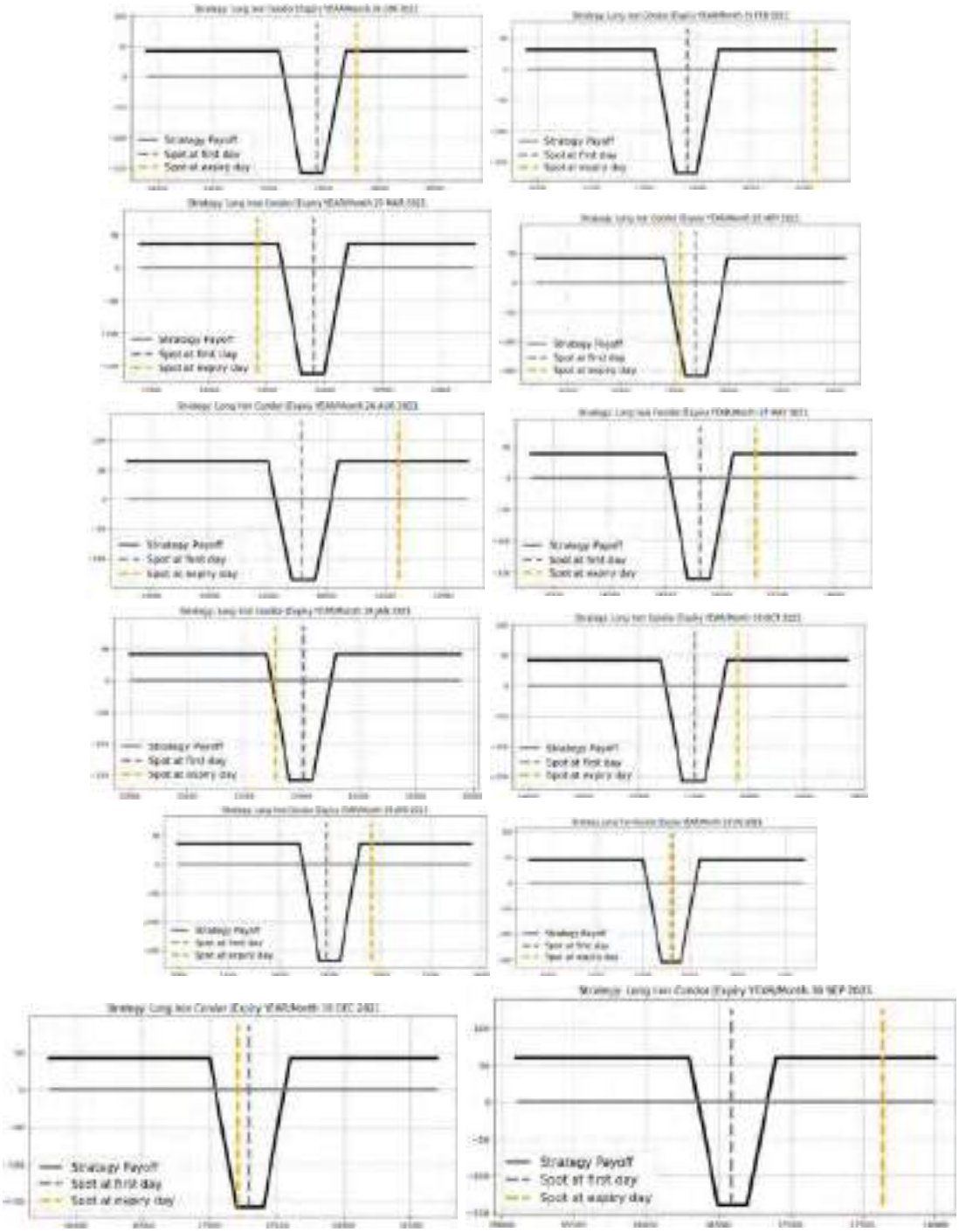

Strategy Payoff
Spot at first day
Spot at expiry day


Strategy Payoff
Spot at first day
Spot at expiry day

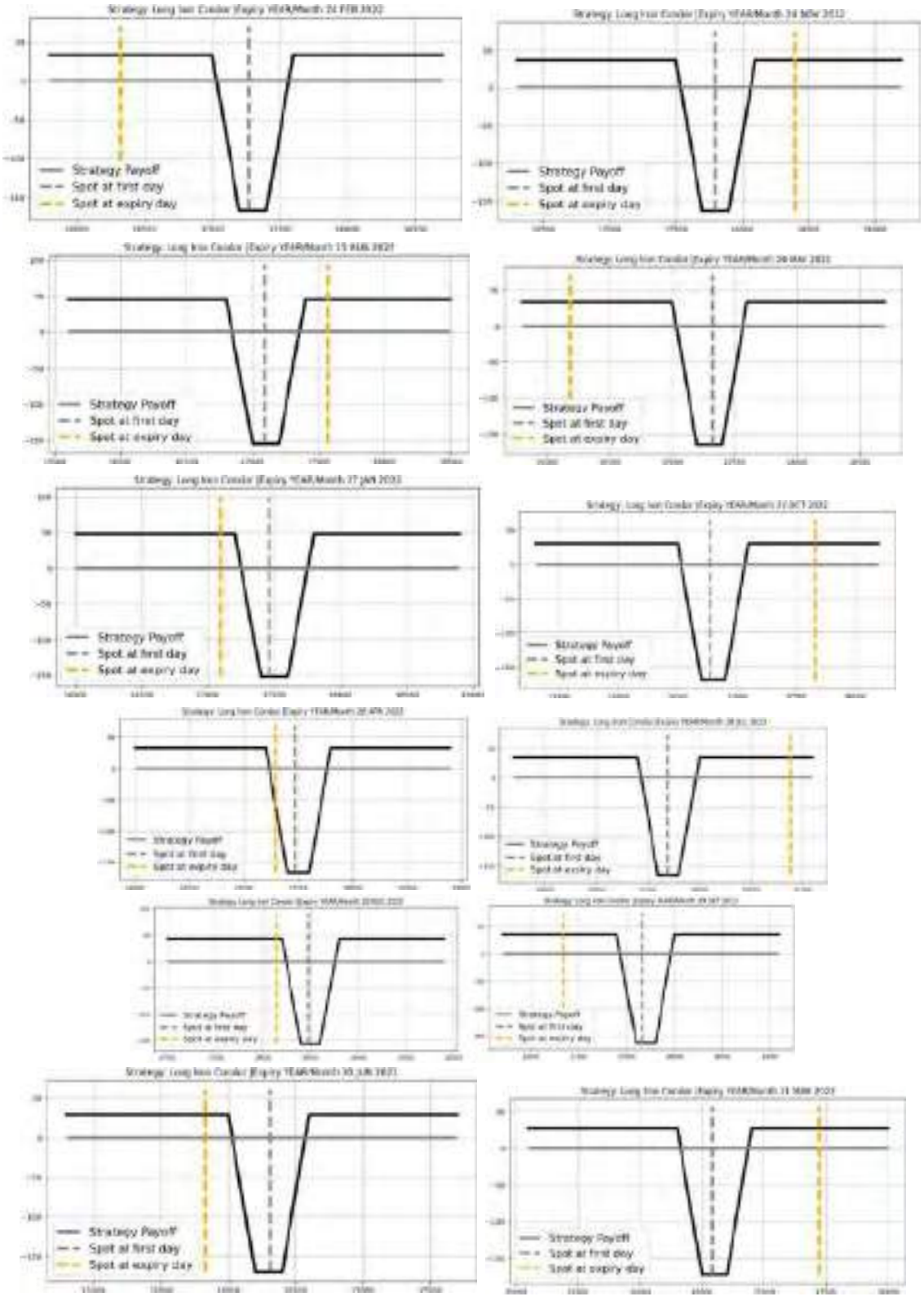

Strategy Payoff
Spot at first day
Spot at expiry day


Strategy Payoff
Spot at first day
Spot at expiry day

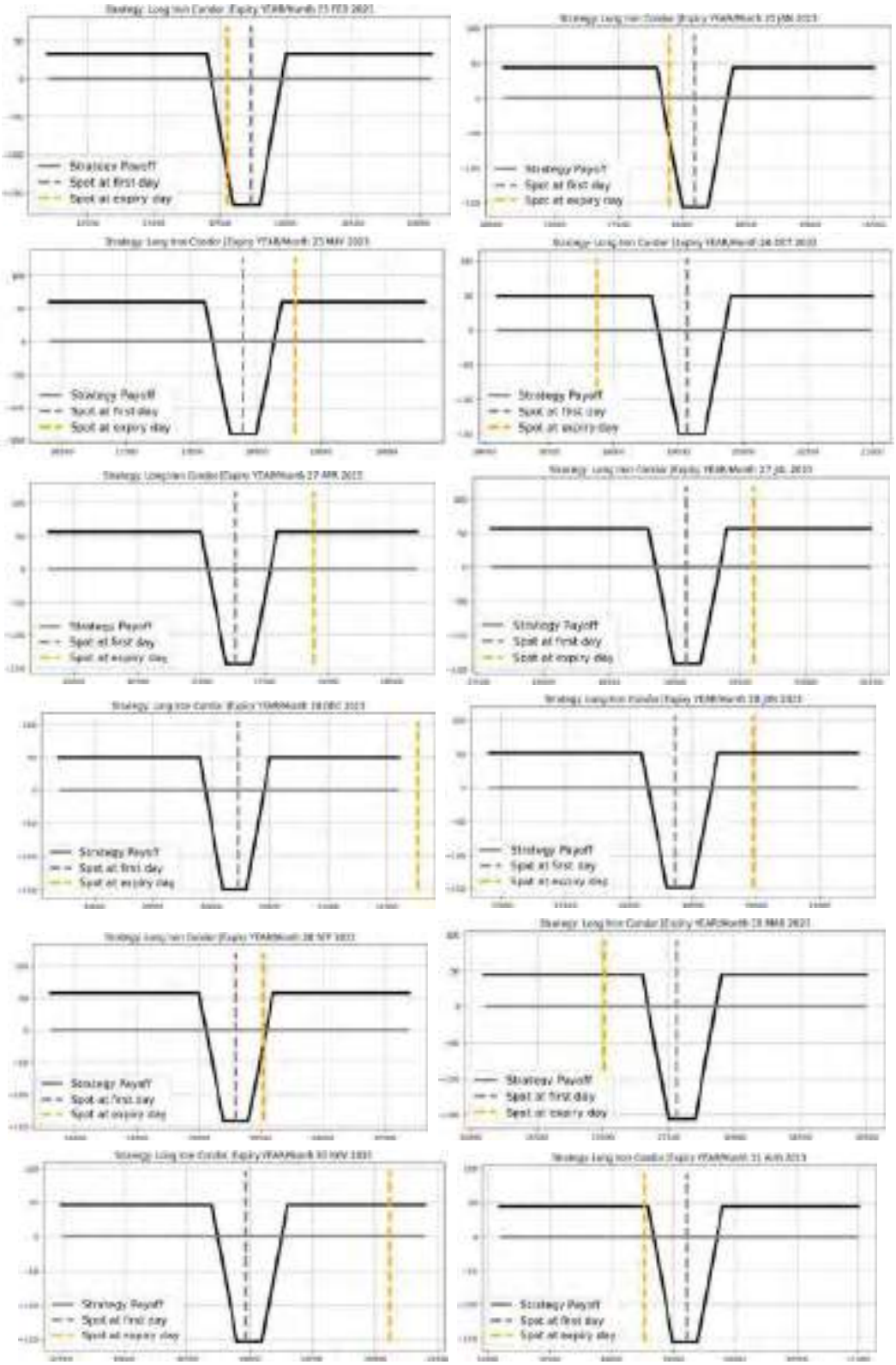
Strategy: Long Iron Condor Testing Year: 2021



Strategy: Long Iron Condor **Testing Year:** 2022

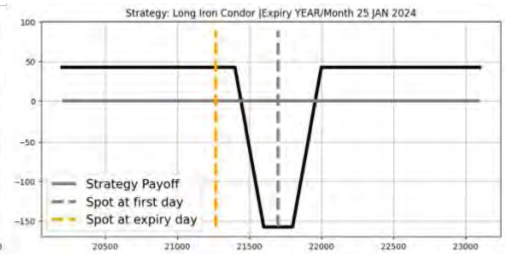


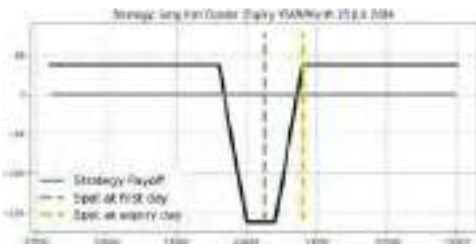
Strategy: Long Iron Condor **Testing Year:** 2023

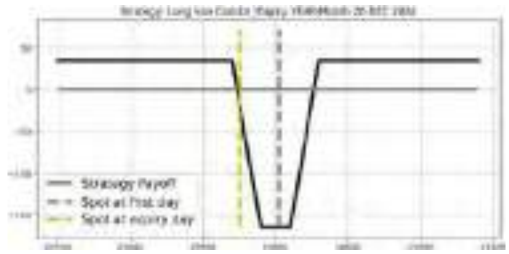


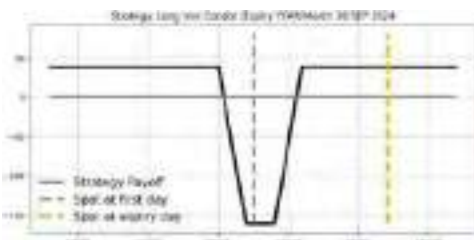
Strategy: Long Iron Condor **Testing Year:** 2024

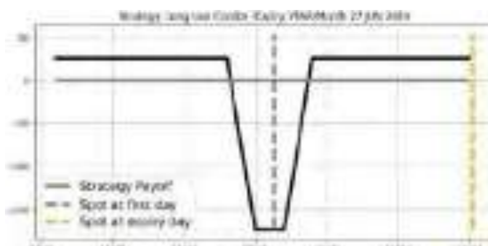

Strategy: Long Iron Condor |Expiry YEAR/Month 25 APR 2024
50
0
-50
-100
-150
Strategy Payoff
Spot at first day
Spot at expiry day
21000
21500
22000
22500
23000
23500
24000

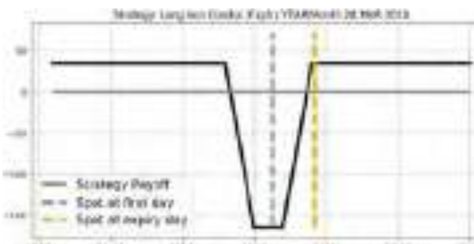

Strategy: Long Iron Condor |Expiry YEAR/Month 25 JAN 2024
100
50
0
-50
-100
-150
Strategy Payoff
Spot at first day
Spot at expiry day
20500
21000
21500
22000
22500
23000

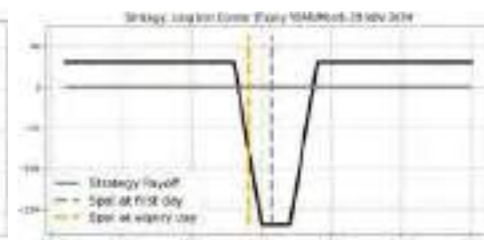


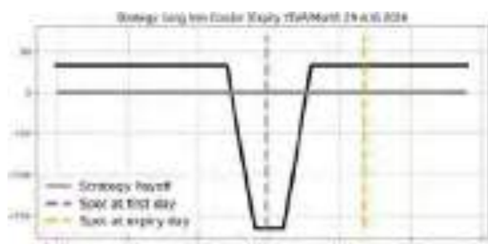


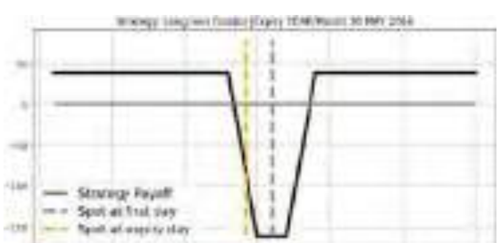




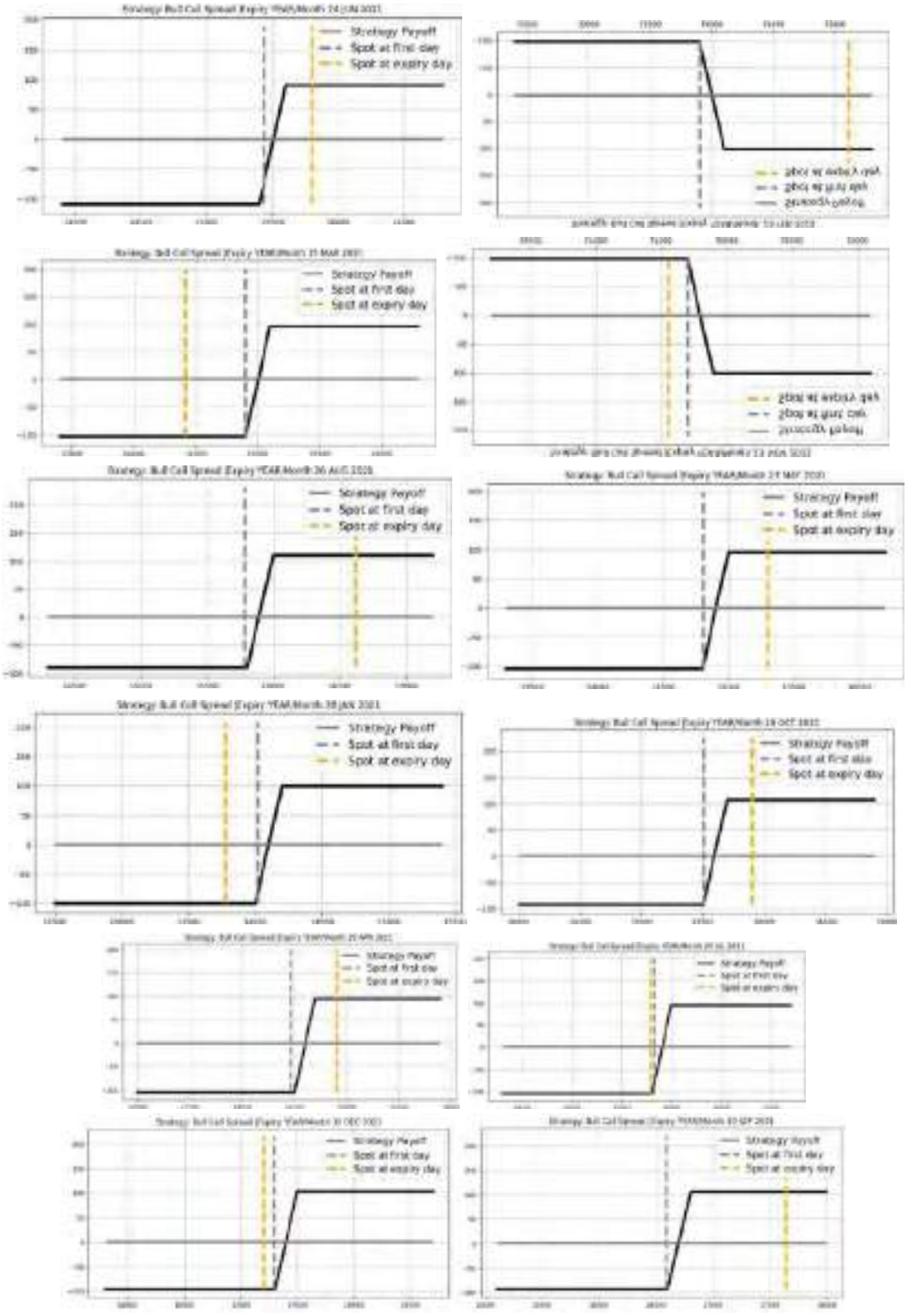




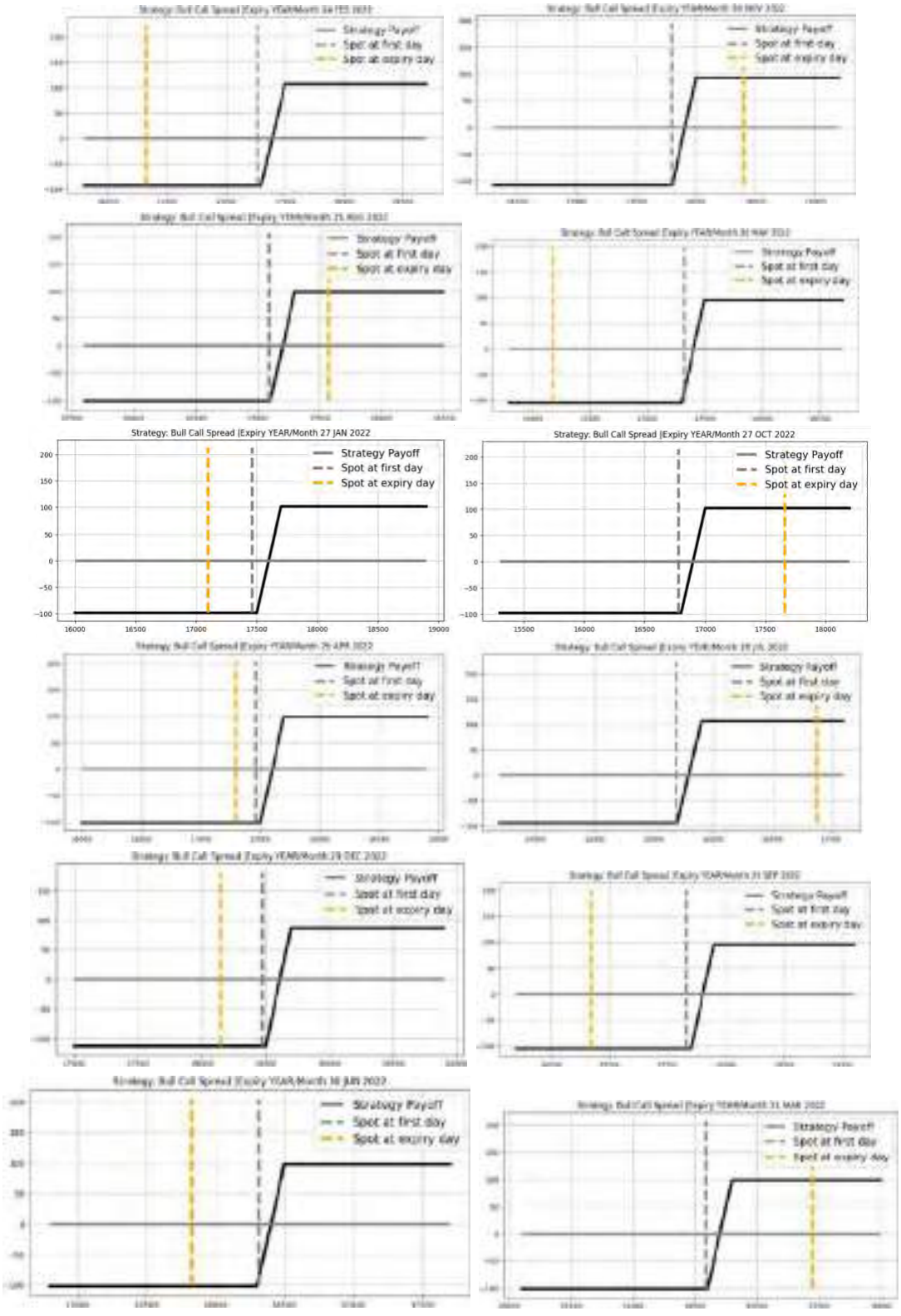




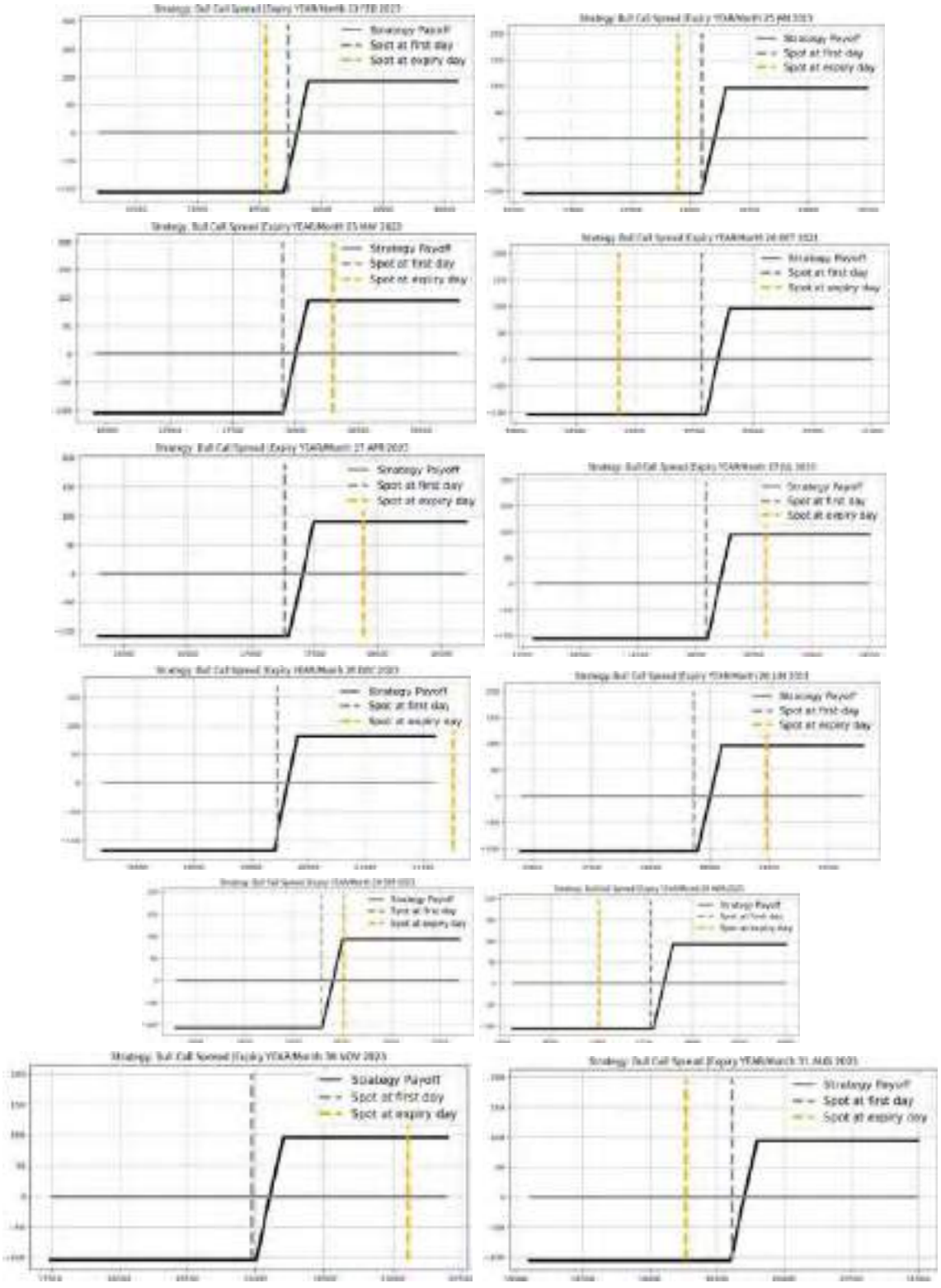
Strategy: Bull Call Spread Testing Year: 2021



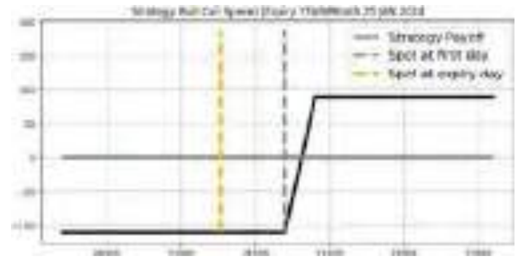
Strategy: Bull Call Spread **Testing Year:** 2022

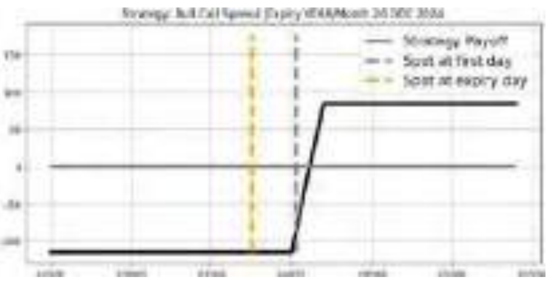


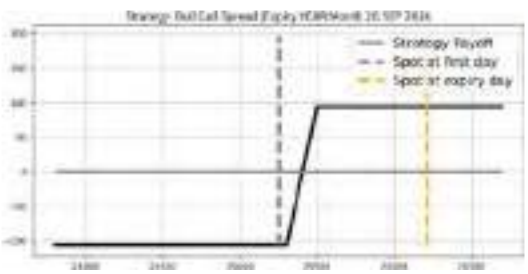
Strategy: Bull Call Spread **Testing Year:** 2023

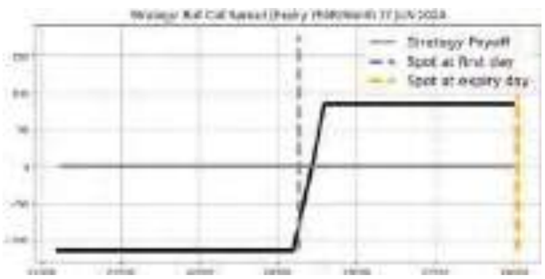


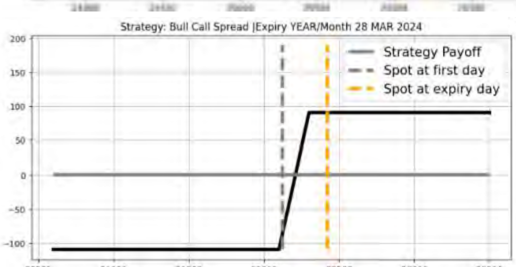
Strategy: Bull Call Spread **Testing Year:** 2024



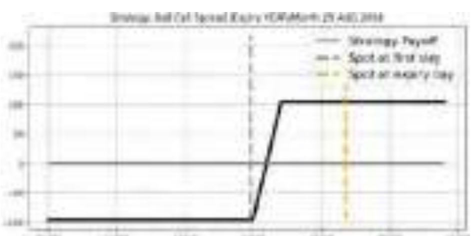




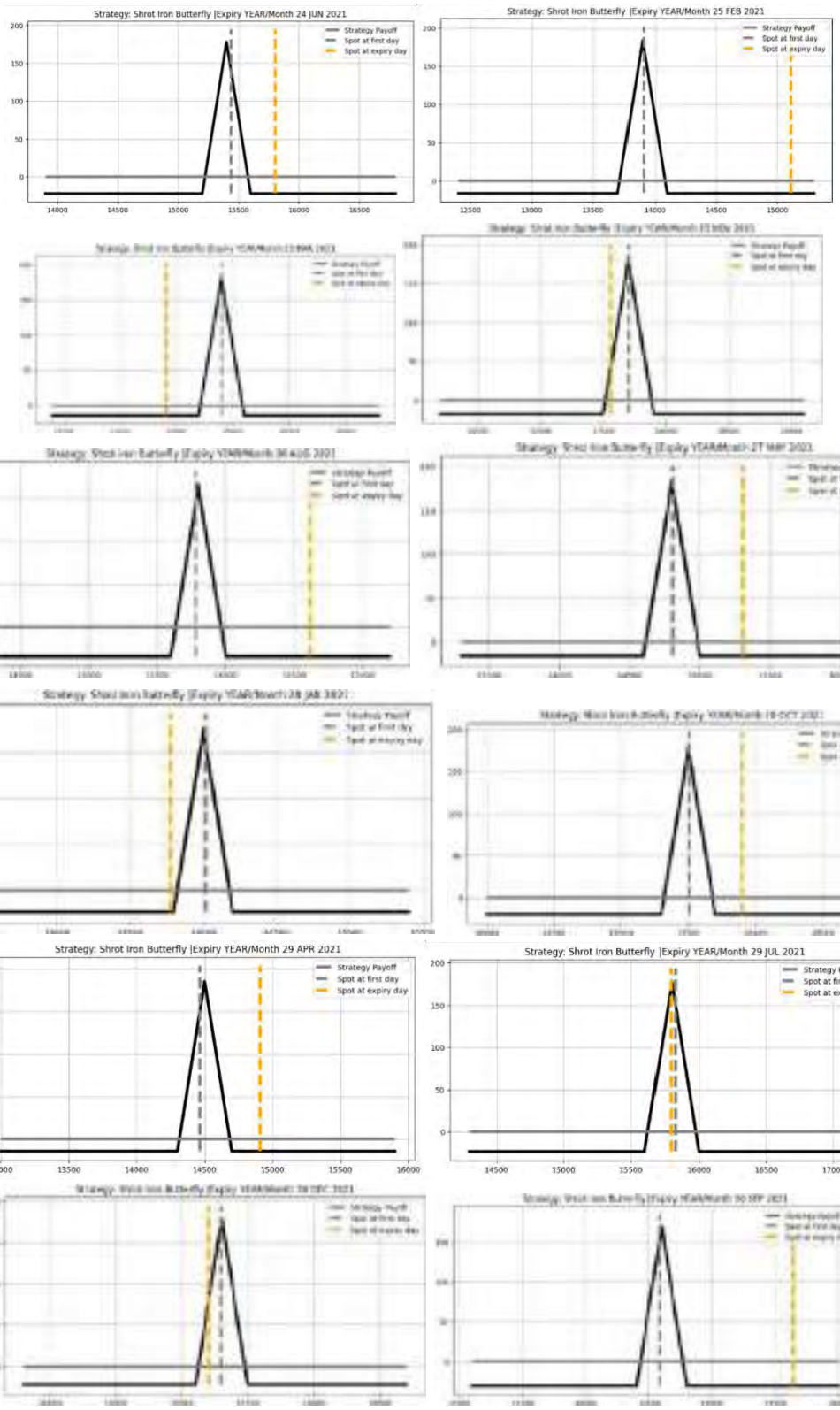



Strategy: Bull Call Spread |Expiry YEAR/Month 28 MAR 2024
Strategy Payoff
Spot at first day
Spot at expiry day
200
150
100
50
0
−50
−100
20500
21000
21500
22000
22500
23000
23500

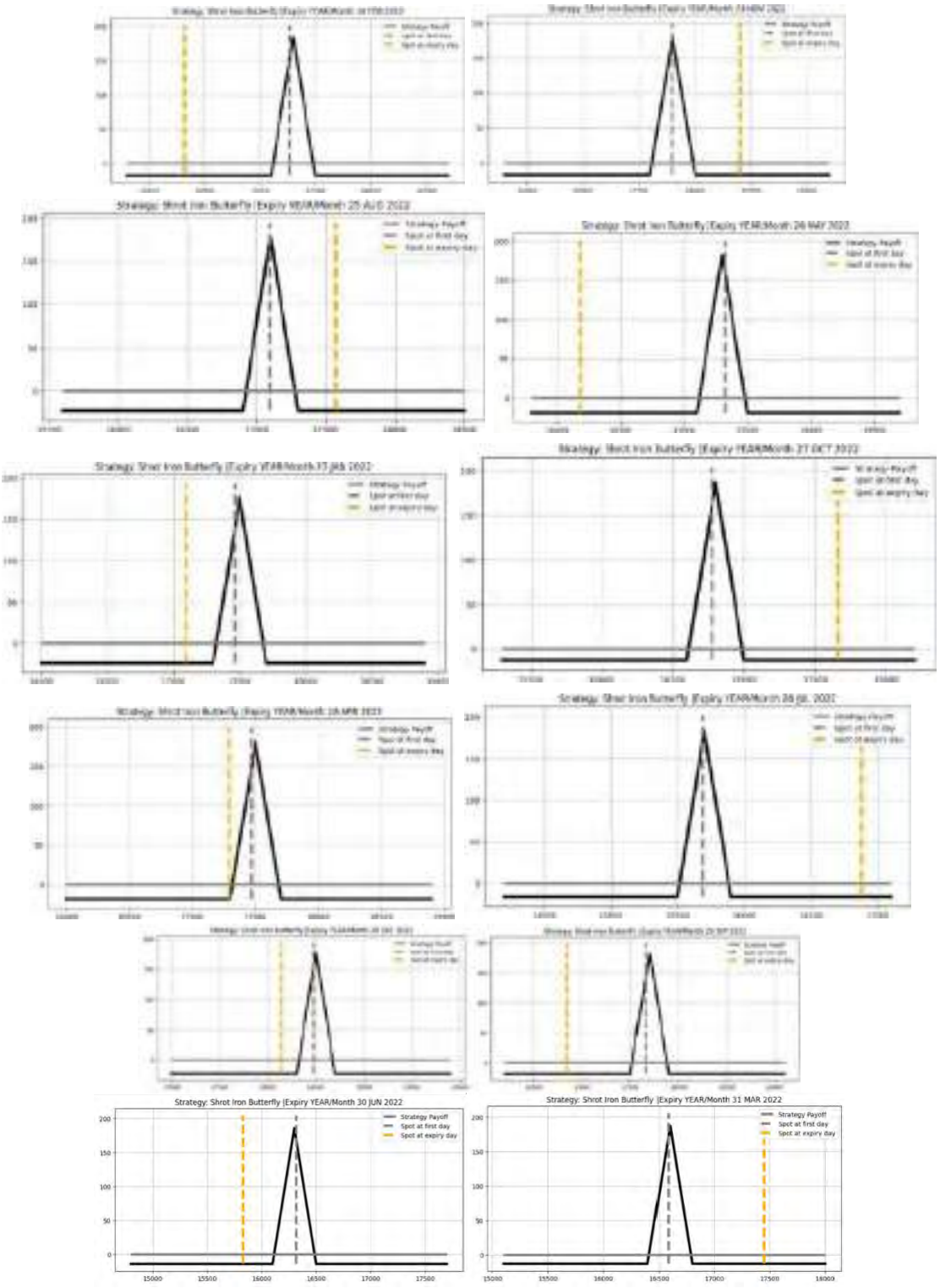

Strategy: Bull Call Spread |Expiry YEAR/Month 28 NOV 2024
Strategy Payoff
Spot at first day
Spot at expiry day
200
150
100
50
0
−50
−100
22500
23000
23500
24000
24500
25000
25500



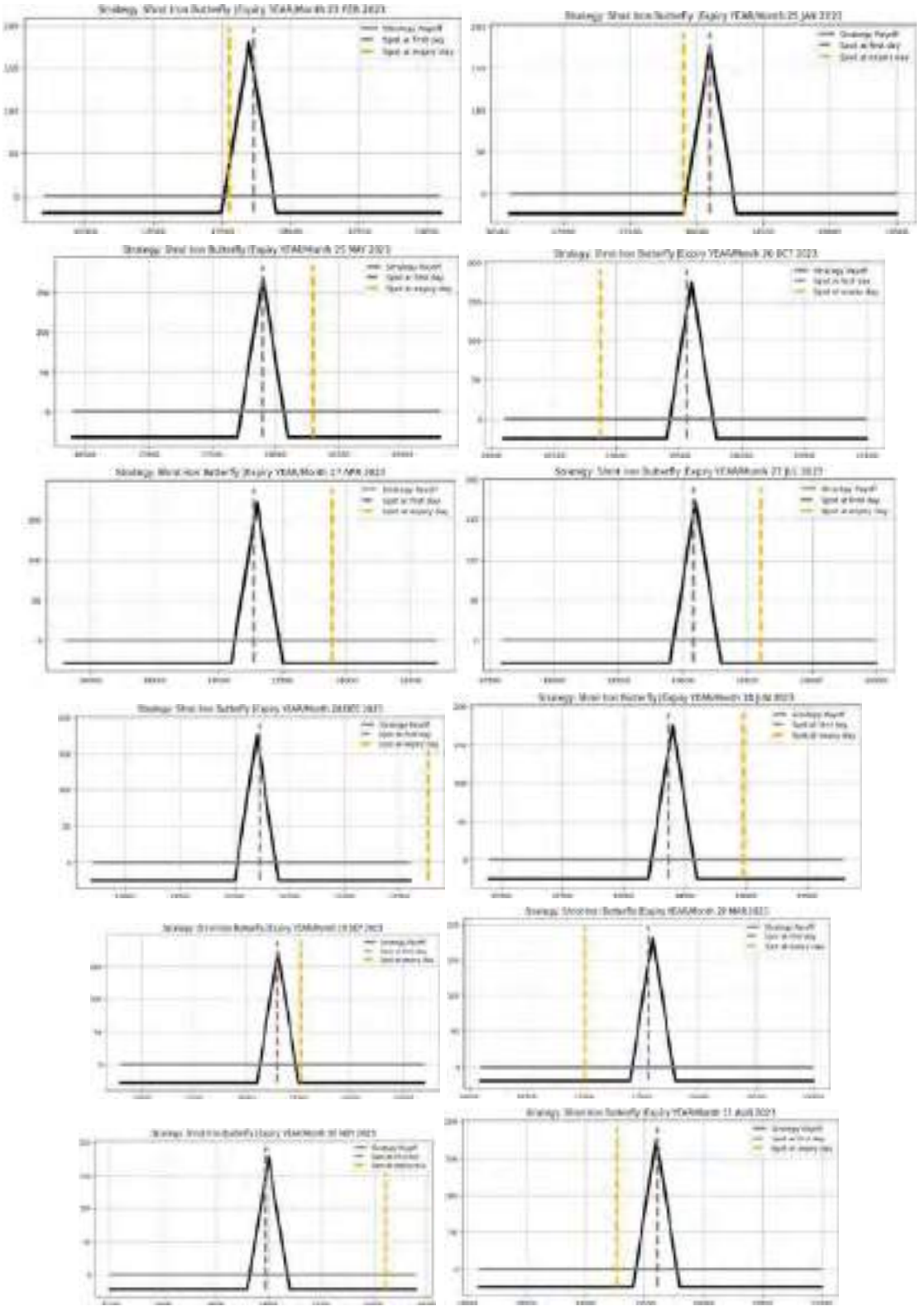
Strategy: Short Iron Condor Testing Year: 2021



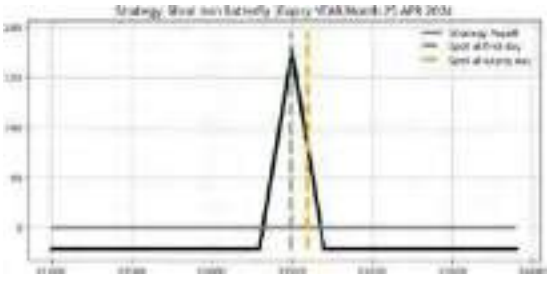
Strategy: Short Iron Condor **Testing Year:** 2022

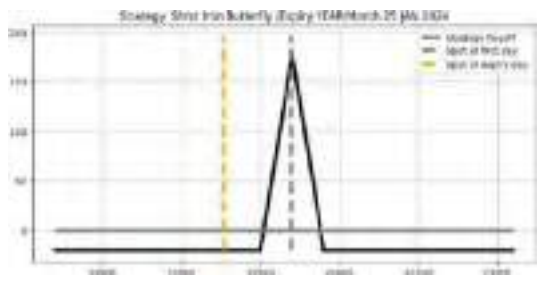


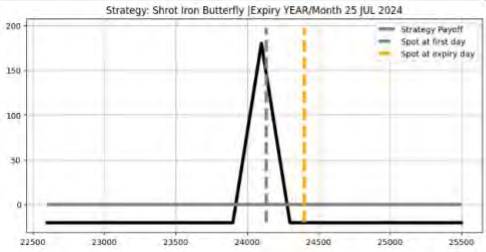
Strategy: Short Iron Condor **Testing Year:** 2023

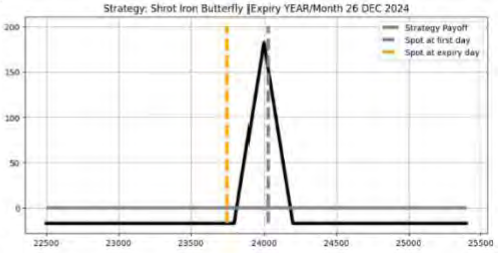


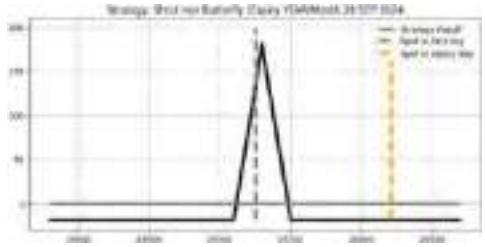
Strategy: Short Iron Condor **Testing Year:** 2024

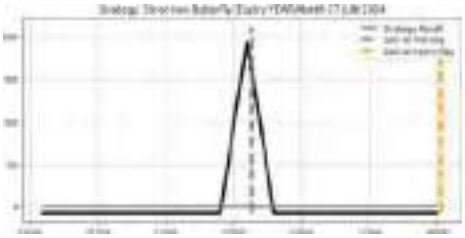


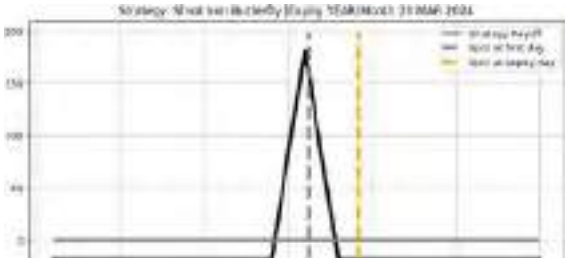


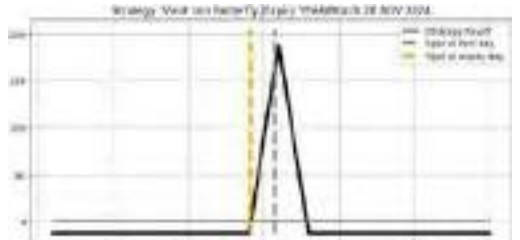

Strategy: Shrot Iron Butterfly |Expiry YEAR/Month 25 JUL 2024
Strategy Payoff
Spot at first day
Spot at expiry day
200
150
100
50
0
22500
23000
23500
24000
24500
25000
25500

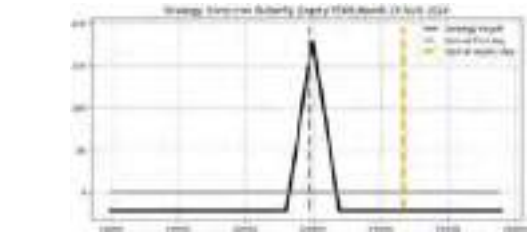

Strategy: Shrot Iron Butterfly |Expiry YEAR/Month 26 DEC 2024
Strategy Payoff
Spot at first day
Spot at expiry day
200
150
100
50
0
22500
23000
23500
24000
24500
25000
25500

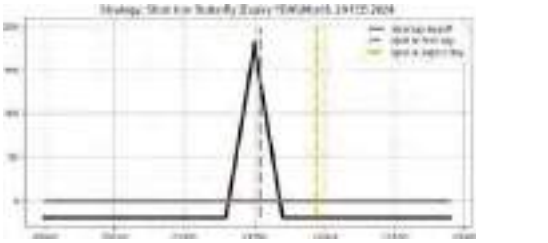


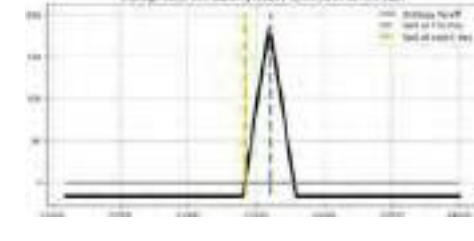


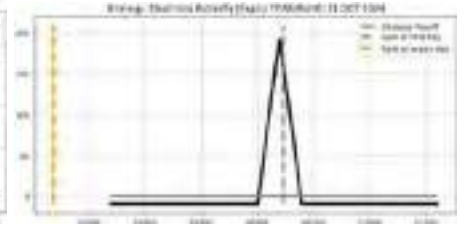




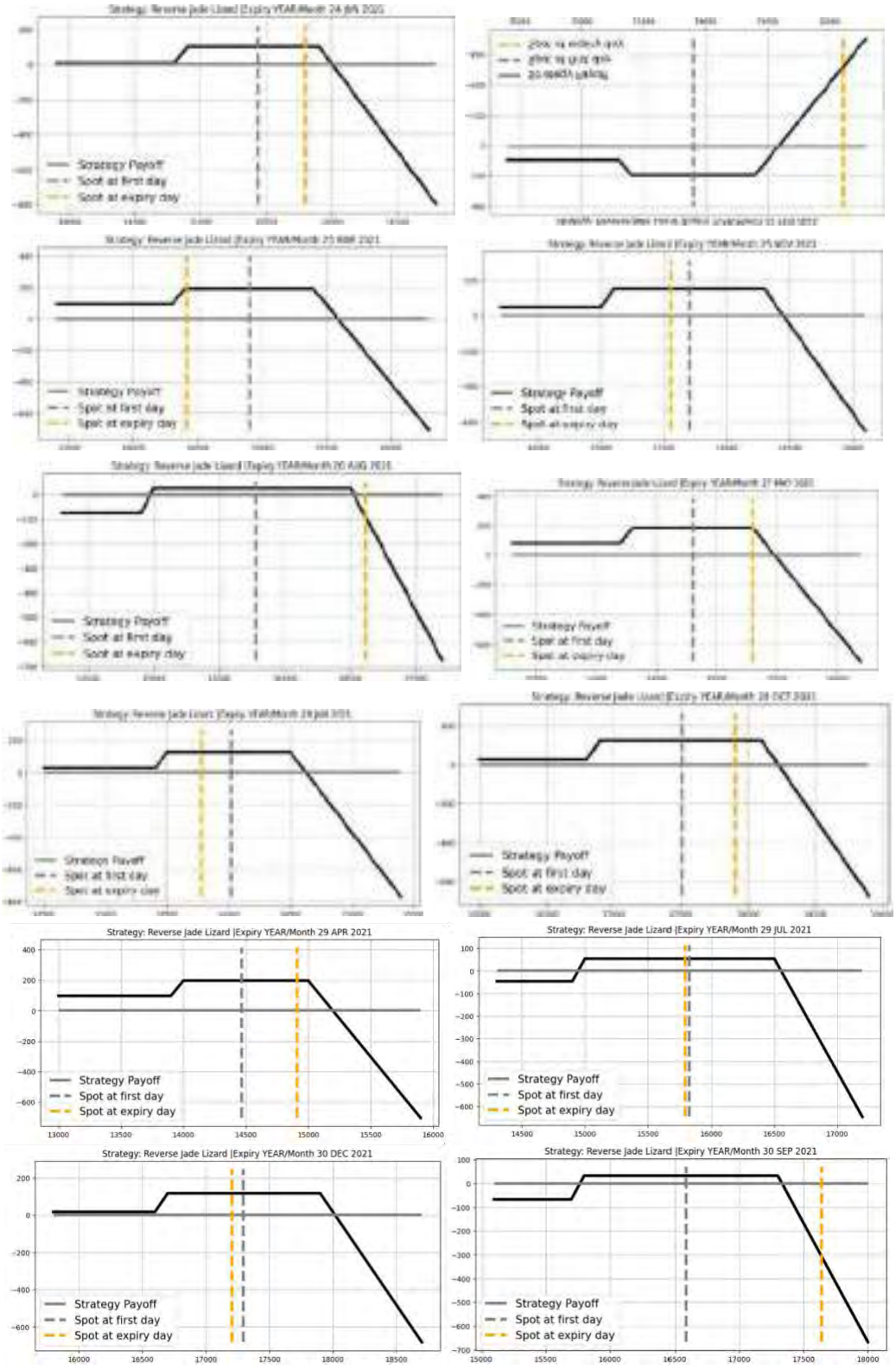




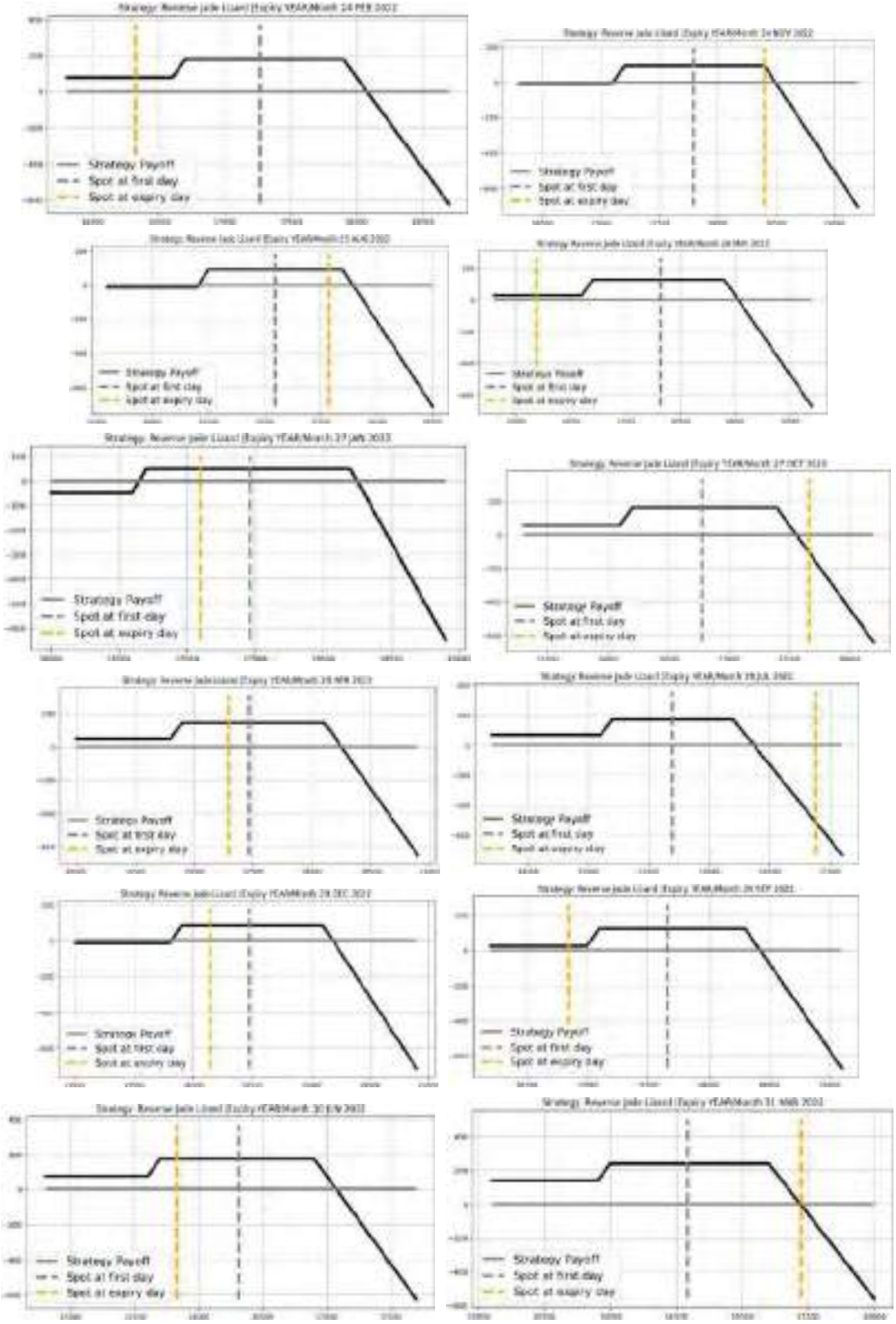




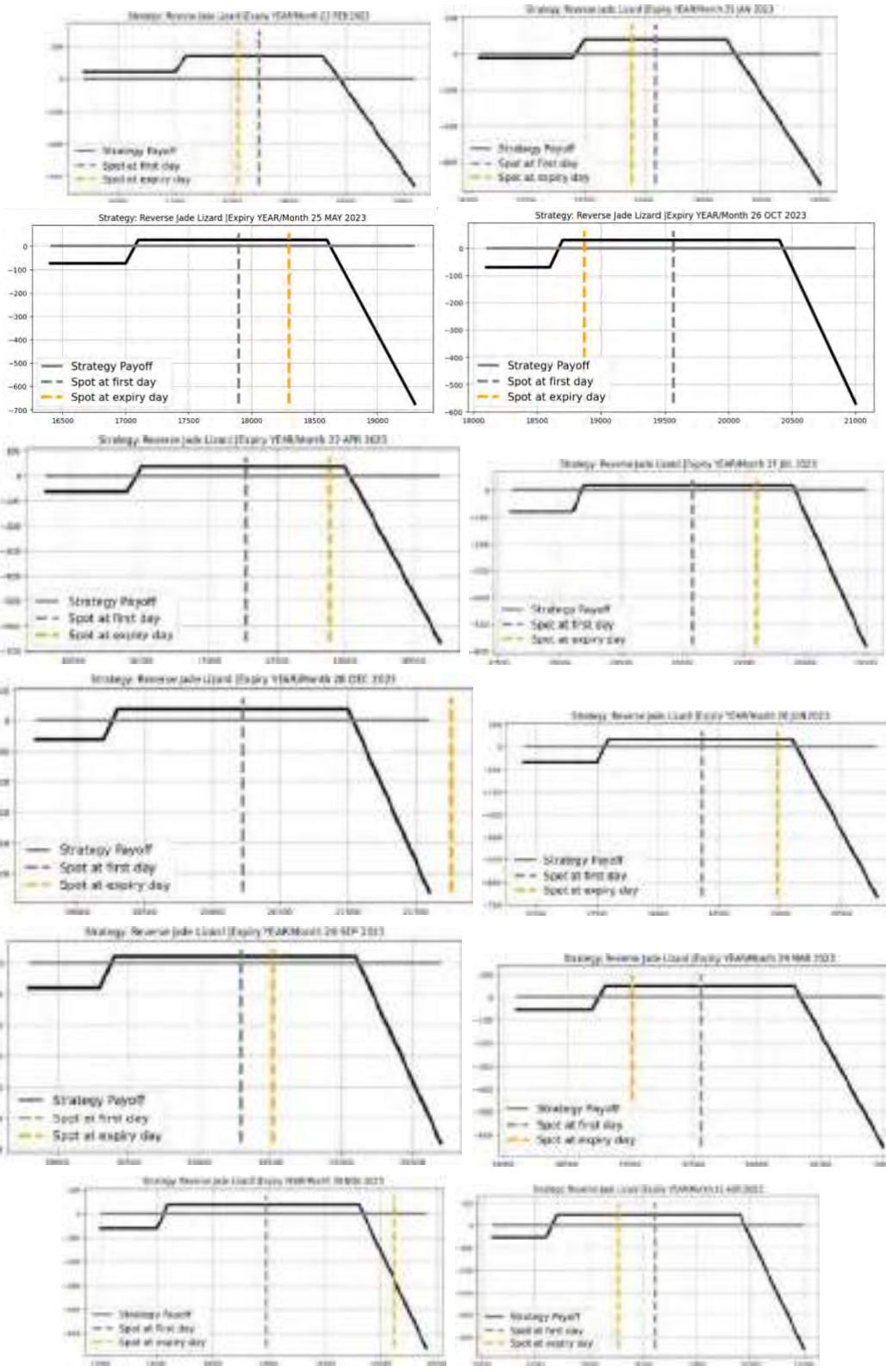
Strategy: Reverse Jade Lizard Testing Year: 2021



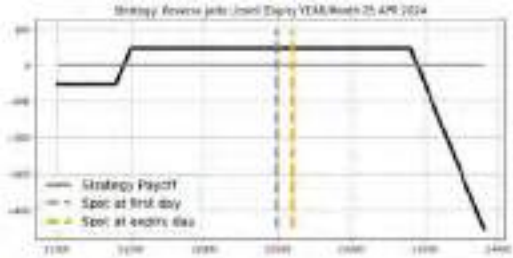
Strategy: Reverse Jade Lizard Testing Year: 2022



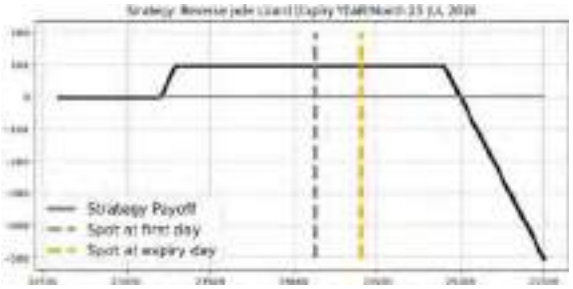
Strategy: Reverse Jade Lizard **Testing Year:** 2023

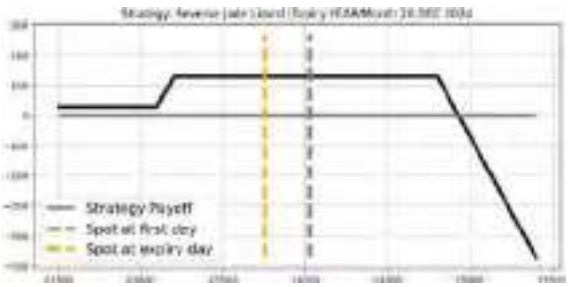


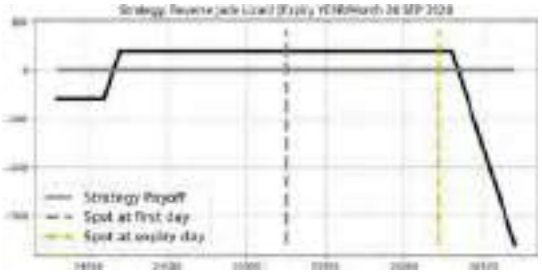
Strategy: Reverse Jade Lizard **Testing Year:** 2024

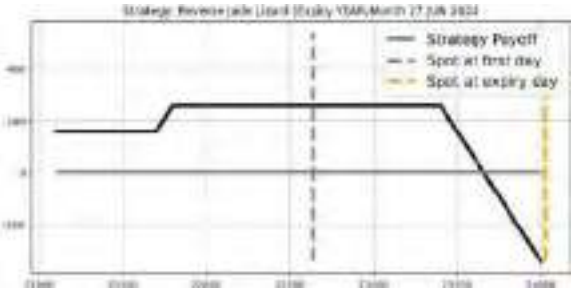

Strategy Payoff
Spot at first day
Spot at expiry day

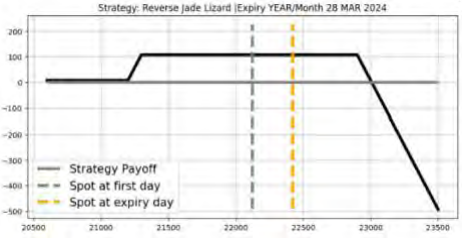

Strategy Payoff
Spot at first day
Spot at expiry day


Strategy: Reverse Jade Lizard |Expiry YEAR/Month 25 JUL 2024
Strategy Payoff
Spot at first day
Spot at expiry day

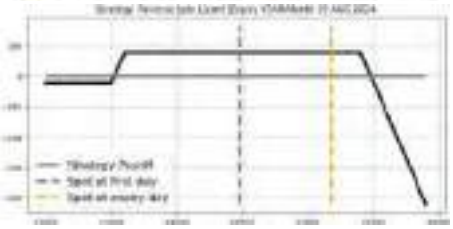

Strategy Payoff
Spot at first day
Spot at expiry day

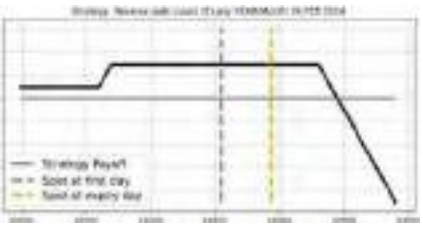

Strategy: Reverse Jade Lizard |Expiry YEAR/Month 26 SEP 2024
Strategy Payoff
Spot at first day
Spot at expiry day

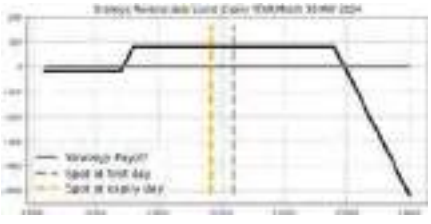

Strategy: Reverse Jade Lizard |Expiry YEAR/Month 27 JUN 2024
Strategy Payoff
Spot at first day
Spot at expiry day

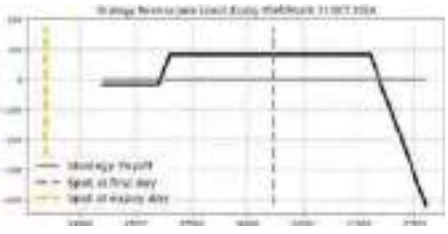

Strategy: Reverse Jade Lizard |Expiry YEAR/Month 28 MAR 2024
Strategy Payoff
Spot at first day
Spot at expiry day


Strategy: Reverse Jade Lizard |Expiry YEAR/Month 28 NOV 2024
Strategy Payoff
Spot at first day
Spot at expiry day

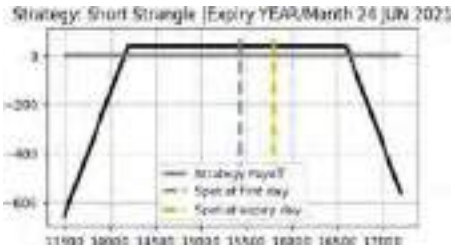


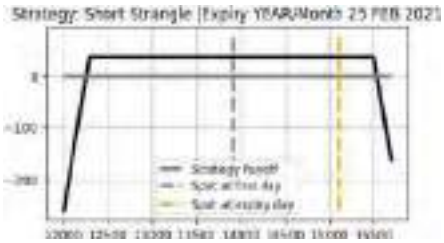


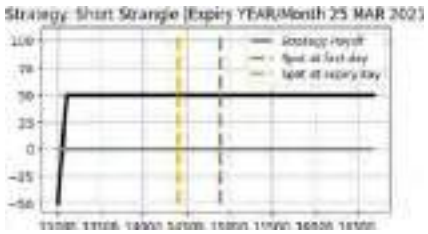


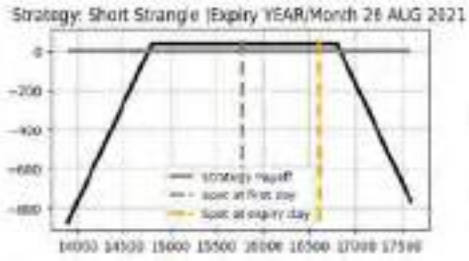


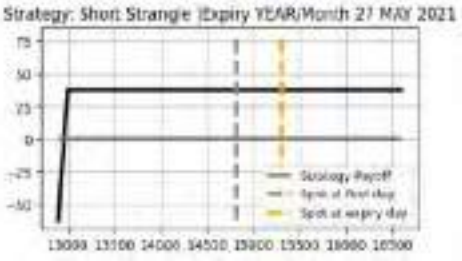
Strategy: Short Strangle Testing Year: 2021

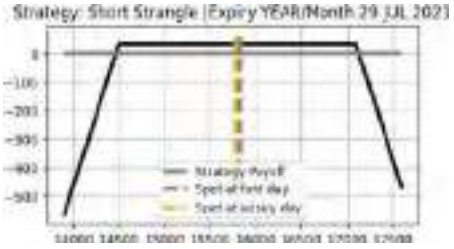


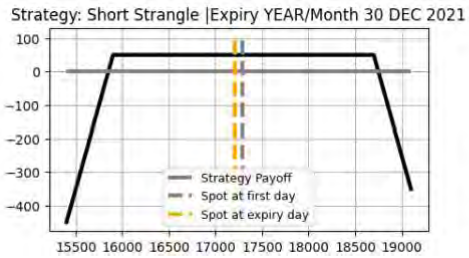


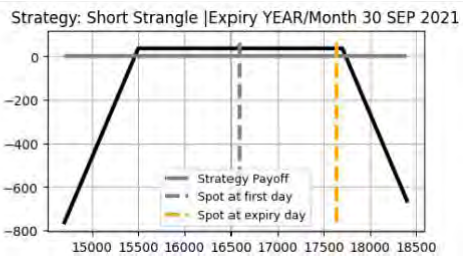




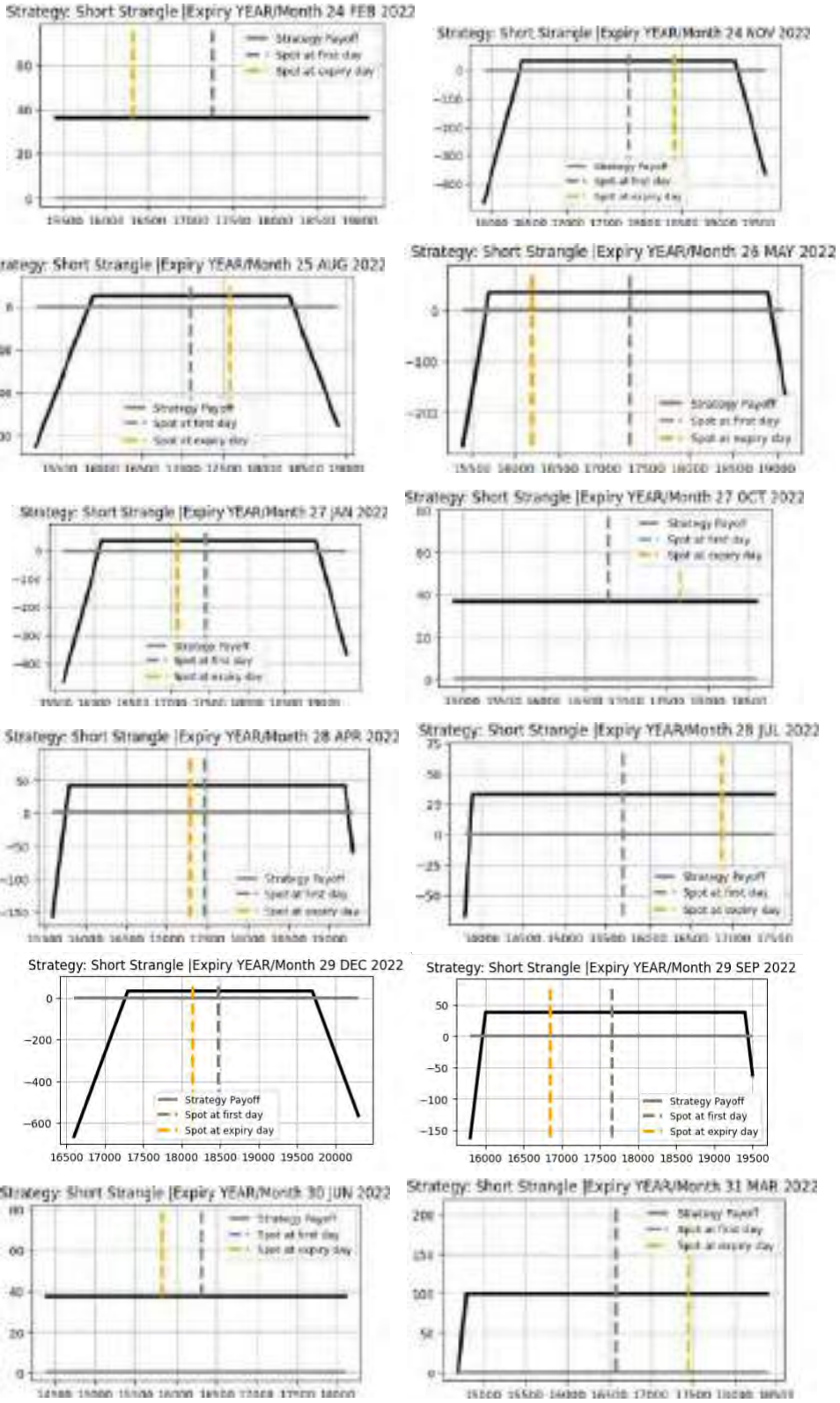




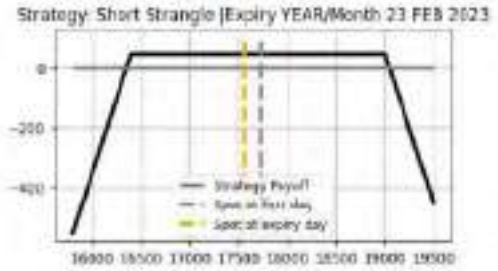


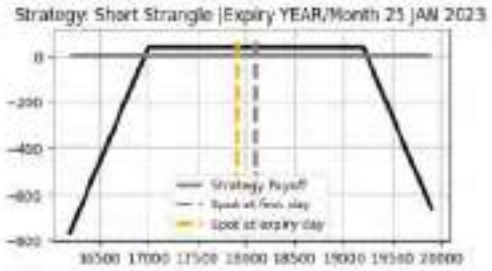


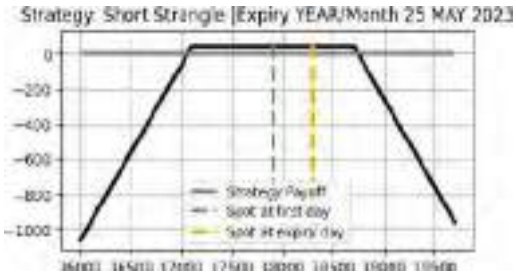
Strategy: Short Strangle **Testing Year:** 2022

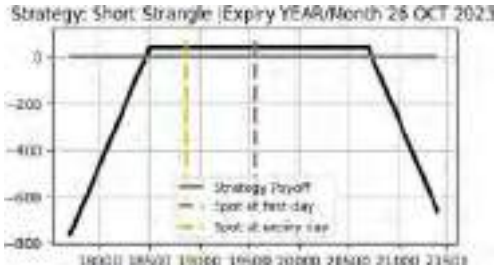


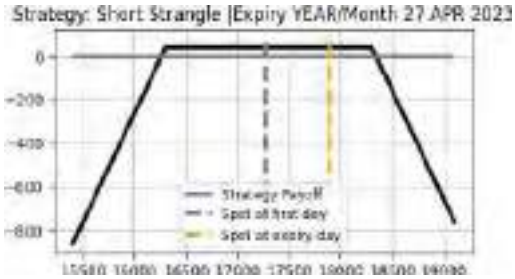
Strategy: Short Strangle **Testing Year:** 2023

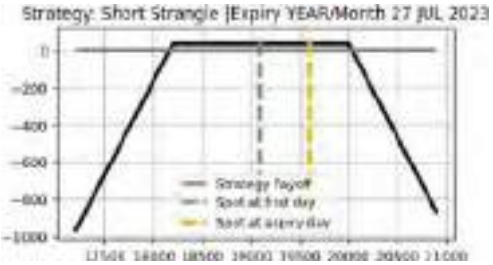


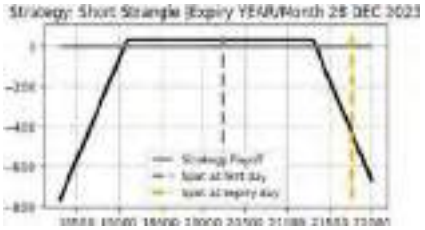


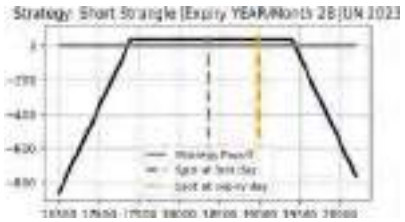


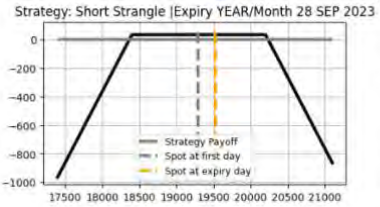


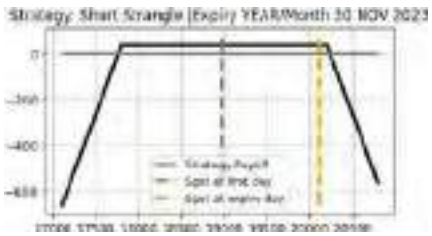






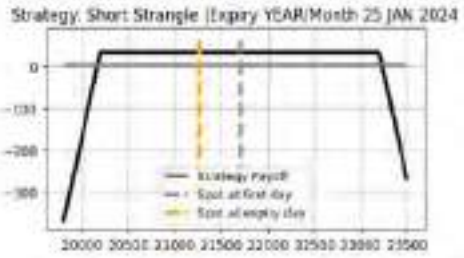


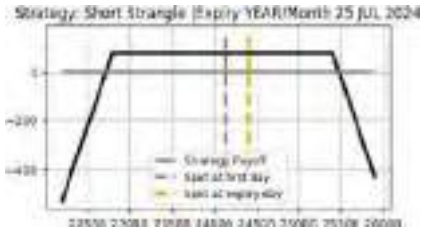




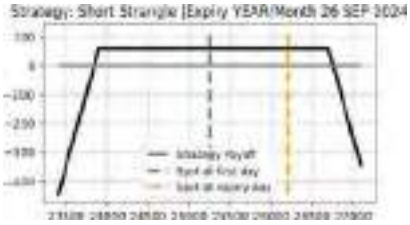
Strategy: Short Strangle **Testing Year:** 2024


Strategy: Short Strangle |Expiry YEAR/Month 25 APR 2024

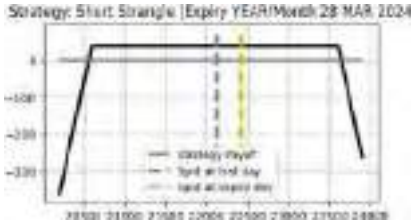

Strategy: Short Strangle |Expiry YEAR/Month 25 JAN 2024

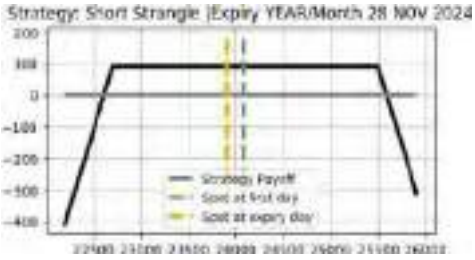

Strategy: Short Strangle |Expiry YEAR/Month 25 JUL 2024


Strategy: Short Strangle |Expiry YEAR/Month 26 DEC 2024


Strategy: Short Strangle |Expiry YEAR/Month 26 SEP 2024

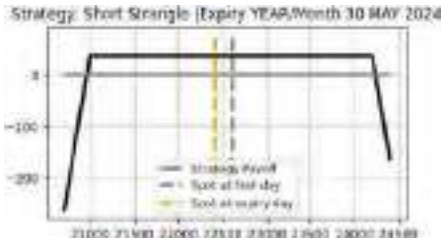

Strategy: Short Strangle |Expiry YEAR/Month 27 JUN 2024

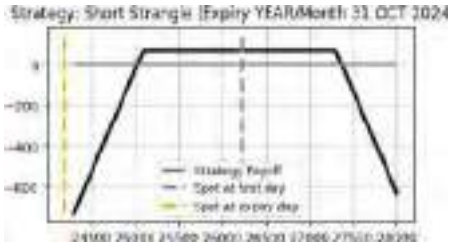

Strategy: Short Strangle |Expiry YEAR/Month 28 MAR 2024


Strategy: Short Strangle |Expiry YEAR/Month 28 NOV 2024

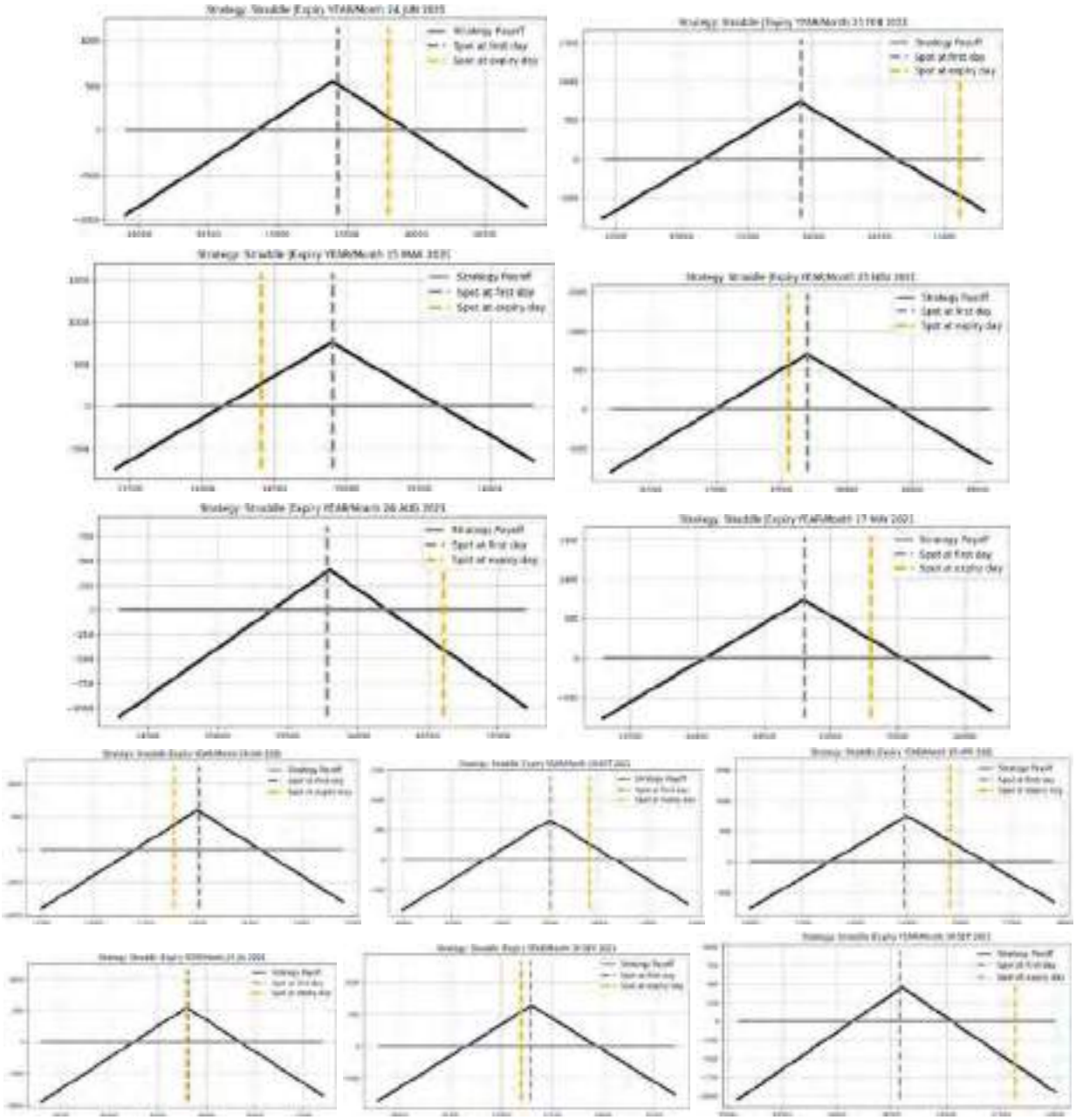

Strategy: Short Strangle |Expiry YEAR/Month 29 AUG 2024


Strategy: Short Strangle |Expiry YEAR/Month 29 FEB 2024

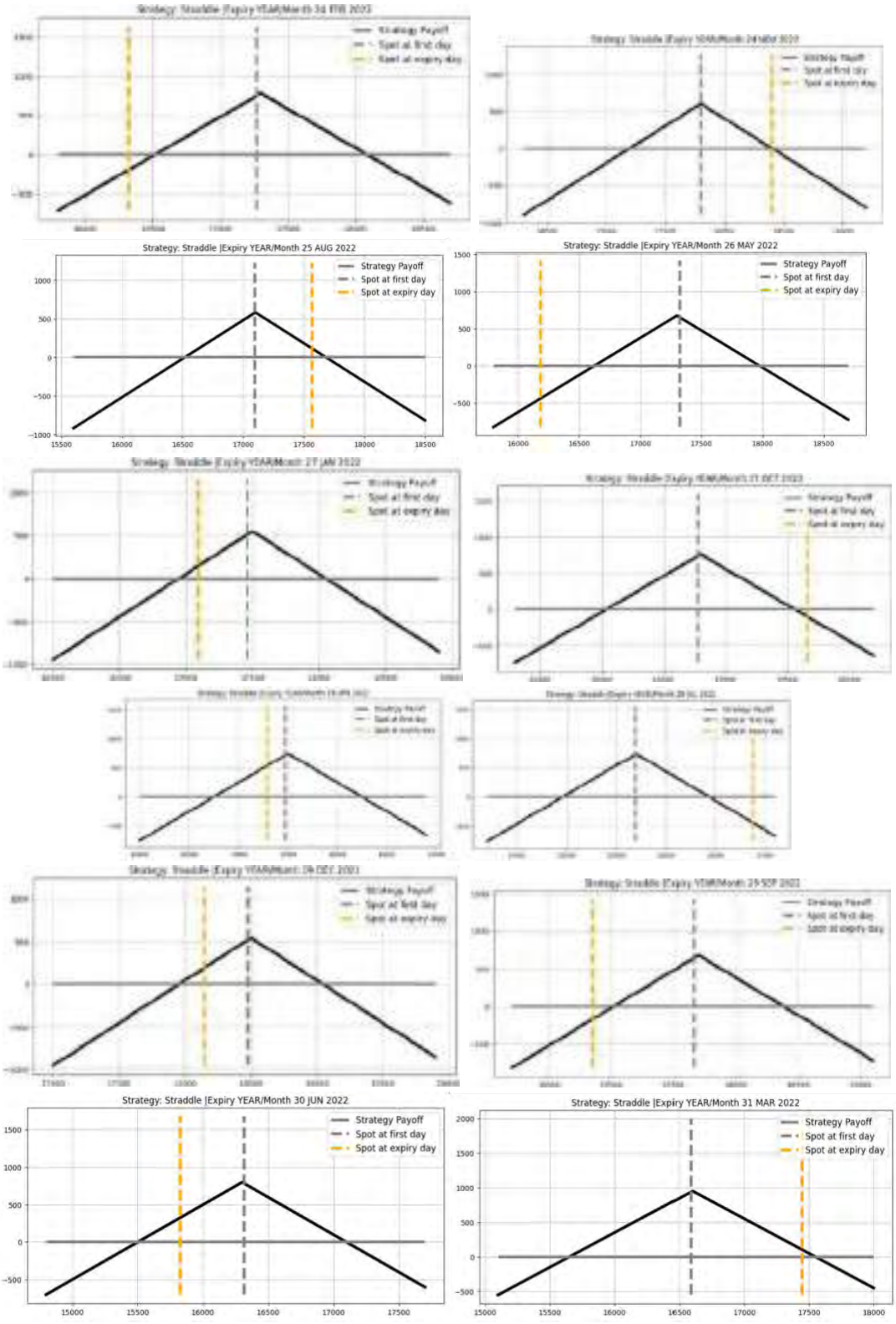

Strategy: Short Strangle |Expiry YEAR/Month 30 MAY 2024


Strategy: Short Strangle |Expiry YEAR/Month 31 OCT 2024

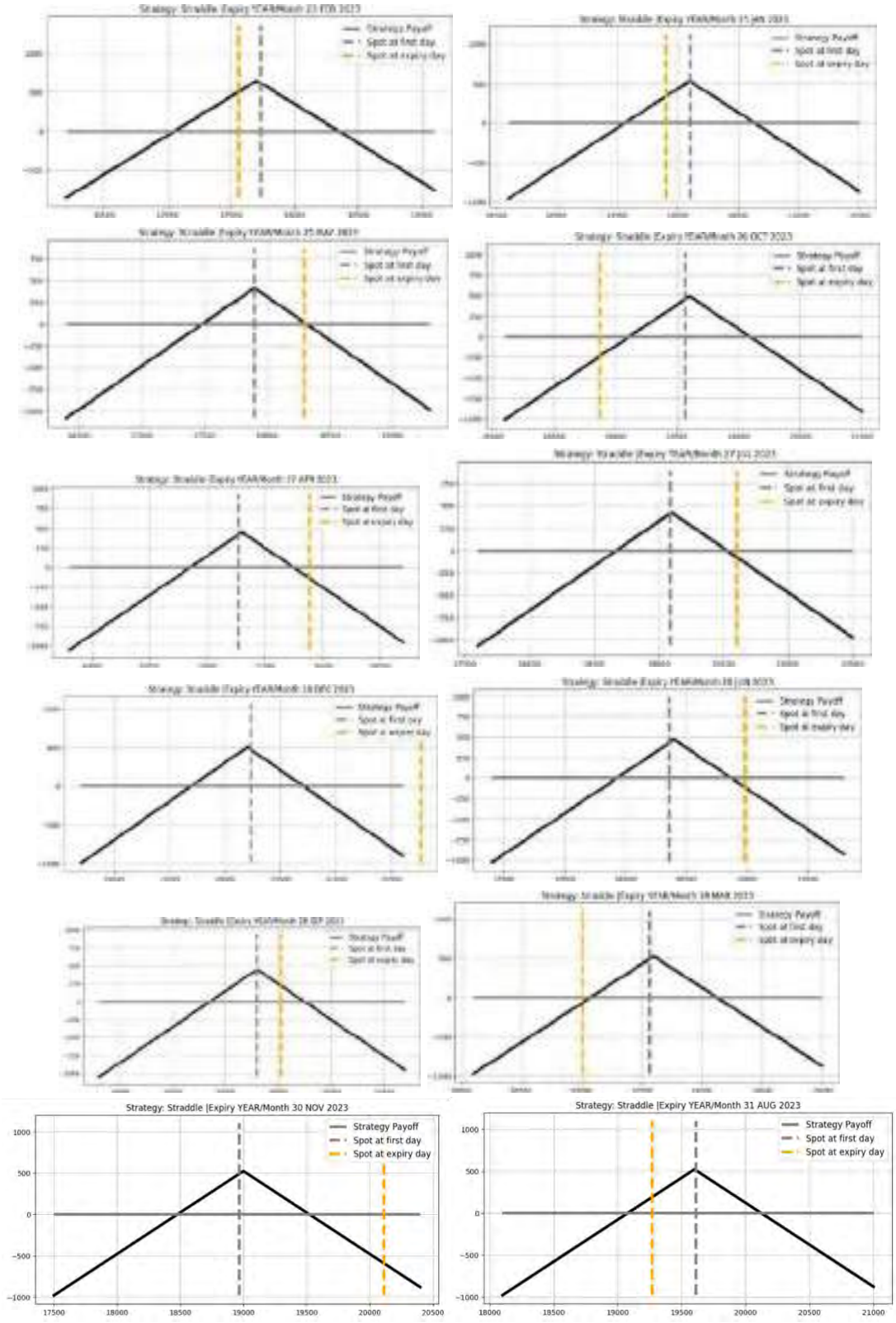
Strategy: Short Straddle Testing Year: 2021



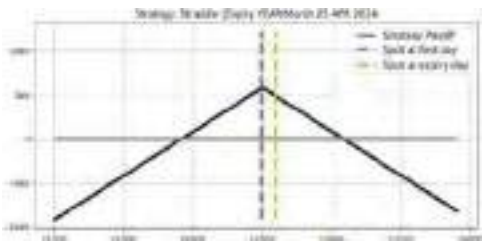
Strategy: Short Straddle **Testing Year:** 2022

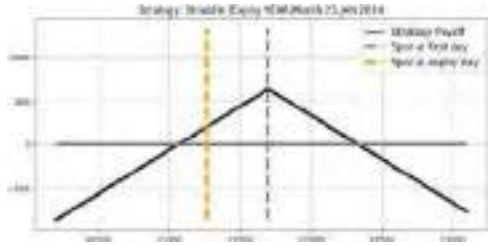


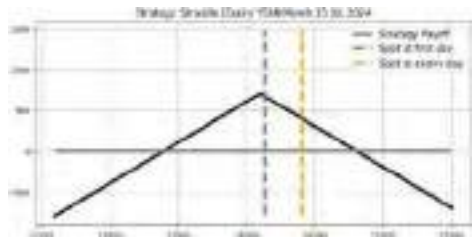
Strategy: Short Straddle **Testing Year:** 2023

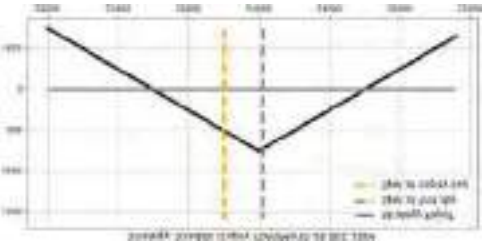


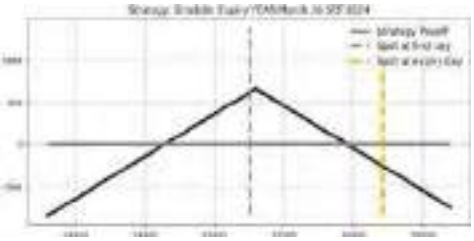
Strategy: Short Straddle **Testing Year:** 2024

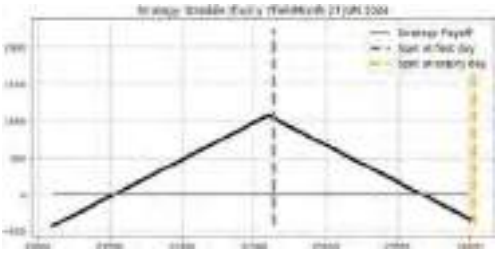


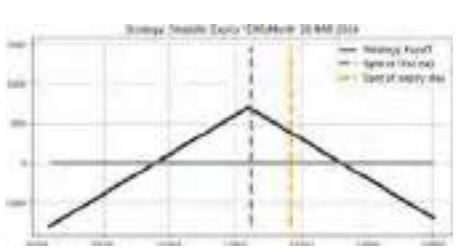


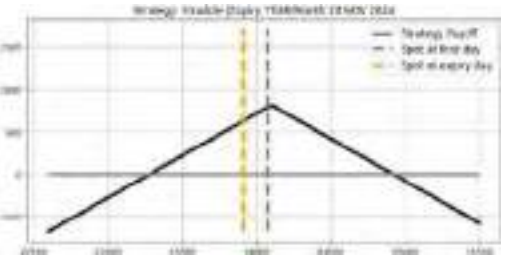


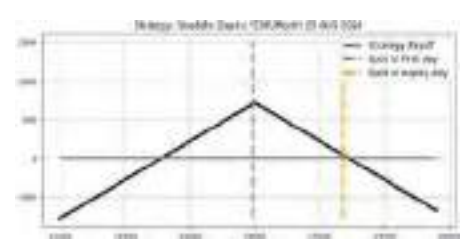


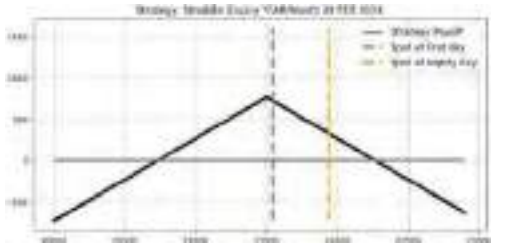


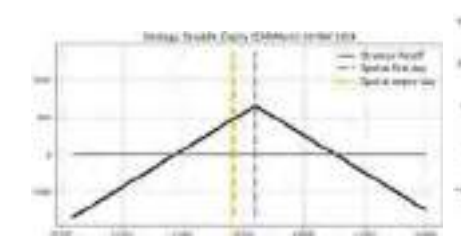


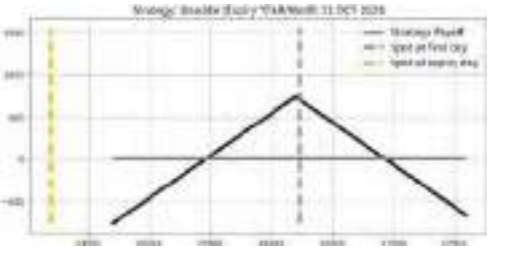




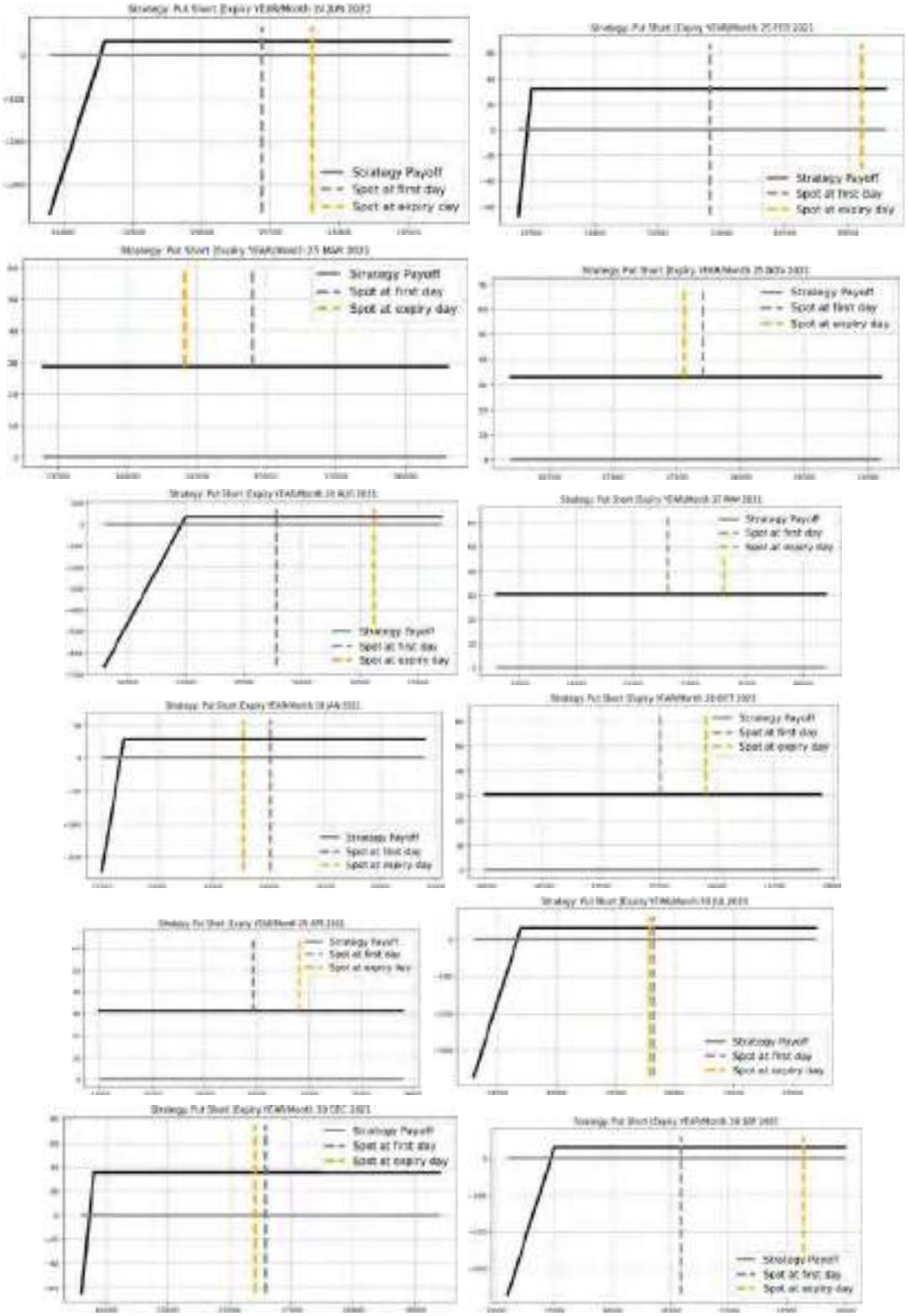




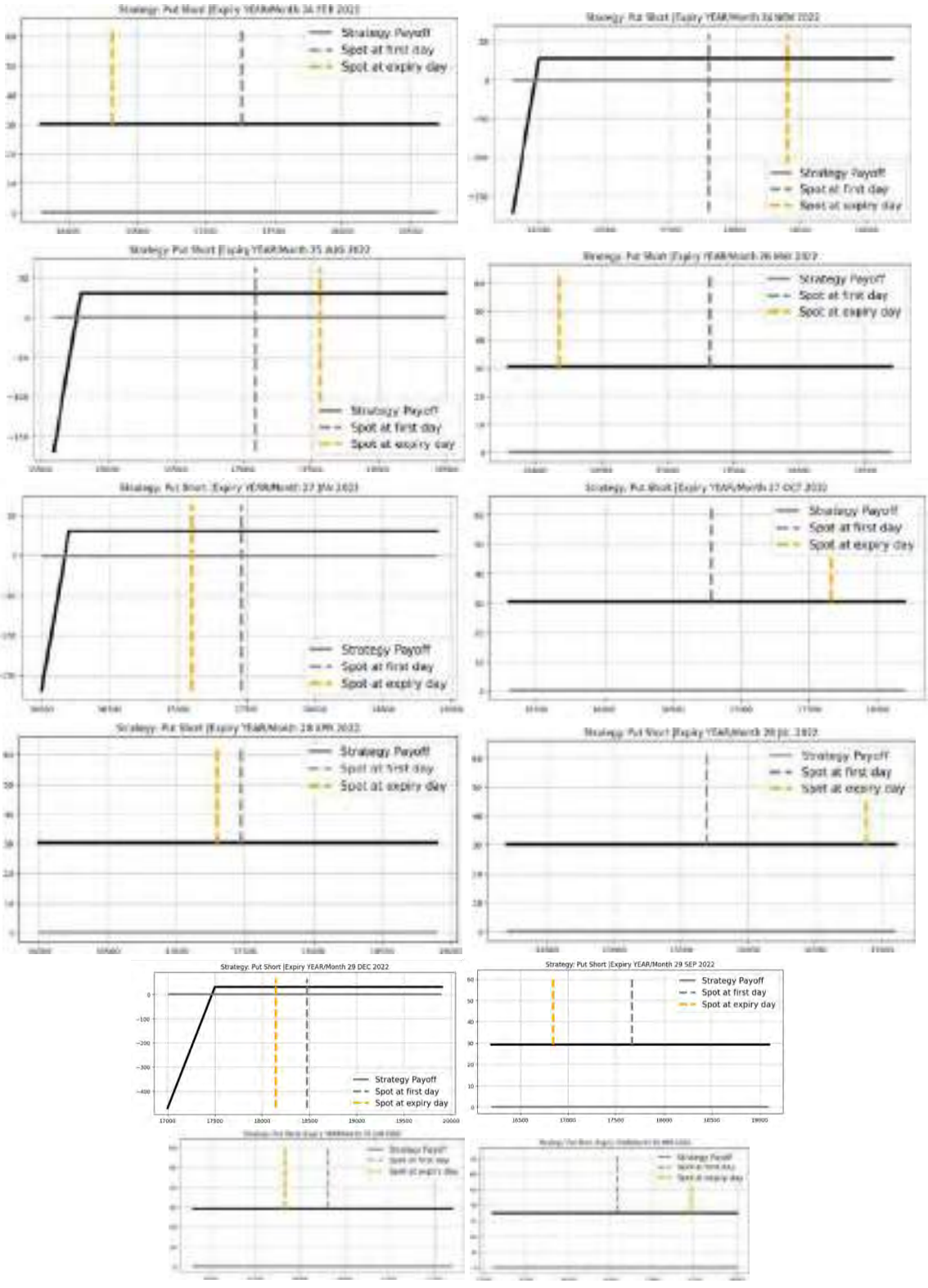




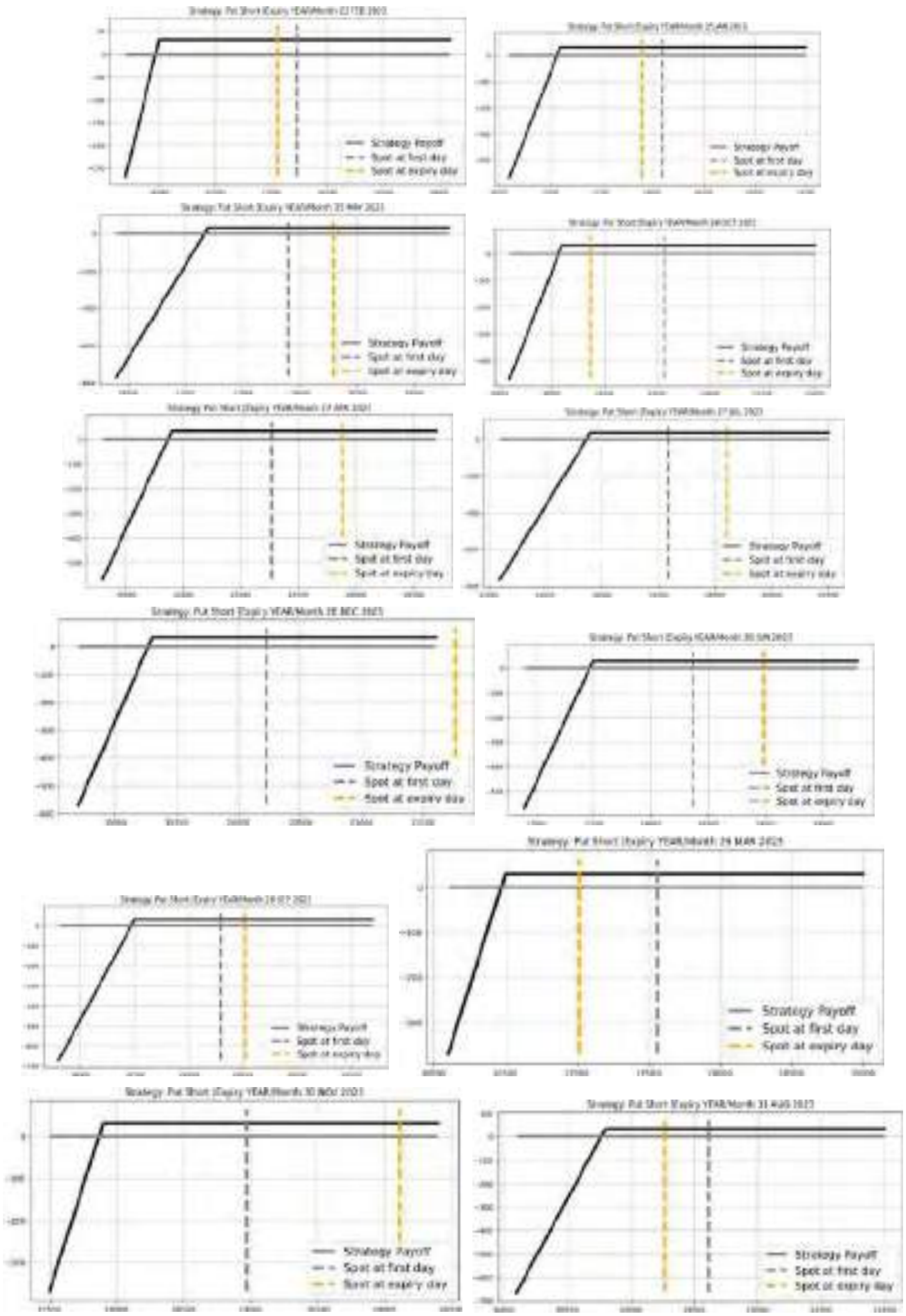
Strategy: Put Short Testing Year: 2021



Strategy: Put Short **Testing Year:** 2022



Strategy: Put Short **Testing Year:** 2023


Strategy Payoff
Spot at first day
Spot at expiry day

Strategy: Put Short **Testing Year:** 2024