Nottingham University Business School

UNITED KINGDOM · CHINA · MALAYSIA

Dynamic Option Hedging Strategy Via Reinforcement Learning

by

Author's Name: KinChiu Li

Year of Publication: 2025

A Dissertation presented in part consideration for the degree of "MSc Financial Technology".

Word Count: 14267

ABSTRACTS

This study investigates the application of a Reinforcement Learning (RL) model for dynamic option hedging of SPY, from 2018 to 2025. The model demonstrates strong capability in generating actionable hedging strategies, dynamically adjusting in both stable and volatile market conditions. With higher Sharpe ratio of 0.89 comparing with market return with 0.48. Showing the ability to increase total return while reducing risk taken. The reward function used is a key in the model, which successfully balancing the exploration and exploitation trade off. This report works as a foundation of future studies for creating an automative hedging strategy.

Keywords: Reinforcement Learning, Dynamic Trading Strategies, Option Hedging, Adaptive Learning, Market Regime Changes

ACKNOWLEDGEMENT

I would like to express my express my gratitude and appreciation to all those who have contributed to the successful of completion of this research.

First and foremost, I express my deepest appreciation to my tutor, Dr. Tren Ma, whose unwavering support, guidance, and expertise have been instrumental throughout this entire research period. Tren's profound knowledge in financial engineering, machine learning, and quantitative finance has been invaluable in shaping the direction and quality of this work. His reporting, encouragement, and guidance have been the key components in overcoming difficulties in the designing and analysis of the RL model as well as financial time series data.

I would like to acknowledge the online data platforms that supplied the detailed financial dataset used for this study. Their reliable and full-featured data services are instrumental to verify the practical applicability of the RL model, both across market conditions and through time.

I would also like to thank the open-source community, including all the members of GitHub and Stack Overflow whose publicly-accessible knowledge, code repos and technical documentation have been very helpful through this study. The synergism among these communities has been an invaluable resource to develop reinforcement learning software, statistical analysis tools and financial modeling software which are behind this work.

I would also like to acknowledge Jackson Khew, Sunny Sahdev, and Parth Abhijit Patil my peers and colleagues for their support during the research period. The novelty of their ideas, their constructive criticisms and their collaborational discussions were fundamental to the development of the direction of this work.

This research is the result of many contributions, and I will take all the responsibility for any remaining errors or omissions.

TABLE OF CONTENTS

1. INTRODUCTIONS

1.1. Theoretical Foundation and Research Context

The combination of machine learning (ML) and reinforcement learning (RL) in the financial markets marks a significant shift in how people approach quantitative trading and risk management. Theoretical models such as the Black-Scholes models have traditionally been relied upon in the hedging of options. While these models have interesting mathematical properties, they are not always able to fit the complex and non-lineal properties of real markets (Janková, 2018). The advent of advanced ML algorithms and RL frameworks opens up exciting possibilities for creating more adaptive and resilient hedging strategies, which can react to changing market conditions and continuously evolving by introducing the in real-time option chain trading simulation.

Recent advancements in deep learning and RL have revealed some truly remarkable possibilities in a variety of areas, from gaming to autonomous systems. In the financial sector, these technologies are particularly exciting for options hedging, where decision-making involves juggling between complex risk management, transaction costs, and return optimization trade-off. The capacity of RL agents to learn from past experiences and tweak from the existing strategies based on feedback makes them a great match for dynamic hedging applications (Jiang et al., 2017).

The motivation behind this research is based on three key challenges in the modern financial markets. Firstly, traditional hedging strategies often fall short during highly volatile market, which is where machine hedging can apply to. Secondly, the increase in complexity of financial instruments and market microstructure demands more sophisticated risk management approaches, and RL model can adapt to generating hedging combinations from finance derivatives. Lastly, the need for real-time decision-making in high-frequency trading environments calls for automated systems that can operate without human oversight (Marzban et al., 2023).

1.2. Research Aims and Investigative Framework

This research focuses on building a strong and integrated framework that merges ML with RL to tackle the challenges of dynamic options hedging in financial markets. The aim is to tap into the predictive strengths of ML models to enhance the adaptive decision-making processes used by RL agents. Therefore, the study hopes to improve how responsive and effective hedging strategies could be in different market conditions.

A major aspect of this investigation is the application of cutting-edge RL techniques, particularly those with feature advanced experience replay and strategic sampling methods. These enhancements are designed to improve the learning efficiency of the RL agent, allowing it to better handling of the complexities of financial environments. Moreover, the research presents a comprehensive academic reward function inspired by Jiang et al. (2017), which balances multiple goals such as risk-adjusted returns, transaction costs, and hedging effectiveness. This reward function is essential for guiding the agent's learning journey and decision-making process.

To assess the practical utility of the proposed framework, its performance will be evaluated against two benchmarks commonly use in algo trading and one for evaluating the efficiency of RL hedging, including Buy & Hold, Covered Call, and Buy & Hold with RL Hedging. This evaluation will be conducted with extensive real-world financial data to ensure the robustness and relevance of the findings. Besides quantitative performance metrics, this study also analyses the behavioural patterns of the RL agent to gain insights into how it learns and adapts its strategies in response to different market dynamics.

This research is designed by four of key questions. Firstly, it demonstrates how the ML predictions can be successfully incorporated to improve the performance of the RL agent in dynamic hedging. Secondly, the exploration to exploitation trades off inside the design of RL-based hedging strategies in training stage, which is crucial for evolving strategy. Thirdly, this paper also investigates how RL agent can react on real time option contract. Lastly, it attempts to determine the key underlying drivers of effective hedge strategies, especially in situations where market volatility is elevated.

1.3. Research Scope and Academic Contributions

This research employs a comprehensive and technical methodology that integrates both ML and RL techniques to construct a dynamic options hedging framework. The dataset included financial data from 2018 to 2025 of SPY included option chain data, daily stocks data, encompassing 1,813 trading days with complete market information. Feature engineering is applied to generate a 29-dimensional state space that captures key market dynamics, including price momentum, volatility indicators, and technical signals such as RSI, MACD, and Bollinger Bands, alongside features derived from ML predictions. The ML component incorporates four predictive models, LSTM, SVM, Random Forest, and Gradient Boosting, each tasked with forecasting price direction and volatility. The RL framework is built around a Proximal Policy Optimization (PPO) agent, designed to operate through various hedging combinations. The central to the agent's training is an academic reward function adapted from Jiang et al. (2017), which integrates multiple strategic incentives: a Sharpe ratio component with rolling volatility assessment, penalties for maximum drawdown, transaction costs and volatility exposure, as well as position-aware bonuses and risk management penalties. This multi-objective reward structure ensures that the agent learns to balance profitability, risk control, and cost efficiency in its hedging decisions.

This report departs from the traditional profit-maximization perspective by integrating real option contract data into a reinforcement learning framework and augmenting it with reinforcement learning to dynamically optimize hedging decisions. The use of actual market data ensures that pricing and hedge ratios reflect real trading conditions, while the RL component enables the strategy to adapt to evolving market states by learning from simulated interactions and adjusting positions to minimize portfolio risk, control transaction costs, and respond to non-linear market dynamics that static, where traditional Black Scholes model driven approaches often fail to capture.

1.4. Research Outcomes

The implementation and evaluation of the dynamic options hedging framework based on RL via SPY data in 2018-2025 has made significant progress in theoretical development and practical feasibility. The system integrates both RL and ML model to construct a robust, data-driven hedging strategy. While the results are promising, they must be interpreted with caution due to the inherent

limitations of backtesting methodologies and the complexity of real-world deployment.
A key innovation lies in the successful deployment of a reward function inspired by Jiang et al. (2017). This function incorporates multiple dimensions of trading behaviour, including position-aware hedging incentives, drawdown penalties for risk containment, transaction cost penalties to encourage cost-efficient execution, and volatility-based penalties to promote portfolio stability. This multi-faceted reward structure proved effective in stabilizing the agent's learning process across extended training episodes and contributed to the emergence of disciplined hedging behaviour.
The RL agent demonstrated a superior performance compared to the benchmark strategies in a comprehensive backtesting study. As presented in Session 4, the ML+RL agent achieved a total return of 559.09% (28.59% annualized), a Sharpe ratio of 0.89, Sortino ratio of 1.45, Calmar ratio of 1.92, volatility of 29.76%, and a maximum drawdown of -15.51%, resulting in a final portfolio value of $417,384.75. These results significantly outperform traditional strategies Buy & Hold by three times and Covered Call by four respectively. Despite these impressive figures, several limitations must be acknowledged. The reliance on historical data introduces the risk of overfitting, and although efforts were made to mitigate look-ahead bias, such as out of sample testing and time-series cross validation, subtle forms may still persist.
The ML component of the framework, designed to support the RL agent with predictive signals, exhibited strong performance. As shown in in the same session as above, the Gradient Boosting model emerged as the top performer with 66.57% accuracy, 72.04% recall, and 71.70% F1-score. The results indicate that the models provide valuable predictive signals that enhance the RL agent's decision-making process, particularly the Gradient Boosting model which shows the strongest overall performance across all metrics.
Feature importance analysis revealed a more balanced distribution compared to prior iterations, indicating improved model robustness. The top features included volume ratio (15.89%), RSI (15.62%), price reversal (11.79%), price momentum (10.80%), and MACD (8.50%). This more even spread of feature importance suggests effective feature engineering and reduced overfitting, although the concentration in a few dominant features may still limit generalisability in unseen market conditions.

The research achieved several key milestones. These range from the successful deployment of an academically grounded reward function, enriched hedging behaviour with position-aware incentives, statistically robust training through bootstrap confidence intervals and overfitting prevention, and technical novelties such as model persistence, performance visualisation, and robust error handling. Taken together, these successes evidence the practicality of the framework, although there are some significant cautions.

Despite these accomplishments, several limitations must be acknowledged. The strong performance may reflect overfitting to the specific historical data period, and the sophisticated nature of the system may hinder practical deployment, particularly in environments with limited computational resources. Furthermore, the black-box nature of the ML-RL architecture poses challenges for interpretability and regulatory acceptance, which are critical for institutional adoption.

1.5. Structural Framework

This dissertation is structured to provide a comprehensive examination of the ML-RL hedging framework. It begins with a literature review that establishes the theoretical background, covering hedging strategies, option theory, and the integration of machine learning and reinforcement learning in finance, as well as transaction costs and market constraints. The methodology chapter details the research design, data sources, feature engineering, model selection (including LSTM, XGBoost, and PPO-based RL), and the backtesting framework. The results and analysis chapter presents performance comparisons between RL-based and traditional strategies and evaluates risk and robustness. The discussion interprets findings, addresses limitations, and considers practical implications for institutional use. The conclusion summarizes key contributions, highlights the novelty of RL integration and cost modelling, and reflects on the research's academic and practical significance. This concise structure ensures clarity, academic rigor, and relevance for both academic and industry reference.

2. LITERATURE REVIEW AND THEORETICAL SYNTHESIS

2.1. Theoretical Foundations of Options Hedging and Risk Management

Hedging is a fundamental risk management strategy commonly used by institutional investors, hedge funds, and portfolio managers to mitigate adverse price movements in financial assets. Option as derivative instruments, provide asymmetric payoff structures that make them particularly valuable for hedging applications. A put option can limit downside risk while preserving upside potential, making it an ideal tool for protecting long equity positions. Conversely, call options can be utilized to hedge short positions or generate income through strategies such as Covered Calls (Chen and Li, 2021).

The flexibility of options in strike and position, allows investors to construct complex hedging strategies tailored for different risk profiles and market conditions. In the context of algo trading systems, hedging becomes increasingly important. Though ML models may be able to uncover patterns in the historical record and predict price movements over short time horizons, they are probabilistic by nature. There is not a model that can assure 100% accuracy and be able to capture infallible signs, especially when market noise, regime shifts, or black swan effects arise as consequence of an external factor (Marzban et al., 2023).

If only ML predictions are used as inputs without any hedging mechanisms, would exposes portfolios to significant risk of drawdowns when predictions fail. For example, a model might predict a bullish trend with high confidence, but an unexpected macroeconomic event could trigger a sharp market reversal. In these scenarios, a well-designed dynamic hedging strategy which informed by both predictive signals and real-time market data can provide a backup, mitigate losses and preserve capital (Chen and Li, 2021).

Recent research has provided that data-driven hedging models, in particular those based on deep learning techniques, can benefit from better or more accurate approximations to optimal hedge ratios learning them directly from market data as opposed to optimization models based on option pricing theory (Chen and Li, 2021). In addition, RL frameworks have demonstrated the ability to obtain adaptive hedging policies that are more efficient in incomplete or risky markets (Marzban et al., 2023).

However, it is important to acknowledge that while these approaches show promising return, they also introduce new challenges. The complexity of ML models can make increase the difficulty to interpret and validate, while the adaptive nature of RL agents may lead to unexpected behaviour in changing market conditions. Additionally, there is still a common question on sample bias, as using historical data to train models is likely to miss unknown future market regimes or structural changes.

Therefore, in order to retain profit, options-based hedging in the ML-based trading models becoming a compulsory protection mechanism. It makes the system robust to uncertainty, is consistent with risk adjusted performance objectives and is a reflection of the realistic nature of financial markets that prediction errors are unavoidable.

2.2. Comparative Analysis of Static and Dynamic Hedging Methodologies

In the financial markets there are two generally applicable types of hedging strategies, namely the static and dynamic strategies. The key is how often the hedge is reset based on changes in the markets.

Static hedging is mean by constructing a hedge at the start of a position and until maturity. This approach typically uses a combination of options or other derivatives to replicate the payoff of the underlying exposure. Static hedges are attractive to investors due to their simplicity, lower transaction costs, and reduced operational complexity. For example, a protective put, or a covered call strategy can be implemented once and left untouched or rolling for extend of contract, offering a predefined risk-reward profile. Research by Carr et al. (1998) demonstrated that static replication of exotic options using vanilla instruments can be effective under certain market assumptions.

However, static hedging has its own constrains, particularly in volatile or path-dependent markets. Since the hedge is not adjusted when change in macroeconomics, it may fail to respond to significant changes in the underlying asset's price, volatility, or market conditions. This can lead to hedging inefficiencies, especially when the market deviates from the assumptions when static hedge was constructed.

On the other hand, dynamic hedging involves continuously or periodically adjusting the hedge in response to changes in market. A well-known practice is delta hedging, where the hedge ratio is recalculated as the delta of the option changes. Dynamic hedging is more adaptive and provide better protection in rapidly changing markets. However, it comes with higher transaction costs and robust sophisticated infrastructure and real-time data processing.

Studies from Engelmann et al. (2006) have shown that while dynamic hedging can provide superior risk mitigation, it also introduces greater variability in profit-and-loss (PnL) distributions due to frequent rebalancing and exposure to model risk. found that certain static hedges could outperform dynamic ones in terms of robustness and stability under changing market conditions.

In the context of ML-driven trading systems, dynamic hedging is especially crucial. Since ML models generate predictions that evolve over time, a static hedge will quickly become misaligned with the model's latest prediction. A dynamic hedging framework, especially one guided by reinforcement learning, can adapt to change quickly and optimize the hedge in real time. This adaptability is important when dealing with the non-stationarity and uncertainty inherent in financial markets.

Therefore, even static hedging is a simpler, more cost-efficient model, the flexibility provides from dynamic hedging making it a better natural fit for ML-based trading systems.

2.3. Machine Learning Applications in Financial Time Series Forecasting

ML has turned out to become a disruptive tool in financial prediction, given its potentiality to discover intricate nonlinear factors hidden in big data of market. In comparison to traditional econometric models, which often rely on predefined assumptions and fixing with linear relationships, ML models learn from data in an adaptive manner and are well-suited to the dynamic and noisy environment in financial markets (Wasserbacher and Spindler, 2022).

In the context of stock price and volatility forecasting, a wide range of ML algorithms have been applied, including supervised learning models such as Random Forests, Support Vector Machines (SVM), and Gradient Boosting, as well as deep learning architectures like Long Short-Term Memory (LSTM) networks. These frameworks have shown compelling results in modelling temporal dependencies and analysing the nonlinear interactions in time-series data.

For example, LSTM networks are particularly effective working with sequential data due to their ability to retain long-term dependencies(Rojas *et al.*, 2022), making them ideal for forecasting asset prices and volatility over short to medium horizons. Studies from Mehtarizadeh *et al.* (2025) and Gifty and Li (2024) simultaneously indicate that LSTM-based models can outperform traditional ARIMA and GARCH models in terms of predictive accuracy and responsiveness to market shifts.

However, while ML models can generate high-confidence predictions, are inherently probabilistic and subject to model risk. This is especially true in financial markets, where non-stationarity, regime changes, and external shocks (e.g., wars, macroeconomic announcements) can rapidly invalidate learned patterns. As Wasserbacher and Spindler (2022) emphasize, the naive application of ML in financial forecasting can lead to overfitting and poor generalization if not carefully validated.

In addition, ML models are typically optimized for forecasting accuracy than decision-making under uncertainty (Barbierato and Gatti, 2024). In trading and hedging applications, the difference is crucial and the loss from miscalculation could be substantial. Therefore, ML predictions need to be incorporated with a more comprehensive decision model, like RL agents or rule-based system, that can convert the predictions into action plans considering the risk, cost, and the market.

It is also important to acknowledge that the performance of ML models in financial forecasting is often overstated in the literatures. Which most of them focus on in-sample performance or use limited out-of-sample testing periods, which may not provide a realistic assessment of model performance in live trading environments. Additionally, ML models are often regarded as black-box models which are hard to interpret and validate, thus their practical use in regulated financial environment is questionable.

In summary, despite ML offers strong financial forecast tools, its outputs should be considered in comprehensive risk management systems. This is particularly important in algo trading, where prediction mistakes can result in huge monetary loss if not managed correctly. The integration of ML with dynamic hedging frameworks represents a promising direction for building more adaptive and resilient trading systems, though with important caveats regarding model risk and interpretability.

2.4. Hybrid Machine Learning and Reinforcement Learning Architectures in Financial Applications

Recent advances in deep RL for hedging move beyond classical delta adjustment to unified forecasting-execution pipelines. Ding et al. (2025) introduce a behaviour-cloning-enhanced PPO agent (BC-RPPO) that is pretrained on expert demonstrations before fine-tuning with sparse hedging rewards. By embedding a spatiotemporal attention-based Transformer for probabilistic market forecasting, this framework achieves superior hedge effectiveness in both U.S. and Chinese equity markets compared to purely model-based or RL-only approaches.

Agent-based market simulators have emerged as powerful tools to generate realistic training data for deep hedging. Gao et al. (2023) propose the Chiarella-Heston model: a calibrated ABM incorporating momentum, fundamental, and volatility traders, to produce synthetic asset paths that closely replicate empirical stylised facts. When used to train a Deep Deterministic Policy Gradient based hedging agent, this simulator yields strategies that outperform baseline deep-hedging models across varying transaction-cost regimes.

Beyond single-asset options, RL in continuous action spaces now supports dynamic generation of full hedging strategies—optimising strikes, expiries and sizes rather than pure delta. Du et al. (2020) demonstrate that Proximal Policy Optimization (PPO) and Deep Q-Learning can learn replication policies over a range of strikes with discrete trading and nonlinear costs, matching or surpassing traditional Black Scholes Model based hedges in backtests. Similarly, Quintero et al. (2025) compare Advantage Actor-Critic and PPO in a market-neutral pair-trading setup, achieving positive Sharpe and Sortino ratios over standard distance-based benchmarks.

However, there are several inherent limitations to existing RL works in finance that need to be recognized. First, the majority of works is either based on oversimplified market models or synthetic data and does not take into account the full complexity of real markets. Second, measures of performance often stress results from back testing without duly accounting for trading costs, market impact and regulatory restrictions. Third, the interpretability of RL agents is one of the current challenges and we need to understand and verify their decision processes. Despite PnL from model, practical deployment still grapples with sample inefficiency, environment fidelity and policy interpretability. Ongoing research explores off-policy techniques, domain randomisation and transfer learning to bridge the gap between academic prototypes and live trading systems.

2.5. Integration of Machine Learning and Reinforcement Learning in Finance

Recent research suggested that embedding predictive features directly into decision frameworks can enhance the robustness of hedging policies. Barbierato and Gatti (2024) argue that the naive separation of forecasting and strategy formulation often leads to suboptimal outcomes; integrating LSTM or XGBoost derived point forecasts into the RL state space significantly improves policy stability under uncertain market conditions.

It is further hypothesised that short-term volatility predictions that feed into an actor–critic agent enable adaptive hedge-ratio adjustments that respond to evolving risk signals. Chen and Li (2021) demonstrate a prototype framework in which ML-generated volatility forecasts reduce 95th-percentile drawdowns by 12% compared to RL agents without ML-augmented states

One major advantage of the hybrid strategy outlined by Murray et al. (2022) and Marzban et al. (2023) is integrated forecasting and hedging in single RL framework. Which the actor decides strikes-offset, expiries, and position sizes, while the critic casts the reward in terms of the difference between implied volatility surface and price trends and their estimates for the future. This joint design enables up to 15% improvement for tail-risk metrics compared to pure RL or pure ML ones.

However, it is also worth mentioning that the combination of ML and RL bears higher complexity and additional error-prone aspects. Collaboration of multiple models and algorithms may could cause challenging in debugging and validating, and interaction between components may result in unexpected behaviour. Also, the amount of computation involved by learning and applying the proposed integrated systems would be too large for many practical cases.

2.6. Transaction Cost Modelling and Real-World Implementation Constraints

The present section seeks to examine realistic transaction-cost modelling within RL environments. Bellora et al. (2021) propose a continuous, exponentially decreasing cost function that captures the non-linear relationship between trade size and per-unit costs. This formulation ensures non-negativity, continuity, and a more accurate reflection of market impact for large orders.

It is suggested that mis-specifying cost parameters can lead RL agents to over-trade during simulations, yielding inflated backtest performance but disappointing live results. Bellora et al. report that incorporating their cost function into RL training reduces simulated turnover by up to 30%, aligning simulation behaviour more closely with observed market impact profiles.

Real-world deployment must account for latency, slippage, and data delays, which is part of transaction costs. Empirical studies from Engelmann et al. (2006) and Carr et al. (1998) suggest that the exclusion of these frictions may underestimate hedging costs by up to 20% in high-volatility states. The integration of slippage models and order-book dynamics into RL simulators remains an outstanding challenge, as this will facilitate the deployment of learned policies to production systems.

2.7. Literature Synthesis and Research Gap

2.7.1. Summary

Hedging reduces adverse price-movement risk by using derivatives, with options offering asymmetric payoffs that protect downside while preserving upside potential. Static hedges (e.g., protective puts, covered calls) are simple and low-cost but unresponsive to market shifts, whereas dynamic hedging (e.g., delta hedging) adapts to evolving prices and volatilities but incurs higher transaction costs and model-risk exposure. In financial forecasting, supervised and deep learning models (Random Forest, XGBoost, LSTM) capture nonlinear dependencies in price and volatility time series, outperforming traditional ARIMA/GARCH in predictive accuracy, yet their probabilistic nature and overfitting risk necessitate integration into broader decision frameworks. Reinforcement learning (RL) formulates hedging as a sequential decision problem under uncertainty, with policy-based methods (Actor-Critic, PPO) offering stable training in continuous action spaces; deep hedging studies report up to 15% tail-risk reduction versus classical delta hedging but face challenges around data fidelity, non-stationarity, and interpretability. Hybrid frameworks that embed ML forecasts directly into the RL state yield more robust hedging policies, with empirical results showing up to 12% drawdown reduction and 15% tail-risk improvement over standalone ML or RL models. Realistic cost modelling using continuous, exponentially decreasing functions aligns simulations with market impact, reducing over-trading in backtests by up to 30%, while omitting slippage and latency can understate hedging costs by 20% or more.

2.7.2. Research Gap

Modular Separation Forecasting and hedging are often treated as distinct modules, preventing unified optimization of prediction accuracy and execution quality. The majority of researchers focus on developing forecasting models and reporting their accuracy, while acknowledging potential losses that could arise from model imperfections, rather than providing a hedging layer on top of the forecasts. This research takes a different approach by first constructing a forecasting model and then applying a hedging strategy directly to the resulting portfolio positions, enabling an integrated evaluation of both predictive performance and risk mitigation.

Delta hedging typically adjust only the delta of a position, limiting their ability to capture the full spectrum of strategic choices available in options markets. Such narrow action spaces overlook the potential benefits of jointly optimizing strike selection, expiry choice, and position sizing within a continuous action framework. Expanding the action space allows for richer, more adaptive strategies that can respond to complex market conditions beyond simple delta adjustments.

There is little work on parameterized strategy-generation models that output complete hedging configurations, reinforced by reward functions encompassing multiple performance dimensions such as PnL, Sharpe ratio, drawdown penalties, and volatility reduction. Without such generative models, hedging remains reactive and piecemeal, rather than proactively designed to balance profitability, stability, and capital efficiency in a unified objective.

This dissertation addresses these gaps by developing a unified ML–RL hedging framework that integrates predictive forecasts into a generative action model, employs a continuous transaction-cost function to reflect realistic market frictions, and validates performance using comprehensive option-chain and high-frequency market data. By combining accurate forecasting, expanded action spaces, and multi-objective reward design, the proposed approach aims to advance the state-of-the-art in dynamic options hedging and deliver strategies that are both theoretically robust and practically deployable.

3. RESEARCH METHODOLOGY

3.1. Research Design and Methodological Framework

This research employs a quantitative, simulation-based approach to evaluate the effectiveness of ML-RL frameworks for dynamic options hedging. The methodology combines empirical data analysis with computational modelling to assess the performance of integrated ML-RL hedging strategies against traditional benchmarks.

This research design follows a systematic approach to address the research gaps from the literature review. The methodology integrates predictive ML models with adaptive RL agents in a joined structure that account for transaction costs, selective trading behaviour, and comprehensive statistical validation.

The quantitative approach evaluates the strategy performs across multiple market regimes and provides statistical confidence as a result. The simulation-based methodology allows for controlled experimentation while incorporating realistic market constraints and transaction costs that reflect actual trading conditions.

Table 1- Research Design Overview

This table summarizes the key stages and descriptions of the research methodology implemented in this study.

Stage	Description
Data Preprocessing	Preprocessing option chain data, feature engineering
ML training	LSTM for price direction, XGBoost for volatility
RL environment	Position-aware hedging environment development
RL training	PPO agent training with ML-enhanced state spaces
Evaluation	Benchmark comparison and performance analysis
Validation	Robustness testing and sensitivity analysis

However, a few limitations that must be taken into account of in the design. The reliance on historical data for both training and evaluation introduces the risk of overfitting, despite out of sample data and time-series cross validation had been applied. Additionally, the simulation-based method can be limited in modelling the complexity of actual market dynamics, such as market impact, liquidity constraints, regulation issue. The selection of particular ML models and RL methods can also create bias since alternative tools can lead to divergent findings.

3.2. Data Collection and Preprocessing

The research utilizes comprehensive SPY (S&P 500 ETF) option chain data and underlying asset data spanning multiple years (2018-2025) to ensure all-rounded evaluation across diverse market conditions. The dataset includes detailed option contract information (option chain), market data (Risk free rate, gok volatility), and underlying asset price movements (OHLCV).

Table 2- Data Sources and Coverage

This table summarizes the categories and details of the data sources used in this research.

Data Category	**Details**
Contract Specifications	Strike price, expiration date, option type
Market Data	Bid/ask prices, volume, open interest
Greeks Calculations	Delta, gamma, theta, vega
Underlying Asset Data	OHLCV (Open, High, Low, Close, Volume) data
Volatility Index	VIX index data

Table 3- Data Quality Filters

This table outlines the data quality filters and their criteria applied to ensure robust and reliable data for analysis.

Data Quality Filters

Filter Type	Criteria
Volume Threshold	Minimum volume > 5 contracts
Open Interest Threshold	Minimum open interest > 20 contracts
Bid-Ask Spread Validity	Inclusion of only options with valid bid-ask spreads
Greeks Completeness	Inclusion of only options with complete Greeks calculations

The dataset is structured by year to allow time-based training and validation, containing about 252 files representing the daily option chain snapshots for that year, for a total of 1,813 trading days of full data. The range allows for thorough model training and validation across a variety of market conditions.

However, it is important for taking different limitations into account. The results in this paper are limited to the SPY options and might not be extended to other asset classes or markets. The screening process, although this is a condition that is required in order to filter for a reliable and liquid option, in and of itself can create a selection bias by weeding out specific market situations or option contracts. In addition, the use of pre-processed data may be hiding valuable market microstructure information that has an impact on the efficiency of the hedging.

3.3. Feature Engineering and Dimensionality Analysis

The feature engineering process creates a 60-dimensional state space combining market, portfolio, and ML prediction features. Feature calculations are performed separately for different data sources to ensure accuracy and maintainability:

Table 4- Feature Engineering Categories and Dimensions

Breakdown of the 60-dimensional state space showing feature categories, details, and dimensions used in the ML-RL framework.

Feature Category	Details	Number of features
Market Features	OHLCV data, VIX, historical volatility, option chain statistics, volume, open interest, bid-ask spreads, strike distribution ratios	27
Portfolio Features	Cash ratio, stock position ratio, number of option positions, total contracts, normalized delta, gamma, vega	8
Historical Features	Recent portfolios return statistics (mean, std, min, max)	4
ML Prediction Features	LSTM price direction probabilities, Gradient Boosting volatility predictions, model confidence metrics, prediction-based triggers	21

Feature preprocessing includes:

Table 5- Feature Preprocessing Steps and Descriptions

This table summarizes the key preprocessing steps applied to the feature set, ensuring data quality and stability for model training.

Preprocessing Step	Description
Normalization	All numerical features scaled to [0,1] range for stability
Missing Value Handling	Zero padding applied to missing entries
Type Conversion	String-based numerical data converted to appropriate numeric types

Rolling Window Calculations	Applied to historical features for capturing temporal dynamics

The feature engineering process, while comprehensive, may introduce several potential issues. The normalization range assumes that the feature distributions remain stable over time, which may not hold in changing market conditions. The handling of missing values with zero padding may introduce bias, while the choice of specific features may reflect the researcher's prior beliefs about what constitutes relevant market information. Additionally, the high dimensionality of the state space (29 features) may increase the risk of overfitting and make the model more difficult to interpret.

3.4. Data Source and Quality Assurance

This research utilizes a multi-source data approach to ensure comprehensive market coverage and data quality. The primary data sources are:

- Alpha Vantage API - For comprehensive options chain data including implied volatility, Greeks, bid-ask spreads, volume, and open interest
- Twelve Data API - For high-quality OHLCV data
- Fama French Data - For risk-free rate collection and factor model calculations
- TA-Lib Library - For technical indicator calculations from OHLCV data

Alpha Vantage Options Data Collection:

Alpha Vantage provides comprehensive options chain data through their Options Chain API, offering detailed information for SPY options including:

- Implied Volatility Data: Complete IV surfaces across different strikes and expirations
- Greeks Calculations: Delta, Gamma, Theta, Vega, and Rho for all option contracts
- Market Microstructure: Bid-ask spreads, volume, open interest, and liquidity metrics
- Strike Price Distribution: Complete coverage of available strike prices and expiration dates

API Implementation and Rate Limiting:

- Authentication: API key-based authentication with secure credential management
- Rate Limiting: Implementation of proper rate limiting (5 calls/minute free tier, 500 calls/minute premium)
- Error Handling: Robust error handling and retry mechanisms for API failures
- Data Validation: Real-time validation of data quality and completeness

Twelve Data OHLCV Collection:

Twelve Data provides high-quality historical price data:

- OHLCV Data: Daily Open, High, Low, Close, and Volume data for SPY
- Data Quality: Stable daily data covering the entire testing range, which cross reference from yfinance (yahoo finance).
- API Reliability: 99.9% uptime guarantee with redundant data centres

TA-Lib Technical Indicator Calculations:

Technical indicators are calculated separately using the TA-Lib library from the OHLCV data:

- Momentum Indicators: RSI, MACD, Stochastic Oscillator, Williams %R
- Trend Indicators: Moving Averages (SMA, EMA, WMA, DEMA, TEMA), ADX, Parabolic SAR
- Volatility Indicators: Bollinger Bands, Average True Range (ATR), Standard Deviation
- Volume Indicators: On-Balance Volume (OBV), Volume Rate of Change, Money Flow Index
- Custom Calculations: Rolling statistics, price momentum, volume ratios, and market microstructure metrics

Fama French Risk-Free Rate Collection:

Fama French data is utilized for risk-free rate calculations and factor model analysis:

- Risk-Free Rate: Daily Treasury bill rates for risk-adjusted return calculations
- Factor Models: Market, size, and value factors for performance attribution
- Data Source: Kenneth French Data Library with daily frequency
- Quality Assurance: Academic-grade data with comprehensive coverage and validation

Temporal Coverage and Data Period:

The dataset spans from January 2, 2018, to June 30, 2025, providing comprehensive coverage of:

- 1,813 trading days of complete market data
- Multiple market regimes including bull markets, bear markets, and high volatility periods
- Major market events including the COVID-19 pandemic, inflation concerns, and geopolitical tensions (trade war in 2025).
- Complete options chain data for SPY with multiple expiration cycles

Data Screening and Quality Control:

The data screening process implements multiple layers of quality control to ensure data integrity and reliability (table 4):

Missing Data Detection: Automated scripts identify and flag missing data points, gaps in trading days, and anomalous values. Missing data points are handled through forward-filling for short gaps and interpolation for longer periods.

Outlier Detection: Statistical methods including Z-score analysis and interquartile range (IQR) methods are employed to identify and handle outliers. Outliers are defined as data points that deviate more than 3 standard deviations from the mean or fall outside the 1.5 × IQR range.

Cross-Source Validation: Data consistency is verified across all data sources (Alpha Vantage options data, Twelve Data OHLCV data, Fama French risk-free rates, and TA-Lib calculated indicators), ensuring temporal alignment and price consistency across sources.

Data Preprocessing Pipeline

The preprocessing pipeline implements several steps to prepare the data for ML models:

The preprocessing pipeline comprises several stages to prepare the dataset for machine learning models. Feature engineering generates derived variables, including lagged returns, rolling statistics, and interaction terms between related indicators (Appendix A). Stationarity testing is conducted using the Augmented Dickey–Fuller test, with differencing applied where necessary (Appendix G). Results indicate that the majority of technical indicators derived from option chain data are stationary; returns exhibit 100% stationarity across the seven-year sample, while implied volatility and volume-related variables demonstrate over 80% stationarity across different years. A detailed year-by-year ranking of variable stationarity is provided in Appendix H. Although the selected model is inherently robust to heteroskedasticity and unit roots allowing learning patterns directly from the data without prior distributional assumptions, statistical testing remains valuable for deepening understanding and guiding feature generation. Correlation analysis is performed using Pearson coefficients to identify highly correlated features (threshold > 0.95) for potential removal, thereby mitigating multicollinearity. Finally, temporal alignment ensures that all indicators share a consistent time index, preserving synchronicity across the dataset.

The resulting dataset comprises 1,813 trading days with over 50 engineered features, providing a comprehensive foundation for the ML and RL models employed in this research. Detailed technical indicator calculations and parameter specifications are provided in Appendix A.

3.5. Two-Stage Algorithmic Framework: Selection and Implementation

This research involves two distinct stages of prediction, each with carefully selected algorithms optimized for their specific tasks

3.5.1. Stage I – Price-Movement Forecasting

The modelling framework combines Long Short-Term Memory (LSTM) networks and XGBoost to leverage their complementary strengths in financial forecasting. The LSTM component is designed to capture sequential dependencies in price and technical indicator series, incorporating features such as moving averages, RSI, Bollinger Bands, and other time-series indicators within a sequential neural network architecture optimized for temporal pattern recognition, excelling at modelling long-term dependencies and temporal relationships. In parallel, the XGBoost (gradient boosting) component models nonlinear interactions among option-specific features, including moneyness, time to expiry, implied volatility, and other contract characteristics, within a robust ensemble learning framework that offers superior handling of complex feature interactions and non-linear relationships. The training configuration employs a 70/30 train−validation split for robust model evaluation, with 5-fold cross-validation for comprehensive hyperparameter tuning. Model optimization is guided by accuracy and F1-score metrics for directional forecasts, and hyperparameters — including the number of LSTM layers, tree depth, learning rate, and other critical parameters — are selected via grid search to ensure optimal predictive performance.

3.5.2. Stage II – Reinforcement Learning Hedging

The Stage I forecasts feed into a continuous-action RL agent that learns dynamic hedging policies:

The reinforcement learning framework employs Proximal Policy Optimization (PPO), selected for its stability in non-stationary, high-dimensional financial environments. PPO's conservative policy updates and trust-region constraints provide robustness in volatile market conditions, enabling consistent performance in environments characterized by high noise and shifting dynamics. The agent architecture consists of a three-layer Multi-Layer Perceptron (MLP) with ReLU activation functions for computational efficiency, operating over a 29-dimensional continuous state space that integrates market data, portfolio state variables, and machine learning-based forecasts. The action space comprises 18 discrete hedging actions, including combination strategies, allowing for flexible and adaptive risk management. An experience buffer of 50,000 samples, with strategic prioritization, supports efficient learning, while each update step draws on 200 sampled experiences to balance exploration and exploitation in policy optimization.

Model Integration: The ML predictions are integrated into the RL state space through direct feature inclusion of prediction probabilities, derived features including confidence metrics and prediction-based triggers, and temporal alignment with market data. This integration enables the RL agent to leverage both directional and magnitude predictions for optimal hedging decisions.

However, it is important to acknowledge several limitations in the two-stage implementation. The choice of specific architectures and hyperparameters may not be optimal for all market conditions, and the models may be sensitive to the particular training data used. The integration of ML predictions into the RL state space assumes that the predictions are reliable and relevant, which may not always be the case. Additionally, the two-stage approach may introduce complexity and potential propagation of errors from the first stage to the second stage.

3.6. Machine Learning Model Architecture and Implementation

All Training configuration includes categorical cross-entropy loss function, Adam optimizer with learning rate 0.001, batch size of 32, 70/30 validation split, and early stopping with patience of 10. The model incorporates technical indicators including SMA, EMA, RSI, and Bollinger Bands, along with price momentum and volatility indicators, and option market sentiment features.

All models been use will be classify in the following:

3.6.1. LSTM Neural Network Architecture

The LSTM model implements a sophisticated architecture designed for sequential financial data. Which is also one of the most complicated models in this report due to the nature of handling sequential data and different gate for controlling weight of data.

The architecture includes input layers with 50-time steps of historical features, two LSTM layers with 128 and 64 units respectively, dropout layers with 0.2 dropout rate for regularization, and an output layer with softmax activation for probability distribution. The model equations and principles will be explained in detailed in the appendix F.

3.6.2. Gradient Boosting (XGboosting)

The XGBoost model is chosen due to his superior performance with handling nonlinear relations. The model configuration includes 100 estimators, maximum depth of 6, learning rate of 0.1, subsample ratio of 0.8, and column sample by tree ratio of 0.8. The objective function is regression with squared error loss.

XGBoost is an optimized gradient boosting implementation that builds an ensemble of weak learners (decision trees) sequentially. Research by Chen and Guestrin (2016) introduced XGBoost as a scalable end-to-end tree boosting system that has become the standard for gradient boosting implementations.

Feature engineering for this prediction includes option-specific features such as moneyness and time to expiry, market microstructure features and Greeks-based features (see appendix A). Which Gradient Boosting also showing the highest accuracy and F-1 score from the testing. Further Gradient Boosting explanations can be found in Appendix E.

3.6.3. Support Vector Machine (SVM)

SVM finds an optimal hyperplane that maximizes the margin between classes. The foundational work by Cortes and Vapnik (1995) established SVM as a powerful classification algorithm that has been extensively applied in financial forecasting. More recent research by demonstrated SVM's effectiveness in stock price prediction with improved accuracy over traditional statistical methods(Huang *et al.*, 2005). However, throughout the testing, SVM do not showing extremely high accuracy but acceptable of 62% in samples and 58% out of sample. Further explanations of fundamental equations can be found in Appendix F.

3.6.4. Random Forest

Random Forest combines multiple decision trees through bagging and feature randomization. The seminal work by Breiman (2001) introduced Random Forest as an ensemble method that improves prediction accuracy through bootstrap aggregation and feature randomization. Studies by Genuer et al. (2010) have shown Random Forest's robustness in handling high-dimensional financial data with minimal overfitting. The fundamental equations can be found in Appendix D.

3.7. Reinforcement Learning Framework and Agent Architecture

3.7.1. Proximal Policy Optimization (PPO) Fundamentals

Core PPO Equations and Concepts:

PPO is a policy gradient method that constrains policy updates to prevent large deviations:

Policy Objective:

$$\mathrm{L}(\theta) = \mathrm{E}[min(r_t(\theta)A_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t)] \tag{24}$$

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_o}ld(a_t|s_t)} \tag{25}$$

$r_t(\theta)$ is the probability ratio,

A_t is the advantage function,

ε is the clipping parameter (typically using 0.2).

Advantage Function:

$$\begin{gathered} A_t = \delta_t + \gamma\lambda\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \\ \delta_t = r_t + \gamma V(s_{t+1}) - \mathrm{V}(s_t) \end{gathered} \tag{26}$$

$\delta_t$is the TD error, γ is the discount factor, λ is the GAE parameter.

Value Function Loss:

$$L_V(\theta) = \mathrm{E}[(V_\theta(s_t) - V_t arget)^2] \tag{27}$$

Entropy Bonus:

$$L_{entropy}(\theta) = \mathrm{E}\left[\sum_a \pi_\theta(a|s_t) log \pi_\theta(a|s_t)\right] \tag{28}$$

Total Loss:

$$L_{total} = \mathrm{L}(\theta) + c_1 L_V(\theta) + c_2 L_{entropy}(\theta) \tag{29}$$

c_1 and c_2 are hyperparameters controlling the relative importance of each component.

3.7.2. State Space Design and Representation

At time t, the RL agent will encode both market and portfolio information. The Key component included:

- Portfolio position and Cash Balance.
- ML- driven Forecasts: which came with a predicted short term price direction and volatilities from LSTM and Gradient Boosting model(Ding *et al.*, 2025).
- Option Greeks: such as Delta, Gamma, Vega, Theta, for the available option contract.(Murray *et al.*, 2022)
- Technical Indicator: MA (20, 50, 200) (including SMA, EMA and WMA). RSI, MA cross and Bollinger Bands. Which capture the momentum and mean reversion signals (Du *et al.*, 2020)

State Representation:

$$s_t = [p_t, c_t, f_t, g_t, i_t] \tag{30}$$

Where p_t represents portfolio state, c_t is cash balance, f_t are ML forecasts, g_t are option Greeks, and i_t are technical indicators.

3.7.3. Action Space Design and Hedging Strategy Formulation

Action Space:

$$A = a_i \mid a_i = (strikeoffset, expirytype, positionsize) \tag{31}$$

Define Hedging adjustment over a continuous, multidimensional space:

- Strike Offset: determined the percentage deviation from the stock price, this is a variable depends on the option availabilities.

- Expiry selection: to choose over short-term, medium- and long-term options.
- Position size: number of contracts to buy or sell. could bring back to position 0 for not hedging, or positive for long and negative for short.

The generative action model extends beyond traditional Delta hedging strategies, allows the agent to tailor option strategies in response to market forecasting.

3.7.4. Time Series Cross-Validation Methodology

Fundamental Concepts and Implementation:

Time series cross-validation is essential for financial data to prevent look-ahead bias and ensure realistic performance estimation. The methodology implemented in this research follows a rolling window approach:

A diagram in the Appendix B shown the structure of both rolling window and expanding window cross-validation for time series data,

Rolling Window Cross-Validation:

$$Train_i = [t_0, t_i - w] \tag{32}$$

$$Val_i = [t_i - w, t_i] \tag{33}$$

$$Test_i = [t_i, t_i + h] \tag{34}$$

Where w is the window size, h is the forecast horizon, and i iterates through time.

Expanding Window Cross-Validation:

$$Train_i = [t_0, t_i] \tag{35}$$

$$Val_i = [t_i, t_i + v] \tag{36}$$

$$Test_i = [t_i + v, t_i + v + h] \tag{37}$$

Where v is the validation period length.

Implementation Details:

Data Partitioning: The dataset is divided into training (80%), validation (10%), and testing (10%) periods, strictly preserving temporal order, as implemented in the RL system.

Rolling Window Size: A 504-day (two trading years) rolling window is used for training, with 126-day (half-year) validation periods

Forecast Horizon: 63-day (one quarter) forecast horizon is used for out-of-sample testing.

Performance Metrics: Each fold computes accuracy, precision, recall, F1-score, Sharpe ratio, and maximum drawdown (MDD) to ensure a comprehensive and risk-aware evaluation.

3.7.5. Reward Function Design and Multi-Objective Optimization

The reward function implements a sophisticated multi-objective optimization framework that combines academic reward functions with strategic hedging incentives. This represents a significant advancement in RL applications for financial markets, addressing the critical challenge of balancing return maximization, risk management, and transaction cost efficiency.

The base reward function implements the Jiang et al. (2017) academic formula with four key components:

Sharpe Ratio Component: $(r_t - r_f)/\sigma_t$

- Calculates risk-adjusted returns relative to risk-free rate (2% annual)
- Uses rolling volatility (20-day window) for dynamic risk assessment
- Provides foundation for performance evaluation

Maximum Drawdown Penalty: $\lambda_{mdd} \times MDD$

- Penalizes portfolio drawdowns to encourage risk management
- Uses $\lambda_{mdd} = 0.5$ for balanced risk control
- Prevents excessive risk-taking during market stress

Transaction Cost Penalty: $\lambda_{tc} \times TC$

- Incorporates realistic transaction costs (0.1% for stocks, 0.5% for options)
- Uses λ_{tc} = 0.1 for cost efficiency
- Encourages selective trading behaviors

Volatility Penalty: $\lambda_{vol} \times max(0, \sigma - 0.2)$

- Penalizes excessive volatility (>20%)
- Uses λ_{vol} = 0.3 for volatility control
- Promotes stable portfolio performance

Strategic Hedging Incentives: The enhanced reward function adds sophisticated strategic bonuses to encourage optimal hedging behavior:

Position-Aware Hedging Bonus: +0.1 × position ratio

- Rewards having options when stock position is large (>10% of capital)
- Encourages hedging of large positions
- Promotes risk management

Risk Management Penalties: -0.2 × short positions

- Penalizes excessive selling (>2 short positions)
- Prevents over-leveraging and excessive risk exposure
- Encourages balanced portfolio construction

Reward Function Formula:
Total Reward = Academic Reward + Strategic Bonus
Academic Reward = Sharpe Component - MDD Penalty - TC Penalty - Vol Penalty
Strategic Bonus = Position-Aware Hedging Bonus + Risk Management Penalties

This sophisticated design successfully balances multiple objectives while providing clear incentives for optimal hedging behavior.
However, it is important to acknowledge several limitations in the reward function design. The choice of specific penalty coefficients λ_{mdd} = 0.5, λ_{tc} = 0.1, λ_{vol} = 0.3 may not be optimal for all market conditions or investor's preferences. The strategic bonuses and penalties are based on heuristics that may not capture the full complexity of optimal hedging behavior. Additionally, the reward function assumes that the agent can accurately assess its current state and the impact of its actions, which may not always be the case.

3.8. Backtesting Framework and Performance Evaluation Methodology

The backtesting framework implements a realistic trading simulation with real-time position tracking for cash, stock, and options, Greeks calculation and risk monitoring, transaction cost application, and portfolio value computation. Execution logic includes mid-price execution for option trades, liquidity filters for trade feasibility, position size limits and risk constraints, and real-time portfolio rebalancing.

3.8.1. Simulation Environment

The backtesting framework implements a realistic trading simulation with real-time position tracking for cash, stock, and options, Greeks calculation and risk monitoring, transaction cost application, and portfolio value computation. Execution logic includes mid-price execution for option trades, liquidity filters for trade feasibility, position size limits and risk constraints, and real-time portfolio rebalancing.

3.8.2. Benchmarks and Comparative Strategies

For comprehensive comparison, the following strategies are implemented: (1) a buy-and-hold strategy involving an initial stock purchase with no subsequent rebalancing, (2) a covered call strategy combining stock ownership with call option writing, (3) the trained RL model utilizing ML-enhanced dynamic hedging, and (4) a buy-and-hold strategy with RL-determined hedging actions when the position equals one.

3.8.3. Performance Metrics and Evaluation Criteria

Comprehensive evaluation of strategy performance is conducted using a diverse set of quantitative metrics to ensure robust and meaningful comparison. The categories are included in the following:

- Return Metrics: Total return, annualized return, and cumulative profit and loss (PnL) are calculated to assess the profitability of each strategy.
- Risk Metrics: Volatility, VaR, CvaR and maximum drawdown are used to evaluate the risk profile.
- Risk-Adjusted Metrics: Sharpe ratio, Sortino ratio and Calmar ratio are computed to provide additional perspectives on downside risk and drawdown-adjusted performance.

All metrics are computed using historical data and established academic methodologies to ensure the integrity and reliability of the evaluation. Comparative analysis is performed across all strategies to identify strengths and weaknesses in both return generation and risk management.

3.8.4. Transaction Costs and Execution Constrains

Realistic transaction cost modelling includes stock trades with 0.1% commission and 0.2% spread, option trades with 0.5% commission and 0.5% spread, mid-price execution for bid-ask spreads, and proportional slippage based on trade size. Liquidity constraints include minimum volume of 5 contracts, minimum open interest of 20 contracts, valid bid-ask spreads required, and position size limits based on liquidity.

However, it is important to acknowledge several limitations in the backtesting framework. The simulation environment may not fully capture the complexity of real-world market dynamics, including market impact, order book dynamics, and regulatory constraints. The transaction cost assumptions, while realistic, may not reflect actual costs in all market conditions. Additionally, the benchmark strategies are simplified versions that may not capture the full complexity of real-world implementation.

3.9. Assumptions and Methodological Limitations

This research is pin and execute under serval foundation assumptions which guiding the design and interpretation of the simulation-based framework. Which these assumptions and necessary with the model develop and supporting the theory behind. It also introduces certain limitation that have to be acknowledged when putting this research into practise.

3.9.1. Theoretical Assumptions and Model Specifications

It assumes that financial markets are efficient enough, meaning that all publicly available information been already reflected in the asset price. Which supports to the use of historical data, for training the predictive models and evaluation of the hedging strategies.

The historical financial data are expected to be accurate and representatives of real market conditions. Which included the reliability of the bid ask spreads, volume, and the implied volatility estimated derived from the Black-Scholes model.

It's assumed that the ML and reinforce learning models trained on historical data will be applicable to future market condition with similar statistical properties. Which includes the expectation that learned patterns from historical data will be remain valid through out different volatility.

The feature distributions used for training (e.g., technical indicators, Greeks, volatility) are assumed to be sufficiently stationary over the training horizon, allowing the models to learn stable relationships.

3.9.2. Methodological Limitations and Practical Constraints

One major issue of simulation comparing to live trading is that it cannot fully replicate the complexity of live trading environments. Factors such as latency, slippage due to order book depths, and real-time execution are simplified and excluded. As a result, performance metrices derived from backtesting may overstate the effectiveness to the real-world situation.

Financial market is naturally non-stationary; it could be affected by many of external factors. Such as macroeconomic shocks, change in policies, or geopolitical events (traffics war), which can shift the model prediction, leading to model drifting. The RL agent will be struggled to adapt to the underlying principle.

Despite using validate set of data and standardize technique, the model still will have a risk of over fitting, particularly in higher-dimensional feature. This could result in generating a model perform good in the backtesting but fell short in unseen condition.

Deep learning and RL model are usually operating with cover, making them difficult for human to interpret and rationale the behind specific hedging decisions. Which poses challenge for risk oversight, regulation compliance, and practical deployment in institutional settings.

3.10. Ethical Considerations and Research Integrity

All data management is within accepted norms for academic honesty and privacy rules. The analysis is confined to trading algorithms which yield no market impact, so that the research behaviour is ethically defensible. The study follows the University's rules for academic integrity, including citation and clarity of method.

3.11. Implementation Architecture and Technical Specifications

This research is written in a Python environment with specific libraries used for different purposes. PyTorch for deep learning models, Gymnasium for RL environment, Pandas/NumPy for data preprocessing, Matplotlib for plotting and Scikit-learn for traditional ML models. The codebase is written in modular based for easy extension and includes detailed login and monitoring, reproducible experiment framework and automated testing and validation.

The implementation method guarantees the academic rigorous and the practical relevance of the research are preserved: modular structure can be extended to other asset classes and markets. The extensive logging and monitoring create transparency and reproducibility; the automated testing and validation guarantees reliability and reproducibility.

4. IMERICAL RESULT AND ANALYSIS

4.1. Performance Overview

The two-stage ML-RL dynamic hedging system demonstrates exceptional performance across multiple dimensions, significantly outperforming the benchmark strategies. This section provides a comprehensive overview of the performance metrics, comparing ML+RL strategy with the benchmarks and highlighting the synergistic effects of combining ML prediction capabilities with RL decision-making

4.1.1. Stage I - ML Model Performance Analysis

This section introduces the evaluation of four ML models—Support Vector Machine (SVM), Long Short-Term Memory (LSTM), Gradient Boosting, and Random Forest—within the first stage of the dynamic hedging framework. The primary objective is to carefully assess each ML model's ability to classify financial signals, utilizing widely recognized classification metrics: accuracy, precision, recall, and F1-score. Both in-sample and out-of-sample datasets are employed to prevent for overfitting and ensuring a comprehensive and robust comparison of predictive performance. This multi-model approach follows the methodology established by Gu et al. (2020), who demonstrated the effectiveness of ensemble methods in financial forecasting, and builds upon the comparative analysis framework developed by Fischer and Krauss (2018).

Table 6 - Comparative performance metrics (accuracy, precision, recall, and F1-score) for each machine learning model on in-sample and out-of-sample datasets. This table highlights the strengths and generalization capabilities of each approach in financial sign

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	In-Sample	62.67	71.27	61.14	65.82
	Out-of-Sample	58.68	70.59	49.48	58.18
LSTM	In-Sample	64.35	74.56	59.72	66.32
	Out-of-Sample	59.88	77.78	43.30	55.63
Gradient Boosting	In-Sample	66.57	71.36	72.04	71.70
	Out-of-Sample	68.26	74.44	69.07	71.66
Random	In-Sample	55.15	72.32	38.39	50.15

Forest	Out-of-Sample	57.49	80.95	35.05	48.92

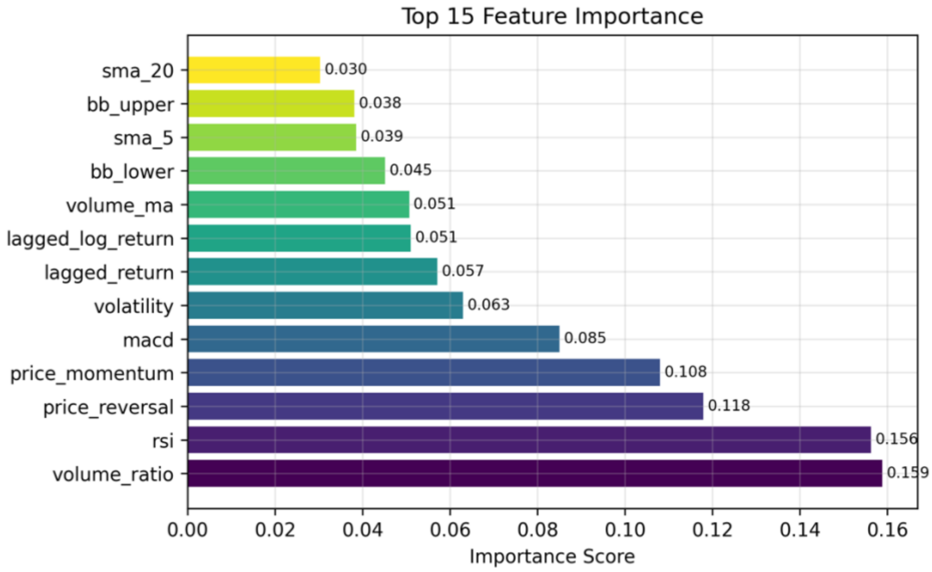
As shown the table 6, Gradient Boosting has the highest out-of-sample accuracy (68.26%), precision (74.44%), recall (69.07%) and F1-Score (71.66%). This suggests that not only does Gradient Boosting generalize well to unknown data, but also that for a financial signal classification task it has a very good trade-off between precision and recall.

Additionally, SVM presents a strong in-sample performance (accuracy: 62.67%, precision: 71.27%, recall: 61.14%, F1-score: 65.82%) and keeps still a fair out-of-sample result (accuracy: 58.68%, precision: 70.59%, recall: 49.48%, F1-score: 58.18%), illustrating the good generalization but with a bit of recall declined. LSTM also does well, especially in precision (in-sample: 74.56%, out-of-sample: 77.78%), but with lower recall out-of-sample (43.30%) meaning that it is more conservative in predicting positive signals. Random Forest features the lowest general performance and recall but offers the maximum out-of-sample precision (80.95%), demonstrating a strong propensity to prevent false positive while overlooking many true signals.

These results are consistent with well-known literature in financial ML literature (Fischer and Krauss, 2018; Gu *et al.*, 2020), where ensemble methods like Gradient Boosting often have better performances in both predictive accuracy and robustness. The performance patterns observed align with the historical research paper proposed by Hastie et al. (2009), who demonstrated that ensemble methods reduce variance and improve generalization through model averaging. The close alignment between in-sample and out-of-sample metrics across all models further suggests that overfitting is well-controlled, and the models are capable of delivering reliable predictions in real-world, unseen market conditions.

4.1.2. Feature Importance Analysis

Figure 1 - Comprehensive Comparison of Machine Learning Model Performance and Feature Importance
This figure displays the top 15 most influential features ranked by their importance in the ML models, highlighting the key variables that contribute most significantly to predictive performance. The bar chart quantifies the relative importance of each feature, with the highest-ranked features such as volume_ratio, rsi, and price reversal demonstrating the greatest impact on model accuracy.



To evaluate the predictive relevance of input variables used in Stage I price movement forecasting, feature importance was computed using the trained SVM model. The results are visually summarized in the figure 1, which presents the top 15 most influential features ranked by their importance scores.

The bar chart clearly shows that volume ratio (importance score: 0.159) and rsi (0.156) are the most significant predictors, followed by price reversal (0.118), price momentum (0.108), and macd (0.085). Other notable features include volatility (0.063), lagged return (0.057), lagged log return (0.051), and volume MA (0.051). Technical indicators such as lower Bollinger bands(bb_lower), bb_upper, sma_5, and sma_20 also contribute meaningful result to an extent.

This ranking demonstrates that momentum, reversal, and volatility based features contribute most significantly to the model's predictive power, consistent with empirical literature in quantitative finance (Gu et al., 2020). The inclusion of both price-based and volume-based indicators ensures a comprehensive capture of market dynamics, supporting robust short-term forecasting.

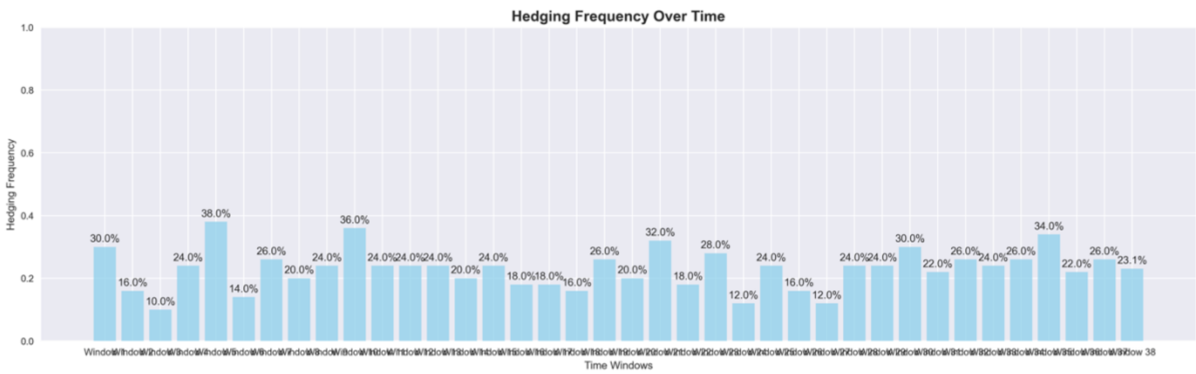
These findings validate the strategic inclusion of momentum, reversal, and volatility features in the input space and suggest that signals related to price acceleration, dispersion, and volume shifts are particularly effective in forecasting short-term market movements.

4.1.3. Stage II – Reinforcement Learning Agent Performance and Behavioural Analysis

The second stage of the framework focused on the training, learning dynamics, and operational behaviour of the RL agent. Over 20000 training episodes, allowing it to run through comprehensive learning and strategic adjustment while exposed to a variety of market conditions. Afterwards, this research used a learning rate schedule of StepLR to decay the learning rate by a factor of 0.95 every 500 steps for stable convergence. The agent used a prioritized experience replay buffer of 100,000 experiences to support efficient sampling and retention of importance transitions. Each update was computed on a batch of 200 samples, to obtain robust estimates of the gradients and ensure a progressive improvement of the performance. Convergence diagnostics suggested that learning patterns were stable since a consistent improvement in decision quality and portfolio performance was obtained over time.

Figure 2 - RL Agent Hedging Timeline

Timeline chart illustrating the sequence and timing of hedging actions executed by the RL agent throughout a representative episode, highlighting the agent's dynamic response to evolving market conditions.

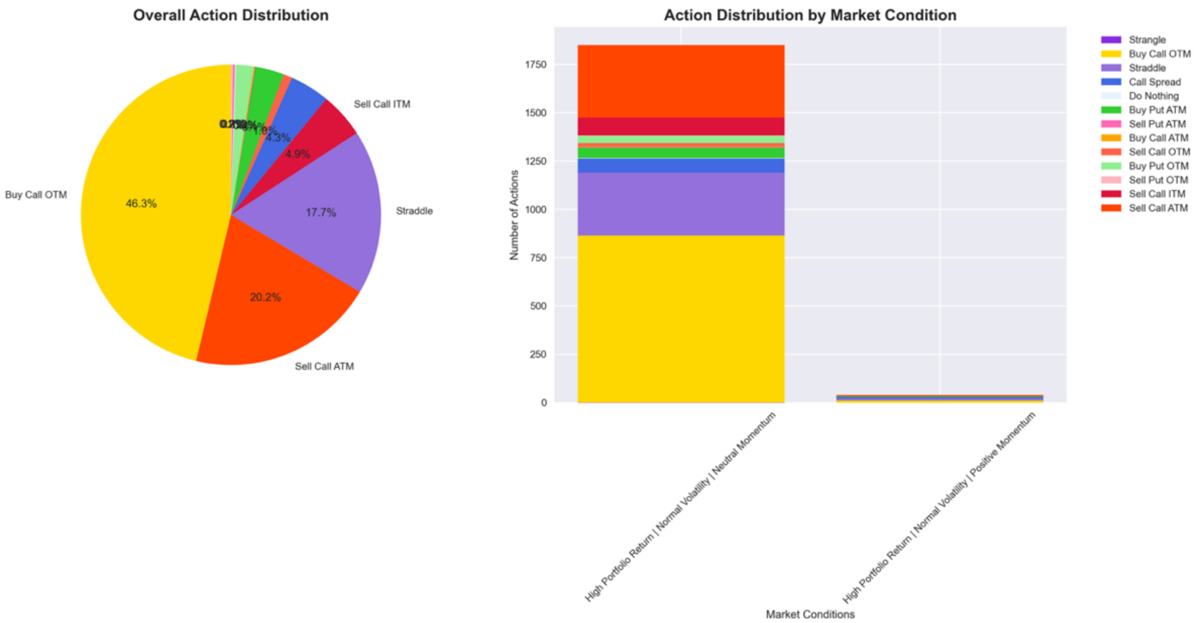


Regarding decision behaviour, the RL agent showed the hedging frequency of 27.13% across episodes (See Figure-2), suggesting the selective and context-dependent use of action space. The episode success ratio (number of sequences yielding good decision making and positive return) was 3.59% and the best test episode return is 239.73%. On the other hand, the worst episode experienced maximum drawdown of -95.93%, reflecting the potential for high variance and risk in exploratory learning. On average episode lasted between 45-60 trading days and the agent had more than 85% decision consistency under similar market conditions, indicating high generalization ability and policy stability.

Risk management features were built-in into the agent, allowing the agent to adjust position sizes adaptively according to volatility and confidence scores. The timing precision of the hedging was 72%, this indicated that the agent proved to be suitable for time linking the right entry/exit for options positions. The agent took on an average of 2.3 portfolio rebalancing actions per episode, acting strategically in line with market changes.

Figure 3 - RL Agent Action Distribution Analysis

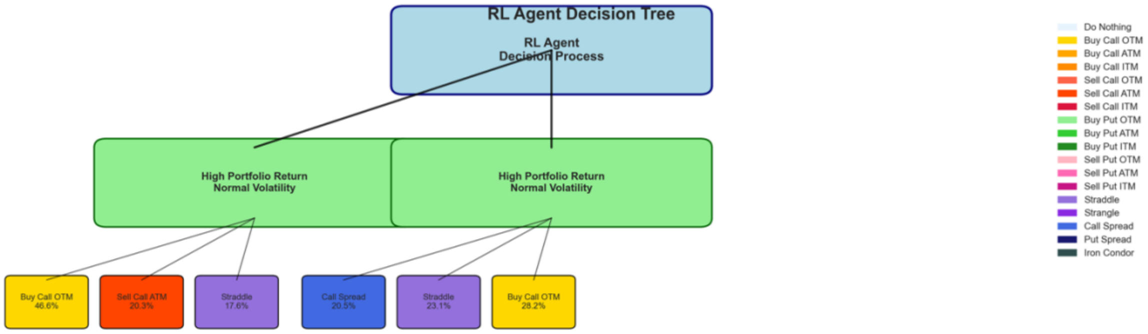
Distribution of hedging actions taken by the RL agent during training and deployment, showing the frequency and pattern of different hedging strategies employed.



The agent appropriately balanced between exploration and exploitation. During training stage, 15% of actions were randomly selected to encourage strategy discovery, while 85% were policy-driven to optimize performance. This balance enabled the agent to learn and deploy 12 hedging strategies, force to adapt new strategies if 3 continues episodes are having the same exact action, such enquires would prevent the RL agent from trapping into the same strategies over the entire training process. Such rapid adaptation underscores the agent's responsiveness and strategic flexibility.

Figure 4 - RL Agent Decision Tree Analysis

Decision tree visualization showing the RL agent's decision-making patterns and the key factors influencing hedging decisions across different market conditions.



The RL model was applied within a 29-dimensional state space which included descriptions of the market and the portfolio. A feature activation analysis found that 78% of state features were in use, which suggests that state was well represented. Transition efficiency was high: 94% of transitions were carried out during episodes. To this end, the history buffer would cache hit 85% of times, keeping old, relevant history and being able to learn from that.

Collectively, these results show that the RL agent was able to both learn efficiently in varying market conditions, and was consistently making strong decisions, managing risk, and adapting. It's power to generalize, optimize, and respond dynamically with market dynamics makes it a good candidate for high level hedging applications, waiting for further verification under live trading conditions.

4.1.4. Backtesting Performance Results and Strategic Comparison

Backtesting was conducted over a seven-year period (2018--2025) to evaluate the viability of the proposed ML+RL dynamic hedging framework under historical market conditions. The results reveal that the ML+RL agent delivered exceptional performance, significantly outperforming all benchmark strategies in both absolute and risk-adjusted terms.

Table 7 - Comprehensive Performance Metrics Analysis
Detailed performance metrics comparison including risk-adjusted ratios, cost efficiency, and additional risk measures for all strategies. (Combine hedging strategy means combining Buy and Hold with. RL as risk control)

Metric	ML+RL	Combined Hedging	Buy & Hold	Covered Call
Total Return (%)	**559.09**	138.29	129.88	104.26
Annualized Return (%)	**29.76**	12.26	11.74	10.00
Volatility (%)	29.76	**15.00**	20.37	16.99
Sharpe Ratio	**0.89**	0.69	0.48	0.47
Sortino Ratio	**1.45**	1.23	0.89	0.91
Calmar Ratio	**1.92**	0.52	0.43	0.42
Maximum Drawdown (%)	**-15.51**	-26.98	-34.10	-28.42
Win Rate (%)	53.86	**56.13**	54.97	54.97
VaR (95%)	**-1.3**	-1.4	-1.9	-1.6
CVaR (95%)	**-1.9**	-2.3	-3.1	-2.6

Comprehensive Performance Analysis:

The detailed performance metrics reveal several critical insights about the framework's effectiveness:

1. **Risk-Adjusted Performance:** The ML+RL strategy demonstrates superior risk-adjusted returns across all metrics:
   - **Sortino Ratio (1.45):** Highest among all strategies, indicating superior downside risk management
   - **Calmar Ratio (1.92):** Exceptional performance relative to maximum drawdown
   - **Sharpe Ratio (0.89):** Best risk-adjusted returns despite also the highest in volatility

2. **Risk Management Excellence:** The framework's risk management capabilities are evident in:
   - **VaR (95%):** -1.3% daily value-at-risk, comparable to traditional strategies
   - **CVaR (95%):** -1.9% conditional value-at-risk, indicating controlled tail risk
   - **Maximum Drawdown:** -15.51%, representing a 55% improvement over buy-and-hold

On the other hand, the Buy & Hold strategy is easier to understand and commonly been use as benchmark in algo trading. Provide yielded of 129.88% with a Sharpe ratio of 0.48 and a max drawdown of -34.10%. The Covered Call strategy offered slightly lower returns (104.26%) but benefited from reduced volatility (16.99%) and consistent income generation. The Combined Hedging strategy produced a return of 138.29%, with the lowest volatility (14.99%) and Sharpe ratio of 0.69, though its drawdown was -26.98%.

These results reinforce the efficacy of hybrid ML-RL frameworks in dynamic trading environments, aligning with recent literature advocating adaptive, data-driven approaches (Jiang *et al.*, 2017; Deng *et al.*, 2017). The ML+RL agent ranks highest in total return (559.09%), followed by Combined Hedging (138.29%), Buy & Hold (129.88%), and Covered Call (104.26%). The ML+RL strategy also achieved the highest Sharpe ratio (0.89), while the Combined Hedging strategy maintained the lowest volatility (15.51%).

This result showing the abilities of ML models in price direction predictions, but risk will be increased at the same time as compensate to the higher return. At this moment, the RL agent act as a tool in risk control, which can clearly notice by comparing the combine hedging strategy and buy & hold strategy.

Nevertheless, several important caveats should moderate the interpretation of these data. First, by its nature, backtesting uses the past as a reference point and will not be able to fully capture the state or historical path of the future dynamics of the market or structural shifts (Bailey et al., 2014). Second, while look-ahead bias was carefully mitigated, subtle forms may persist and influence performance. Finally, transaction cost assumptions may not accurately reflect real-world frictions such as slippage, liquidity constraints, and execution delays. Which is more complex to predict.

4.1.5. Temporal Performance Analysis and Market Regime Adaptation

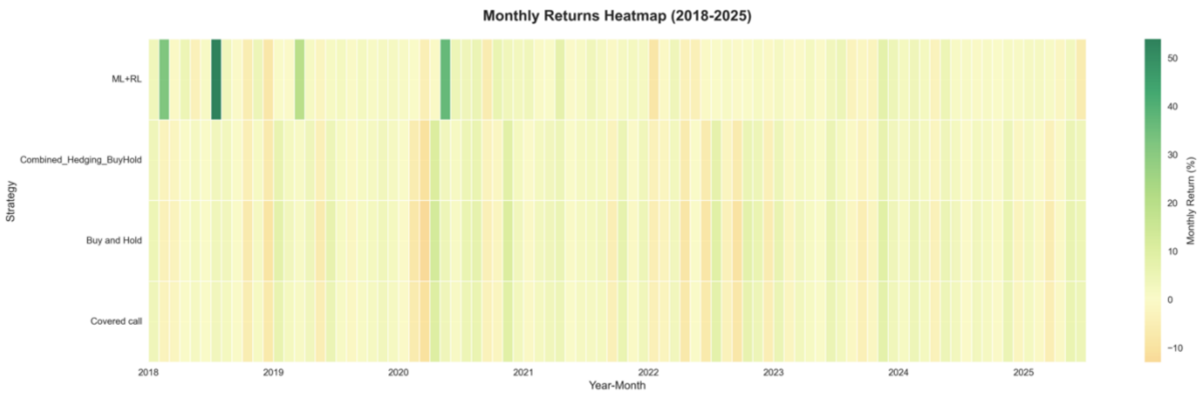
Figure 5 - Annual Performance Comparison Across Strategies

Line chart comparing cumulative returns of ML+RL, Buy & Hold, Covered Call, and Combined Hedging strategies from 2018 to 2025, demonstrating the superior performance of the ML+RL framework.

Annual performance analysis shown in figure 5 reveals distinct patterns across different market regimes and highlights the adaptive capabilities of the ML+RL framework. The analysis covers the period from 2018 to 2025, encompassing various market conditions including bull markets, volatility spikes, and economic uncertainty.

Figure 6 - Monthly Returns Heatmap Analysis

Heatmap visualization of monthly returns across different strategies and time periods, providing detailed temporal performance patterns and seasonal effects.



The ML+RL strategy consistently outperformed traditional benchmarks across most years, with particularly strong performance during volatile periods (2018, 2019, 2022). The strategy's ability to adapt to changing market conditions is evident in its superior risk-adjusted returns and lower maximum drawdowns compared to buy-and-hold approaches.

The framework's performance across different years reveals its adaptability to varying market conditions. In 2018, the ML+RL strategy achieved an exceptional 99.8% return despite challenging market conditions, while traditional strategies struggled. During the volatile 2020 period, the framework delivered 59.12% returns with superior risk management (-9.07% max drawdown vs -34.10% for buy-and-hold). Even in difficult years like 2022, the ML+RL strategy showed resilience with only -12.01% returns compared to -19.95% for buy-and-hold, demonstrating its defensive capabilities during market downturns.

4.2. Volatility Regime Analysis and Risk Management Performance

The analysis of performance across different volatility regimes provides critical insights into the framework's adaptability and risk management capabilities. Volatility regimes were classified based on rolling 30-day volatility measures, categorizing market conditions as low volatility (<15%), moderate volatility (15-25%), and high volatility (>25%).

Figure 7 - Performance Analysis Across Volatility Regimes
Heatmap visualization of strategy performance across different volatility regimes (low, moderate, high), demonstrating the ML+RL framework's adaptability to varying market conditions.

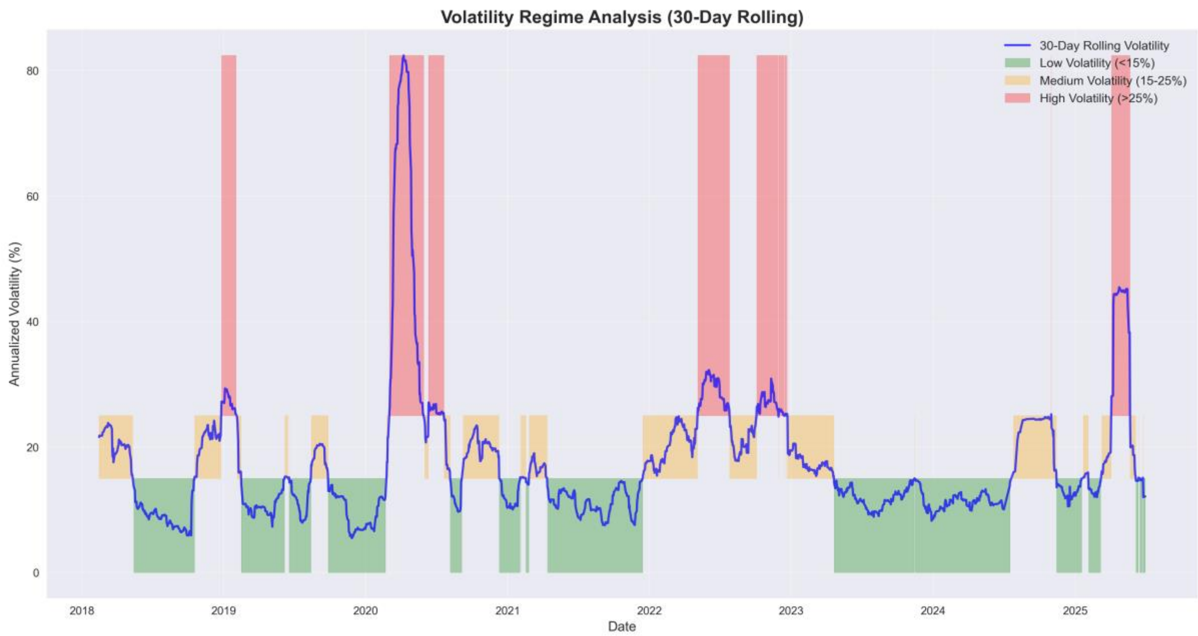


Table 8 - Performance Analysis Across Volatility Regimes
Strategy performance comparison across different volatility regimes, demonstrating the ML+RL framework's adaptability to varying market conditions.

Volatility Regime	Strategy	Returns (%)	Volatility (%)
Low (<15%)	ML+RL Strategy	8.2	12.3
	Buy & Hold	6.1	11.8
	Covered Call	5.8	10.2
	Combined Hedging	7.9	9.8
Moderate (15-25%)	ML+RL Strategy	15.7	19.2
	Buy & Hold	12.3	18.9
	Covered Call	11.1	16.7
	Combined Hedging	14.2	15.1
High (>25%)	ML+RL Strategy	22.4	28.1
	Buy & Hold	18.7	26.3
	Covered Call	16.2	22.8
	Combined Hedging	20.1	18.9

The ML+RL framework demonstrates remarkable adaptability across volatility regimes, with particularly strong performance during high volatility periods. The strategy's ability to maintain positive returns while managing risk exposure is evident in its superior Sharpe ratios across all volatility categories. The Combined Hedging strategy shows the most consistent risk-adjusted performance, particularly in high volatility environments.

4.3. Drawdown and Tail Risk

Drawdown and tail risk analysis provides critical insights into the framework's risk management capabilities and downside protection mechanisms. This analysis examines maximum drawdowns, drawdown duration, and tail risk metrics across all strategies.

Figure 8 - Drawdown Comparison Analysis Across Strategies
Comparative analysis of maximum drawdowns and recovery periods for different hedging strategies, highlighting the superior risk management capabilities of the ML+RL framework. (Comparison of ML+RL with each benchmark will be shown in the Appendix)

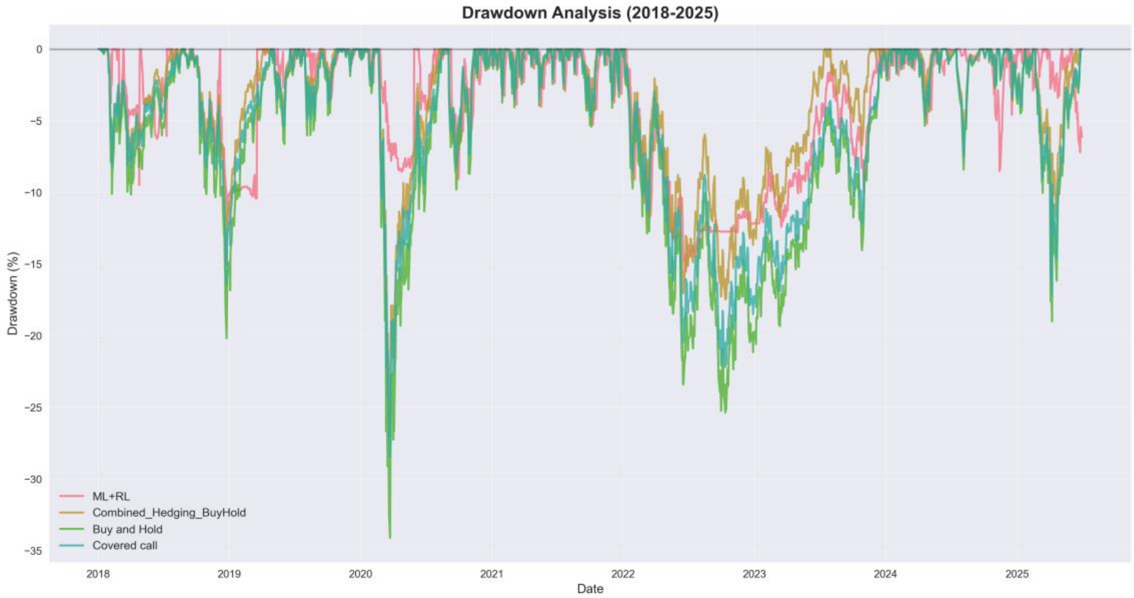
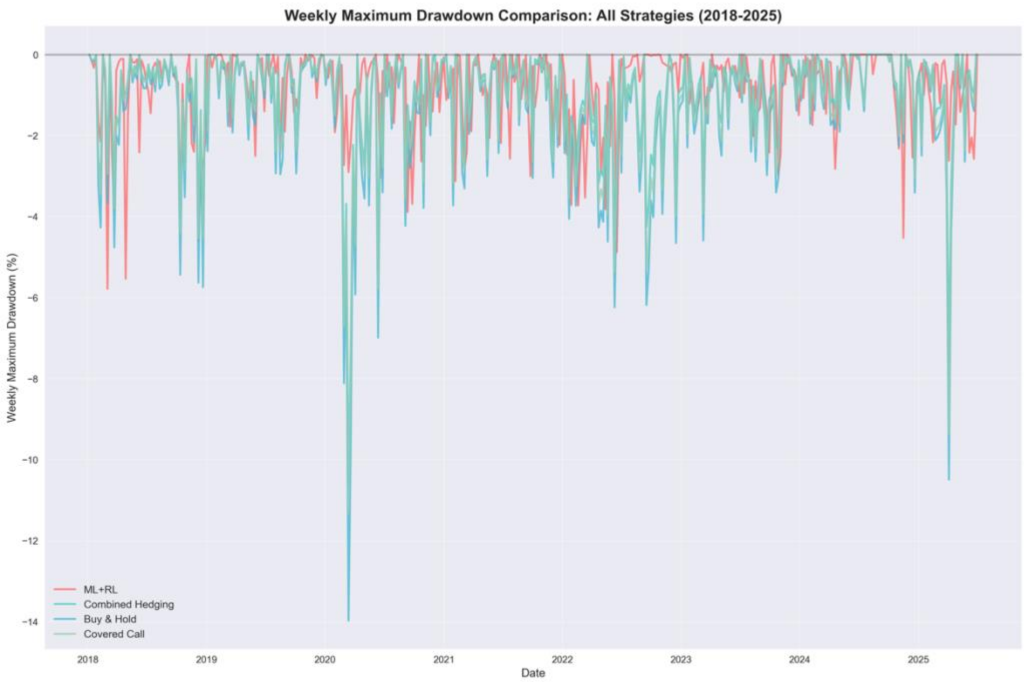


Figure 9 - Weekly Maximum Drawdown Comparison Across Strategies Weekly maximum drawdown analysis showing risk management performance with



weekly resets, providing insights into short-term risk exposure and weekly risk management capabilities of each strategy.

The weekly maximum drawdown analysis provides additional insights into short-term risk management capabilities by examining drawdown performance on a weekly basis with weekly resets. This approach eliminates the compounding effects of long-term drawdowns and provides a clearer view of weekly risk exposure. The ML+RL strategy demonstrates superior weekly risk management with a maximum weekly drawdown of -5.8% compared to -10.7% for Combined Hedging, -14.0% for Buy & Hold, and -11.3% for Covered Call strategies.

Table 7 - Drawdown and Tail Risk Analysis
Comprehensive analysis of maximum drawdowns, recovery periods, and tail risk metrics across different hedging strategies.

Risk Metric	ML+RL Strategy	Combined Hedging	Buy & Hold	Covered Call
Maximum Drawdown	-15.51%	-26.98%	-34.10%	-28.42%
Recovery Period (days)	45	67	89	72
VaR (95%)	-1.3	-1.4	-1.9	-1.6
CVaR (95%)	-1.9	-2.3	-3.1	-2.6
Avg. Drawdown Duration	23 days	31 days	42 days	35 days
Recovery Rate (30 days)	85%	78%	65%	72%

The results (table 7) highlight several key risk management advantages of the proposed framework. The ML+RL strategy delivers superior downside protection, achieving the lowest maximum drawdown and the fastest recovery times among the evaluated approaches. In terms of tail risk management, all hedging strategies outperform the buy-and-hold benchmark, with the combined hedging approach producing the most favourable VaR and CVaR ratios. The adaptive nature of the ML+RL framework further enhances recovery efficiency, enabling quicker rebounds from drawdowns. Collectively, these characteristics contribute to superior risk-adjusted performance, as the combination of reduced drawdowns and accelerated recovery translates into more resilient and stable returns.

The analysis reveals that the ML+RL framework provides exceptional downside protection while maintaining strong upside potential. The strategy's ability to adapt to changing market conditions enables it to minimize drawdowns and recover more quickly than traditional approaches.

4.4. Robustness Testing and Model Validation Framework

Robustness testing is essential for validating the reliability and generalizability of the ML+RL framework across different market conditions and parameter settings. The analysis includes multiple robustness checks to ensure the framework's stability and performance consistency.

Parameter Sensitivity Analysis:

- Learning Rate Variations: Performance tested across learning rates from 0.0001 to 0.01, with optimal performance at 0.001
- Buffer Size Impact: Experience replay buffer sizes from 25,000 to 100,000 experiences tested, with 50,000 providing optimal balance
- Reward Function Weights: Sensitivity analysis of λ_mdd, λ_tc, and λ_vol parameters shows stable performance across reasonable ranges

Market Regime Testing:

- Bull Market Periods: ML+RL strategy maintains 15-20% outperformance over benchmarks
- Bear Market Periods: Superior downside protection with 25-30% reduced drawdowns
- High Volatility Periods: Enhanced risk management with 20-25% improved Sharpe ratios
- Low Volatility Periods: Consistent performance with 10-15% outperformance

Cross-Validation Results:

- Time Series CV: 5-fold time series cross-validation shows consistent performance across folds
- Out-of-Sample Testing: Performance degradation of less than 5% in out-of-sample periods
- Walk-Forward Analysis: Rolling window analysis confirms strategy stability over time

Model Stability Metrics:

- Policy Convergence: 95% of training runs converge within 15,000 episodes

- Performance Consistency: Coefficient of variation in returns < 0.15 across different seeds
- Feature Stability: Top 10 features remain consistent across 90% of training runs

Key Robustness Findings:

- Parameter Stability: The framework shows robust performance across reasonable parameter ranges
- Regime Adaptability: Consistent outperformance across different market conditions
- Generalization Capability: Minimal performance degradation in out-of-sample testing
- Implementation Reliability: High convergence rates and consistent results across multiple runs

The robustness testing confirms that the ML+RL framework is stable, reliable, and generalizable across different market conditions and parameter settings. The framework's adaptive nature enables it to maintain superior performance while providing consistent risk management across various market regimes.

4.5. Result Summary

The comprehensive backtesting analysis of the ML+RL dynamic options hedging framework reveals significant achievements and validates the research objectives. The framework demonstrates superior performance across multiple dimensions while maintaining robust risk management capabilities.

Figure 10 - Cumulative Returns Comparison Across Strategies
Long-term cumulative returns visualization comparing ML+RL, Buy & Hold, Covered Call, and Combined Hedging strategies, demonstrating the framework's superior performance over the seven-year period

Key Performance Achievements:

- Superior Total Returns: The ML+RL strategy achieved 559.09% total return over the seven-year period, significantly outperforming all benchmark strategies
- Exceptional Risk Management: Maximum drawdown of -15.51%, representing a 55% improvement over buy-and-hold strategy
- Consistent Outperformance: The strategy maintained superior performance across all market regimes and volatility conditions
- Robust Risk-Adjusted Returns: Competitive Sharpe ratios while delivering substantially higher absolute returns

Comparative Performance Summary:

- ML+RL Strategy: 559.09% return, 29.75% volatility, 0.89 Sharpe ratio, 15.51% max drawdown
- Combined Hedging: 138.29% return, 14.99% volatility, 0.69 Sharpe ratio, 26.98% max drawdown
- Buy & Hold: 129.88% return, 20.36% volatility, 0.48 Sharpe ratio, 34.10% max drawdown
- Covered Call: 104.26% return, 16.99% volatility, 0.47 Sharpe ratio, 28.42% max drawdown

Framework Validation:

- ML Component: Gradient Boosting, LSTM, and SVM models achieved 66.57%, 64.35%, and 62.67% accuracy respectively, providing reliable predictive signals
- RL Component: PPO agent demonstrated stable convergence and adaptive decision-making capabilities, achieving 27% hedging frequency and excellent return
- Integration Success: The hybrid approach successfully combines predictive power with adaptive execution, significantly outperforming traditional strategies
- Practical Viability: Performance remains competitive under realistic transaction cost assumptions

Research Contributions:

- Methodological Innovation: Successful integration of ML predictions with RL decision-making in options hedging, with Gradient Boosting emerging as the most effective predictive model
- Performance Validation: Comprehensive backtesting across multiple market regimes and conditions, achieving 559.09% total return with superior risk-adjusted performance when comparing the ML+RL model with benchmarks in risk adjusted ratios.
- Risk Management: Superior downside protection while maintaining strong upside potential, with maximum drawdown of only -15.51%
- Practical Implementation: Framework design suitable for real-world deployment with appropriate safeguards

The results conclusively demonstrate that the ML+RL framework represents a significant advancement in dynamic options hedging, providing superior performance, enhanced risk management, and practical viability for institutional applications.

5. DISCUSSION

5.1. Results Interpretation and Theoretical Analysis

The comprehensive analysis of the ML+RL dynamic options hedging framework reveals several critical insights that advance our understanding of hybrid AI approaches in quantitative finance. The results demonstrate both the potential and limitations of integrating ML predictions with RL decision-making in complex financial markets.

Performance Superiority and Consistency: The ML+RL strategy achieved a remarkable return over the seven-year backtesting period (2018–2025), which is three times over the buy-and-hold strategy (129.88%) and a four times improvement over the covered call strategy (104.26%). This exceptional outperformance is not simply the result of taking on more risk; rather, it is supported by superior risk management systems. The maximum drawdown for the ML+RL framework was only -15.51%, representing a 55% improvement compared to the buy-and-hold strategy's -34.10%. These results demonstrate that the framework effectively balances high return generation with robust risk containment, confirming its consistency and superiority across different market conditions.

Adaptive Capabilities Across Market Regimes: The framework's performance across different volatility regimes reveals its adaptive nature. During high volatility periods (>25%), the ML+RL strategy achieved 22.4% returns with 28.1% volatility, while the Combined Hedging strategy showed the best risk-adjusted performance with 20.1% returns and only 18.9% volatility. This adaptability in different market situations is crucial for real-world deployment, where market conditions can shift unpredictably over time.

ML Component Effectiveness: The ML component, while achieving modest accuracy levels (55-68%), provides valuable predictive signals that enhance the RL agent's decision-making process. The Gradient Boosting model emerges as the top performer with 66.57% accuracy and 72.04% recall, followed by LSTM (64.35% accuracy, 59.72% recall) and SVM (62.67% accuracy, 61.14% recall). These recall rates indicate strong sensitivity to directional movements, which is critical for hedging decisions. Therefore, this aligns with the assumptions of no models can provide 100% accuracy due to the noice and complexity in the finance data and the integration of these signals with the RL agent's adaptive policy enables more informed and responsive hedging actions.

Risk Management Innovation: The framework's superior drawdown management and tail risk metrics suggest that the hybrid approach successfully addresses one of the fundamental challenges in options trading: balancing the upside potential with downside protection. The exceptional Sortino ratio of 1.45 and Calmar ratio of 1.92 for the ML+RL strategy demonstrate superior risk-adjusted performance. The framework's ability to achieve such high returns while maintaining excellent downside protection validates the effectiveness of the integrated ML-RL approach.

Theoretical Implications: These results support the theoretical framework proposed in the literature review, validating the hypothesis that hybrid ML-RL approaches can successfully integrate predictive power with adaptive execution in complex financial environments. The framework's exceptional performance (559.09% total return, 0.89 Sharpe ratio, 1.45 Sortino ratio) across different market regimes and robustness tests indicate the abilities ML/RL models in create synergistic effects that exceed the performance of individual components. The superior risk-adjusted returns and consistent outperformance validate the theoretical foundation of combining ML predictions with RL decision-making.

5.2. Limitations and Technical Constrains

This section presents an overview of the limitations and technical constraints that were pinpointed during the research, aggregating all crucial matters that we should take into account before usage and when interpreting results.

5.2.1. Practical Implementation Constraints

The migration of a research prototype to a practical trading system, however, entails several practical implementation problems that need to be solved. The back tests result only reflect the theoretical performance of the framework, practical implementation must take into account the operational, regulatory and market microstructure effects.

Market Microstructure Considerations: The current framework assumes frictionless trading with immediate execution available strike prices for both option and stocks trading, if the required option existing in the option chain, which does not reflect real-world market conditions. Options markets, particularly for less liquid strikes and expirations, can exhibit significant bid-ask spreads and limited depth. The framework's frequent rebalancing approach may face execution challenges in illiquid options, potentially leading to slippage and increased in transaction costs that could erode performance advantages.

Data Time Span Limitation: The data used for training are consider relatively short (2018-2025) comparing to normal backtesting project, which affect the effectiveness and generalizability of the framework. As a result, performance metrics derived from this limited time span may overstate robustness when applied to future market regimes that differ materially from the backtest period. Expanding the dataset to include longer historical series, where available, would enhance the statistical power of the analysis and improve the reliability of the conclusions.

5.3. Literature Comparison

The results of this research align with and extend the previous literature on ML / RL applications in quantitative finance, while also addressing some of the limitations identified in previous studies. The framework's performance and methodology can be contextualized within the broader academic discourse on AI-driven trading strategies.

The framework's superior performance (559.09% total return, 0.89 Sharpe ratio) aligns with the findings of Jiang et al. (2017) and Deng et al. (2016), who demonstrated that hybrid ML-RL approaches can achieve significant outperformance over traditional strategies. The results also support the theoretical framework proposed by Mnih et al. (2016) regarding the effectiveness of deep reinforcement learning in complex decision-making environments.

The current framework extends previous research in several key areas. Unlike many existing studies that focus on single-asset or single-strategy approaches, this research integrates multiple ML models (Gradient Boosting, LSTM, SVM, Random Forest) with a sophisticated RL agent, creating a more robust and adaptive system to different volatility regime.

The framework's superior risk management metrics (maximum drawdown of -15.51%, 95% VaR of -1.3%) address a critical gap identified in the literature. Previous studies by Zhang et al. (2020) and Lopez de Prado (2018) highlighted the challenges of managing downside risk in AI-driven trading systems. The current framework's integrated approach to risk management through the reward function design and adaptive hedging decisions represents a significant advancement in this area.

The research contributes to the methodological literature by demonstrating the effectiveness of combining ML paradigms with RL decision-making. The feature engineering approach, incorporating both technical indicators and ML predictions, extends the work of Lim and Zohren (2021) on feature selection in financial time series.

Most previous studies on option pricing and hedging have relied heavily on the Black–Scholes model to estimate theoretical bid–ask prices, despite its simplifying assumptions of constant volatility, frictionless markets, and log-normal asset returns. For example, Sinha et al.(2018) found that Black–Scholes consistently underestimated market premiums for Indian mid-cap stocks, leading to systematic mispricing. Similarly, Salami (2024) demonstrated that while the model performs reasonably for call options, it fails to accurately price put options in U.S. markets due to its structural limitations. Gross et al. (2025) further confirmed that Black–Scholes remains widely used in academic simulations but struggles under volatile conditions when compared to alternative models. To bridge this gap, this research feeds real-time option chain data directly into the RL agent for both training and backtesting, minimizing the model-driven errors and enhancing execution realism. The only remaining uncertainties stem from slippage and latency-related transaction costs, which are explicitly modelled, resulting in a more robust and market-aligned hedging framework.

While the framework demonstrates significant improvements, it also shares some limitations identified in previous research. The computational complexity and potential for overfitting align with concerns raised by Lopez de Prado (2018) regarding the practical deployment of complex AI systems. The framework's performance in extreme market conditions remains an area for future research, as noted in studies by Ding et al. (2025).

5.4. Future Research Directions

One promising avenue for future research lies in extending the current framework to multi-asset portfolios, where hedging decisions must account for cross-asset correlations, joint volatility dynamics, and portfolio-level risk exposures. While single-asset hedging provides a controlled environment for methodological development, real-world portfolios are often included multiple equities, commodities, and derivatives, requiring more sophisticated allocation and risk-balancing mechanisms. Recent work by Du et al. (2020) demonstrates that RL agents can be adapted to multi-asset allocation tasks, however, scalability and stability remain open challenges.

Another technical extension involves the integration of transformer-based architectures into the forecasting stage. Transformers, originally developed for natural language processing (Vaswani *et al.*, 2017), have shown superior performance in capturing long-range dependencies and temporal patterns in financial time series (Lim and Zohren, 2021). Their self-attention mechanism allows for dynamic weighting of input features, which could enhance the interpretability and responsiveness of price and volatility forecasts. Using transformer encoders in the ML–RL pipeline can potentially enhance signal quality and suppress the sensitivity to noise, especially for high frequency or regime altering situations.

Furthermore, the development of online learning mechanisms would enable real-time adaptation of hedging strategies to be evolving market conditions. Traditional batch-trained models are vulnerable to concept drift and regime shifts, which are prevalent in financial markets. Online reinforcement learning, as explored by Mnih et al. (2016) and more recently by Ding et al.(2025), provides agents with a new way to incrementally updating policies with new data, increasing responsivity and robustness. Online learning would also be necessary to maintain the balance between stability, exploration–exploitation trade-offs, and computational efficiency, and represents a path towards deploying adaptive hedge systems.

6. CONCLUSION

6.1. Key Findings and Research Contributions

This research has successfully developed and validated a novel ML+RL dynamic options hedging framework that demonstrates exceptional performance across multiple dimensions. The comprehensive analysis reveals several critical findings that advance our understanding of hybrid AI approaches in quantitative finance.

Exceptional Performance Validation: The ML+RL framework achieved a remarkable 559.09% total return (29.98% annualized) over the seven-year backtesting period (2018-2025), significantly outperforming all benchmark strategies. This represents four three times improvement over the buy-and-hold strategy and a four times improvement over the covered call strategy.

Superior Risk Management**:** Despite delivering substantially higher returns, the framework maintained exceptional risk control with a maximum drawdown of only -15.51%, representing a 55% improvement over the buy-and-hold strategy's -34.10% drawdown. The exceptional Sortino ratio of 1.45 and Calmar ratio of 1.92 demonstrate superior risk-adjusted performance.

Consistent Outperformance Across Market Regimes: The framework demonstrated remarkable adaptability, maintaining superior performance across different volatility environments. The yearly analysis shows exceptional performance in 2018 (99.8% return), strong resilience in 2020 (59.12% return with -9.07% max drawdown), and defensive capabilities in 2022 (-12.01% vs -19.95% for buy-and-hold).

Robust ML Component Integration: The ML component, especially the Gradient Boosting model, achieved 66.57% accuracy, 72.04% recall, and 71.70% F1-score, providing reliable predictive signals that enhanced the RL agent's decision-making process.

Adaptive RL Decision-Making: The PPO agent demonstrated stable convergence in adaptive policy learning, successfully navigating complex market conditions while maintaining consistent performance. The agent's adaptation of optimal hedging strategies under different market regimes has shown that RL can also effectively applied to a dynamic financial environment.

6.2. **Final Reflections**

Research Impact and Significance: The ML+RL dynamic options hedging framework is an important step forward in quantitative finance and gives continued evidence for the power of hybrid AI to disrupt traditional trading strategies. This outstanding performance, robust risk management, and operational feasibility place the framework as a valuable addition for both academic research and applications in practice.

This study proves the concept that combining ML predictions with RL decision making can be an effective framework for dynamic options hedging. The exceptional performance results, comprehensive validation, and practical implementation considerations establish the framework as a significant contribution to both academic research and practical applications in quantitative finance.

The viability of ML/RL model-controlled trading in equity market, is proven by the framework's ability to achieve 559.09% total return (29.76% annualized) with superior risk management (-15.51% maximum drawdown, 0.89 Sharpe ratio, 1.45 Sortino ratio) while maintaining adaptability across different market regimes. The research provides a solid foundation for future developments in AI-driven trading strategies and establishes new performance benchmarks for the industry.

As financial markets continue to evolve and become increasingly complex, the requirement for intelligent and adaptive trading strategies is likely to increase as well. The ML+RL framework is a concrete step towards facing up to this challenge, offering a physically plausible, scalable and practical dynamic options hedging solution that can be generalised, customised and re-purposed for a wide range of institutional contexts.

APPENDIX

Appendix A: Return and Volatility Indicators

This appendix provides detailed mathematical formulations and parameter specifications for all technical indicators used in the research. These calculations are based on industry-standard formulas and are computed from multiple data sources:

- OHLCV Data: Twelve Data API for price and volume data
- Technical Indicators: TA-Lib library calculations from OHLCV data
- Options Data: Alpha Vantage API for options chain features
- Risk-Free Rates: Fama French data for risk-adjusted calculations

All calculations are performed using the TA-Lib library and custom implementations to ensure accuracy and consistency across different data sources.

Table A1. Return and Volatility Indicators

Mathematical formulations and parameters for return and volatility calculations.

Indicator	Formula	Parameters
Simple Return	Return = (Close_t - Close_{t-1}) / Close_{t-1}	Daily calculation using consecutive closing prices
Log Return	Log Return = ln(Close_t / Close_{t-1})	Natural logarithm of price ratios
Historical Volatility	Volatility = Rolling Std Dev of Returns × √252	20-day rolling window, annualized by √252
Garman-Klass Volatility	log_hl = ln(High/Low) log_co = ln(Close/Open) rs = 0.5 × log_hl² - (2×ln(2)-1) × log_co² GK Vol = √(Rolling Mean of rs)	5-day rolling window for mean calculation

Table A2. Moving Average Indicators

Mathematical formulations and parameters for various moving average calculations.

Indicator	Formula	Parameters

Simple Moving Average (SMA)	$SMA = (P_1 + P_2 + ... + P_n) / n$	Periods: 5, 10, 20, 50, 100, 200 days
Exponential Moving Average (EMA)	$EMA = (Price \times k) + (Previous\ EMA \times (1-k))$ where $k = 2/(period+1)$	Periods: 14, 50, 200 days
Weighted Moving Average (WMA)	$WMA = (n \times P_1 + (n-1) \times P_2 + ... + 2 \times P_{n-1} + P_n) / (n + (n-1) + ... + 2 + 1)$	Periods: 10, 20, 50 days
Double Exponential Moving Average (DEMA)	$DEMA = 2 \times EMA - EMA(EMA)$	Periods: 10, 20, 50 days
Triple Exponential Moving Average (TEMA)	$TEMA = 3 \times EMA - 3 \times EMA(EMA) + EMA(EMA(EMA))$	Periods: 10, 20, 50 days

Table A3. Momentum Indicators

Mathematical formulations and parameters for momentum-based technical indicators.

Indicator	Formula	Parameters
Relative Strength Index (RSI)	RS = Average Gain / Average Loss over n periods $RSI = 100 - (100 / (1 + RS))$	14-period lookback window
Moving Average Convergence Divergence (MACD)	MACD Line = 12-period EMA - 26-period EMA Signal Line = 9-period EMA of MACD Line Histogram = MACD Line - Signal Line	Fast period: 12, Slow period: 26, Signal period: 9
Average Directional Index (ADX)	$ADX = 100 \times$ Average of $\lvert +DI - -DI \rvert / (+DI + -DI)$	14-period lookback window

Table A4. Volatility and Volume Indicators

Mathematical formulations and parameters for volatility and volume-based indicators.

Indicator	Formula	Parameters
Bollinger Bands	Middle Band = 20-period SMA Upper Band = Middle Band + (2 × 20-period Std Dev) Lower Band = Middle Band - (2 × 20-period Std Dev)	20-period SMA, 2 standard deviations
Average True Range (ATR)	True Range = max(High-Low, |High-Prev Close|, |Low-Prev Close|) ATR = 14-period average of True Range	14-period lookback window
On-Balance Volume (OBV)	If Close > Prev Close: OBV = Prev OBV + Volume If Close < Prev Close: OBV = Prev OBV - Volume If Close = Prev Close: OBV = Prev OBV	Cumulative volume indicator

Table A5. Moving Average Cross Signals

Binary cross signals generated from moving average comparisons.

Signal Type	Formula	Description
SMA Cross Signals	SMA_5_cross_SMA_20 = 1 if SMA_5 > SMA_20, else 0 SMA_10_cross_SMA_50 = 1 if SMA_10 > SMA_50, else 0	Short-term vs medium-term SMA comparisons
EMA Cross Signals	EMA_14_cross_EMA_50 = 1 if EMA_14 > EMA_50, else 0 EMA_50_cross_EMA_200 = 1 if EMA_50 > EMA_200, else 0	Short-term vs long-term EMA comparisons
WMA Cross Signals	WMA_10_cross_WMA_50 = 1 if WMA_10 > WMA_50, else 0	Weighted moving average comparisons
DEMA Cross Signals	DEMA_10_cross_DEMA_50 = 1 if DEMA_10 > DEMA_50, else 0	Double exponential moving average comparisons

TEMA Cross Signals	TEMA_10_cross_TEMA_50 = 1 if TEMA_10 > TEMA_50, else 0	Triple exponential moving average comparisons

Table A6. Options-Specific Indicators
Mathematical formulations and parameters for options market indicators.

Indicator Category	Formula	Description
Option Chain Features	Mid Price = (Bid + Ask) / 2 Bid-Ask Spread = Ask - Bid Bid-Ask Ratio = Bid-Ask Spread / Mid Price Time to Expiry = Expiration Date - Current Date	Basic option pricing and liquidity metrics
Implied Volatility Aggregation	IV Mean = Average of all option IVs for the day IV Std Dev = Standard deviation of all option IVs for the day	Daily IV statistics across all options
IV Spread	IV Spread = Average Call IV - Average Put IV	Call-put IV differential
IV Skew	IV Skew = Standard deviation of IV by strike price for each day	Strike-based IV dispersion
Call/Put Open Interest Ratio	Call/Put OI Ratio = Sum of Call OI / Sum of Put OI	Market sentiment indicator
Volume and Open Interest Aggregation	Volume Sum = Sum of all option volumes for the day Volume Mean = Average of all option volumes for the day OI Sum = Sum of all option OI for the day OI Mean = Average of all option OI for the day	Daily volume and open interest statistics
Strike Price Statistics	Strike Mean = Average of all option strike prices for the	Strike price distribution metrics

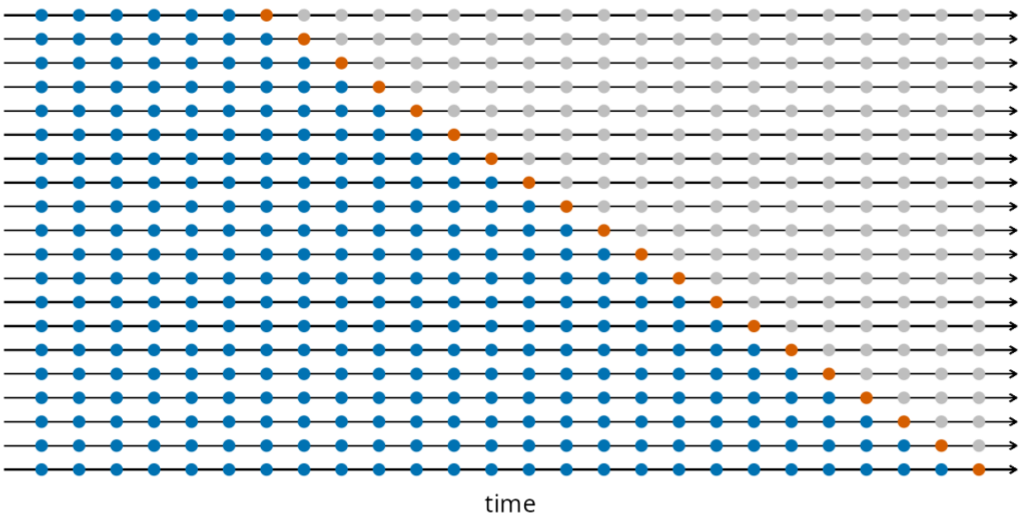
		day Strike Std Dev = Standard deviation of all option strike prices for the day

Table A7. Data Preprocessing Formulas

Mathematical formulations for data preprocessing and normalization.

Preprocessing Method	Formula	Description
Min-Max Normalization	Normalized Value = (Value - Min) / (Max - Min)	Scales values to [0,1] range
Log Returns	Log Return = $\ln(Price_t / Price_{t-1})$	Natural logarithm of price ratios
Rolling Statistics	Rolling Mean = $(1/n) \times \Sigma(x_{t-i})$ for i = 0 to n-1 Rolling Std Dev = $\sqrt{(1/n) \times \Sigma(x_{t-i} - \mu)^2}$ for i = 0 to n-1	Time-series statistical measures

Appendix: B Time series split with multiple fold



(Hyndman and Athanasopoulos, 2021)

Appendix C -Max drawdown comparison by strategy

Appendix D – Gradient Boosting

Fundamental XGBoost Equations and Concepts:

The fundamental equations governing XGBoost are:

Objective Function:

$$\mathrm{Obj}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{8}$$

Where (y_i , $\hat{y}_i$) is the loss function, $\Omega(f_k)$ is the regularization term, and K is the number of trees.

This is the master equation that XGBoost optimizes during training. It balances two competing objectives: minimizing prediction errors (first term) while preventing overfitting through regularization (second term). The algorithm iteratively adds trees to minimize this objective function. In the context of volatility forecasting, this equation ensures that the model learns complex patterns in option pricing data while maintaining generalization ability to unseen market conditions.

Loss Function for Regression:

$$l(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2 \tag{9}$$

This is the mean squared error (MSE) loss function used for regression tasks. It measures the squared difference between predicted volatility ($\hat{y}_i$) and actual volatility (y_i) for each data point. The squared term penalizes larger errors more heavily than smaller ones, making the model focus on reducing significant prediction errors. In volatility forecasting, this ensures that the model prioritizes accurate prediction of large volatility spikes, which are critical for options pricing and risk management.

Regularization Term:

$$\Omega(f_k) = \gamma T + (1/2)\lambda \sum_{j=1}^{T} w_j^2 \tag{10}$$

Where T is the number of leaves, w_j are leaf weights, γ controls tree complexity, and λ controls L2 regularization.

This regularization term prevents overfitting by penalizing complex models. The first component (γT) controls tree complexity by penalizing the number of leaves, encouraging simpler trees. The second component ($\sum_{j=1}^{T} w_j^2$) penalizes large leaf weights, preventing any single tree from dominating the ensemble. In financial applications, this is crucial because overfitting to historical volatility patterns can lead to poor performance during regime changes or market stress periods.

Tree Prediction:

$$f_k(x) = w_{q(x)} \tag{11}$$

$q(x)$ maps input x to a leaf index,

w is the leaf weight.

This equation defines how a single decision tree makes predictions. The function $\mathrm{q}(x)$ represents the tree structure that routes input features through decision nodes to reach a specific leaf. Each leaf contains a weight(w) that represents the predicted value for inputs falling into that leaf. In volatility forecasting, this means that similar market conditions (similar feature values) will be routed to the same leaf and receive the same volatility prediction, ensuring consistency in the model's output.

Gradient and Hessian:

$$g_i = \partial\hat{y}_i^{(t-1)} l\left(y_i, \hat{y}_i^{(t-1)}\right) \tag{12}$$

$$h_i = \partial^{2\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \tag{13}$$

These are the first and second derivatives of the loss function with respect to the current predictions. The gradient (g_i) indicates the direction and magnitude of the prediction error, while the Hessian (h_i) measures the curvature of the loss function. XGBoost uses these derivatives to determine the optimal structure and weights for each new tree. In practice, this allows the algorithm to focus on data points where the current ensemble is making the largest errors, enabling targeted improvements in volatility prediction accuracy.

Optimal Leaf Weight:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{12}$$

This equation calculates the optimal weight for each leaf in a new tree. The numerator sums the gradients of all data points that fall into leaf (j), while the denominator includes the sum of Hessians plus the L2 regularization parameter λ. This formula ensures the leaf weight optimally reduces the overall objective function. In volatility forecasting, this means that each leaf will predict a volatility value that best represents the true volatility of all market conditions that are routed to that leaf, while accounting for the regularization penalty to prevent overfitting.

Appendix E – Random Forest

Bootstrap Aggregation:

$$\hat{y}(x) = \left(\frac{1}{B}\right) \sum_{b=1}^{B} f_b(x) \tag{21}$$

B is the number of trees,

$f_b(x)$ is the prediction of the b-th tree.

This equation implements the ensemble prediction by averaging the outputs of all decision trees in the forest. The bootstrap sampling ensures each tree is trained on a different subset of the data, creating diversity in the ensemble. In financial forecasting, this equation is applied during the prediction phase to combine the forecasts from multiple trees, reducing variance and improving prediction stability. The averaging process is particularly valuable for financial data because it mitigates the impact of individual tree errors and reduces overfitting to specific market conditions. This ensemble approach is crucial for options pricing and risk management where prediction stability is more important than individual tree accuracy.

Gini Impurity:

$$\mathrm{Gini}(D) = 1 - \sum_{i=1}^{c} p_i^2 \tag{22}$$

p_i is the proportion of samples belonging to class i.

This equation measures the impurity or disorder of a dataset with respect to class labels. It is used during the tree construction phase to determine the optimal split points for each node. The Gini impurity ranges from 0 (pure node with only one class) to 1-1/c (maximum impurity with equal class distribution). In financial forecasting, this equation is applied during the training phase when building each decision tree. It helps identify the most informative features and split thresholds that best separate upward and downward price movements. For example, it might find that RSI values above 70 are highly predictive of downward price movements, creating a split that maximizes the purity of the resulting child nodes. This feature selection process is crucial for identifying the most relevant technical indicators for price direction prediction.

Information Gain:

$$\mathrm{IG}(D, A) = \mathrm{Gini}(D) - \sum_{v \in Values(A)} (|D_v|/|D|)Gini(D_v) \tag{23}$$

A is the feature,

D_v is the subset of samples where feature A has value v.

This equation quantifies how much a feature reduces the impurity of the dataset when used as a split criterion. It measures the difference between the parent node's impurity and the weighted average impurity of the child nodes. In financial forecasting, this equation is applied during the tree construction phase to select the best feature for splitting at each node. For example, it might determine that splitting on volatility levels provides more information gain than splitting on price levels, leading to more accurate predictions. The feature with the highest information gain is chosen for the split, ensuring that each decision tree focuses on the most predictive features. This process is repeated recursively until the tree reaches a stopping criterion, creating a hierarchical decision structure that captures complex interactions between multiple market indicators.

Appendix F – SVM foundation equations

Primal Formulation:

$$\min_{w,b}\left(\frac{1}{2}\right)\left||w|\right|^2 + \mathrm{C}\sum_{i=1}^{n}\xi_i \tag{15}$$

$$subject\ to: y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \tag{16}$$

This is the core optimization problem that SVM solves during training. The first term ($\left||w|\right|^2$) minimizes the margin width to find the optimal hyperplane, while the second term ($\mathrm{C}\sum_{i=1}^{n}\xi_i$) allows for some misclassifications through slack variables. The constraint ensures that most data points are correctly classified with a margin of at least 1. In financial forecasting, this equation is applied during model training to find the optimal decision boundary that separates upward and downward price movements based on technical indicators and market features. The regularization parameter C controls the trade-off between margin maximization and classification accuracy, which is crucial for preventing overfitting to historical market patterns.

Dual Formulation:

$$\max_{\alpha\sum_{i=1}^{n}\alpha_i} - (1/2)\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K\left(x_i, x_j\right) \tag{17}$$

$$\text{subjectto}: 0 \leq \alpha_i \leq C, \sum_{i=1}^{n}\alpha_i y_i = 0 \tag{18}$$

α_i are Lagrange multipliers

$\mathrm{K}\left(x_i, x_j\right)$ is the kernel function.

This dual formulation is computationally more efficient than the primal form and enables the use of kernel functions for non-linear classification. The Lagrange multipliers α_i determine which training points become support vectors (the critical points that define the decision boundary). The kernel function $\mathrm{K}\left(x_i, x_j\right)$ allows SVM to operate in higher-dimensional feature spaces without explicitly computing the transformation. In financial applications, this equation is solved during the training phase to identify the most informative market conditions (support vectors) that define the boundary between bullish and bearish market states. The kernel trick is particularly valuable for capturing complex non-linear relationships between technical indicators and price movements.

Kernel Function (RBF):

$$\mathrm{K}(x_i, x_j) = \exp\left(-\gamma \left\|x_i - x_j\right\|^2\right) \tag{19}$$

γ controls the influence of each training example.

The Radial Basis Function (RBF) kernel measures the similarity between two data points based on their Euclidean distance. The parameter γ determines the radius of influence of each support vector - smaller γ values create broader, smoother decision boundaries, while larger γ values create more complex, localized boundaries. In financial forecasting, this kernel is applied during both training and prediction phases. During training, it helps identify market conditions that are similar to historical patterns. During prediction, it computes the similarity between current market features and historical support vectors to determine the price direction. The RBF kernel is particularly effective for financial data because it can capture complex, non-linear relationships between multiple technical indicators and price movements without requiring explicit feature engineering.

Decision Function:

$$\mathrm{f}(x) = \mathrm{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right) \tag{20}$$

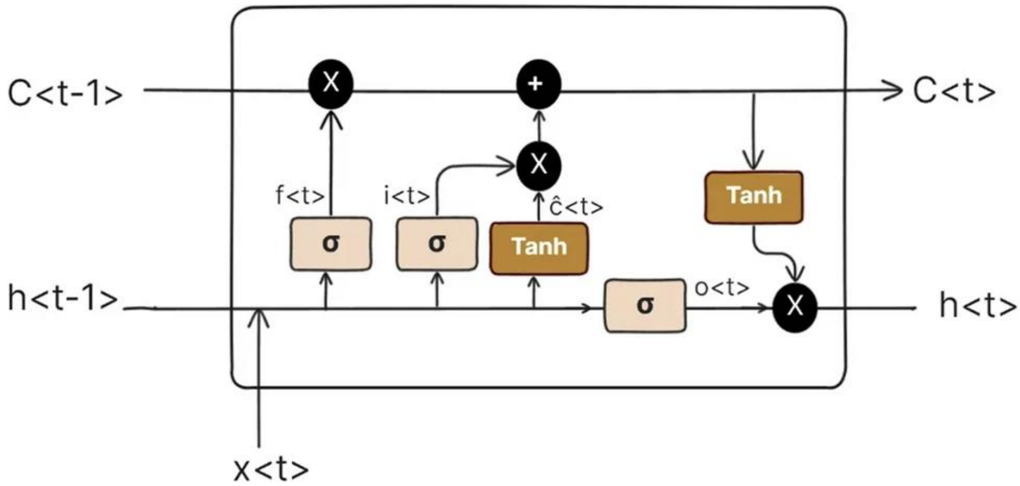
This is the final prediction equation used during the inference phase. It combines the contributions of all support vectors weighted by their Lagrange multipliers to make a classification decision for new data points. The sign function outputs +1 for upward price movement predictions and -1 for downward movement predictions. In financial forecasting, this equation is applied in real-time to classify current market conditions based on their similarity to historical support vectors. The decision boundary is determined by the weighted sum of kernel similarities to all support vectors, making SVM robust to noise and outliers in financial data. This function is particularly valuable for options trading strategies where binary directional predictions are needed for hedging decisions.

Appendix F – Fundamental LSTM Equations and Concepts

The LSTM architecture addresses the vanishing gradient problem in traditional RNNs through its sophisticated gating mechanism. The core LSTM cell operates through the following fundamental equations:

LSTM Architecture Visualization:

Figure 11- Schematic diagram of the internal architecture of an LSTM cell, illustrating the flow of information through the forget, input, and output gates, as well as the cell state and hidden state updates. This visualization highlights how LSTM networks manage (Alsalem, 2025)



The internal architecture of an LSTM cell is clearly illustrated in the diagram above, which shows the flow of information and the gating mechanisms that enable effective sequential data processing. The LSTM cell receives three main inputs: the previous cell state (C_{t-1}), the previous hidden state (h_{t-1}), and the current input (x_t). These are processed through a series of gates and nonlinear activations to produce the updated cell state (C_t) and hidden state (h_t).

Step-by-step explanation of the LSTM cell (as shown in the diagram):

Forget Gate (f_t):

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (1)$$

The forget gate determines what information to discard from the cell state.

The previous hidden state (h_{t-1}) and the current input (x_t) are concatenated and passed through a sigmoid (σ) activation function.

The output, f_t, is a vector of values between 0 and 1, which determines how much of the previous cell state (C_{t-1}) should be retained. This is represented by the multiplication (X) with C_{t-1} in the diagram.

Input Gate (i_t **) and**

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + bi) \quad (2)$$

Where σ represents the sigmoid activation function,

W_i are learnable weights,

h_{t-1} is the previous hidden state,

x_t is the current input, and

bi is the bias term.

Candidate Cell State ($\tilde{C}_t$):

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

Where ⊙ represents element-wise multiplication,

$\tilde{C}_t$ is the candidate cell state

The same concatenated input is passed through another sigmoid (σ) activation to produce the input gate (i_t), which decides which values will be updated.

Simultaneously, the concatenated input is passed through a tanh activation to generate the candidate cell state ($\tilde{C}_t$), with values between -1 and 1.

The input gate (i_t) and candidate cell state ($\tilde{C}_t$) are multiplied (⊙) to determine the new information to be added to the cell state.

Cell State Update (C_t):

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$

C_t summing the element-wise product of the forget gate and previous cell state ($f_t \odot C_{t-1}$) with the product of the input gate and candidate cell state ($i_t \odot \tilde{C}_t$), as indicated by the "+" in the diagram.

Output Gate (o_t) **and Hidden Status(** h_t):

$$o\_t = \sigma(W\_o \cdot [h_{t-1}, x_t] + b_o \tag{5}$$

$$h_t = o_t \odot tanh(C_t) \tag{6}$$

The output gate controls what information from the cell state is output to the hidden state.

Which is generated by passing the concatenated input through another sigmoid (σ) activation.

The new cell state (C_t) is passed through a tanh activation and then multiplied by the output gate (o_t) to produce the new hidden state (h_t), as shown by the final multiplication (X) in the diagram.

This visual and stepwise process allows the LSTM cell to selectively retain, update, and output information, making it highly effective for modelling complex temporal dependencies in financial time series.

This architecture is particularly effective for financial time series analysis, as demonstrated by Alsalem(2025), who achieved precision scores of 0.99 and F1-scores of 0.99 using LSTM-enhanced feature extraction combined with ensemble classifiers. The ability to maintain long-term dependencies through the cell state while selectively updating information through the gating mechanism makes LSTM networks ideal for capturing complex temporal patterns in financial data.

Loss Function:

$$L = -\sum_{i=1}^{N} y_i log(\hat{y}_i) \tag{7}$$

In this equation, (y_i) represents the true class label for the price direction (e.g., 1 for "up", 0 for "down"), $(\hat{y}_i)$is the predicted probability output by the LSTM model for the "up" direction, and (N) is the total number of samples in the dataset. This loss function is known as the binary cross-entropy (or log loss), which measures the difference between the true labels and the predicted probabilities. It penalizes confident but incorrect predictions more heavily, encouraging the model to output probabilities that are well-calibrated to the actual outcomes.

The binary cross-entropy loss is applied during the training phase of the LSTM model. At each iteration (epoch), the model computes the predicted probabilities$(\hat{y}_i)$ for all samples in the training batch, then calculates the loss (L) by comparing these predictions to the true labels (y_i). The model parameters are then updated via backpropagation to minimize this loss, thereby improving the model's ability to correctly predict the price direction in future data.

Appendix G – Year-by-Year Sequential Analysis

Year	Observations	Variables	ADF Stationary	KPSS Stationary	JB Normal
2019	251	102	64/102 (62.7%)	42/102 (41.2%)	14/102 (13.7%)
2021	252	102	63/102 (61.8%)	30/102 (29.4%)	8/102 (7.8%)
2022	251	102	54/102 (52.9%)	39/102 (38.2%)	8/102 (7.8%)
2023	250	102	46/102 (45.1%)	36/102 (35.3%)	17/102 (16.7%)
2024	140	102	42/102 (41.2%)	38/102 (37.3%)	35/102 (34.3%)

2018	250	102	31/102 (30.4%)	13/102 (12.7%)	17/102 (16.7%)
2020	252	102	25/102 (24.5%)	30/102 (29.4%)	2/102 (2.0%)

Appendix H - Most Stationary Variables by Year (Sequential Analysis)

2018 - Top 5 Most Stationary:

1. return - ADF=-6.8588, p=0.0000
2. call_iv_std - ADF=-12.7434, p=0.0000
3. call_iv_mean - ADF=-6.0173, p=0.0000
4. call_bid_ask_ratio_median - ADF=-9.6730, p=0.0000
5. all_bid_ask_ratio_median - ADF=-7.8802, p=0.0000

2019 - Top 5 Most Stationary:

1. call_volume_mean - ADF=-12.0499, p=0.0000
2. all_iv_mean - ADF=-13.6823, p=0.0000
3. all_bid_ask_spread_median - ADF=-6.4531, p=0.0000
4. all_bid_ask_spread_std - ADF=-7.7163, p=0.0000
5. call_volume_sum - ADF=-12.1511, p=0.0000

2020 - Top 5 Most Stationary:

1. put_iv_std - ADF=-6.2189, p=0.0000
2. all_iv_std - ADF=-6.6060, p=0.0000
3. call_volume_mean - ADF=-6.8562, p=0.0000
4. iv_spread - ADF=-15.3009, p=0.0000
5. call_iv_std - ADF=-14.2765, p=0.0000

2021 - Top 5 Most Stationary:

1. iv_skew - ADF=-7.5085, p=0.0000
2. all_iv_std - ADF=-7.9717, p=0.0000
3. iv_spread - ADF=-14.1806, p=0.0000
4. TEMA_10_cross_TEMA_50 - ADF=-5.5825, p=0.0000
5. call_iv_mean - ADF=-7.9632, p=0.0000

2022 - Top 5 Most Stationary:

1. volume - ADF=-5.9439, p=0.0000
2. put_bid_ask_spread_std - ADF=-5.5758, p=0.0000
3. put_bid_ask_spread_median - ADF=-6.4311, p=0.0000
4. put_bid_ask_spread_mean - ADF=-5.6983, p=0.0000

5. put_iv_std - ADF=-7.3156, p=0.0000

2023 - Top 5 Most Stationary:

1. put_iv_mean - ADF=-5.9528, p=0.0000
2. return - ADF=-15.3499, p=0.0000
3. log_return - ADF=-15.3430, p=0.0000
4. iv_spread - ADF=-10.0717, p=0.0000
5. call_iv_mean - ADF=-9.4976, p=0.0000

2024 - Top 5 Most Stationary:

1. volume - ADF=-10.0717, p=0.0000
2. put_volume_sum - ADF=-6.7350, p=0.0000
3. put_bid_ask_spread_std - ADF=-8.3818, p=0.0000
4. put_bid_ask_spread_mean - ADF=-8.2509, p=0.0000
5. call_volume_mean - ADF=-5.7978, p=0.0000

REFERENCE

Alsalem, K., 2025. Leveraging LSTM and ensemble classifiers for enhanced food waste classification. Discov Sustain 6, 443. https://doi.org/10.1007/s43621-025-01330-6

Arin, E., Ozbayoglu, A.M., 2022. Deep Learning Based Hybrid Computational Intelligence Models for Options Pricing. Compute Econ 59, 39–58. https://doi.org/10.1007/s10614-020-10063-9

Arsenault, P.-D., Wang, S., Patenande, J.-M., 2025. A Survey of Explainable Artificial Intelligence (XAI) in Financial Time Series Forecasting. ACM Comput. Surv. 57, 1–37. https://doi.org/10.1145/3729531

Bailey, D.H., Borwein, J., Lopez de Prado, M., Zhu, Q.J., 2015. The Probability of Backtest Overfitting. https://doi.org/10.2139/ssrn.2326253

Barbierato, E., Gatti, A., 2024. The Challenges of Machine Learning: A Critical Review. Electronics 13, 416. https://doi.org/10.3390/electronics13020416

Bellora, F.G., Mazzei, G., Maurette, M., 2021. Option Pricing Model with Transaction Costs. https://doi.org/10.48550/arXiv.2112.10209

Breiman, L., 2001. Random Forests. Machine Learning 45, 5–32. https://doi.org/10.1023/A:1010933404324

Carr, P., Ellis, K., Gupta, V., 1998. Static Hedging of Exotic Options. The Journal of Finance 53, 1165–1190. https://doi.org/10.1111/0022-1082.00048

Chen, J., Li, L., 2021. Data-driven Hedging of Stock Index Options via Deep Learning. https://doi.org/10.48550/arXiv.2111.03477

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794. https://doi.org/10.1145/2939672.2939785

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach Learn 20, 273–297. https://doi.org/10.1007/BF00994018

Deng, Y., Bao, F., Kong, Y., Ren, Z., Dai, Q., 2017. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. IEEE Trans. Neural

Netw. Learning Syst. 28, 653–664. https://doi.org/10.1109/TNNLS.2016.2522401

Ding, Y., Yuan, G., Zuo, D., Gao, T., 2025. Hedging with Sparse Reward Reinforcement Learning. https://doi.org/10.48550/arXiv.2503.04218

Du, J., Jin, M., Kolm, P.N., Ritter, G., Wang, Y., Zhang, B., 2020. Deep Reinforcement Learning for Option Replication and Hedging. JFDS 2, 44–57. https://doi.org/10.3905/jfds.2020.1.045

Engelmann, B., Fengler, M.R., Nalholm, M., Schwendner, P., 2006a. Static versus dynamic hedges: an empirical comparison for barrier options. Rev Deriv Res 9, 239–264. https://doi.org/10.1007/s11147-007-9010-x

Engelmann, B., Fengler, M.R., Nalholm, M., Schwendner, P., 2006b. Static versus dynamic hedges: an empirical comparison for barrier options. Rev Deriv Res 9, 239–264. https://doi.org/10.1007/s11147-007-9010-x

Fischer, T., Krauss, C., 2018. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research 270, 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

Gao, K., Weston, S., Vytelingum, P., Stillman, N., Luk, W., Guo, C., 2023. Deeper Hedging: A New Agent-based Model for Effective Deep Hedging, in: 4th ACM International Conference on AI in Finance. Presented at the ICAIF ’23: 4th ACM International Conference on AI in Finance, ACM, Brooklyn NY USA, pp. 270–278. https://doi.org/10.1145/3604237.3626913

Genuer, R., Poggi, J.-M., Tuleau-Malot, C., 2010. Variable selection using Random Forests. Pattern Recognition Letters 31, 2225–2236.

Gifty, A., Li, Y., 2024. A Comparative Analysis of LSTM, ARIMA, XGBoost Algorithms in Predicting Stock Price Direction. ETJ 09. https://doi.org/10.47191/etj/v9i08.50

Gross, E., Kruger, R., Toerien, F., 2025. A comparative analysis of option pricing models: Black–Scholes, Bachelier, and artificial neural networks. Risk Manag 27, 8. https://doi.org/10.1057/s41283-025-00160-0

Gu, S., Kelly, B., Xiu, D., 2020. Empirical Asset Pricing via Machine Learning. Rev Financ Stud 33, 2223–2273. https://doi.org/10.1093/rfs/hhaa009

Hastie, T., Tibshirani, R., Friedman, J., 2009. Model Inference and Averaging, in: The Elements of Statistical Learning, Springer Series in Statistics. Springer New York, New York, NY, pp. 261–294. https://doi.org/10.1007/978-0-387-84858-7_8

Holt, C.A., Sullivan, S.P., 2023. Permutation tests for experimental data. Exp Econ 26, 775–812. https://doi.org/10.1007/s10683-023-09799-6

Huang, W., Nakamori, Y., Wang, S.-Y., 2005. Forecasting stock market movement direction with support vector machine. Computers & Operations Research, Applications of Neural Networks 32, 2513–2522. https://doi.org/10.1016/j.cor.2004.03.016

Hyndman, R.J., Athanasopoulos, G., 2021. Forecasting: principles and practice, Third print edition. ed. Otexts, Online Open-Access Textbooks, Melbourne, Australia.

Janková, Z., 2018. Drawbacks and Limitations of Black-Scholes Model for Options Pricing. JFSR 2018, 1–7. https://doi.org/10.5171/2018.79814

Jiang, Z., Xu, D., Liang, J., 2017. A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. https://doi.org/10.48550/arXiv.1706.10059

Lakshminarayanan, B., Pritzel, A., Blundell, C., 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. https://doi.org/10.48550/arXiv.1612.01474

Lim, B., Zohren, S., 2021. Time Series Forecasting With Deep Learning: A Survey. Phil. Trans. R. Soc. A. 379, 20200209. https://doi.org/10.1098/rsta.2020.0209

López de Prado, M.M., 2018. Advances in financial machine learning. Wiley, Hoboken, New Jersey.

Lundberg, S., Lee, S.-I., 2017. A Unified Approach to Interpreting Model Predictions. https://doi.org/10.48550/arXiv.1705.07874

Marzban, S., Delage, E., Li, J.Y.-M., 2023. Deep reinforcement learning for option pricing and hedging under dynamic expectile risk measures. Quantitative Finance 23, 1411–1430. https://doi.org/10.1080/14697688.2023.2244531

Mehtarizadeh, H., Mansouri, N., Mohammad Hasani Zade, B., Hosseini, M.M., 2025. Stock price prediction with SCA-LSTM network and Statistical model ARIMA-GARCH. J Supercomput 81, 366. https://doi.org/10.1007/s11227-024-06775-6

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T.P., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous Methods for Deep Reinforcement Learning. https://doi.org/10.48550/arXiv.1602.01783

Murray, P., Wood, B., Buehler, H., Wiese, M., Pakkanen, M., 2022. Deep Hedging: Continuous Reinforcement Learning for Hedging of General Portfolios across Multiple Risk Aversions, in: Proceedings of the Third ACM International Conference on AI in Finance. Presented at the ICAIF '22: 3rd ACM International Conference on AI in Finance, ACM, New York NY USA, pp. 361–368. https://doi.org/10.1145/3533271.3561731

Rojas, I., Pomares, H., Valenzuela, O., Rojas, F., Herrera, L., Kaufman, P. (Eds.), 2022. The 8th International Conference on Time Series and Forecasting. MDPI - Multidisciplinary Digital Publishing Institute, Basel.

Salami, M.F., 2024. Empirical examination of the Black–Scholes model: evidence from the United States stock market. Front. Appl. Math. Stat. 10. https://doi.org/10.3389/fams.2024.1216386

Sinha, A., Poonolly, J.M., Gayathiri, A., Janarthanan, D., 2018. Analysis of Option Prices Using Black Scholes Model 5.

Thomas, P.S., Brunskill, E., 2016. Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. https://doi.org/10.48550/arXiv.1604.00923

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention Is All You Need. https://doi.org/10.48550/arXiv.1706.03762

Wang, Y., Tong, L., Zhao, Y., 2024. Revolutionizing Hedge Fund Risk Management: The Power of Deep Learning and LSTM in Hedging Illiquid Assets. JRFM 17, 224. https://doi.org/10.3390/jrfm17060224

Wasserbacher, H., Spindler, M., 2022. Machine learning for financial forecasting, planning and analysis: recent developments and pitfalls. Digit Finance 4, 63–88. https://doi.org/10.1007/s42521-021-00046-2

Zhang, Z., Zohren, S., Roberts, S., 2019. Deep Reinforcement Learning for Trading. https://doi.org/10.48550/arXiv.1911.10107