الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
YAHIA FARES UNIVERSITY OF MEDEA



FACULTY OF SCIENCES

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCES

Graduation Project Report Submitted for the Degree of

Master in Computer Science

Option: Applied Artificial Intelligence

Application of Artificial Intelligence Algorithms in Gain Optimization Problems

Proposed and Directed by

Dr. Mohamed Yassine Hayi

Prepared and Presented by

Mechentel Abdelghani
Yahiaoui Achouak

Defended on June 2026, in front of the jury composed of:

Dr. Tibrkek Allal: Yahia Fares University – President
Dr. Babali Baligh: Yahia Fares University – Examiner

Academic Year: 2025 / 2026

ACKNOWLEDGEMENTS

First and foremost, we thank « **ALLAH** », for granting us the faith, patience, courage, health, and all the means necessary throughout our years of study, and for allowing us to complete this humble work.

We would like to express our deepest gratitude to our **parents**, who stood by us throughout all these years with their endless patience, care, and sacrifices. Their unconditional support and encouragement have been our greatest strength and the foundation of everything we have achieved.

We would also like to extend our sincere thanks to our supervisor, **Dr. Mohamed Yassine Hayi**, for his guidance, availability, and valuable advice throughout the realization of this project.

Finally, we would like to thank every person who tried to help us and gave from their time and effort throughout all our years of study, whether near or far, their support did not go unnoticed and will never be forgotten.

Thank you.....

Abstract

The cryptocurrency market is volatile, presenting both a unique challenge and an opportunity for algorithmic traders. This study evaluates six algorithmic trading models — Deep Q-Network (DQN), Proximal Policy Optimization (PPO), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Random Forest (RF), and XGBoost — across five assets (Bitcoin, Ethereum, Cardano, Solana, and Binance Coin) and five investment horizons (10 days, 1 month, 3 months, 6 months, and 12 months), using 2024 out-of-sample market data. The proposed framework addresses two key limitations in prior work: (1) reinforcement learning agents are restricted to a fixed three-action space without position sizing, and (2) the absence of multi-horizon evaluation frameworks.

The models are evaluated on return, Sharpe ratio, and maximum drawdown. Key quantitative findings include: DQN achieves +36.59% on SOL at 1 month and +49.52% on BNB at 3 months, outperforming all other models by a large margin in trending markets. PPO delivers the highest single-asset return of +74.60% on ADA at 12 months, demonstrating its ability to learn when to stay out of bearish markets. RF records the tightest drawdown control, with a maximum drawdown of only –2.05% on BNB at 1 month, validating its capital preservation capability. No single model dominates across all assets and horizons; the optimal choice depends on the interplay between market regime, asset behaviour, and investment horizon. All six models are deployed in Trademind IA, a real-time AI trading dashboard available on mobile and desktop.

This work opens a clear path toward a dynamic, regime-aware model selection framework that automatically assigns the best model to the current market condition. Future work should extend the evaluation to multiple market cycles and incorporate asset-specific on-chain features.

Résumé

Le marché des cryptomonnaies est volatil, ce qui représente à la fois un défi unique et une opportunité pour les traders algorithmiques. Cette étude évalue six modèles de trading algorithmique — Deep Q-Network (DQN), Proximal Policy Optimization (PPO), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Random Forest (RF) et XGBoost — sur cinq actifs (Bitcoin, Ethereum, Cardano, Solana et Binance Coin) et cinq horizons d'investissement (10 jours, 1 mois, 3 mois, 6 mois et 12 mois), en utilisant des données de marché hors échantillon pour l'année 2024. Le cadre proposé répond à deux limitations majeures des travaux antérieurs : (1) les agents d'apprentissage par renforcement sont restreints à un espace d'action fixe à trois actions sans gestion de la taille des positions, et (2) l'absence de cadres d'évaluation multi-horizons.

Les modèles sont évalués selon le rendement, le ratio de Sharpe et le drawdown maximal. Les principaux résultats quantitatifs sont les suivants : DQN atteint +36,59% sur SOL à 1 mois et +49,52% sur BNB à 3 mois, surpassant tous les autres modèles avec une large avance dans les marchés en tendance. PPO enregistre le rendement individuel le plus élevé sur un actif, avec +74,60% sur ADA à 12 mois, démontrant sa capacité à apprendre quand rester en dehors du marché en période baissière. RF affiche le meilleur contrôle du drawdown, avec un drawdown maximal de seulement –2,05% sur BNB à 1 mois, validant sa capacité de préservation du capital. Aucun modèle unique ne domine sur l'ensemble des actifs et des horizons ; le choix optimal dépend de l'interaction entre le régime de marché, le comportement de l'actif et l'horizon d'investissement. Les six modèles sont déployés dans Trademind IA, un tableau de bord de trading en temps réel basé sur l'intelligence artificielle, disponible sur mobile et ordinateur.

Ce travail ouvre une voie claire vers un cadre dynamique de sélection de modèles tenant compte du régime de marché, qui attribue automatiquement le meilleur modèle à la condition de marché actuelle. Les travaux futurs devraient étendre l'évaluation à plusieurs cycles de marché et intégrer des caractéristiques on-chain spécifiques à chaque actif.

ملخص

تتسم سوق العملات المشفرة بتقلبات حادة، مما يشكّل تحديًا وفرصة فريدة للمتداولين الخوارزميين. تقيّم هذه الدراسة ستة نماذج للتداول الخوارزمي — شبكة Q العميقة (DQN)، وتحسين السياسة القريبة (PPO)، والذاكرة طويلة قصيرة المدى (LSTM)، والوحدة التكرارية ذات البوابات (GRU)، والغابة العشوائية (RF)، وXGBoost — عبر خمسة أصول (بيتكوين، وإيثيريوم، وكاردانو، وسولانا، وعملة بينانس)، وخمسة آفاق استثمارية (10 أيام، وشهر واحد، وثلاثة أشهر، وستة أشهر، و 12 شهراً)، باستخدام بيانات السوق خارج العينة لعام 2024. يعالج الإطار المقترح قيدين رئيسيين في الأعمال السابقة: (1) اقتصار وكلاء التعلم المعزز على فضاء إجراءات ثلاثي ثابت دون تحديد حجم المركز، و(2) غياب أطر التقييم متعددة الآفاق.

تُقيَّم النماذج وفق العائد، ونسبة شارب، والحد الأقصى للسحب. تشمل النتائج الكمية الرئيسية: يحقق DQN عائدًا قدره +36.59% على SOL خلال شهر واحد، و +49.52% على BNB خلال ثلاثة أشهر، متفوقًا على جميع النماذج الأخرى بفارق كبير في الأسواق ذات الاتجاه الواضح. يحقق PPO أعلى عائد فردي بلغ +74.60% على ADA خلال 12 شهراً، مما يُظهر قدرته على التعلّم متى يجب الابتعاد عن الأسواق الهابطة. يحقق RF أفضل تحكم في السحب، بحد أقصى للسحب لا يتجاوز -02.05% على BNB خلال شهر واحد، مما يؤكد قدرته على حفظ رأس المال. لا يتفوق نموذج واحد على البقية عبر جميع الأصول والآفاق الزمنية؛ فالخيار الأمثل يعتمد على التفاعل بين نظام السوق، وسلوك الأصل، والأفق الاستثماري. تُطبَّق جميع النماذج الستة على منصة Trademind IA، وهي لوحة تداول ذكاء اصطناعي تعمل في الوقت الفعلي ومتاحة على الهواتف المحمولة وأجهزة الحاسوب.

يفتح هذا العمل طريقًا واضحًا نحو إطار ديناميكي واعٍ بنظام السوق يختار تلقائيًا النموذج الأنسب للظروف السوقية الحالية. ينبغي للأعمال المستقبلية توسيع نطاق التقييم ليشمل دورات سوقية متعددة، ودمج الميزات الخاصة بالأصل على السلسلة (on-chain).

Table of Contents

List of Figures 4
List of Tables 6
List of Abbreviations 7

General Introduction **8**

Chapter 1: Algorithmic Trading **12**
1.1 Introduction 12
1.2 Related Concepts 12
1.2.1 Trading 12
1.2.2 Candlesticks 12
1.2.3 Bullish and Bearish Candlesticks 14
1.2.4 Position 14
1.2.5 Stop Loss and Take Profit 14
1.2.6 Blockchain and Cryptocurrency Mining 14
1.2.7 Volatility 15
1.2.8 Cryptocurrencies 15
1.2.8.1 Bitcoin (BTC) 15
1.2.8.2 Ethereum (ETH) 15
1.2.8.3 Binance Coin (BNB) 15
1.2.8.4 Cardano (ADA) 15
1.2.8.5 Solana (SOL) 16
1.3 Technical Indicators 16
1.3.1 Trend Indicators 16
1.3.1.1 Simple Moving Average (SMA) 16
1.3.1.2 Exponential Moving Average (EMA) 16
1.3.1.3 The Moving Average Convergence Divergence (MACD) 17
1.3.2 Momentum Indicators 17
1.3.2.1 Momentum (MOM) 17
1.3.2.2 Rate of Change (ROC) 17
1.3.2.3 Relative Strength Index (RSI) 17
1.3.3 Volatility Indicators 18
1.3.3.1 Bollinger Bands 18
1.3.3.2 Average True Range (ATR) 18
1.3.3.3 Rolling Volatility 18
1.3.4 Volume Indicators 18
1.4 Trading Strategies 19
1.4.1 Buy and Hold 19
1.4.2 Trend Following 19
1.4.3 Mean Reversion 19
1.4.4 Momentum Trading 19
1.5 Artificial Intelligence in Financial Trading 19
1.6 Conclusion 20

Chapter 2: State of Art **22**
2.1 Introduction . . . 22
2.1.1 Selection Methodology . . . 22
2.2 Smart Trading Using Technical Analysis on the Crypto Market . . . 22
2.2.1 Methodology . . . 22
2.2.2 Results and Findings . . . 23
2.2.3 Relation to Our Work . . . 23
2.3 Deep Reinforcement Learning in Cryptocurrency Trading . . . 23
2.3.1 Methodology . . . 23
2.3.2 Results and Findings . . . 23
2.3.3 Relation to Our Work . . . 24
2.4 Reinforcement Learning Pair Trading . . . 24
2.4.1 Methodology . . . 24
2.4.2 Results and Findings . . . 24
2.4.3 Relation to Our Work . . . 25
2.5 Forecasting Cryptocurrency Prices Using Deep Learning Approach . . . 25
2.5.1 Methodology . . . 25
2.5.2 Results and Findings . . . 26
2.5.3 Relation to Our Work . . . 26
2.6 Machine Learning Approaches to Trading Optimization . . . 26
2.6.1 Methodology . . . 26
2.6.2 Results and Findings . . . 27
2.6.3 Relation to Our Work . . . 27
2.7 Summary of Related Works . . . 27
2.8 Conclusion . . . 29

Chapter 3: Proposed Approach and Methodology **31**
3.1 Introduction . . . 31
3.2 Dataset . . . 31
3.2.1 Data Sources . . . 31
3.2.2 Data Splitting Strategy . . . 32
3.2.3 Data Preprocessing . . . 33
3.2.3.1 Daily Data Cleaning . . . 33
3.2.3.2 Hourly Data Cleaning . . . 33
3.2.3.3 Missing Values and Feature Scaling . . . 34
3.3 Model Architectures . . . 34
3.3.1 Deep Q-Network (DQN) . . . 35
3.3.2 Action Space . . . 35
3.3.3 Observation Space . . . 35
3.3.4 Reward Function . . . 36
3.3.5 Architecture . . . 37
3.3.6 Hyperparameters . . . 38
3.4 Proximal Policy Optimization (PPO) . . . 38
3.4.1 Action Space . . . 38
3.4.2 Observation Space . . . 39
3.4.3 Reward Function . . . 39
3.4.4 Hyperparameters . . . 41
3.5 Long Short-Term Memory (LSTM) . . . 41
3.5.1 Label Construction . . . 42
3.5.2 Loss Function . . . 42
3.5.3 Buy Threshold . . . 42
3.5.4 Exit Strategy . . . 42
3.5.5 Position Sizing . . . 43
3.5.6 Architecture . . . 43
3.5.7 Hyperparameters . . . 44
3.6 Gated Recurrent Unit (GRU) . . . 45
3.6.1 Architecture . . . 45
3.7 Random Forest . . . 46

3.7.1 Label Construction . . . 47
3.7.2 Class Imbalance . . . 47
3.7.3 Exit Strategy . . . 47
3.7.4 Hyperparameters . . . 47
3.8 XGBoost . . . 47
3.8.1 Class Imbalance . . . 47
3.8.2 Exit Strategy . . . 48
3.8.3 Buy Threshold . . . 48
3.8.4 Hyperparameters . . . 48
3.9 Evaluation Framework . . . 48
3.9.1 Test Periods . . . 48
3.9.2 Evaluation Metrics . . . 49
3.9.3 Benchmark . . . 49
3.10 Conclusion . . . 49

Chapter 4: Results and Evaluation **51**
4.1 Introduction . . . 51
4.2 Experimental Setup . . . 51
4.2.2.1 Development Environment . . . 51
4.2.2.2 Libraries and Frameworks . . . 51
4.3 Data Visualization . . . 52
4.3.3.1 Historical Price Evolution Plot . . . 52
4.3.3.2 Price evolution in 2024 Plot . . . 52
4.3.3.3 Technical Indicators in 2024 . . . 53
4.3.3.4 Rolling 30 Day Annualized Volatility . . . 54
4.3.3.5 Maximum Drawdown by Quarter . . . 55
4.4 Experimental Results and Comparative Evaluation . . . 56
4.4.4.1 per Asset Comparative Analysis . . . 56
4.4.1.1 Bitcoin (BTC) . . . 56
4.4.1.2 Ethereum (ETH) . . . 58
4.4.1.3 Cardano (ADA) . . . 61
4.4.1.4 Solana (SOL) . . . 64
4.4.1.5 Binance Coin (BNB) . . . 68
4.4.4.2 Cross Asset Comparative Analysis . . . 72
4.4.2.1 Best Model per Horizon . . . 73
4.4.2.2 Average Returns Across All Assets . . . 73
4.5 Discussion . . . 74
4.5.5.1 Dynamic Position Sizing Enhances DQN's Short-to-Medium Return . . . 74
4.5.5.2 Reinforcement Learning Agents Outperform Supervised Models in Bearish Regimes . . . 74
4.5.5.3 Machine learning Excel at Capital Preservation . . . 75
4.5.5.4 Ethereum Is the Hardest Asset for Algorithmic Trading . . . 75
4.5.5.5 No Single Model Dominates Across All Assets and Horizons . . . 75
4.5.5.6 Implementation . . . 76
4.5.6.1 Signal Configuration Panel . . . 77
4.5.6.2 Performance Dashboard and Chart . . . 78
4.5.6.3 Trade Execution Panel . . . 79
4.5.5.7 Conclusion and Future Work . . . 80

Bibliography **81**

List of Figures

1.1 A Typical Candle . . . 13
1.2 5 minutes candlestick graph . . . 14
3.1 Hourly data collection script . . . 32
3.2 Daily data cleaning script . . . 33
3.3 Hourly Data Cleaning script . . . 34
3.4 XGBoost Feature scaling . . . 34
3.5 DQN feature normalization . . . 34
3.6 DQN normalization clipping . . . 34
3.7 Observation vector for the DQN agent . . . 36
3.8 Reward Function for the DQN agent . . . 37
3.9 DQN Architecture . . . 38
3.10 Observation vector for the PPO agent . . . 39
3.11 Reward Function for the PPO agent . . . 40
3.12 PPO agent actor and critic architecture . . . 41
3.13 LSTM Label Construction . . . 42
3.14 LSTM loss Function . . . 42
3.15 LSTM Exit Strategy . . . 43
3.16 LSTM Position Sizing . . . 43
3.17 LSTM architecture . . . 44
3.18 GRU architecture . . . 46
3.19 Random Forest Label Construction . . . 47
3.20 XGBoost Buy Threshold . . . 48
4.1 Historical Price evolution plot . . . 52
4.2 Price evolution in 2024 plot . . . 53
4.3 Technical indicators for BTC closing prices in 2024 . . . 54
4.4 Rolling 30 Day Annualized Volatility in 2024 . . . 55
4.5 Maximum Drawdown by Quarter in 2024 . . . 55
4.6 LSTM Performance vs Buy & Hold on BTC over 12 Months (2024) . . . 56
4.7 XGBoost Performance vs Buy & Hold on BTC over 12 Months (2024) . . . 57
4.8 DQN Performance vs Buy & Hold on BTC over 1 Months . . . 57
4.9 PPO Performance vs Buy & Hold on BTC over 10 days . . . 58
4.10 XGBoost Performance vs Buy & Hold on ETH over 10 days . . . 59
4.11 GRU Performance vs Buy & Hold on ETH over 6 month . . . 59
4.12 DQN Performance vs Buy & Hold on ETH over 3 month . . . 60
4.13 PPO Performance vs Buy & Hold on ETH over 12 month . . . 60
4.14 XGBoost Performance vs Buy & Hold on ADA over 12 month . . . 62
4.15 LSTM Performance vs Buy & Hold on ADA over 12 month . . . 62
4.16 DQN Performance vs Buy & Hold on ADA over 6 month . . . 63
4.17 PPO Performance vs Buy & Hold on ADA over 12 month . . . 64
4.18 RF Performance vs Buy & Hold on SOL over 12 month . . . 65
4.19 GRU Performance vs Buy & Hold on SOL over 6 month . . . 66
4.20 LSTM Performance vs Buy & Hold on SOL over 3 month . . . 66
4.21 DQN Performance vs Buy & Hold on SOL over 1 month . . . 67
4.22 PPO Performance vs Buy & Hold on SOL over 12 month . . . 67
4.23 XGBoost Performance vs Buy & Hold on BNB over 12 month . . . 69
4.24 RF Performance vs Buy & Hold on BNB over 1 month . . . 69
4.25 GRU Performance vs Buy and Hold on BNB over 1 month . . . 70
4.26 DQN Performance vs Buy & Hold on BNB over 3 month . . . 71

4.27 PPO Performance vs Buy & Hold on BNB over 12 month 72
4.28 Trademind IA landing page on mobile 77
4.29 Signal configuration panel on mobile 78
4.30 Main trading dashboard on mobile 79
4.31 Simulated trade execution panel on mobile 80

List of Tables

1.1 Key Characteristics of Major Cryptocurrencies 16
2.1 Comparative summary of related works on cryptocurrency trading and forecasting 28
3.1 Overview of Collected Data per Asset 32
3.2 DQN action space: discrete actions and their descriptions 35
3.3 Reward function components 36
3.4 Hyperparameters used for DQN training 38
3.5 Action space for the PPO agent 39
3.6 Reward function components 39
3.7 Hyperparameters used for PPO training 41
3.8 Sell conditions for closing a position 43
3.9 Training hyperparameters for the LSTM model 45
3.10 Random Forest hyperparameters 47
3.11 Hyperparameters for the gradient boosting model 48
3.12 Evaluation metrics used to assess model performance 49
4.1 Libraries and Descriptions 51
4.2 BTC Model Performance by Horizon 58
4.3 ETH Model Performance by Horizon 61
4.4 ADA Model Performance by Horizon 64
4.5 SOL Model Performance by Horizon 68
4.6 BNB Model Performance by Horizon 72
4.7 Best Performance by Horizon (across all assets) 73
4.8 Average Return per Model by Horizon (across all assets) 74
4.9 Recommended Models by Investment Horizon and Market Condition 76

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
RL	Reinforcement Learning
DQN	Deep Q-Network
PPO	Proximal Policy Optimization
A2C	Advantage Actor Critic
RPPO	Recurrent Proximal Policy Optimization
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
Bi-LSTM	Bi-Directional Long Short-Term Memory
TCNN	Temporal Convolutional Neural Network
RNN	Recurrent Neural Network
RF	Random Forest
XGBoost	Extreme Gradient Boosting
SVR	Support Vector Regression
PCA	Principal Component Analysis
B&H	Buy and Hold
DCA	Dollar Cost Averaging
OHLCV	Open, High, Low, Close, Volume
ROI	Return on Investment
MDD	Maximum Drawdown
PnL	Profit and Loss
SL	Stop Loss
TP	Take Profit
SMA	Simple Moving Average
EMA	Exponential Moving Average
MACD	Moving Average Convergence Divergence
RSI	Relative Strength Index
MOM	Momentum
ROC	Rate of Change
ATR	Average True Range
BB	Bollinger Bands
TR	True Range
BTC	Bitcoin
ETH	Ethereum
ADA	Cardano
SOL	Solana
BNB	Binance Coin
LTC	Litecoin
DOGE	DogeCoin
PoW	Proof of Work
PoS	Proof of Stake
PoH	Proof of History
PoSA	Proof of Staked Authority
DeFi	Decentralized Finance
NFT	Non-Fungible Token
DAO	Decentralized Autonomous Organization
API	Application Programming Interface
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
MAE	Mean Absolute Error

Continued on next page

Abbreviation	**Meaning**
R^2	R-squared (Coefficient of Determination)

General Introduction

Cryptocurrencies are becoming increasingly prevalent and experiencing exponential growth around the world. They are popular due to decentralization, security, anonymity, investment potential, and global accessibility. A 2023 survey from the University of Chicago (Weber et al., 2023) found that ownership among American households increased from 3% in 2021 to 11% in 2022, reaching 12% by late 2023. The 24/7, global, and highly liquid nature of cryptocurrency markets makes them particularly well suited to algorithmic intervention, where human traders cannot realistically monitor price action around the clock. As novel assets, the lack of historical stability and price fluctuations poses an extremely high-impact risk to investors. According to Delfabbro et al. (2021), many coins lost 80% of their value during the 2017–2018 crypto market crash, leading investors who did not sell at the peak to lose most of their gains. This shows that it is not only market volatility, but also a lack of effective gain optimization and decision-making strategies that affect results.

Machine learning (ML), deep learning (DL), and reinforcement learning (RL) are currently the most robust approaches to address these challenges. Adedigba et al. (2025) conducted a comparative analysis of predictive models and studied different machine learning techniques to determine the most effective approach to maximizing profits in cryptocurrency markets. Similarly, Park & Seo (2022) demonstrated the potential of deep learning through an action recommendation model specifically designed for cryptocurrency profit maximization.

Despite the growing adoption of artificial intelligence in financial trading, existing approaches to cryptocurrency gain optimization remain limited in scope, methodology, and practical applicability, leaving a gap between academic experimentation and deployable, real-time trading tools. Three critical gaps in particular remain unaddressed in the existing literature. First, reinforcement learning agents are consistently limited to a fixed three-action space (Buy, Sell, Hold), which ignores the capital allocation decisions that real traders make on every entry (Jay & Berlanga, 2024; Tay & Lim, 2024). Second, evaluation frameworks are almost exclusively single-horizon, making it impossible to know whether a model's performance is specific to one time scale or generalizes across investment periods (Seabe et al., 2023; Adedigba et al., 2025). Third, most studies test on a single asset, obscuring whether results reflect a model's true capability or simply the behavior of one particular cryptocurrency during one particular market phase (Tay & Lim, 2024; Seabe et al., 2023).

Beyond these scope limitations, existing studies also suffer from methodological weaknesses. Many rely on a classic train-test split that ignores temporal order and overestimates real-world performance (Adedigba et al., 2025). Furthermore, most research reports only predictive accuracy metrics such as R^2 and MSE, while omitting transaction costs, position sizing, and risk management — the factors that determine whether a strategy is actually profitable in practice (Jay & Berlanga, 2024).

To address these gaps, this work makes three concrete contributions. First, we extend the DQN action space to four actions by introducing dual position sizing (Buy 30% and Buy 70%), enabling the agent to modulate its market exposure based on signal confidence. Second, we evaluate all six models across five investment horizons (10 days, 1 month, 3 months, 6 months, and 12 months) on five cryptocurrency assets, producing the most comprehensive multi-horizon comparison in the existing literature. Third, all models are deployed in **Trademind IA**, a real-time trading dashboard that translates research findings into a practical decision tool accessible on both mobile and desktop.

This work is relevant to several audiences. For the algorithmic trading literature, it provides the most comprehensive multi-model, multi-asset, multi-horizon comparison to date, offering an empirical reference point against which future studies can be benchmarked. For practitioners and retail traders, the deployment of all six models in a single, unified dashboard — Trademind IA — bridges the gap between academic backtesting and a usable, real-time decision-support tool. Finally, for the broader field of AI-driven finance, the explicit comparison between reinforcement learning agents and supervised models across different market regimes offers practical guidance on when each family of models should be trusted, and when it should not.

This study is limited to five major cryptocurrencies (BTC, ETH, ADA, SOL, BNB) and a single calendar year (2024) of out-of-sample data. Results should therefore be interpreted as indicative of model behavior under the specific market regimes observed during this period — a bullish Q1, a corrective Q2, consolidation in Q3, and a Q4 rally — rather than as universal performance guarantees applicable to all market cycles or all cryptocurrencies. Similarly, the feature set used across all models is restricted to price-, volume-, and technical-indicator-based features; asset-specific on-chain metrics (e.g., gas usage, staking yields, DeFi total value locked) are outside the scope of this work.

This research aims to develop a trading system capable of executing strategies based on ML, DL, and RL prediction across multiple time horizons with realistic risk controls, in order to identify the model that delivers the best risk-adjusted performance at each horizon. We hypothesize that model performance is regime-dependent rather than universal, such that no single algorithm will dominate across all assets and investment horizons. To operationalize this goal, the following specific objectives guide the remainder of this work:

- Develop a backtesting framework that evaluates XGBoost, Random Forest, LSTM, GRU, PPO, and DQN across five time horizons (10 days, 1 month, 3 months, 6 months, 12 months) and five cryptocurrency assets (BTC, ETH, ADA, SOL, BNB).
- Expand the DQN action space by introducing a dual position sizing mechanism (Buy 30% and Buy 70%), allowing the agent to adjust market exposure based on signal confidence, and apply realistic risk management rules — including stop loss, take profit, and risk-based position sizing — for the supervised models.
- Compare the three model families using the Sharpe ratio, maximum drawdown, and total return, and benchmark all results against a buy-and-hold strategy to evaluate whether the models add meaningful value under real market conditions.

This document is organized as follows. Chapter 1 presents the theoretical foundations of algorithmic trading, cryptocurrency markets, and the technical indicators used as model inputs. Chapter 2 reviews the five most closely related works and positions our contribution within the current state of the art. Chapter 3 describes the dataset, preprocessing pipeline, model architectures, and evaluation framework. Chapter 4 presents and discusses the experimental results across all models, assets, and horizons, and introduces the Trademind IA platform.

CHAPTER 1:

Algorithmic Trading

1.1 Introduction

This field is rich in complex concepts and technical terms; therefore, it is necessary to explain the different key principles and terms related to cryptocurrencies and trading before presenting the results and methodology to provide a clear understanding. In this section, key terms and cryptocurrencies are explained. By defining this vocabulary, the reader will be prepared for the subsequent section of methodology and results.

1.2 Related Concepts

This section defines and places the key concepts underlying the research. It includes the basic principles of trading and the key features of cryptocurrencies.

1.2.1 Trading

Trading is the act of buying or selling a given asset in a financial market (Capital Research Brasil, 2020). This asset can be a stock, future indexes, currency pairs, cryptocurrencies, and others. These buy and sell operations are performed by a trader, who does it through a trading platform (Capital Research Brasil, 2020).In any market, traders are divided into two categories: retail traders, which are natural persons, and institutional traders, which include banks, financial institutions, and funds, representing the largest part of the trading volume.

The simplest trading strategy is Buy and Hold. It consists of buying an asset and holding it for a long period of time (Beers, 2020). Another simple and popular strategy is Dollar Cost Averaging (DCA). In DCA, the investor divides the total amount to be invested in equal parts that will be used to purchase an asset periodically, regardless of the current price (Hayes, 2022). This period can be weekly, monthly, etc. The goal is to reduce the impact of volatility on the overall purchase.

1.2.2 Candlesticks

There are seven key elements that shape a candlestick: the open, high, low, close, the upper and lower wicks (shadows), and the spread, as shown in Figure 1.1. According to Coulling (2013).

A candle captures four key price points: Open (the first price of the period), High (the highest price of the period), Low (the lowest price of the period), and Close (the last price of the period), which represent price movement over a fixed time period called a time frame. This time frame ranges from one minute to one month.

A bullish candle (typically green or blue) indicates that the closing price is higher than the opening price, meaning the market moved upward over that period. A bearish candle (typically red or black) indicates that the closing price is lower than the open, defining a downward move.

Figure 1.1. Illustrates the anatomy of a candlestick used in financial charts, showing the Open, High, Low, and Close prices, along with the upper/lower wicks (shadows) and the spread (difference between open and close).

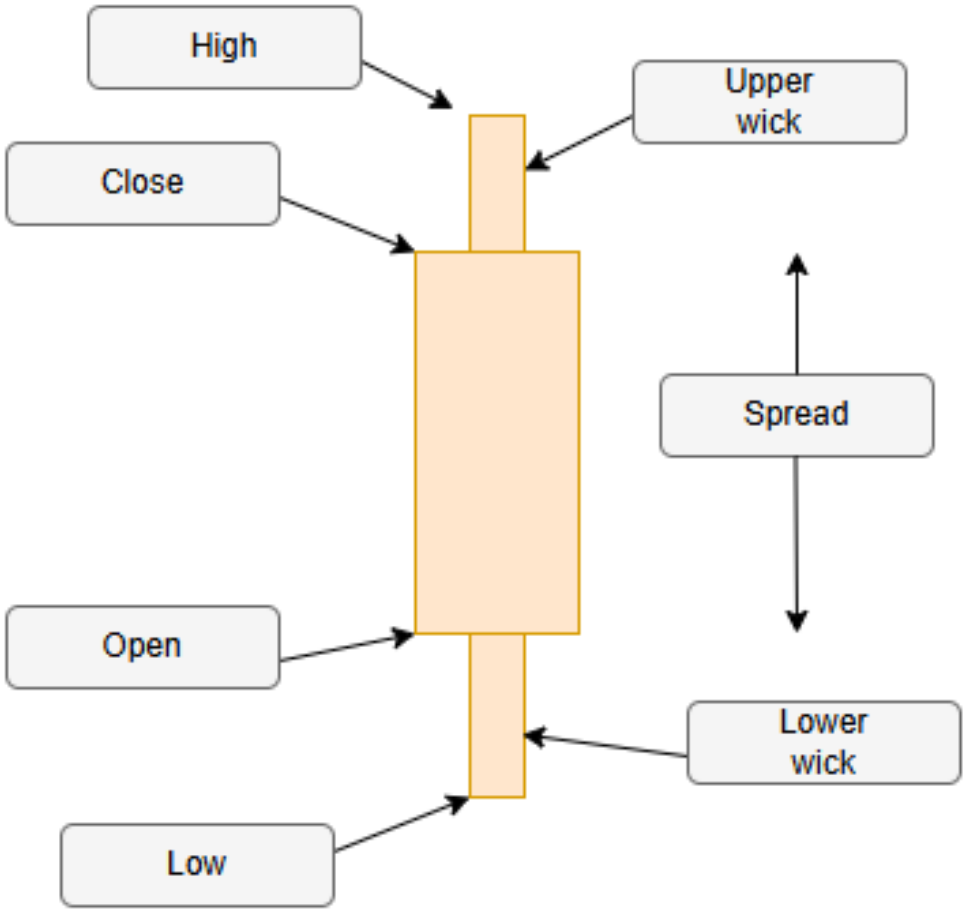


Figure 1.1: A Typical Candle

Figure 1.2. Shows a real example of a price chart using candlesticks at 5-minute intervals, displaying how price moves over time with green (bullish) and red (bearish) candles.

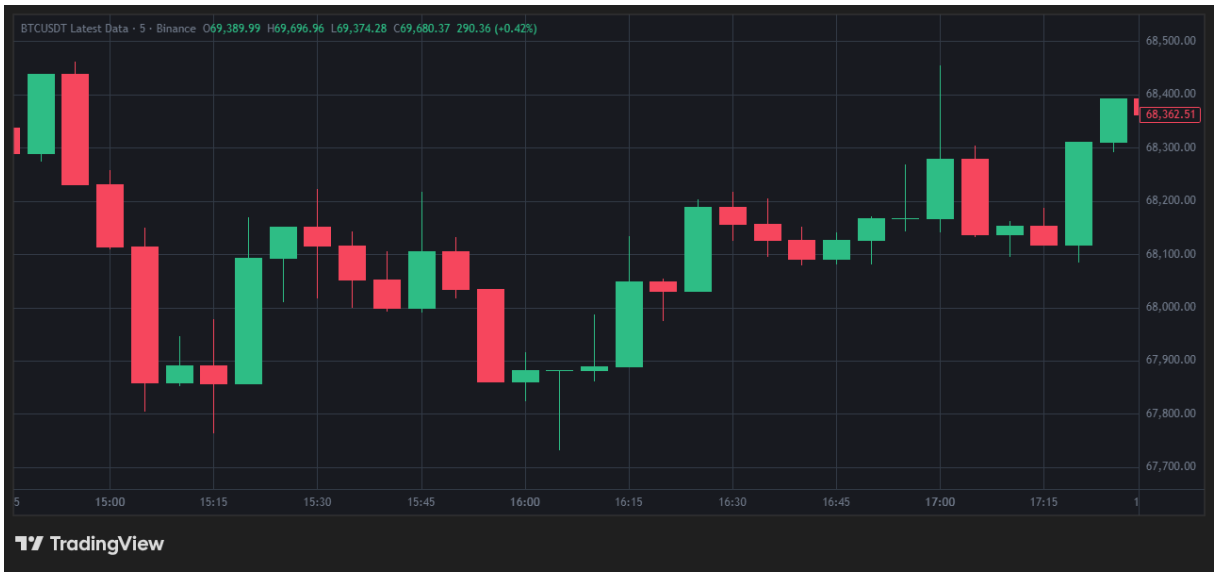


Figure 1.2: 5 minutes candlestick graph

1.2.3 Bullish and Bearish Candlesticks

A bullish candlestick forms when the closing price exceeds the opening price, signaling buying pressure and upward momentum, and is typically represented in green or blue. Conversely, a bearish candlestick forms when the closing price falls below the opening price, indicating selling pressure and downward momentum, usually represented in red or black (Coulling, 2013). Together, these two candle types provide traders with an immediate visual interpretation of market sentiment over any given time frame, allowing for rapid assessment of price direction and trend.

1.2.4 Position

A position in financial trading is the amount of a financial asset that is owned by a trader at any given time. Opening a position means entering a trade by buying or selling an asset, while closing a position means exiting that trade. The fluctuation in market prices affects the value of a position, resulting in unrealized profits or losses until the position is closed.

A long position is the most fundamental used position in financial markets, in which a trader buys a financial asset expecting the price will rise in the future. The trader realizes a profit when the price exceeds the entry price and incurs a loss if the price falls below it.

A short position is a more advanced trading strategy that allows traders to profit from declining markets, in which a trader sells a financial asset with the expectation that its price will fall in the future. The trader realizes a profit when the price decreases below the entry price and incurs a loss if the price rises above it.

1.2.5 Stop Loss and Take Profit

A stop loss is one of the most fundamental risk management tools in trading that defines the lower boundary. It automatically closes a position when the price reaches a predetermined level, limiting the trader's maximum loss on a position.

A take profit is the complement of the stop-loss. It defines the upper boundary of a trade rather than the lower boundary. It automatically closes a position when the price reaches a predefined level to secure realized profits.

1.2.6 Blockchain and Cryptocurrency Mining

A blockchain is a distributed ledger technology that supports most cryptocurrencies. This technology records transactions transparently and immutably across a network of computers, making it resistant to tampering. It consists of a continuous chain of blocks; each block contains a set of transaction records, a timestamp, and a cryptographic reference to the previous block, known as a hash. Cryptocurrency mining

is the process by which new transactions are validated and added to the blockchain. It also creates new cryptocurrency coins as rewards. Mining has two main functions: it produces new coins according to a predetermined schedule and preserves the security and integrity of the blockchain network.

1.2.7 Volatility

Volatility is the degree of variation in the price of a financial asset over time, used as a measure of market risk and uncertainty. It measures how much and how rapidly an asset's price moves up and down.

1.2.8 Cryptocurrencies

Cryptocurrencies are digital decentralized assets, unlike traditional currencies controlled by central banks. These digital assets use a technology called blockchain, a public digital ledger that records all transactions governed by mathematical rules. They rely on cryptographic techniques to secure transactions and control the creation of new units.

1.2.8.1 Bitcoin (BTC)

Bitcoin is the first cryptocurrency created in 2008 by Nakamoto(Nakamoto, 2008), who defines it as an electronic payment system based on cryptography and proof instead of trust. This cryptocurrency uses blockchain technology, where transactions are verified by mining, in which miners race to solve computationally hard math problems in exchange for new bitcoins. Many people call Bitcoin "digital gold" due to the capped total supply of 21 million units and a deflationary supply schedule. This schedule reduces the block reward issued to miners by half every four years in an event called the halving.

1.2.8.2 Ethereum (ETH)

Ethereum, introduced by Buterin(Buterin, 2015), is a programmable blockchain that uses smart contracts, which are self-executing code on the blockchain that automatically enforces agreements without requiring an intermediary. This feature gave rise to an entirely new category of applications, including DeFi protocols, NFTs, and DAOs (Schär, 2021).

On September 15 2022, in an event called "The Merge", Ethereum transitioned from Proof-of-Work (PoW) to Proof-of-Stake (PoS), in which validators are chosen to create new blocks based on the amount of ETH they hold and stake as collateral (Ethereum Foundation, 2022). This transition reduced energy consumption by 99.99% (from 80 TWh to 0.01 TWh), eliminated GPU mining, reduced block reward income by 97%, and cut new ETH issuance by 90% (from 13,000 to 1,600 ETH/day)(Buterin et al., 2021).

1.2.8.3 Binance Coin (BNB)

In 2017, Binance launched its coin, known as Binance Coin BNB, the most popular cryptocurrency exchange by trading volume due to lower trading fees on Binance. It is now the core of BNB Chain for smart contracts and decentralized apps, using a Proof of Staked (BNB SmartChain, 2022)Authority consensus mechanism that combines elements of Proof of Stake and delegated authority to enable faster, cheaper transactions, but at the cost of reduced decentralization.

Every three months, Binance destroys some of its coins to push the price up, though its value is also tied to Binance's performance. This makes it one of the most exchange-dependent major cryptocurrencies in the market.

1.2.8.4 Cardano (ADA)

In 2017, Charles Hoskinson, the co-founder of Ethereum (Hoskinson, 2017), founded Cardano (ADA). ADA is the native cryptocurrency of the Cardano network. Unlike most blockchain platforms, all protocol development in Cardano is based on peer-reviewed academic research. It was the first Proof-of-Stake mechanism (Ouroboros) to be scientifically proven secure(Kiayias et al., 2017), and it is unique because its major decisions are based on peer-reviewed academic research. It has a structured, multi-phase roadmap which is : settlement, decentralisation, smart contracts, scalability, and governance. In 2021, Smart contracts were introduced and written in a language based on Haskell known as Plutus, which allows formal mathematical verification of code correctness.

1.2.8.5 Solana (SOL)

Solana is a high-speed blockchain project launched in 2020 by Anatoly Yakovenko(Yakovenko, 2018). Combines Proof of History (PoH), which records event order without full consensus at every step, and Proof-of-Stake (PoS) (Solana Foundation, 2020), which processes thousands of transactions per second at a fraction of a cent. That solves the slow speeds and high fees issues of Ethereum. making it a competitor to Ethereum for high-throughput applications such as DeFi, NFTs, and Web3 gaming. However, suffered multiple network interruptions, which have raised concerns about its overall reliability.

Table 1.1: Key Characteristics of Major Cryptocurrencies

Cryptocurrency	**Launch**	**Consensus**	**Key Feature**
Bitcoin (BTC)	2009	Proof of Work	Digital gold, fixed supply
Ethereum (ETH)	2015	Proof of Stake	Smart contracts, programmable blockchain
Binance Coin (BNB)	2017	Proof of Staked Authority	Exchange utility, burn mechanism
Cardano (ADA)	2017	Proof of Stake (Ouroboros)	Academic research-driven development
Solana (SOL)	2020	Proof of History + PoS	High throughput, low transaction cost

1.3 Technical Indicators

Murphy (1999), in his foundational work Technical Analysis of the Financial Markets, explains that technical indicators are mathematical calculations applied to historical price and volume data to help traders identify market conditions, confirm trends, and generate trading signals. Technical trading rules have shown mixed profitability across different markets (Lento, 2007; Brock et al., 1992).

There are four categories of technical indicators: trend indicators, momentum indicators, volatility indicators, and volume indicators.

1.3.1 Trend Indicators

Trend indicators are designed to identify the direction and strength of a sustained market movement. They smooth out short-term price fluctuations to reveal the underlying direction of the market, making them among the most widely used tools in technical analysis.

1.3.1.1 Simple Moving Average (SMA)

Trend indicators define the direction and strength of a market movement (Brock et al., 1992). The most common technical indicator used in technical analysis is the **Simple Moving Average (SMA)**, which calculates the average closing price over a defined number of periods by giving equal weight to each data point in the window. A 7-day SMA smooths out short-term noise and exposes the short-term direction, while a 30-day SMA captures the medium-term trend.

$$SMA = \frac{P_1 + P_2 + \cdots + P_n}{n} \tag{1}$$

Where:

- P_i = Price at each candle within the period i,
- n = number of time periods.

1.3.1.2 Exponential Moving Average (EMA)

traders also use the **Exponential Moving Average (EMA)**, applies more weight to recent prices to react faster to new information while still smoothing out random noise. The weighting factor is calculated as 2 ÷ (n + 1).

$$EMA_t = (P_t \times k) + (EMA_{t-1} \times (1 - k)) \tag{2}$$

Where:

- P_t = current price.

- EMA_{t-1} = previous day's EMA.
- $k = \frac{2}{n+1}$.
- n = number of periods.

1.3.1.3 The Moving Average Convergence Divergence (MACD)

Extending the idea of EMA, **The Moving Average Convergence Divergence (MACD)** (Appel, 1979; Chong & Ng, 2014), calculates the difference between a 12-period EMA and a 26-period EMA by subtracting the slower one from the faster one. When the result is positive, short-term momentum exceeds long-term momentum, indicating that upward momentum is dominant. When it is negative, the opposite is true.

$$\text{MACD Line} = EMA_{12} - EMA_{26} \tag{3}$$

A second line, called the **signal line**, is simply a 9-period exponential moving average (EMA) of the MACD line itself (Aspray, 1986). it used to generate actionable trading signals. When the MACD crosses above the signal line, it is interpreted as a bullish signal. When it crosses below, it suggests bearish momentum is taking over.

$$\text{Signal Line} = EMA_9(\text{MACD Line}) \tag{4}$$

1.3.2 Momentum Indicators

Momentum indicators focus on the speed and strength of price changes, how fast the price is moving rather than simply its direction, helping traders identify whether a trend is gaining or losing energy and whether an asset is overextended in either direction.

Momentum (MOM) and **Rate of Change (ROC)**are the most fundamontal Momentum indicators. Both measure price change over a defined period.

1.3.2.1 Momentum (MOM)

MOM calculates the difference between the current price and the price n periods ago

$$\text{MOM} = P_t - P_{t-n} \tag{5}$$

However, it uses the absolute price term, a value of 500 carries a different meaning depending on whether Bitcoin is trading at 5,000 dollars or 50,000 dollars. In the first case it represents 10% gain, while only 1% in the second case, making it an unreliable measure when comparing momentum across different time periods.

1.3.2.2 Rate of Change (ROC)

The Rate of Change expresses the same concept as a percentage to solve the Momentum (MOM) problem.

$$\text{ROC} = \frac{P_t - P_{t-n}}{P_{t-n}} \times 100 \tag{6}$$

1.3.2.3 Relative Strength Index (RSI)

The most widely used momentum indicator is **the Relative Strength Index (RSI)** (Wilder, 1978), which measures the ratio of average gains to average losses over a 14-period lookback window and normalized to a scale between 0 and 100 using this formula:

$$\text{RSI} = 100 - \frac{100}{1 + \text{RS}} \tag{7}$$

where

$$\text{RS} = \frac{\text{Average Gain}}{\text{Average Loss}}$$

over the selected period (typically 14 days).

1.3.3 Volatility Indicators

Volatility indicators measure price fluctuation over a given period. While low volatility means a more predictable market, High volatility means large price swings and higher risk.

1.3.3.1 Bollinger Bands

One of the most widely adopted tools is the **Bollinger Bands**(Bollinger, 2001), which consist of three lines plotted directly on the price chart: a middle band, upper band, and lower band.

The middle band shows the average price and acts like a support or resistance line.

When the price touches the upper band, it may be overbought, while a touch of the lower band suggests a oversold.

$$\begin{aligned} \text{Middle Band} &= \text{SMA}(n) \\ \text{Upper Band} &= \text{SMA}(n) + k \times \sigma \\ \text{Lower Band} &= \text{SMA}(n) - k \times \sigma \end{aligned} \tag{8}$$

where:

- $\text{SMA}(n)$ is the simple moving average over n periods (typically 20).
- σ is the standard deviation of price over the same n periods.
- k is the number of standard deviations (typically 2).

1.3.3.2 Average True Range (ATR)

(Wilder, 1978) developed a volatility indicator called **Average True Range (ATR)** that quantifies market volatility by decomposing the entire range of a price movement over a defined lookback period, capturing both the intraday spread between the high and low of each session and any price gaps that may occur between the close of one session and the open of the next.

The True Range (TR) for period t is defined as:

$$\text{TR}_t = \max\left(\text{H}_t - \text{L}_t,\ |\text{H}_t - \text{C}_{t-1}|,\ |\text{L}_t - \text{C}_{t-1}|\right) \tag{9}$$

where:

- H_t = current period's high,
- L_t = current period's low,
- C_{t-1} = previous period's close.

The Average True Range is then a moving average of the True Range values, typically over $n = 14$ periods:

$$\text{ATR}_t = \frac{1}{n} \sum_{i=1}^{n} \text{TR}_{t-i+1} \tag{10}$$

1.3.3.3 Rolling Volatility

In addition to Bollinger Bands and ATR, there is also **rolling volatility** computed at two horizons, 7 days and 30 days, calculated as the standard deviation of daily returns over the respective window:

$$\text{Rolling Volatility} = \sigma(\text{Return}_t, \text{Return}_{t-1}, \ldots, \text{Return}_{t-n}) \tag{11}$$

1.3.4 Volume Indicators

Volume represents the total number of units of an asset traded during a given period. It tells traders how much conviction is behind that move. The most commonly adopted is the volume ratio, which is calculated by dividing the current period's volume by its 20-day moving average:

$$\text{Volume Ratio} = \frac{\text{Volume}_t}{\text{SMA}(\text{Volume}, n)} \tag{12}$$

1.4 Trading Strategies

A trading strategy is a clear set of rules that tells a market participant when and how to buy or sell a financial asset. Strategies differ in their underlying assumptions, time horizons, and levels of automation, but all aim to beat a passive benchmark while managing risk. The academic and professional literature classifies trading strategies into several major categories based on how prices move over time.

1.4.1 Buy and Hold

Buy-and-hold strategy is the most widely referenced benchmark, Where Investors purchase an asset and hold it permanently, ignoring short-term price movements. This strategy is based on the hypothesis that markets are efficient, which makes it impossible to consistently beat the market through active trading (Fama, 1970). In the context of Bitcoin, buy-and-hold has performed very well over multiple years.

1.4.2 Trend Following

Trend-following strategy builds on the hypothesis that rising assets tend to continue upword and falling assets continue to downword(Covel, 2009). Moving averages, the Moving Average Convergence Divergence indicator, and channel breakout systems are the most common technical indecators used to implement this strategy, helping traders identify the direction of the dominant market movement and then enter trades aligned with that direction. This strategy perform well in strongly directional markets but also tends to produce losses in during periods of sideways.

1.4.3 Mean Reversion

Mean reversion strategy assumes that extreme asset movements tend to return to their long-term average, which means a price rise above its historical mean is expected to fall back toward that level, and vice versa(Poterba & Summers, 1988). Unlike trend-following, this approach is implemented using indicators such as the Relative Strength Index (RSI), Bollinger Bands, and z-score deviations. This strategy is effective in range-bound markets or when asset prices display strong statistical persistence around a central value.

1.4.4 Momentum Trading

Momentum trading uses the rate of change of price, rather than on the price's direction relative to a moving average, as trend following does. On the hypothesis that recent asset changes are predictive of future asset changes over short horizons, this approach buys assets that have proved strong recent performance and sells assets that have underperformed (Jegadeesh & Titman, 1993). This strategy is measured through indicators such as the Rate of Change, the Momentum oscillator, and percentage returns over rolling windows.

1.5 Artificial Intelligence in Financial Trading

Artificial intelligence has fundamentally changed financial trading by allowing systems to learn from past market data and adjust to changing market conditions (Aldridge, 2013). AI-powered trading systems can process huge amounts of data at high speed, make decisions without emotional bias, and recognize complex non-linear patterns.

Trading bots execute trades based on rules or learned strategies, representing an automated software agent. These bots trade across various assets such as stocks, forex, and cryptocurrencies (Ahmad et al., 2021; Detzel et al., 2018). Also, it can process market movements in milliseconds, much faster than human capability. Nowadays, AI-powered algorithmic trading accounts for a significant portion of daily trading volume (Aldridge, 2013).

Sentiment analysis was unlocked by AI, which uses natural language processing models to analyze news articles and social media to predict price movements. Also, machine-learning-based predictive models are used to forecast asset prices, detect irregularities, and identify profitable trading opportunities (Yang et al., 2020). In the context of risk management, AI systems continuously monitor portfolio exposure and dynamically adjust positions to limit losses and protect capital.

In addition, the integration of AI has given rise to high-frequency trading systems, robo-advisors, and portfolio optimization engines that trade with minimal human involvement (Aldridge, 2013). The main

artificial intelligence paradigms underlying these systems are machine learning, deep learning, and reinforcement learning, which provide specific capabilities in prediction, pattern recognition, and sequential decision-making (Sutton & Barto, 2018).

1.6 Conclusion

In conclusion, the theoretical and technological foundations of our project have been comprehensively outlined in this chapter. First, it explored the cryptocurrency ecosystem and blockchain technology, showing the most profound shift in modern finance markets and the opportunities they offer in investment and financial access. Then, the chapter analyzed the essential mechanisms of trading, with a special focus on position management and loss limitation strategies such as stop-loss, which form the basis of any disciplined trading approach.

Furthermore, this section presented the most widely used technical indicators in financial market analysis, which serve to build the input signals for our models. Finally, we outlined the increasing importance of artificial intelligence in the field of trading via automated trading bots, algorithmic trading systems, and approaches based on machine learning, deep learning, and reinforcement learning.

All of these elements will allow us to identify the methodological approaches adopted in the literature, position the conceptual foundation of our work and clearly position our contribution within the current state of research in artificial intelligence applied to cryptocurrency trading.

CHAPTER 2:

State of Art

2.1 Introduction

Due to the increasing use of artificial intelligence in financial markets, there has been a notable rise in research dedicated to automating and optimizing cryptocurrency trading. Instead of relying on fixed rules or manual analysis, today's approaches leverage machine learning and reinforcement learning to learn adaptive strategies directly from market data, explicitly targeting the goal of maximizing returns while managing risk.

This chapter offered an overview of five closely related works, that were selected for their direct relevance to our thesis. Each selected study addresses the same fundamental objective of gain optimization in cryptocurrency markets. In addition, the related works covers the key model families upon which our comparative study is built, which are DQN, PPO, LSTM, GRU, XGBoost and Random Forest. In this section, each work is analyzed in terms of its methodology, experimental results, and the specific insights it contributes to the broader problem of building intelligent, profit-oriented trading systems. Together, these works define the state of the art against which our contribution is positioned.

2.1.1 Selection Methodology

The five studies reviewed in this chapter were selected following a systematic procedure. We searched the following databases: Google Scholar, Scopus, and the SSRN e-library, using keywords: *"cryptocurrency trading"*, *"reinforcement learning"*, *"deep learning"*, *"machine learning"*, *"gain optimization"*, and *"profit maximization"*. The inclusion criteria were: (1) peer-reviewed articles or working papers from 2020–2025, (2) studies that directly compare at least two AI models on cryptocurrency data, and (3) works that evaluate performance using financial metrics (return, Sharpe, drawdown). We excluded studies that only report predictive accuracy (e.g., RMSE, MAE) without trading simulation. The five papers selected represent the most relevant and methodologically sound comparative studies in this domain, covering the key model families (DQN, PPO, LSTM, GRU, RF, XGBoost) that we also employ.

2.2 Smart Trading Using Technical Analysis on the Crypto Market

The cryptocurrency market is characterized by price volatility, continuous 24/7 trading, and a sensitivity to external factors ranging from regulatory announcements to social media sentiment, making the cryptocurrency market one of the most difficult environments for traditional trading strategies which use fixed rules or manually tuned technical indicators. In response to these obstacles, Jay & Berlanga (2024) performed a systematic comparative study that targeted the evaluation of how three distinct families of intelligent agents, which are: Deep Q-Network (DQN), Long Short-Term Memory (LSTM), and Random Forest (RF), perform in cryptocurrency trading that uses technical analysis. They directly oriented their work toward building a profitable trading strategy rather than simply predicting prices, by optimizing trading decisions measured through real financial metrics such as total return, Sharpe ratio, and maximum drawdown.

2.2.1 Methodology

In this paper, a cautiously designed pipeline enabled all models to be trained and evaluated under the same market regimes and data, guaranteeing a fair comparison across model families. The model families used in this study include DQN (reinforcement learning), which interacts with a simulated backtesting framework and receives rewards based on portfolio performance; LSTM, which learns temporal patterns in price and indicator sequences to generate forward-looking trading signals; and Random Forest (supervised learning), an ensemble classifier that combines many decision trees to make trading decisions. The models are trained on standard technical indicators to issue buy, sell, or hold signals within a backtesting framework.

The models were evaluated using a methodical backtesting procedure on a dataset that includes several pairs of cryptocurrency. To cover both profitability and risk, performance was measured by Total Return, Annualized Return, Annualized Volatility, Sharpe Ratio, Sortino Ratio, Maximum Drawdown, and Calmar Ratio.

2.2.2 Results and Findings

The results show that no single model masters every evaluation measure, illustrating how different algorithmic methods have complementary nature. The DQN achieved the highest total return among all models in the study, while the LSTM led in the Sharpe ratio category, demonstrating superior profitability and risk-adjusted performance. This supports the hypothesis that reinforcement learning agents are better suited to the objective of gain maximization than models trained on prediction error minimization, because RL optimizes directly for cumulative reward. The LSTM adjusted successfully to changing market trends by generating consistent returns over time while capturing sequential dependencies in price and indicator data. The Random Forest agent was the most efficient in controlling downside risk but not the most profitable. it record the lowest maximum drawdown among all three models. This is good for risk-averse investors seeking capital preservation.

A considerable finding is the need for flexible and varied trading strategies, because no single technical indicator is optimal for every model or market situation. In the end, the authors recommend a hybrid of DQN, LSTM, and RF could furnish more robust performance than any single model on its own.

2.2.3 Relation to Our Work

The relevance of this study to our work is evident from the fact that it benchmarks precisely the same models — DQN, LSTM, and Random Forest – on the task of cryptocurrency trading with the aim of maximizing gains. As with Jay & Berlanga (2024), our study compares multiple algorithms under similar backtesting conditions, evaluating them with respect to the same financial measures. We differ from them in that our work adds PPO, GRU and XGBoost to the mix of models compared, offering a wider perspective on both RL and DL architectures; and also in that we consider a more complex environment with additional features like volatility regime information, trend strength and position sizing decisions. The conclusions about the relative strengths of DQN and RF made by Jay & Berlanga (2024) are valuable to us as an empirical reference point against which we can evaluate our own findings.

2.3 Deep Reinforcement Learning in Cryptocurrency Trading

In the past years, we have seen an exponential increase in the number of people participating in the cryptocurrency market with the goal of automating investment decision-making through AI. Tay & Lim (2024) have introduced a complete automatic trading system for cryptocurrencies, which uses Deep Learning based models to forecast prices and deep reinforcement learning agents to make decisions in a simulation trading environment. They wanted to prove that a system which consists of deep learning-based forecasting models and deep reinforcement learning agents can generate positive returns consistently regardless of the current market situation. This is because many other automated AI trading systems use supervised learning, which aims to maximize accuracy but fails to optimize profitability.

2.3.1 Methodology

The system was built in two stages. In the first stage, three time-series forecasting models were constructed to predict future cryptocurrency prices: a Temporal Convolutional Neural Network (TCNN), a Long Short-Term Memory Network (LSTM), and a Gated Recurrent Unit Network (GRU). These models were trained on historical price data for eight cryptocurrencies: Bitcoin, Ethereum, BinanceCoin, DogeCoin, Cardano, Polygon (MATIC), Avalanche (AVAX), and WAVES, their predictions were used to generate forward price estimates. In the second stage, the predicted prices were combined with cryptocurrency sentiment data collected via the Alternative.me API, historical price data, technical indicators, and account state information to form the input state for three deep reinforcement learning agents: Deep Q-Network (DQN), Advantage Actor Critic (A2C), and Recurrent Proximal Policy Optimization (RPPO). Each agent was initialized with $1,000 of capital per cryptocurrency and was trained to take one of three actions — Buy, Sell, or Hold — at each time step.

2.3.2 Results and Findings

From the results of experiments, one can see that DQN gave the best return on investment for all cryptocurrencies. In total, DQN made an income of $740, which means that it provided an ROI of 12.33% on average. In particular, its performance was especially good on BinanceCoin where its ROI reached 63.98%. On the other hand, both A2C and RPPO showed negative ROI during the experiment. Such

results may be explained by the ineffectiveness of the policy gradient method used by them due to their inability to effectively work with the reward mechanism implemented by the authors of the work. Thus, value-based approaches (e.g., DQN) performed better than actor-critic ones in such a discrete action space. The use of sentiment and price prediction as additional inputs turned out to be a meaningful decision as well. It should be noted that the process of comparing different agents is rather complicated since their performance differs significantly depending on the cryptocurrency used and on the accuracy of price predictions produced by the LSTM/GRU models.

2.3.3 Relation to Our Work

Another study relevant to our research work is the recent work done by Tay & Lim (2024). This work is very close in terms of reference to our current project, as both works employ DQN, LSTM, and GRU in pursuit of the same objective which is optimization of cryptocurrency trading with return on investment used as the measure of performance.However, while Tay & Lim (2024) use LSTM and GRU as price forecasting modules feeding into the DQN agent, our work deploys them as independent trading agents that directly generate trading decisions. Similar to Tay & Lim (2024), our study relies on a two-stage approach where technical indicators and other relevant market information is derived from the raw price data and then fed into the environment observation space. The major difference between the design of our reinforcement learning environment, as opposed to the work of Tay & Lim (2024), lies in the way the action space is designed. In particular, while the action space in Tay & Lim (2024) involves only traditional three-actions – Buy, Sell, and Hold – we have decided to implement a position sizing feature through dual buys (Buy 30%, Buy 70%). These additions to the action space represent a significant architectural change that differentiates our DQN-based trading agent, and was not present in the work of Tay & Lim (2024).

Finally, the architecture of our comparative framework involves a third model – PPO, as opposed to the two models used by Tay & Lim (2024) – and an additional reference model – Random Forest, as opposed to the reference baseline used in their work. Additionally, we propose a novel reward shaping technique addressing the apparent weakness of RL agents to detect long-term market trends observed in the results of Tay and Lim's paper.

2.4 Reinforcement Learning Pair Trading

The pairing strategy is one of the oldest quantitative methods, which utilizes the correlation between two stocks to detect and profit from temporary deviations in prices. According to Yang & Malik (2024), classical rule-based pair trading has proven successful in the past but lacks an important degree of flexibility; its entry and exit thresholds are fixed and thus incapable of adapting to changes in market volatility, which is especially pronounced for cryptocurrencies. This paper examines whether reinforcement learning is capable of solving this problem, by creating flexible trading algorithms, which not only specify when to trade but also how to scale trades. The research is based on a pairing of two denominated currency pairs involving Bitcoin (BTC-GBP and BTC-EUR).

2.4.1 Methodology

To address the problem of pair trading, the researchers developed their own RL environments, where the agents were tasked with controlling the difference between two crypto pairs. Two configurations were offered by the researchers: RL1, which learned solely the entry and exit strategies, and RL2, a dynamic scaling strategy, which learned both the entry/exit strategies and the quantity invested at each point in time. To conduct the research, the researchers used two different types of agents: the A2C agent and its variations based on PPO and DQN. In addition, the reward function was adjusted to better simulate actual trading goals, taking into account transaction fees (0.02%) and the risk-free interest rate (5.5%). Of special importance in the paper under discussion was the definition of the observation and action space, since, according to the authors, the quality of the RL state representation plays a key role in policy learning.

2.4.2 Results and Findings

The classic non-RL technique based on the Gatev et al. (2006) approach resulted in 8.33% annualized gain. The RL-based techniques resulted in gains ranging from 9.94% to 31.53%, depending on the used

algorithm and hyperparameters. RL2 with dynamic scaling was the most profitable among those, showing that not only time but size of orders is an important aspect to learn in order to have a good trading system. As for the performance of individual RL algorithms, A2C yielded the best results, whereas PPO achieved only a marginal positive return in the RL2 configuration and DQN resulted in significant losses in all tested scenarios. Finally, the study found that lower transaction costs result in larger number of executed trades whereas higher transaction costs lead to a more prudent behavior in terms of executing trades – again, a desirable characteristic of any trading system. The authors concluded that the use of RL in trading is effective, especially since it is able to identify better opportunities than traditional techniques judging by the profit per trade metric.

2.4.3 Relation to Our Work

Firstly, Yang & Malik (2024) work can be connected to our project in terms of algorithms, which are applicable for cryptocurrency trading tasks. As mentioned above, we decided to employ reinforcement learning algorithms, such as DQN and PPO, as the core components of our trading system. Thus, our research is based on similar premises since both works use reinforcement learning algorithms, which are effective when used for crypto trading tasks. However, there are some distinctions between our approaches that should be highlighted. For instance, the idea of dynamic scaling, which implies that the agent can decide not only about the time of position adjustment but also about the amount of money that should be invested into a specific asset, is implemented in the design of the action space of the DQN. In other words, the agent does not have a simple choice of Buy, Sell, and Hold. Instead, our DQN agent has four options : Buy 30%, Buy 70%, Sell, and Hold. The decision to choose among these options will depend on how confident the agent feels after analyzing the incoming signal and the data in the observation space. The PPO agent, on the other hand, is limited to the standard three-action space Buy, Sell, and Hold. This agent has the same position size. The main distinction between Yang & Malik (2024) paper and our project lies in the type of trading, the authors focus on pair trading where the price spread between two correlated assets is used as a basis for decision-making. In turn, our system is aimed at single-asset trading that utilizes more sophisticated technical and momentum-based features. Besides, the feature spaces of the DQN and PPO agents differ significantly in their complexity and the types of market data they incorporate.

2.5 Forecasting Cryptocurrency Prices Using Deep Learning Approach

The fact that cryptocurrency price series have the character of being nonlinear and non-stationary makes their analysis quite problematic from the perspective of using classical approaches such as ARIMA and exponential smoothing for forecasting purposes. Seabe et al. (2023)tried to tackle this problem by considering and comparing three different architectures of recurrent neural networks: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bi-Directional LSTM (Bi-LSTM), when predicting the exchange rates of three main cryptocurrencies: Bitcoin (BTC), Ethereum (ETH), and Litecoin (LTC). Their research was conducted in the journal called Fractal and Fractional and constitutes one of the most detailed direct comparisons between the two aforementioned model architectures on the subject matter of cryptocurrencies.

2.5.1 Methodology

The dataset was downloaded from Yahoo Finance and includes 5 years of data starting from January 2018 and ending January 2023. This gives us a period of 5 years during which we get daily prices of BTC, ETH, and LTC. Data was split according to the train:test ratio of 80:20, where training data covers the period of January 2018-December 2021, while testing covers January 2022-January 2023. In preparation to the training phase, minMax scaling was conducted on the feature space of all assets. For each one of the three mentioned architectures, namely LSTM, GRU, and Bi-LSTM, models were independently trained. RMSE and MAPE scores were used as evaluation metrics for all models. Bi-LSTM architecture, as compared to other types of Recurrent Neural Networks, allows us to process sequences in both directions - forward and backward, meaning that it can find temporal dependencies not only in previous but also in future parts of the sequence. Moreover, it has been proved by the authors that there exist correlations among these three assets, measured by Pearson correlation coefficient; interdependencies have proven

strong enough to share some architectural decisions between them. The problem of missing values was handled by forward imputation approach.

2.5.2 Results and Findings

The outcomes demonstrated superior performance of bi-directional deep learning models compared to the traditional recurrent neural network techniques when predicting cryptocurrency prices. For instance, Bi-LSTM showed consistent performance in generating Mean Absolute Percentage Error (MAPE) values as low as 0.036 for most cryptocurrencies, outperforming significantly the GRU and LSTM models. The Bi-LSTM algorithm was the most accurate and efficient, specifically achieving the lowest error scores for Bitcoin, Ethereum, and Litecoin. Meanwhile, the GRU model was identified as the second-best performing architecture in the study. Bidirectional flows provided more consistent forecasting results regardless of market condition than the unidirectional algorithms. Bi-directional architectures incorporating forward and backward data processing performed better and combined their advantages for better prediction accuracy.

In conclusion, the study highlighted significant advantages of using Bi-LSTM techniques as highly accurate alternatives to standard LSTM and GRU solutions for predicting cryptocurrency prices. Moreover, the researchers recommended utilizing deep learning algorithms to identify concealed patterns in historical data for more efficiency. It is worth noting that the authors stated that future research on the topic should focus on social media and trading volumes. They even mentioned exploring the effect of tweets and sentiments.

2.5.3 Relation to Our Work

The research by Seabe et al. (2023) provides empirical evidence validating our choice to separately investigate the performance of LSTM and GRU as models. The finding in the paper that GRU outperformed LSTM in terms of forecasting accuracy when applied to cryptocurrency prices could serve as a basis for comparison and interpretation of results of our model performance. However, contrary to Seabe et al.'s research, in which LSTM and GRU were only evaluated for their prediction accuracy using measures like RMSE and MAPE, we consider performance of the models in a framework of maximizing gains using ROI, Sharpe Ratio, and maximum drawdown as evaluation criteria. In respect of the asset coverage, our studies have several points of similarity: both cover several popular cryptocurrencies such as Bitcoin and Ethereum. While Seabe et al.'s research evaluates the models using data from the years in which Bitcoin, Ethereum, and Litecoin gained popularity, our study covers Bitcoin, Ethereum, Cardano, Solana, and Binance coin, thus including a wider selection of assets that have gained popularity relatively recently. In addition, while evaluating the models over several years, including the bear market phase, our study covers the data only from the year 2024.

2.6 Machine Learning Approaches to Trading Optimization

The cryptocurrency exchange market has particular characteristics, including round-the-clock transactions, internationality, volatility, and market immaturity, that make common techniques of financial analysis insufficient. In response to these challenges, Adedigba et al. (2025) conducted a broad comparative study that was not limited to a particular model or cryptocurrency, but rather employed several prediction algorithms across 30 different cryptocurrencies over a period of four years, with the goal of predicting prices and providing trading recommendations. This paper can be considered one of the most recent and methodically sound comparative studies involving machine learning applications in cryptocurrency trading.

2.6.1 Methodology

In this research, the historical prices and volumes for 30 cryptocurrencies were obtained from Yahoo Finance during the last four years. To reduce the number of dimensions in the data set, the researchers used PCA for the first step and applied K-means clustering to determine groups of cryptocurrencies and representative assets. These steps show the knowledge of how to work with high-dimensional financial data and use the most suitable methods for data processing. Four machine learning algorithms were built on selected representative assets, namely Gradient Boosting, XGBoost, Support Vector Regression, and LSTM networks. Performance measures such as MAE, RMSE, and R-squared were calculated to

compare model accuracy. In addition, trading signals were generated based on the prediction results using moving average crossover and machine learning methods. It is important to note that trading signals were generated by both techniques because of their different nature, which allowed the authors to test the effectiveness of combining the results of two methods.

2.6.2 Results and Findings

The outcomes demonstrated superior performance of ensemble-based models compared to the deep learning techniques when predicting cryptocurrency prices. For instance, Gradient Boosting and XGBoost showed consistent performance in generating R^2 of about 0.98 for most cryptocurrencies, outperforming significantly the SVR and LSTM models. The XGBoost algorithm was the most accurate and efficient with a mean accuracy of 67.2% and an F1-score of 0.63 for 7-day prediction periods. Meanwhile, the LSTM model had the largest peak recall score of 73.4%, making it more sensitive to new trends. Moving averages provided more consistent trading signals regardless of market condition than the machine learning algorithms; however, the latter were more responsive to price changes. Hybrid strategies incorporating moving averages and machine learning signals performed better and combined their advantages for better trading opportunities. In conclusion, the study highlighted significant advantages of using ensemble techniques as cost-efficient alternatives to deep learning solutions for predicting cryptocurrency prices. Moreover, the researchers recommended combining the strengths of statistical and machine learning approaches in hybrid trading systems for more efficiency. It is worth noting that the authors stated that future research on the topic should focus on developing adaptive algorithms for online training. They even mentioned reinforcement learning.

2.6.3 Relation to Our Work

The contribution of Adedigba et al. (2025) deals with the supervised learning portion of our model space – i.e., application of ensemble and predictive algorithms in the context of trading optimization and offers a theoretical basis for a comparison of machine learning predictive abilities on cryptocurrency data. The finding that, despite its superior predictive accuracy, the LSTM algorithm fails to outperform Gradient Boosting and XGBoost in recall and trend detection ability is of particular relevance to the interpretation of the relative effectiveness of the LSTM, GRU, and Random Forest portions of our experiment. More crucially, however, the authors state clearly that reinforcement learning is a logical progression from predictive machine learning in this context – which is exactly where our current study goes. While the former paper concludes with developing a framework for extracting profitable signals from ML predictions, we go one step further by including two types of reinforcement learning agents (DQN and PPO) into our backtesting setup and evaluating the results based on the same financial metrics. Thus, our study directly builds upon that of Adedigba et al. (2025). but applies it on a more recent sample and thus, to a different market regime.

2.7 Summary of Related Works

Table 2.1 provides a structured comparison of the five reviewed studies across eight dimensions: assets traded, training and testing periods, dataset sources, models used, action space design, input features, evaluation metrics, and key findings. This synthesis highlights the methodological diversity across the literature while also revealing the common limitations that motivate our proposed approach, particularly the absence of position sizing in reinforcement learning action spaces and the lack of multi-horizon evaluation frameworks.

Table 2.1: Comparative summary of related works on cryptocurrency trading and forecasting

Study	Assets	Train/Test Period	Dataset Source	Models	Action Space	Features & Indicators	Metrics	Findings
Jay & Berlanga (2024)	Multiple crypto pairs	Not disclosed	Not disclosed	DQN, LSTM, RF	Buy/Sell/Hold (no sizing)	RSI, MACD, BB, MA, Volume	Return, Sharpe, Sortino, MDD, Calmar	DQN: best return & Sharpe. RF: lowest MDD. LSTM: most consistent.
Tay & Lim (2024) & Dig. Econ.	BTC, ETH, BNB, DOGE, ADA, WAVES	Train: 2017–2022 Test: 2023	Binance API + Alternative.me ∼2 190 rows × 6	TCNN, LSTM, GRU, DQN, A2C, RPPO	Buy/Sell/Hold (no sizing)	OHLCV, sentiment, predicted prices. *No TA.*	ROI, profit ($), win/loss	DQN: +$740, ROI 12.3%. BNB: 63.98%. A2C & RPPO: negative.
Yang & Malik (2024)	BTC-GBP / BTC-EUR spread	Test: Dec 2023 (minute-level)	Binance API	DQN, PPO, A2C	Entry/Exit + dynamic sizing (RL2)	Price spread, balance, position, cost 0.02%. *No TA.*	Ann. profit, profit/trade, Sharpe	RL: 9.94%–31.53% vs 8.33% baseline. RL2 best. A2C > DQN > PPO.
Seabe et al. (2023)	BTC, ETH, LTC	Train: 2018–2021 Test: 2022–2023 (80/20)	Yahoo Finance ∼1 461 rows × 3	LSTM, GRU, Bi-LSTM	*N/A* — forecasting only	Closing price, MinMax norm., 10-step lookback. *No TA.*	RMSE, MAPE, accuracy	Bi-LSTM: best accuracy. GRU ≥ LSTM. All beat baselines.
Adedigba et al. (2025)	30 cryptocurrencies	Train: 4-yr Test: held-out (7-day horizon)	Yahoo Finance ∼1 460 rows × 30	XGBoost, GB, SVR, LSTM	Buy/Sell via MA crossover. *No RL.*	PCA features, K-means, MA signals. *No RSI/-MACD/BB.*	MAE, RMSE, R^2, F1, recall	XGBoost: R^2≈0.98, acc 67.2%. LSTM recall: 73.4%. Authors recommend RL.

2.8 Conclusion

The chapter reviewed five closely related papers which provide the scientific basis of our research. Among those are Supervised Models such as Random Forest and XGBoost; Deep Learning models such as LSTM, GRU; as well as Reinforcement Learning models such as DQN, PPO and A2C. Consistently, the analysis of all these methods showed that there is no one-size-fits-all solution to the problem at hand (Chu, 2025) and the choice of the algorithm depends on particular conditions of the crypto market.

There were two main limitations found in the relevant studies which led us to develop our own approach. On the one hand, in the majority of cases, the authors rely on the limited three-action strategy of DQN agent, namely 'Buy', 'Sell' and 'Hold'. In addition, no action is provided to manage the position size depending on the strength of the signal, as shown in Jay & Berlanga (2024) and Tay & Lim (2024). On the other hand, the lack of testing the developed algorithms over different time horizons reduces their practical value. This is especially evident in Jay & Berlanga (2024), Seabe et al. (2023) and Adedigba et al. (2025). Both shortcomings have been addressed. Based on the concept of dynamic scaling presented by Yang & Malik (2024), our action space was extended to incorporate two position sizing actions of 'Buy 30%' and 'Buy 70%'. Furthermore, all five algorithmic frameworks were examined using different test horizons on five cryptocurrencies.

CHAPTER 3:

Proposed Approach and Methodology

3.1 Introduction

As mentioned previously, none of the model families considered can be called optimal for all types of markets. Also, as a result of an overview of the existing literature, two key drawbacks were highlighted: the lack of position sizing when choosing actions in reinforcement learning models, and the lack of evaluation for several different time horizons. This chapter will introduce the solution to the stated problem. Firstly, six families of models were developed and tested: DQN, PPO, LSTM, GRU, Random Forest, and XGBoost on five cryptocurrencies: Bitcoin, Ethereum, Cardano, Solana, and Binance Coin with uniform settings and conditions. In this research, the DQN agent becomes the central innovation in terms of the architecture, since in the action space, it has two additional buy actions: Buy 30% and Buy 70%, to make the size of positions based on the model's confidence level. Training of all six model families is conducted using data until the end of December 2023. The testing of models on real data occurs in the 2024 year, out of the sample. In order to develop the model, it is necessary to begin by describing the dataset used, consisting of five major cryptocurrency pairs, as well as the methods of preprocessing them before using in further machine learning algorithms. Then, it is essential to highlight the feature engineering pipeline, which generates features of the technical and momentum nature. The next step is the description of the architecture and configuration of each model family. The design, choice of parameters, and reason for the chosen configuration should be provided for each algorithm. When designing the trading environment for RL agents, the Gymnasium library will be used, which allows creating the observation space, action space, and reward function. For supervised and ensemble models, a hybrid strategy will be described, according to which the algorithm's task will be exclusively generating buy signals, and sell signals will be generated using predetermined criteria of stop loss and take profit based on technical analysis indicators. In conclusion, the evaluation process will be explained, with metrics being Return on Investment, Sharpe ratio, Maximum Drawdown, and Win Rate. Four time horizons will be considered: 10 days, 1 month, 3 months, 6 months, and 1 year.

3.2 Dataset

This section describes the data used to train, validate, and test all six models. It covers the sources from which the raw price data was collected, the splitting strategy applied to ensure a strict temporal separation between training and testing periods, and the preprocessing steps carried out to prepare the data for each model family.

3.2.1 Data Sources

The dataset obtained in the course of the analysis was gathered from two sources of information, dependent on the evaluation horizon. For the evaluation horizons 1 month, 3 months, 6 months, and 12 months, the manual collection of OHLCV (open, high, low, close, volume) daily data from the Investing website has been conducted for each of the following cryptocurrencies: BTC (Bitcoin), ETH (Ethereum), ADA (Cardano), SOL (Solana), and BNB (Binance Coin). Investing provides clean and easily accessed datasets of daily history for various cryptocurrency pairs. For the evaluation horizon 10 days only, the collection of hourly OHLCV data has been conducted through programming means, using the CCXT library as an interface for obtaining the Binance exchange API data. The dedicated Python program has been written to collect the necessary data in the form of 1-hour candles, requesting 1000 candles per API call until the full history is received, as shown in Figure 3.1. Hourly data has been used for the 10-day evaluation horizon due to its greater granularity: evaluating a period of 10 days based on daily data results in receiving only 10 data points, which is not sufficient to evaluate any price dynamics or generate any trading signals in any of the six models.

```
# =========================
# CONFIG
# =========================
exchange = ccxt.binance()

symbol = 'BTC/USDT'
timeframe = '1h'

limit = 1000

since = exchange.parse8601('2018-01-01T00:00:00Z')
all_data = []

print("Start downloading...")

while True:
    try:
        ohlcv = exchange.fetch_ohlcv(symbol, timeframe, since, limit)

        if not ohlcv:
            break

        all_data += ohlcv
        since = ohlcv[-1][0] + 1

        print(f"Collected: {len(all_data)} candles")
        time.sleep(0.5)

    except Exception as e:
        print("Error:", e)
        time.sleep(5)
        continue
```

Figure 3.1: Hourly data collection script

Table 3.1 summarizes the five cryptocurrency assets used in this study, along with their trading symbols and the start dates of both the daily and hourly datasets.

Table 3.1: Overview of Collected Data per Asset

Asset	**Symbol**	**Daily Dataset Start Date**	**Hourly Dataset Start Date**
Bitcoin	BTC/USD	01-01-2017	07-11-2017
Ethereum	ETH/USD	01-01-2017	17-08-2017
Cardano	ADA/USD	31-12-2017	28-01-2017
Solana	SOL/USD	13-07-2020	28-01-2017
Binance Coin	BNB/USD	09-11-2017	28-01-2017

3.2.2 Data Splitting Strategy

All datasets had a strict temporal split applied in order to avoid data leakage and ensure the out-of-sample evaluation of all the experiments. The split strategy is different for the supervised and reinforcement learning models since there are major differences in the training procedures of these two groups.

All supervised and machine learning models – LSTM, GRU, Random Forest, and XGBoost – have three splits. In their training datasets, there is all available data up to August 2023. The validation set has data from September to December 2023 and is used only to monitor training progress during early stopping in the case of LSTM and GRU and for thresholds' tuning in the case of RF and XGBoost. Test data covers the year 2024.RL models – DQN and PPO – have all available data up to December 2023 as their training data. There is no separate validation set as the training process implicitly includes regularizing steps through the exploration-exploitation mechanism.

In both cases, normalization parameters are calculated based only on training data, while there is no re-fitting on the validation and test sets to ensure that no future information influences the preprocessing step. The test period is the same for all models covering the year 2024 with five horizons: 10 days,

1 month, 3 months, 6 months, and 12 months, enabeling a fair comparison under the same market conditions.

3.2.3 Data Preprocessing

Before passing the unprocessed OHLCV data into any algorithm, this data was processed through several processes to ensure consistency, stability, and compatibility with the standards of each model's input data.

3.2.3.1 Daily Data Cleaning

Formatting adjustments were needed to use the daily data downloaded from Investing.com. First of all, European numeric formatting was used, where the period was utilized as a thousands separator while the comma served as a decimal mark; therefore, it was necessary to change those to a float format. The date was stored in a string format DD/MM/YYYY and then set to be an index of the resulting DataFrame in chronological order. Volume was imported in abbreviated form as thousands ('K'), millions ('M'), and billions ('B'). A customized function was written to transform volume data into a complete numeric value, as shown in Figure 3.2. Finally, had a clean OHLCV DataFrame ready to be used.

```
for col in ["close", "open", "high", "low"]:
    df[col] = (
        df[col]
        .str.replace(".", "", regex=False)
        .str.replace(",", ".", regex=False)
        .astype(float)
    )
df.head()

def clean_volume(v):
    if pd.isna(v):
        return v
    if isinstance(v, str):
        v = v.strip()
        multiplier = 1
        if v.endswith("K"):
            multiplier = 1_000
            v = v[:-1]
        elif v.endswith("M"):
            multiplier = 1_000_000
            v = v[:-1]
        elif v.endswith("B"):
            multiplier = 1_000_000_000
            v = v[:-1]
        v = v.replace(",", ".")
        return float(v) * multiplier
    return float(v)

df["volume"] = df["volume"].apply(clean_volume)
```

Figure 3.2: Daily data cleaning script

3.2.3.2 Hourly Data Cleaning

The hourly data that we retrieved from Binance using its API and CCXT was already provided in a clean numerical format. All timestamps have been transformed into datetime objects from Unix milliseconds, while any duplicated entries that could appear due to paginated API access have been eliminated using drop duplicates. No additional data cleansing was needed, as shown in Figure 3.3.

```
df['timestamp'] = pd.to_datetime(df['timestamp'], unit='ms')

df = df.drop_duplicates()
```

Figure 3.3: Hourly Data Cleaning script

3.2.3.3 Missing Values and Feature Scaling

All rows that had missing values were then deleted by using the function dropna() after the calculation of the indicators. It was expected since the nature of the rolling window-based indicators like the RSI (periods 14), MACD (periods 26), and Bollinger Bands (periods 20), which have NaN for their first few values due to insufficient historical data.

Before training the supervised learning and ensemble methods models – Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Random Forest and XGBoost – the StandardScaler was fitted on the training set and used for the validation and testing set without refitting. This guarantees that the scaling will be based strictly on statistics calculated on the training data without introducing any future information through normalization, as illustrated in Figure 3.4.

```
# SCALE

sc = StandardScaler()
X_train = sc.fit_transform(train_df[FEATURES])
X_test = sc.transform(test_df[FEATURES])
```

Figure 3.4: XGBoost Feature scaling

As for the reinforcement learning models, Deep Q-Network (DQN) and Proximal Policy Optimization (PPO), feature normalization was conducted using mean and standard deviation on the basis of the training period, as shown in Figure 3.5, followed by clipping within the range of [-5, 5] to reduce the risk of large outliers, which could affect the gradients of the neural networks learning process, as illustrated in Figure 3.6.

```
FEAT_MEAN = train_df[FEATURES].mean().values.astype(np.float32)
FEAT_STD  = train_df[FEATURES].std().values.astype(np.float32) + 1e-8
```

Figure 3.5: DQN feature normalization

```
norm = (raw - FEAT_MEAN) / FEAT_STD
norm = np.clip(norm, -5, 5)
```

Figure 3.6: DQN normalization clipping

3.3 Model Architectures

In this section, three categories of models are discussed: reinforcement learning models (DQN, PPO), deep learning sequence models (LSTM, GRU), and machine learning ensemble models (Random Forest, XGBoost). Each model used in this work is detailed by architecture, configuration, and training procedure. It should be noted that the configurations presented for each model represent the primary setup used as a baseline. In practice, several components were adjusted across investment horizons to better align each model with the market dynamics of that specific time scale.

3.3.1 Deep Q-Network (DQN)

The Deep Q-Network (DQN) (Mnih et al., 2015)is a reinforcement learning algorithm that estimates the value of taking action a in state s, denoted as $Q(s, a)$, using a deep neural network. Unlike classical Q-learning, which uses a table to store Q-values for all possible state-action pairs, DQN uses a neural network to process large and complex state spaces such as market data.

To ensure stable training, DQN uses two mechanisms: experience replay, which stores past experiences in a memory buffer (Liu & Zou, 2017) and samples randomly from it during training to avoid learning from correlated consecutive steps, and a target network, which is a separate copy of the main network that is updated periodically to provide consistent learning targets.

the DQN agent is implemented using the Stable-Baselines3(Raffin et al., 2021) library with a custom Gymnasium trading environment(Brockman et al., 2016). The agent is trained on OHLCV data with 10 features including RSI, MACD, SMA, Bollinger Band, ATR, EMA, volatility, momentum, Volumr ratio and rate of change.

3.3.2 Action Space

While regular DQN models have utilized an action space that involves three types of actions: Buy, Sell, and Hold, our DQN model is set up on an action space of four types of actions as follows:

Table 3.2: DQN action space: discrete actions and their descriptions

Action	**Description**	**Capital Invested**
0	Buy 30% (small position — uncertain signal)	30% of available cash
1	Buy 70% (large position — strong signal)	70% of available cash
2	Sell — close full position	—
3	Hold — do nothing	—

The ability to choose among these actions constitutes the principal architectural innovation in this model, which takes the idea of dynamic scaling from Yang & Malik (2024). The availability of two entry levels allows the trading strategy to adjust itself based on the certainty of the information received, instead of allocating the same amount of money on each entry.

3.3.3 Observation Space

The observation vector or the state vector given to the agent at each timestep contains a sliding window of 10 past timesteps of the normalized feature vector, concatenated with 3 state variables :

$$o_t = [\text{features}_{t-10:t},\ \text{in_pos},\ \text{cash_ratio},\ \text{pos_size}] \tag{13}$$

where:

$\text{features}_{t-10:t}$	a flattened matrix of shape 20×10=200 values
in_pos	$\in \{0, 1\}$ indicates whether the agent currently holds a position
cash_ratio	= cash/initial_capital represents the fraction of capital remaining as cash
pos_size	is the current position value as a fraction of total portfolio value

20 features were used in DQN with 10 timestep which makes the observation size: (20×10)+3=203 dimensions

```
def _obs(self):
    raw  = self.df.iloc[self.step_idx - self.window:self.step_idx][FEATURES].values.
    astype(np.float32)
    norm = (raw - FEAT_MEAN) / FEAT_STD
    norm = np.clip(norm, -5, 5)

    in_pos     = 1.0 if self.units > 0 else 0.0
    cash_ratio = self.cash / self.initial_capital
    price      = float(self.df.loc[self.step_idx, "close"])
    total_val  = self.cash + self.units * price
    pos_size   = (self.units * price) / (total_val + 1e-8)

    obs = np.concatenate([norm.flatten(), [in_pos, cash_ratio, pos_size]])
    return np.nan_to_num(obs).astype(np.float32)
```

Figure 3.7: Observation vector for the DQN agent

3.3.4 Reward Function

The reward function is constructed to directly reward profitable trading behavior and discourage detrimental behavior, including holding, forced selling, and unrealized loss:

$$r_t = 200 \cdot \log\left(\frac{V_t}{V_{t-1}}\right) + r_{\text{sell}} + r_{\text{horizon}} + r_{\text{unrealized}} + r_{\text{friction}} + r_{\text{idle}} \tag{14}$$

where :

Table 3.3: Reward function components

Component	**Condition**	**Value**
r_{sell}	Profitable sell	$+60 \times$ profit_pct
r_{sell}	Losing sell	$+60 \times$ profit_pct (negative)
r_{horizon}	Forced exit at loss	-0.50
r_{horizon}	Forced exit at profit	-0.05
$r_{\text{unrealized}}$	Unrealized gain $> 2\%$	$+0.15$
$r_{\text{unrealized}}$	Unrealized loss $< -2\%$	-0.25
r_{friction}	Buy 30% action	-0.001
r_{friction}	Buy 70% action	-0.003
r_{idle}	Holding cash	-0.001

For an increase in the portfolio value, the log return will be positive, meaning that the agent is awarded. For a reduction in the portfolio value, the log return will be negative, meaning that the agent will be punished. The multiplication of the log returns by 200 increases the contrast between right and wrong actions in training(Ng et al., 1999; Sattarov et al., 2026).

Figure 3.8 shows the DQN reward function implementation, where each market action is assigned a reward or penalty to guide the agent toward profitable trading decisions.

```
# --- Reward ---
total_val = self.cash + self.units * price
log_ret   = np.log((total_val + 1e-8) / (prev_val + 1e-8))
reward    = 200.0 * log_ret

if action == 2 and self.trade_log:
    pct = self.trade_log[-1].get("profit_pct", 0) or 0
    reward += 60.0 * pct
    if pct < 0:
        reward += 60.0 * pct

if horizon_triggered and self.trade_log:
    pct = self.trade_log[-1].get("profit_pct", 0) or 0
    if pct < 0:
        reward -= 0.50
    else:
        reward -= 0.05

if self.units > 0:
    unrealized = (price - self.entry_price) / (self.entry_price + 1e-8)
    reward    += 0.05 * unrealized
    if unrealized > 0.02:
        reward += 0.15
    elif unrealized < -0.02:
        reward -= 0.25

if action == 0:
    reward -= 0.001
if action == 1:
    reward -= 0.003

if self.units == 0 and action == 3:
    reward -= 0.001
```

Figure 3.8: Reward Function for the DQN agent

3.3.5 Architecture

The architecture of the DQN agent integrates all the components described in the previous sections. The network receives the observation state s_t as input, which is passed through three linear layers with ReLU activations, and outputs a Q-value estimate $Q(s, a)$ for each action. The action with the highest Q-value is then chosen by the optimal policy and executed in the environment, as shown in Figure 3.9

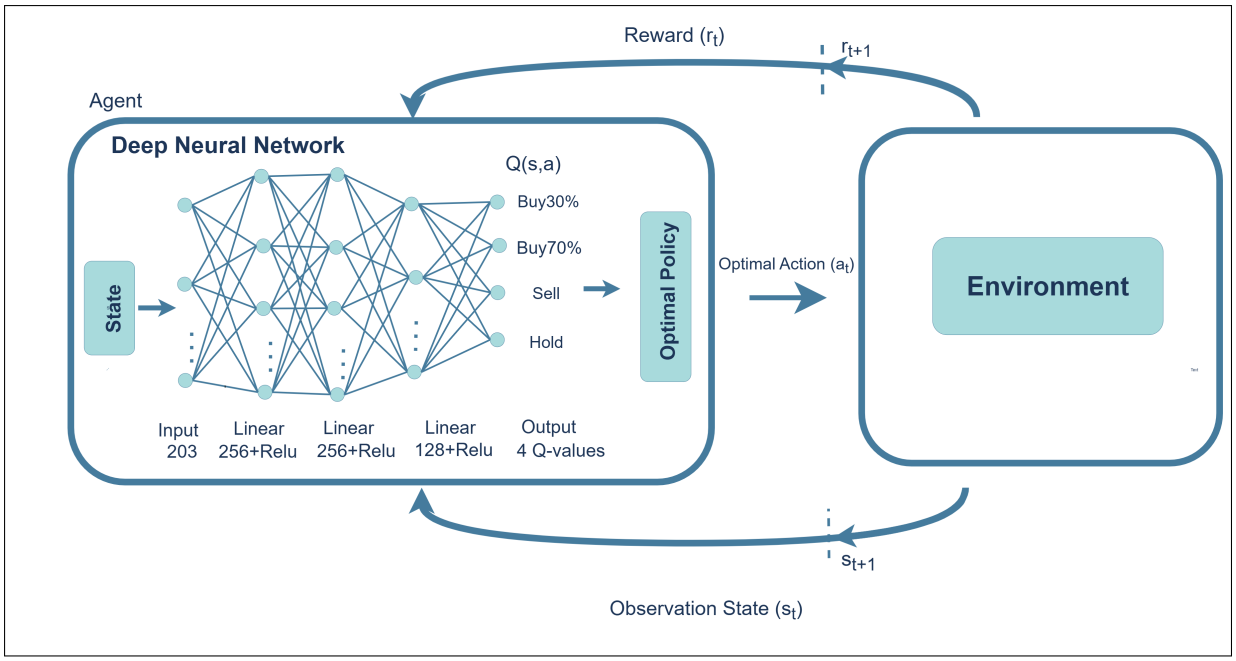


Figure 3.9: DQN Architecture

3.3.6 Hyperparameters

The DQN model was configured with the hyperparameters shown in Table 3.4

Table 3.4: Hyperparameters used for DQN training

Parameter	**Value**
Learning rate	3×10^{-4}
Replay buffer size	200 000
Batch size	512
Discount factor γ	0.97
Exploration fraction	0.35
Final exploration ϵ	0.15
Target network update interval	1 000 steps
Training frequency	Every 4 steps
Total timesteps	1 000 000
Device	GPU

3.4 Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) (Schulman et al., 2017), is a policy gradient algorithm that directly learns a policy $\pi(a|s)$ in order to transfer the states to actions, instead of estimating the action values, as DQN does. PPO instead restricts the policy update by clipping the probability ratio between the new and the old policy so that the update does not go too far away from the last update in each iteration. This clipping mechanism makes PPO more stable and sample efficient than previous one. PPO collects a fixed number of steps interacting with the environment, computes advantages with a critic network, updates the actor and critic simultaneously, then discards the collected data. This is different from DQN which relies on experience replay and a target network.

In the current project, the PPO agent will be realized with the help of Stable-Baselines3 but with the same Gymnasium environment(Brockman et al., 2016) used by the DQN algorithm.

3.4.1 Action Space

The PPO agent uses a standard three-action space:

Table 3.5: Action space for the PPO agent

Action	Description	Capital Invested
0 — Buy	Enter position	30%
1 — Sell	Close full position	—
2 — Hold	Do nothing	—

3.4.2 Observation Space

The observation vector given to the agent at each timestep contains a sliding window of 10 past timesteps of the normalized feature vector, concatenated with two state variables :

$$o_t = [\text{features}_{t-10:t},\ \text{in_pos},\ \text{cash_ratio}] \tag{15}$$

10 features were used in PPO with 10 timestep which makes the observation size: (10×10)+2=102 dimensions

```
def _obs(self):
    raw = self.df.iloc[
        self.step_idx - self.window : self.step_idx
    ][FEATURES].values.astype(np.float32)

    norm = (raw - FEAT_MEAN) / FEAT_STD
    norm = np.clip(norm, -5, 5)

    in_pos     = 1.0 if self.units > 0 else 0.0
    cash_ratio = self.cash / self.initial_capital

    obs = np.concatenate([norm.flatten(), [in_pos, cash_ratio]])
    return np.nan_to_num(obs, nan=0.0, posinf=0.0, neginf=0.0).astype(np.float32)
```

Figure 3.10: Observation vector for the PPO agent

3.4.3 Reward Function

The reward function for the PPO model is similar to that of the DQN reward function outlined in the previous section but few differences due to the simplified nature of its actions and constant position sizing:

$$r_t = 200 \cdot \log\left(\frac{V_t}{V_{t-1}}\right) + r_{\text{sell}} + r_{\text{unrealized}} + r_{\text{friction}} \tag{16}$$

Table 3.6: Reward function components

Component	Condition	Value
r_{sell}	Profitable sell	$+50 \times$ profit_pct
r_{sell}	Losing sell	$+40 \times$ profit_pct (negative)
$r_{\text{unrealized}}$	Positive unrealised PnL	$+0.02 \times$ unrealized
$r_{\text{unrealized}}$	Negative unrealised PnL	$+0.05 \times$ unrealized (negative)
r_{friction}	Buy action	-0.01

Figure 3.11 illustrates the reward signal received by the PPO agent across different market actions, highlighting how profitable trades, unrealized PnL, and friction penalties collectively shape the learning objective

```
# --- REWARD ---
log_ret = np.log((total_val + 1e-8) / (prev_val + 1e-8))
reward  = 200.0 * log_ret

if action == 1 and self.trade_log:
    pct     = self.trade_log[-1].get("profit_pct", 0) or 0
    reward += 50.0 * pct
    if pct < 0:
        reward += 40.0 * pct

if self.units > 0:
    unrealized = (price - self.entry_price) / (self.entry_price + 1e-8)
    if unrealized > 0:
        reward += 0.02 * unrealized
    else:
        reward += 0.05 * unrealized

if action == 0:
    reward -= 0.01

# ----------------------------------------

self.portfolio_values.append(total_val)
self.actions_taken.append(int(action))
self.step_idx += 1
done = self.step_idx >= len(self.df) - 1
return self._obs(), float(reward), done, False, {}
```

Figure 3.11: Reward Function for the PPO agent

PPO employs two separated networks, the actor network and the critic network. The actor network takes the observation state s_t as input and outputs a probability distribution $\pi(a|s)$ over the three available actions (Buy, Sell, Hold), from which an action is sampled during training. The critic network receives the same input and produces a single scalar value $V(s)$ which is an estimate of the expected return from state s_t. Both networks are of the same architecture of two fully connected linear layers with ReLU activations but play fundamentally different roles: the actor takes the decision of what action to take whereas the critic evaluates how good is the current state (Konda & Tsitsiklis, 2003).The output of the critic is used to calculate the advantage estimate $\hat{A}_t$ which determines how much better or worse the action taken was than expected, and is used to update the actor via the clipped surrogate objective. This is shown in Figure 3.12.

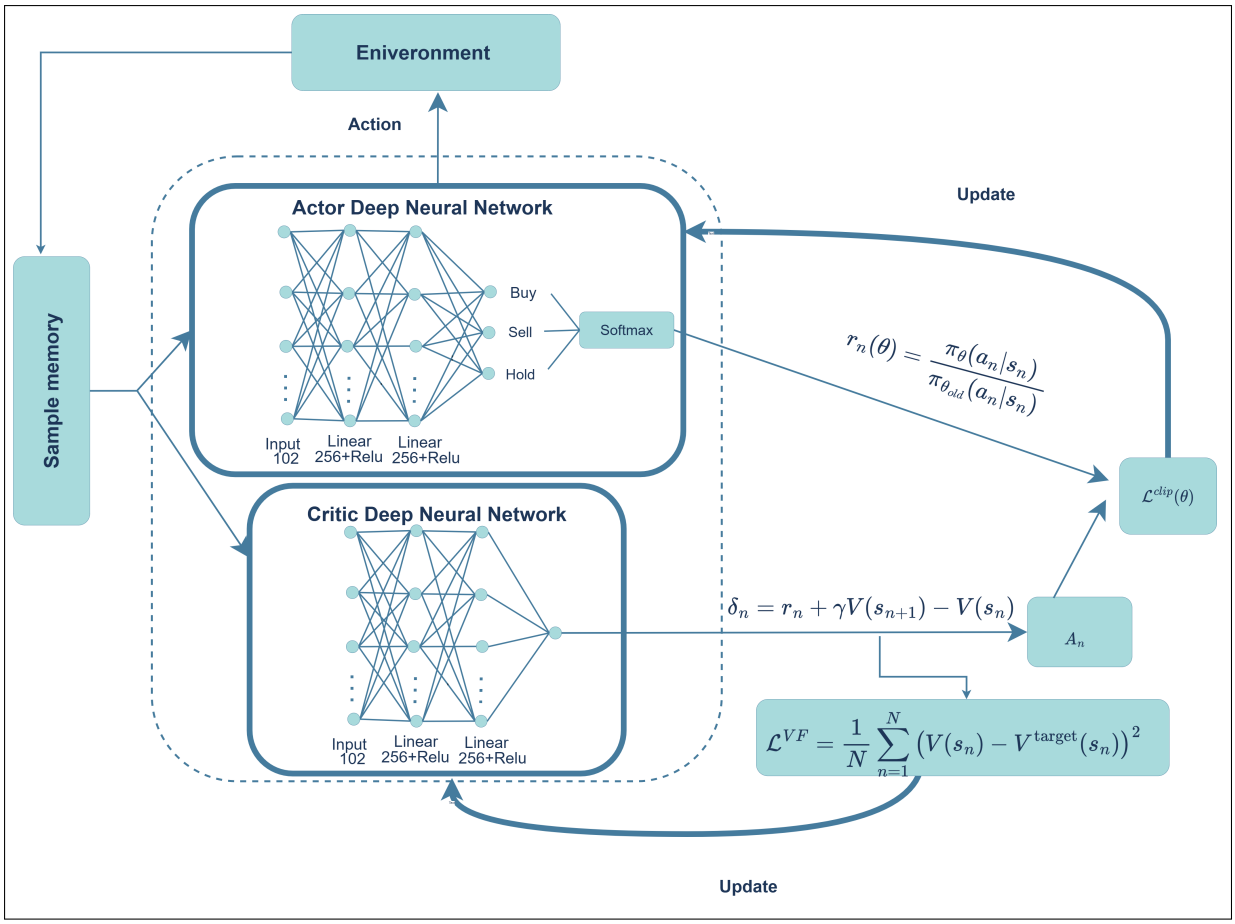


Figure 3.12: PPO agent actor and critic architecture

3.4.4 Hyperparameters

The PPO model was configured with the hyperparameters shown in Table 3.7

Table 3.7: Hyperparameters used for PPO training

Parameter	Value
Learning rate	2×10^{-4}
Rollout steps n	2 048
Batch size	256
Discount factor γ	0.995
Entropy coefficient	0.005
Clip range	0.2
Total timesteps	1 000 000
Device	GPU

3.5 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture introduced by (Hochreiter & Schmidhuber, 1997)for learning patterns in sequential data over long time horizons. In contrast to traditional RNNs, which suffer from the vanishing gradient problem and struggle to remember information over long sequences, LSTM introduces a gating mechanism that includes three gates: the input gate, the forget gate, and the output gate. These gates regulate the information that is to be stored, discarded, or passed forward through the memory cell of the network. This allows the memory cell to store short-term as well as long-term dependencies of the time series data. This makes LSTM

especially suitable for financial time series, where market patterns often depend on context accumulated over many past timesteps.

In this research, LSTM is applied for the purpose of generating binary buy signals instead of forecasting prices: the system is trained to predict whether the 3-day return following the window of market features exceeds a threshold level.

3.5.1 Label Construction

For every timestep i, we calculate the forward 3-day return using the formula below:

$$r_i = \frac{p_{i+3} - p_i}{p_i} \tag{17}$$

In case $r_i > \theta$, where θ is the 33rd percentile of absolute returns calculated on the training data set, the label is set equal to BUY (class 0). Otherwise, NO-BUY (class 1) will be used.

```
fwd    = df["close"].pct_change(3).shift(-3).dropna()
THOLD  = float(fwd.abs().quantile(0.33))
```

Figure 3.13: LSTM Label Construction

3.5.2 Loss Function

Focal Loss (Lin et al., 2017) is applied to deal with class imbalance and increase BUY recall instead of the conventional cross-entropy function:

$$\mathcal{L}_{\text{focal}} = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{18}$$

Where $\gamma = 2.0$ and $\alpha = 0.6$

This can downweight easy negative examples and focus training on difficult BUY signal.

```
# FOCAL LOSS -- alpha=0.6 rewards BUY recall

class FocalLoss(nn.Module):
    def __init__(self, gamma=2.0, alpha=0.6):
        super().__init__()
        self.gamma = gamma
        self.alpha = alpha

    def forward(self, logits, targets):
        ce = nn.functional.cross_entropy(logits, targets, reduction='none')
        pt = torch.exp(-ce)
        at = torch.where(targets == 0,
                         torch.tensor(self.alpha),
                         torch.tensor(1 - self.alpha))
        return (at * (1 - pt) ** self.gamma * ce).mean()
```

Figure 3.14: LSTM loss Function

3.5.3 Buy Threshold

The threshold used for inference at the buy level is chosen from the 20th percentile of buy probability predictions for the validation dataset and labeled as BUY. The advantage here is that we use a dynamic threshold based on the confidence distribution of the model instead of a static value.

3.5.4 Exit Strategy

the rule-based system controls sell decisions which operates independently from the LSTM model:

Figure 3.15 illustrates the rule-based exit strategy, showing how the four sell conditions interact to determine the optimal point at which an open position is closed.

Table 3.8: Sell conditions for closing a position

Rule	Condition
Stop Loss	Position PnL $\leq -6\%$
Take Profit	Position PnL $\geq +15\%$
RSI + MACD filter	RSI > 72 and MACD histogram < 0
Horizon exit	Position held for more than horizon days

```
def should_sell(price, entry, rsi_v, macd_h, step, entry_step, horizon):
    pnl = (price - entry) / (entry + 1e-8)
    if pnl <= -STOP_LOSS_PCT:                   return True, "stop_loss"
    if pnl >=  TAKE_PROFIT_PCT:                  return True, "take_profit"
    if rsi_v > RSI_OVERBOUGHT and macd_h < 0:    return True, "rsi_macd"
    if (step - entry_step) >= horizon:           return True, "horizon"
    return False, ""
```

Figure 3.15: LSTM Exit Strategy

3.5.5 Position Sizing

The number of units to be bought or sold is calculated using this formula:

$$n_{\text{units}} = \min\left(\frac{V \times r_{\%}}{p \times \text{SL}_{\%}} \times s_c,\ \frac{V \times \text{maxpos}}{p}\right) \tag{19}$$

where:

- V: current portfolio value.
- $r_{\%} = 2\%$: maximum risk per trade.
- $\text{SL}_{\%} = 6\%$: stop loss percentage.
- s_c: confidence scaling factor.
- $\text{maxpos} = 40\%$: maximum position size (fraction of portfolio).

```
# POSITION SIZING -- risk-based

def calc_position_size(portfolio_value, price, confidence, buy_thr):

    risk_dollars = portfolio_value * PORTFOLIO_RISK_PCT
    base_units   = risk_dollars / (price * STOP_LOSS_PCT)
    conf_scale   = 1.0 + min((confidence - buy_thr) / buy_thr, 1.0)
    scaled_units = base_units * conf_scale
    max_units    = (portfolio_value * MAX_POSITION_PCT) / price
    return min(scaled_units, max_units)
```

Figure 3.16: LSTM Position Sizing

3.5.6 Architecture

The LSTM model architecture is designed to learn the temporal dependencies in the financial time series data. The network takes as input a sequence of 10 timesteps, each described by 10 engineered features, yielding a 10×10 observation matrix. This input is fed to one layer of LSTM with 96 hidden units, which allows the model to remember relevant past patterns with its recurrent memory cells. To avoid overfitting, a dropout layer with rate $p = 0.35$ is added after the LSTM output(Srivastava et al., 2014). The representation obtained is then fed into a fully connected layer that reduces the dimensionality from 96

to 48 neurons, followed by a ReLU activation function and a lighter dropout layer with $p = 0.25$. Finally, a linear output layer transforms the 48-dimensional representation to 2 output logits, corresponding to the binary classification of market signals. The complete architecture is illustrated in Figure 3.17.

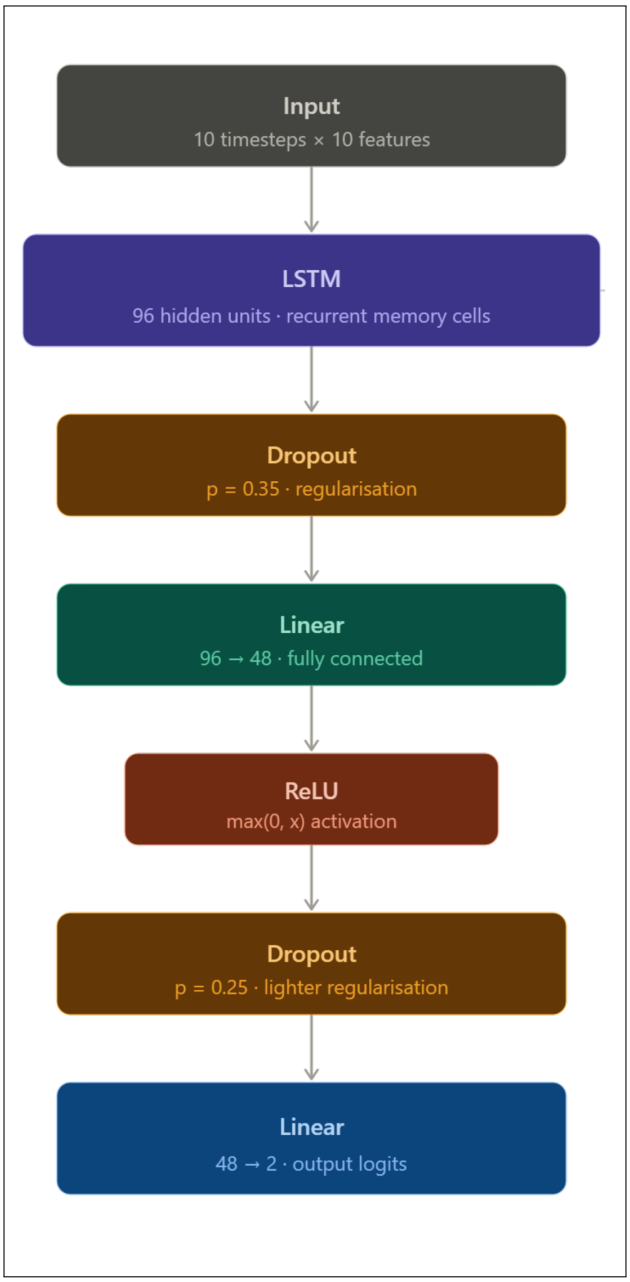


Figure 3.17: LSTM architecture

3.5.7 Hyperparameters

The LSTM model was configured with the hyperparameters shown in Table 3.9.The Adam optimizer (Kingma & Ba, 2015) was used with a cosine annealing learning rate scheduler and early stopping (Prechelt, 1998) with a patience of 15 epochs to prevent overfitting.

Table 3.9: Training hyperparameters for the LSTM model

Parameter	**Value**
Optimizer	Adam
Learning rate	3×10^{-4}
Weight decay	10^{-4}
Batch size	32
Max epochs	80
Early stopping patience	15 epochs
LR scheduler	CosineAnnealing ($T_{\max} = 60$, $\eta_{\min} = 10^{-5}$)
Class balancing	WeightedRandomSampler

3.6 Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is a recurrent neural network architecture proposed by Cho et al. (2014) as a simplification of the Long Short-Term Memory (LSTM) network. Like the LSTM, the GRU is designed to capture long range temporal dependencies in sequential data, making it particularly well suited for financial time series modelling (Goodfellow et al., 2016; Hamayel & Owda, 2021). However, GRU does this in a more compact way replacing the three gates of the LSTM, that is the input gate, forget gate, and output gate with just two gates, that is the **reset gate** and the **update gate**(Elsayed et al., 2019). The reset gate controls the degree of forgetting of the previous hidden state and the update gate controls the degree of adding new information into the current hidden state.Furthermore, this reduction in gating complexity also results in fewer trainable parameters than the LSTM, which usually results in faster training, lower memory consumption and reduced risk of overfitting.

3.6.1 Architecture

The GRU network architecture shown in Figure 3.18 is similar to the LSTM architecture described in Section 3.5.6.the only difference being that the LSTM recurrent cell is replaced by a GRU cell.

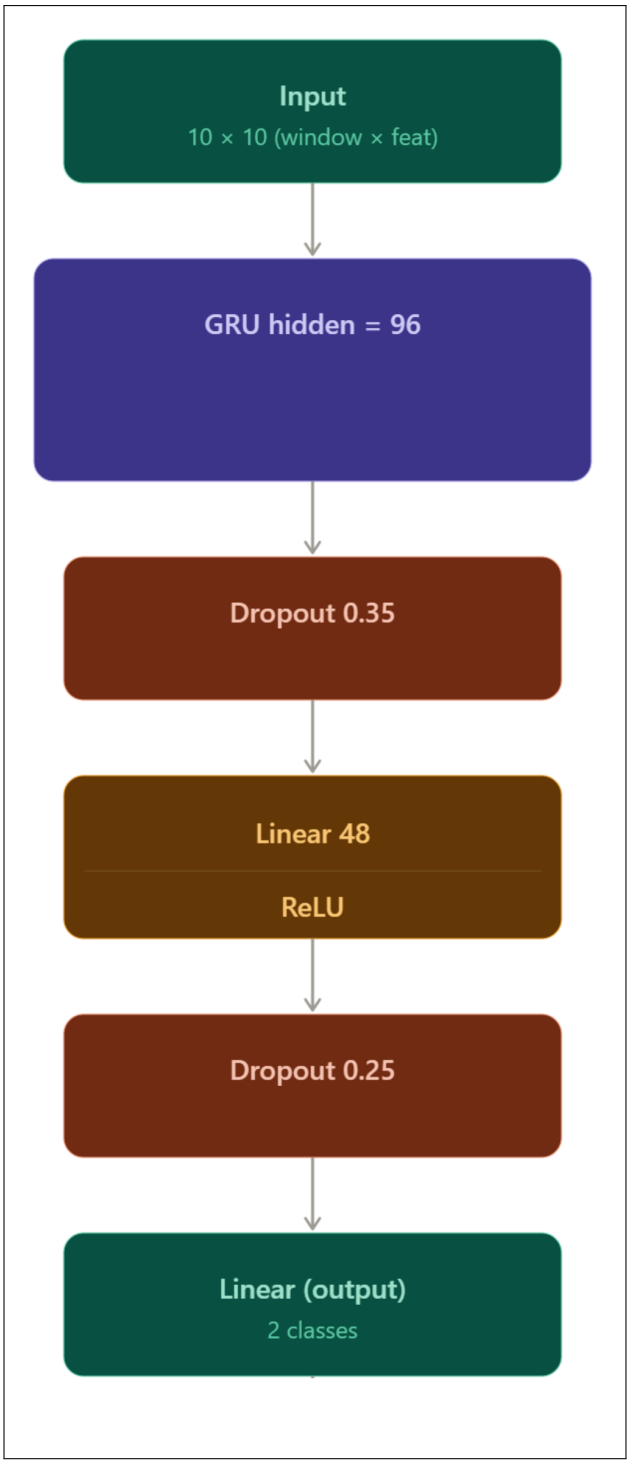


Figure 3.18: GRU architecture

3.7 Random Forest

Random Forest is used for classification in the present study as a classifier for BUY signal (Breiman, 2001), where an input feature vector of 7 technical variables is provided to the network to determine if the forward three candle returns are more than a certain threshold value of 0.0005, resulting in either a class 0 signal (BUY) or class 1 signal (NO-BUY). The deep learning models require a sequence of windowed inputs, whereas Random Forest takes a flat input vector per time step using its non-parametric properties.

3.7.1 Label Construction

The construction of the labeling algorithm is similar to LSTM and GRU. the fixed value of $\theta = 0.0005$ is employed instead of using the adaptive approach which depends on the distribution of returns from the training dataset, as shown in Figure 3.19.

```
def make_labels(prices, th=0.0005):
    y = []
    for i in range(len(prices) - 3):
        r = (prices[i+3] - prices[i]) / prices[i]
        y.append(0 if r > th else 1)
    return np.array(y)
```

Figure 3.19: Random Forest Label Construction

3.7.2 Class Imbalance

The class imbalance is addressed using the class_weight="balanced" attribute(Chawla et al., 2002), where each observation is weighted by the inverse proportion of its class in the training process.

3.7.3 Exit Strategy

The same as LSTM and GRU: rule-based stop loss (6%), take profit (15%), RSI+MACD filter and risk-based position sizing (2% portfolio risk per trade, maximum 40%) are used for exit strategy.

3.7.4 Hyperparameters

The Random Forest model was configured with the hyperparameters shown in Table 3.10

Table 3.10: Random Forest hyperparameters

Parameter	Value
Number of estimators	300
Max depth	10
Min samples leaf	20
Class weight	balanced
Random state	42

3.8 XGBoost

In this paper, XGBoost is utilized as a binary classifier for BUY signals (Chen & Guestrin, 2016)using the same 7 technical variables and label creation as that of the Random Forest model. the model determines if the future 3-candle return exceeds 0.0005, resulting in either a BUY (0) or NO-BUY (1) signal.

3.8.1 Class Imbalance

Class imbalance handled in XGBoost is the scale_pos_weight parameter, which is calculated as the ratio of samples belonging to class "NO-BUY" to class "BUY". This upweights the minority BUY class during gradient computation.

3.8.2 Exit Strategy

The same as LSTM, GRU and Random forest

3.8.3 Buy Threshold

As shown in Figure 3.20, Xgboost uses a simpler heuristic threshold:

$$\theta_{\text{buy}} = \bar{p}_{\text{val}} \times 0.95 \tag{20}$$

Where :

$\bar{p}_{\text{val}}$ is the mean probability of BUY prediction on the validation dataset. This makes the threshold lower than the mean level of confidence, offering some improvement in the recall of BUY over the default 0.5 threshold.

```
BUY_THR = float(val_probs.mean()) * 0.95
print(f"   BUY threshold : {BUY_THR:.3f}")
```

Figure 3.20: XGBoost Buy Threshold

3.8.4 Hyperparameters

The XGBoost model was configured with the hyperparameters shown in Table 3.11

Table 3.11: Hyperparameters for the gradient boosting model

Parameter	**Value**
Number of estimators	300
Max depth	3
Learning rate	0.03
Random state	42
Subsample	0.7
colsample	0.7

3.9 Evaluation Framework

This section describes the testing methodology used to assess the performance of all six models. It covers the five time horizons over which each model was evaluated, the financial metrics used to measure performance, and the benchmark strategy against which all results are compared.

3.9.1 Test Periods

All six models have been tested over five time horizons that have been chosen from the same year 2024 to provide out-of-sample analysis of the models. The horizons are created to capture model behavior across different market durations from very short-term to full-year:

10 days: March 1–10, 2024 1 month: March 2024 3 months: March–May 2024 6 months: March–August 2024 12 months: January–December 2024

The year 2024 has been chosen for the reason that this particular year offered varied market conditions in just one year, including a bullish run during Q1, corrective action in Q2, consolidation in Q3, and finally a rally in Q4. This year thus provided tough testing conditions.

One of the main contributions of this research has been in the multi-horizon testing aspect. All other studies have considered only one horizon at a time when assessing their system performance.

3.9.2 Evaluation Metrics

Performance is assessed using three financial metrics. The Sharpe Ratio, introduced by Sharpe (1994), measures risk-adjusted return. Maximum Drawdown, as defined by Magdon-Ismail & Atiya (2004), captures the worst peak-to-trough loss over the evaluation period. Average Profit per Trade measures the efficiency of each entry signal.

Table 3.12: Evaluation metrics used to assess model performance

Metric	**Formula**	**What it Measures**
Sharpe Ratio	$\frac{\mu_r}{\sigma_r} \times \sqrt{ann}$	Risk-adjusted return
Maximum Drawdown	$\min\left(\frac{V_t - V_{peak}}{V_{peak}}\right)$	Worst peak-to-trough loss
Avg Profit per Trade	$\frac{1}{N}\sum_{i=1}^{N} profit_pct_i$	Efficiency per entry signal

Where:

μ_r = mean of daily returns, σ_r = standard deviation of daily returns, ann = number of annual periods
V_t = portfolio value at time t, V_{peak} = running maximum of portfolio value.
N = total number of closed trades, profit_pct$_i$ = percentage profit of trade i.

3.9.3 Benchmark

The performance of all models is compared with the Buy and Hold method, implemented using the same asset, the same testing period, and a starting investment capital of $4,000. The Buy and Hold method refers to a model in which an investor purchases and holds the investment until the end without trading.

3.10 Conclusion

All the necessary steps for creating our trading algorithm were provided in this chapter from collecting and preprocessing the dataset, through building our models, and finally evaluating our experiment. This chapter aimed to ensure that our experiment would be fair and reliable in evaluating the performance of each of the six models on five different cryptocurrencies across five different time frames.

CHAPTER 4:

Results and Evaluation

4.1 Introduction

In this chapter, the experimental results of all six models — DQN, PPO, LSTM, GRU, Random Forest, and XGBoost — developed in Chapter 3 are analyzed over five time horizons and five cryptocurrency assets. The main purpose of this experiment is to make a thorough comparison of all models under equal testing conditions, in accordance with the performance measures established in the evaluation methodology. The chapter begins with a visual analysis of the 2024 market data to contextualize the testing environment, then presents results per individual cryptocurrency, and concludes with a cross-asset comparative analysis that summarizes the overall findings across all models and horizons.

4.2 Experimental Setup

This section describes the software environment and tools used to implement and evaluate all six models.All experiments were conducted under identical configurations to ensure fair and reproducible comparisons across models and assets.

4.2.2.1 Development Environment

All experiments were executed in Python programming language through two different notebook-based cloud services. which are Google Colab and Kaggle Notebooks, since they allow free GPU usage. Both platforms used Python 3.10 in the form of a Jupyter notebook with identical library version installations.

4.2.2.2 Libraries and Frameworks

The following libraries were used across all six models: Stable-Baselines3 (Raffin et al., 2021), Gymnasium (Brockman et al., 2016), PyTorch (Paszke et al., 2019), Scikit-learn (Pedregosa et al., 2011), Pandas (McKinney, 2010), NumPy (Harris et al., 2020), and Matplotlib (Hunter, 2007). Table 4.1 provides a description of each library and its role in the experiment.

Table 4.1: Libraries and Descriptions

Library	**Description**
Stable-Baselines3	Reinforcement learning library providing ready-to-use implementations of DQN and PPO
Gymnasium	Toolkit for building custom reinforcement learning environments
PyTorch	Deep learning framework used to build and train the LSTM and GRU architectures
Scikit-learn	Machine learning library used for Random Forest, feature scaling, and evaluation metrics
Pandas	Data manipulation library used for loading, cleaning, and processing time series data
NumPy	Numerical computing library used for feature engineering and array operations
Matplotlib	Plotting library used to generate all portfolio curves and result visualizations
CCXT	Cryptocurrency exchange library used to collect hourly OHLCV data from Binance

4.3 Data Visualization

Before discussing the findings from the models, this section will give a visual look at the market environments that all the models faced throughout the year 2024. It is important to understand the price behavior and the pattern of the asset's volatility, as presented in six visualizations.

4.3.3.1 Historical Price Evolution Plot

The figure 4.1 below presents the closing price of all five assets from their respective listing dates through early 2026. There are some findings that come out of this long-term analysis. First, Ethereum and Binance have achieved the highest growth levels since 2017, reaching an index value of over 10,000 by 2024, due to their earlier listings and continuous growth trend.On the other hand, Solana had an impressive rally in 2021 but was followed by a correction in 2022.Finally, Bitcoin demonstrated relatively stable trends of growth while cardano was the weakest performer in the group by the end of the period. in conclusion, the shaded area for 2024 indicates that the test period occurred in an upswing cycle of all assets after the bear market of 2022-2023, means that all assets were in an active growth phase.

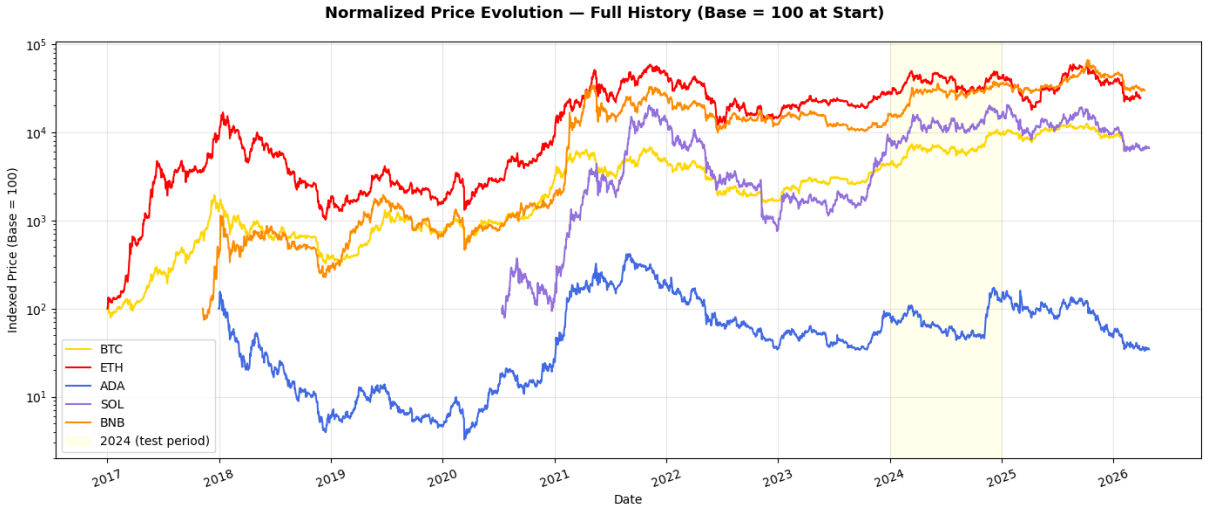


Figure 4.1: Historical Price evolution plot

4.3.3.2 Price evolution in 2024 Plot

The figure 4.2 below presents the analysis to the year 2024, There are three distinct market regimes. The first is a strong upwards move throughout Q1, during which the BTC, ETH, BNB, and SOL moved to levels between 125% and 200% above their January level by March dua to the Bitcoin halving event. The second is a corrective move during Q2, where in most coins moved substantially below their March high, the most notable of which is Cardano, whose price fell to about 50% of its January level. The third is an additional move upwards during Q4, alongside the US election, in which BTC made new high and BNB surpassed 225% of its January value.

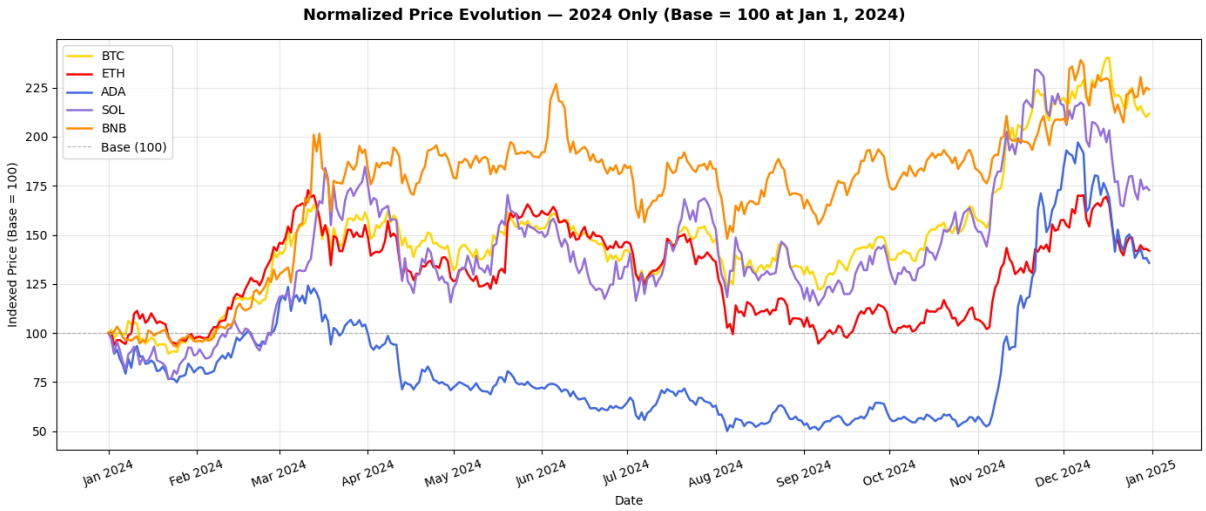


Figure 4.2: Price evolution in 2024 plot

4.3.3.3 Technical Indicators in 2024

As shown in Figure 4.3, this section presents four technical indicators for BTC closing prices in 2024, organized into four plots with a shared time dimension.

In the first plot, the closing prices are shown along with the Bollinger Bands calculated using the 20-day window. Notice that during both March and November/December rallies, the Bollinger Bands increased greatly, signifying increased volatility in these periods. Closing prices hit or even broke through the upper bound twice in quarter 1, quarter 2 and quarter 4, indicating overbought situations, while the price fell below the lower band during the August correction period.

Next plot was the 14-day RSI, where we observe that RSI values crossed 70 four times : in February, then again in March, July, September, October, and November, indicating an overbought situation, while falling below 30 five times, which indicates an oversold situation.

In the third plot, we observe the MACD histogram, which shows strongly positive values during quarter 1 and quarter 4, indicating bullish momentum, and shows negative values during quarter 2 and quarter 3 with brief recoveries, thus giving a very clear signal about the trend direction and momentum at all times. In the final plot, we display the daily BTC volume, which had notable peaks in November and December. This confirms that both rallies indeed were real events and not noise.

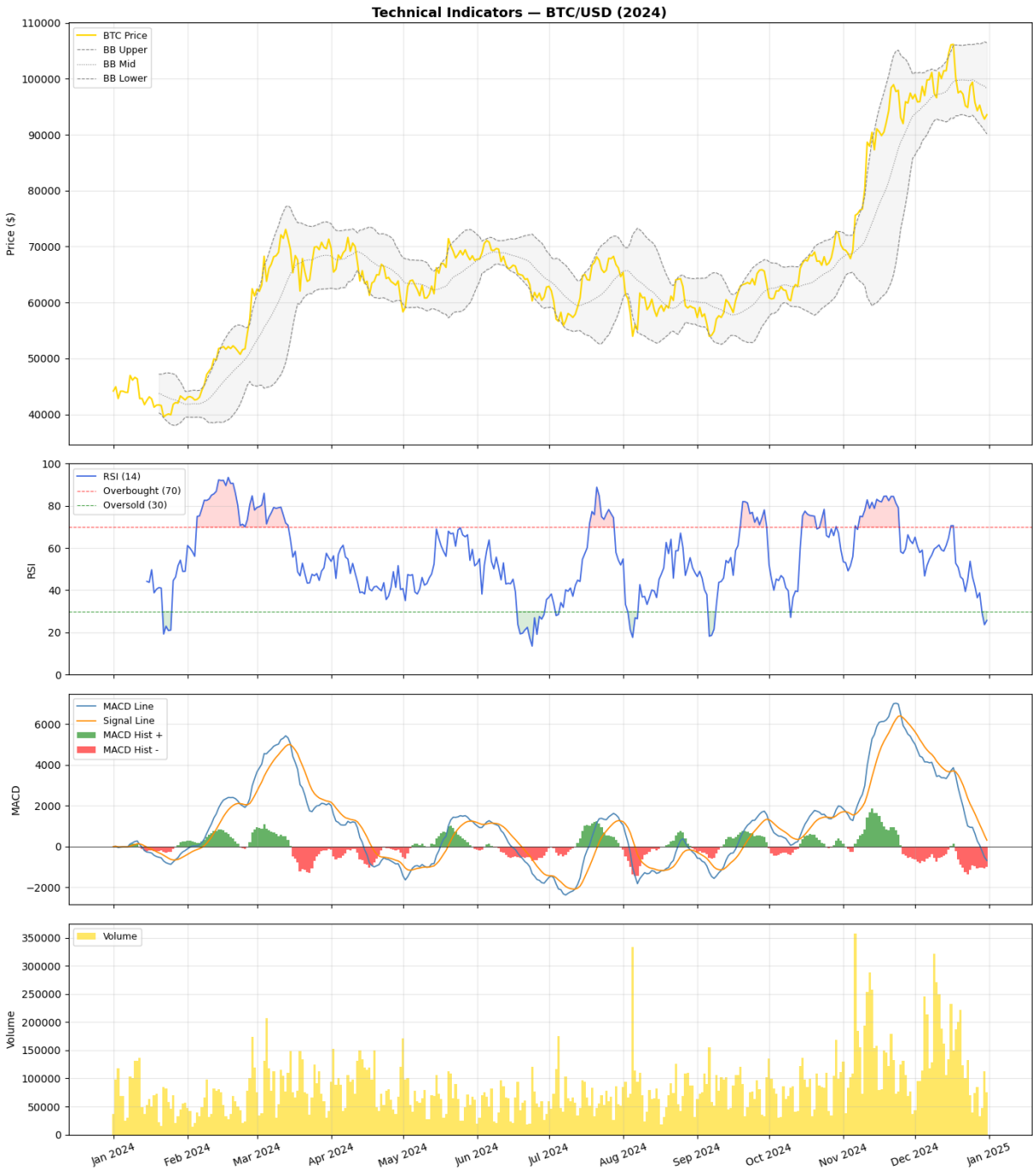


Figure 4.3: Technical indicators for BTC closing prices in 2024

4.3.3.4 Rolling 30 Day Annualized Volatility

The Figure 4.4 below shows the 30 day rolling annualized volatility of all five assets during the entire year of 2024. There are several noticeable things.

Firstly, all five assets had a sharp increase in volatility in March – April 2024, which became the most volatile time for all assets at once during the year. Secondly, volatility decreased sharply in May – June 2024. Thirdly, there was also another sharp increase in volatility in November – December 2024, which corresponds to the post-election Q4 rally.Speaking about the level of volatility in the year under consideration, cardano demonstrated the maximum volatility among all assets considered in 2024. The maximum value was reached in late November and amounted to around 1.2, twice more than the volatility of Ethereum in the same month. At the same time, Solana and Ethereum demonstrated relatively high volatility levels, while Bitcoin and Binance Coin were the least volatile.

In conclusion, assets with higher volatility levels are harder to trade effectively due to increased unpredictability (Baur & Dimpfl, 2018).

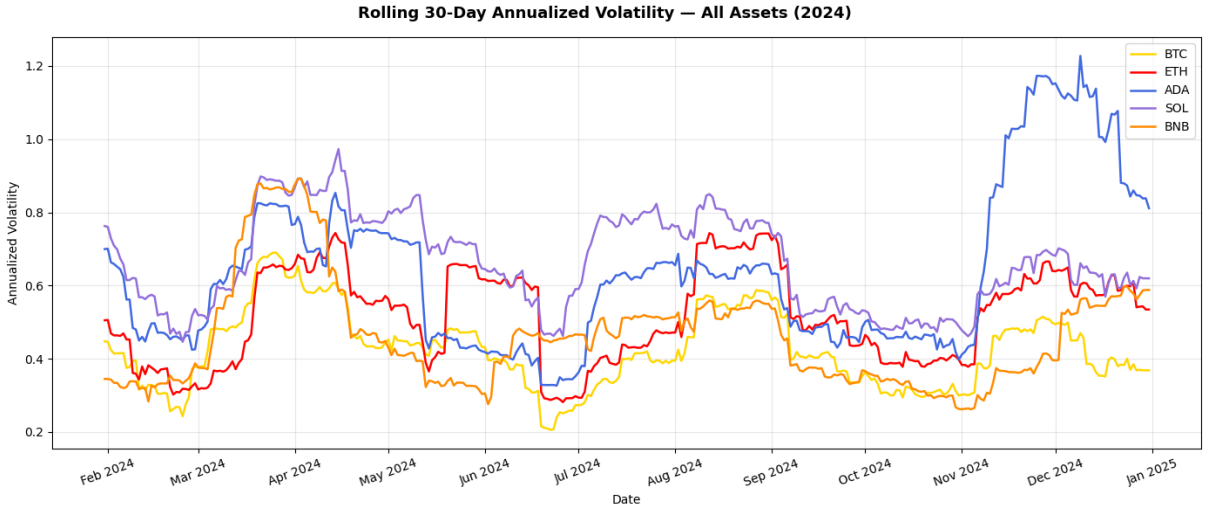


Figure 4.4: Rolling 30 Day Annualized Volatility in 2024

4.3.3.5 Maximum Drawdown by Quarter

Figure 4.5 shows the maximum drawdown recorded by each asset for each quarter of 2024. The bar chart reveals that no quarter was free of significant drawdowns in all assets, but the intensity varied by period and asset. Quarter 2 (April–June) was the most challenging quarter. Cardano recorded a maximum drawdown of −39%, Solana −34%, Ethereum −22%, Bitcoin −18% and Binance Coin −20%. This confirms that quarter 2 was the dominant risk period in the study. Quarter 3 (July–September) was similarly difficult, with Ethereum and Solana recording drawdowns of −37% and −32% respectively, while Bitcoin and Binance Coin had better performance with a drawdown of approximately −20%. Quarter 1 and quarter 4 showed the most moderate drawdowns, consistent with the upward trends observed during those periods. This graph can directly explain the relative performance of models in the test periods.

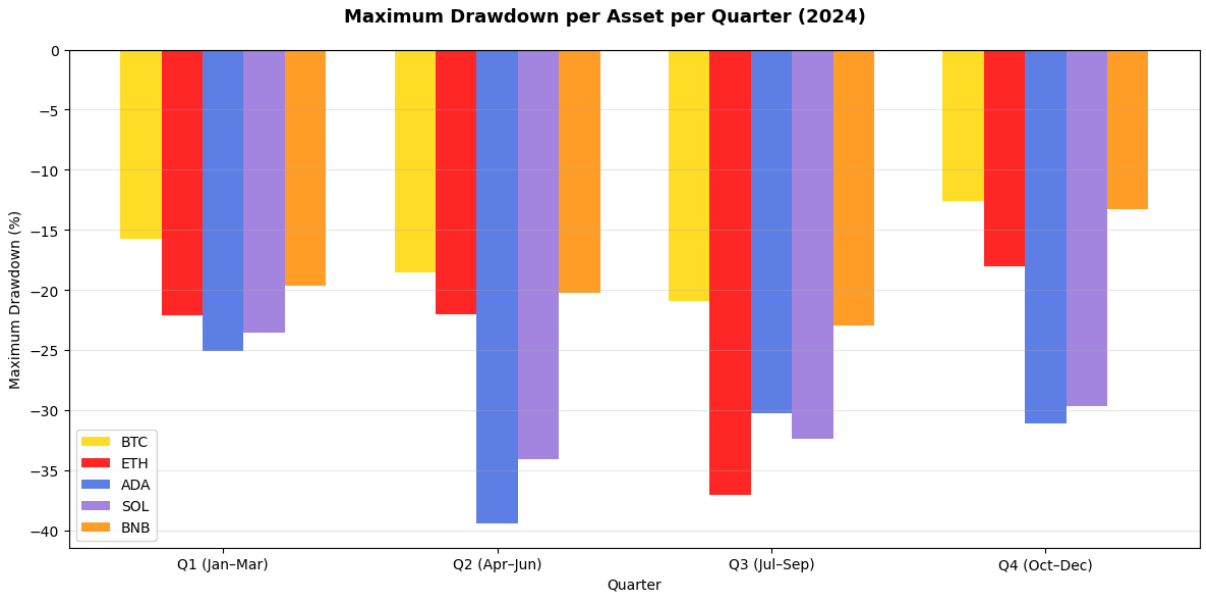


Figure 4.5: Maximum Drawdown by Quarter in 2024

4.4 Experimental Results and Comparative Evaluation

This section presents and analyzes the backtesting results of all six models across five cryptocurrency assets and five investment horizons, evaluated on 2024 out-of-sample market data using performance measures.

4.4.4.1 per Asset Comparative Analysis

The performance results of the mentioned six models, namely RF, XGBoost, LSTM, GRU, DQN and PPO are shown below for five cryptos, with the time periods at 10 days, one month, three months, six months, and twelve months. For each of these cryptos, the returns along with the worst maximum drawdowns, the sharpe ratios are provided and The returns of buy and hold strategy (B&H).

4.4.1.1 Bitcoin (BTC)

The table 4.2 below presnts the most liquid asset in the set and the most studied is Bitcoin. Its return profile shows the different market conditions during the testing period, such as trending up and down and sideways movement.

The LSTM network proves to be the strongest performer in BTC portfolios, generating positive returns in four horizons and achieving the lowest worst drawdown (-8.16 %) among all models. It can be concluded that LSTM, being the sequence-learning model, shows strong performance in recognizing patterns in Bitcoin's price trends and predicting continuation patterns. In addition, LSTM delivers the same Best Sharpe Ratio as Random Forest (4.29), proving that consistency is the key to success in this particular portfolio. While GRU, which shares architecture similarities with LSTM, demonstrates inferior performance in most horizons and generates a drawdown of (-9.12%), losing to LSTM in Bitcoin, as shown in Figure 4.6.

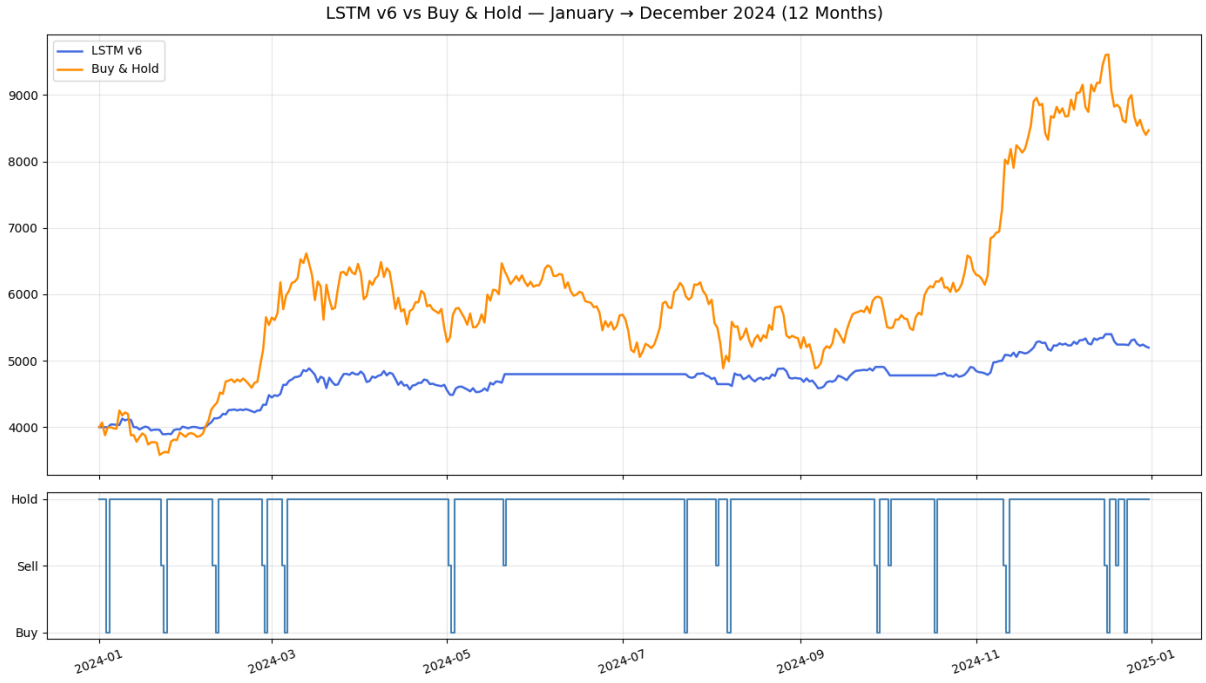


Figure 4.6: LSTM Performance vs Buy & Hold on BTC over 12 Months (2024)

RF and XGBoost demonstrate similar results over the short to medium-term with +4.40% and almost equal 1 month profits (+10.13% and +10.18%). The RF average drawdown of (-14.05%) and (-12.32%) in XGBoost. Both are remain controlled, which supports the notion about capital preservation in machine learning stated by Jay & Berlanga (2024). XGBoost Higher 12 months profits (+16.93% in RF and +24.31% in XGBoost) correspond to the capabilities of gradient boosting to identify long-term trends that RF cannot catch due to the shallowness of trees, explain the superiority of XGBoost over RF, as shown in Figure 4.7.

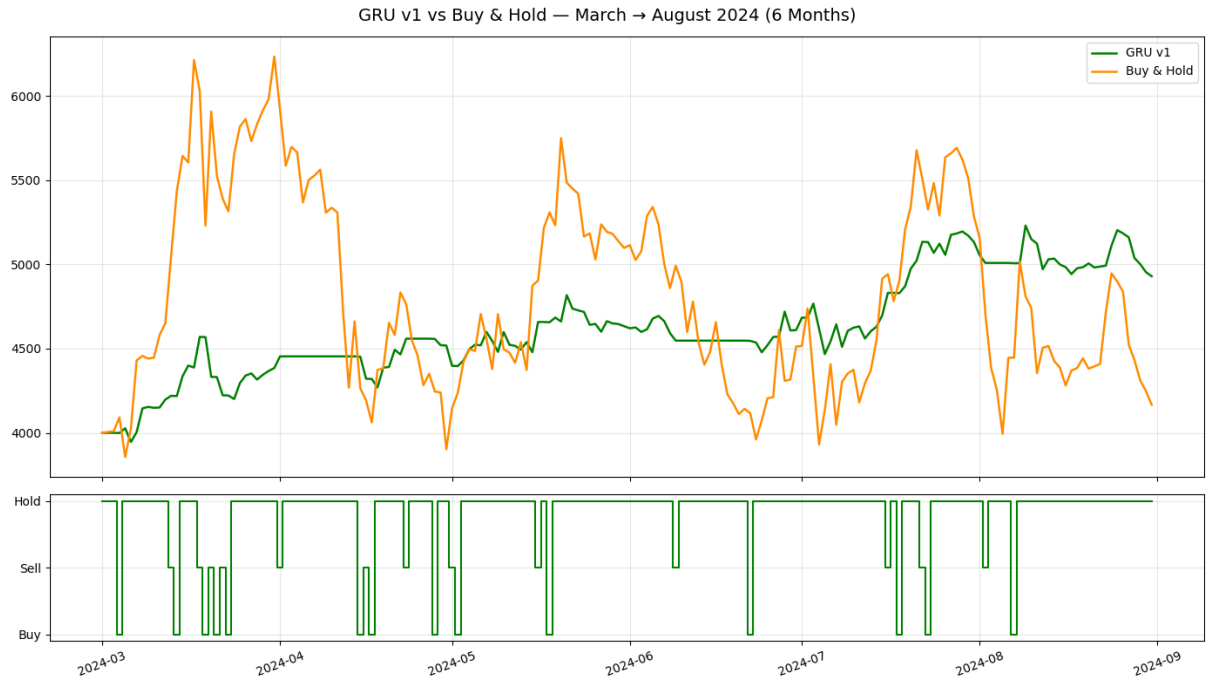


Figure 4.7: XGBoost Performance vs Buy & Hold on BTC over 12 Months (2024)

In the figure 4.8 below, DQN shows an inconsistent performance pattern while dealing with BTC. The best result of +16.91% at 1M is higher than all other models results for the same horizon and even outperforms Buy-and-Hold strategy by +14.32%, thus showing that DQN can add value through active trading decisions trades and dynamic position sizing. At the same time, DQN becomes the only model with negative 10D profits (-0.93%) and demonstrates poor performance at 3M (- 3.36%). This proves high sensitivity to whipsaws. The worst drawdown of (- 20.58%) is the highest value for all considered strategies with BTC, meaning that aggressive position sizing in DQN brings the greatest risks when signal quality is not consistent enough. With proper time to converge on a strategy, DQN provides the best 12M profits (+39.75%).This proves that when given enough time to master a strategy, DQN could continue to generate profits throughout the test period.

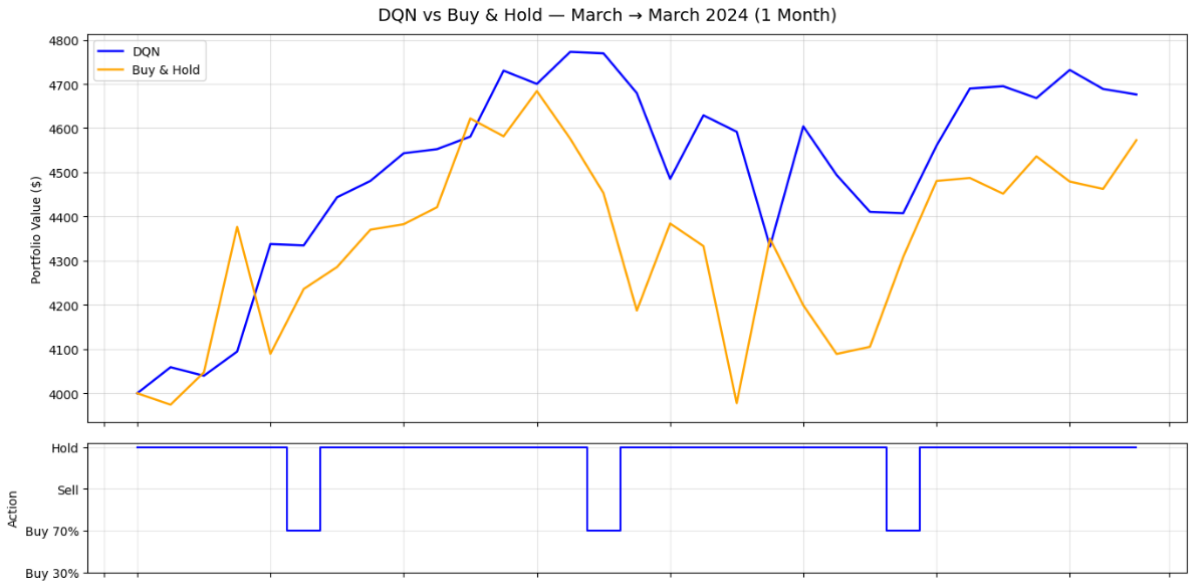


Figure 4.8: DQN Performance vs Buy & Hold on BTC over 1 Months

With respect to the Best Sharpe ratio, PPO is ranked highest with an outstanding 6.05 for BTC,

suggesting strong risk-adjusted performance over the 10-day horizon, as shown in Figure 4.9. Nevertheless, its negative 1 month return of (− 1.57%), which is the only negative value among all models besides DQN, emphasizes PPO's conservative in generating signals at a medium horizon.

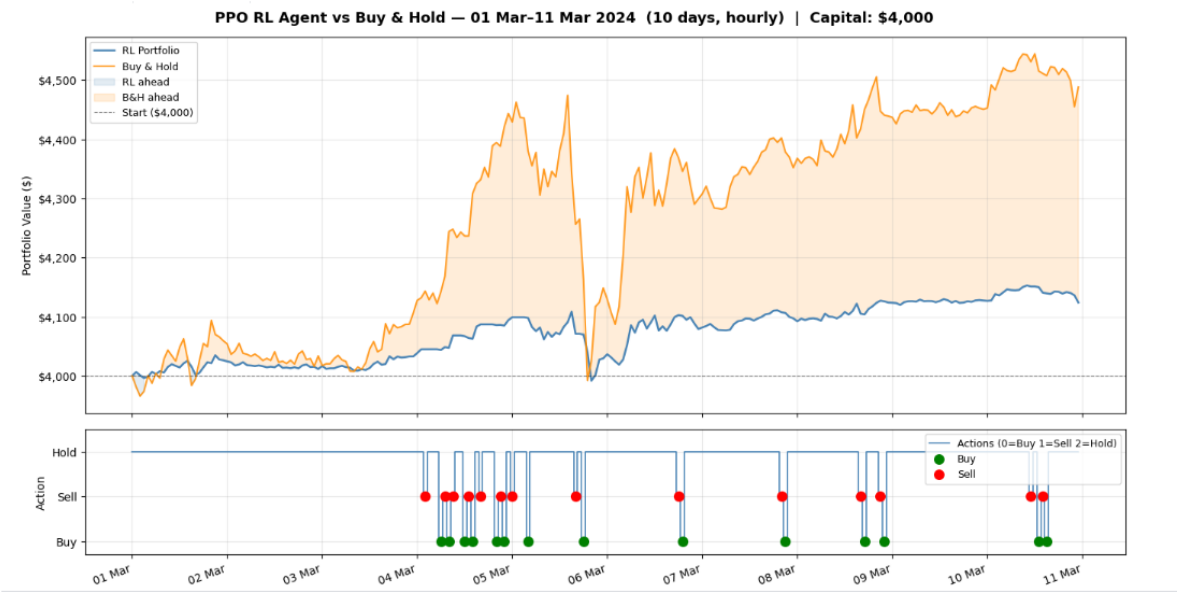


Figure 4.9: PPO Performance vs Buy & Hold on BTC over 10 days

Table 4.2: BTC Model Performance by Horizon

Model	Return (%)					worst DD (%)	Best Sharpe
	10D	1M	3M	6M	12M		
RF	+4.40	+10.13	+0.59	-2.24	+16.93	-14.05	4.29
XGBoost	+4.40	+10.18	+7.26	+0.53	+24.31	-12.32	4.18
LSTM	+1.64	+10.33	+7.88	+6.46	+29.91	-8.16	4.29
GRU	+1.02	+10.23	+7.83	+2.88	+24.92	-9.12	4.21
DQN	-0.93	+16.91	-3.36	+4.17	+39.75	-20.58	3.45
PPO	+3.10	-1.57	+4.22	-1.05	+13.66	-15.92	6.05
B&H	+12.20	+14.32	+8.23	-5.48	+111.75	—	—

Summary Finding
In summary, The LSTM model is the most consistent model for BTC, showing positive returns in four out of five horizons, having the lowest worst DD of (-8.16%), and Best Sharpe of 4.29. The DQN model is the only model that has beaten B&H in 1M (+16.91% vs +14.32%) but has the highest worst DD of -(20.58%). In 12 months, all models perform below B&H (+111.75%) and DQN was the best-performing model with (+39.75%)because of the strong bull market in 2024.

4.4.1.2 Ethereum (ETH)

The table 4.3 below shows that Ethereum is the most challenging asset in this study. The volatile nature of the asset and its uncertain price movements in the test period made most models lose at medium term investment horizons.

RF and XGBoost reached their best performance with (+12.44%) at 10 days which is the highest return among all models on ETH in this horizon, as shown in Figure 4.10. They also achieved their highest Sharpe ratio of 7.24 in the entire study, yet they were still not able to exceed the buy-and-hold strategy at (+14.79%). For both models, their performance was inconsistent across the different

horizons. Positive at 3 months (+3.89%, +2.96%) and 12 months (+9.70%, +9.82%), while negative at 1 month (−3.71%, −7.64%) and 6 months (−7.87%, −5.69%). The above indicates that RF and XGBoost are sensitive to Ethereum's changing market conditions and are unable to provide consistent directional signals across all horizons. worst drawdowns of −21.46% and −20.51% suggest expensive trading due to high volatility levels.

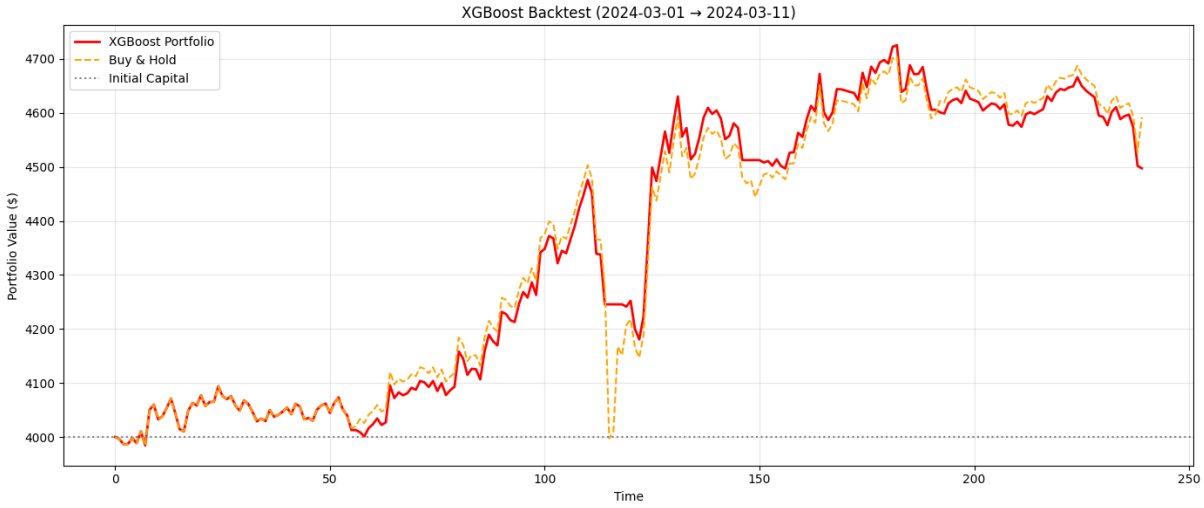


Figure 4.10: XGBoost Performance vs Buy & Hold on ETH over 10 days

LSTM and GRU were ineffective on ETH at any horizon except 10 days and 12 months. GRU was superior to LSTM with the lower worst drawdown of (-17.86%) versus (-21.57%), implying that GRU outperformed LSTM in capital preservation on ETH. LSTM and GRU posted their worst Sharpe ratio at 1.23 and 1.18 on ETH, respectively, indicating that the sequential structure fails to forecast directionality in choppy and untrendy markets, as shown in Figure 4.11.

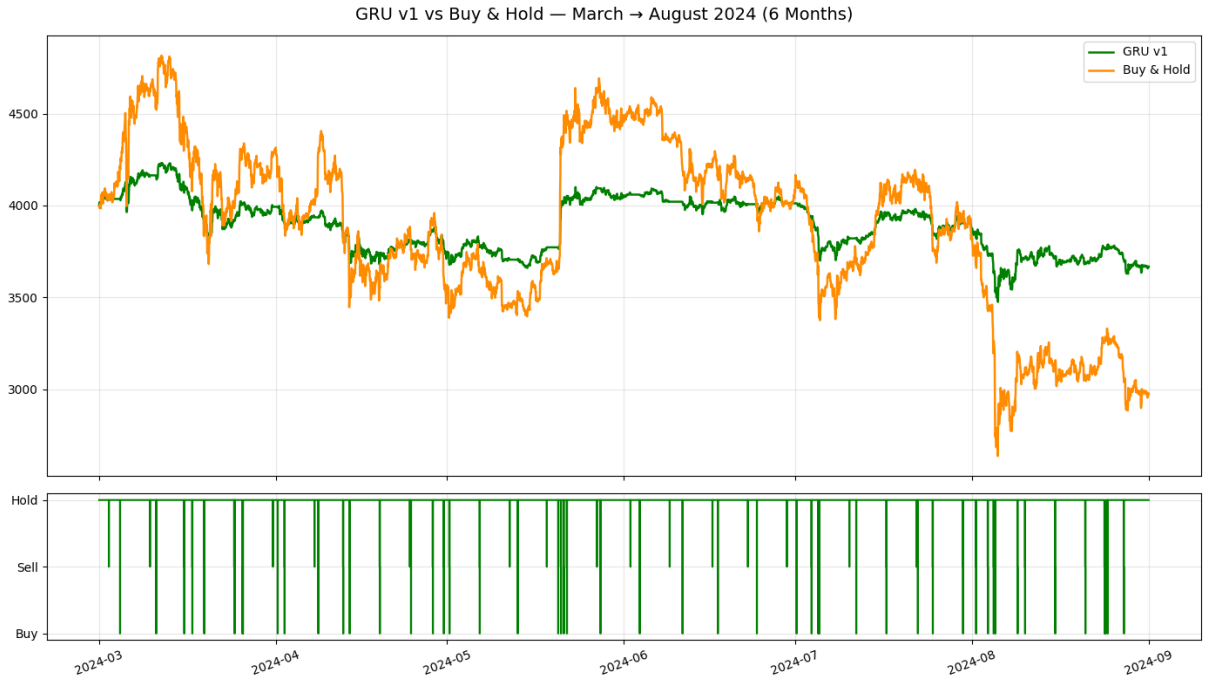


Figure 4.11: GRU Performance vs Buy & Hold on ETH over 6 month

DQN generated its highest 3-month score of (+16.20%) on ETH, which was greater than B&H (+11.35%), as shown in Figure 4.12. It implies that DQN can improve returns on ETH with active

trading decisions and dynamic portfolio management strategies, but this advantage was not sustained at longer horizons because DQN suffered at 6 months and 12 months, recording (-19.18%) and (-3.35%), respectively.DQN also experienced the deepest worst drawdown of (-31.05%), suggesting that DQN's aggressive position sizing was costly under sudden changes in market conditions.

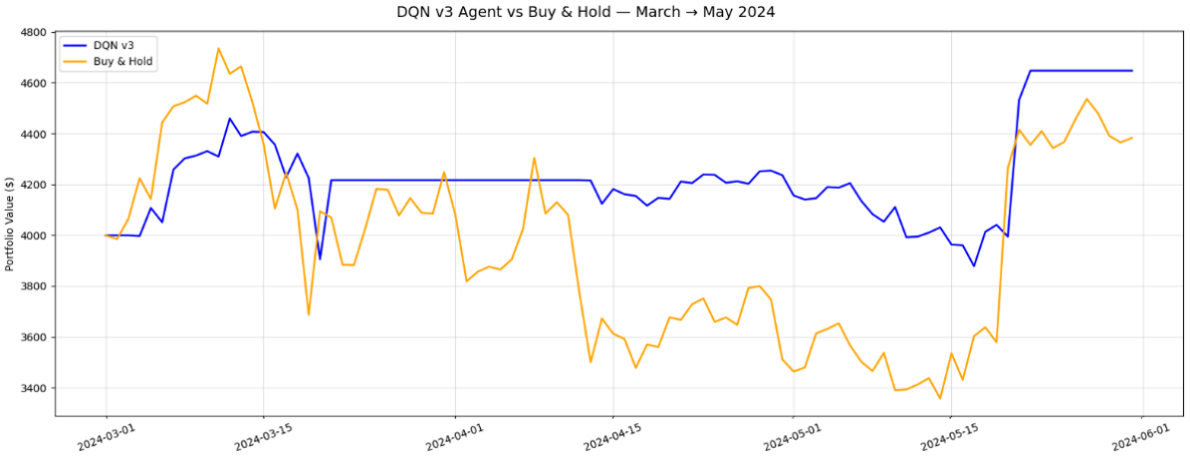


Figure 4.12: DQN Performance vs Buy & Hold on ETH over 3 month

PPO achieved positive returns of (+0.68%) at 1 month, (+4.46%) at 3 months, and (+13.17%) at 12 months, making it the second-best-performing algorithmic model at 12 months behind GRU (+14.38%), as shown in Figure 4.13. The Sharpe ratio of PPO is 5.53, reflecting good risk-adjusted performance. PPO obtained returns of (−11.15%) at 6 months, better than B&H (−25.62%), but not the best.

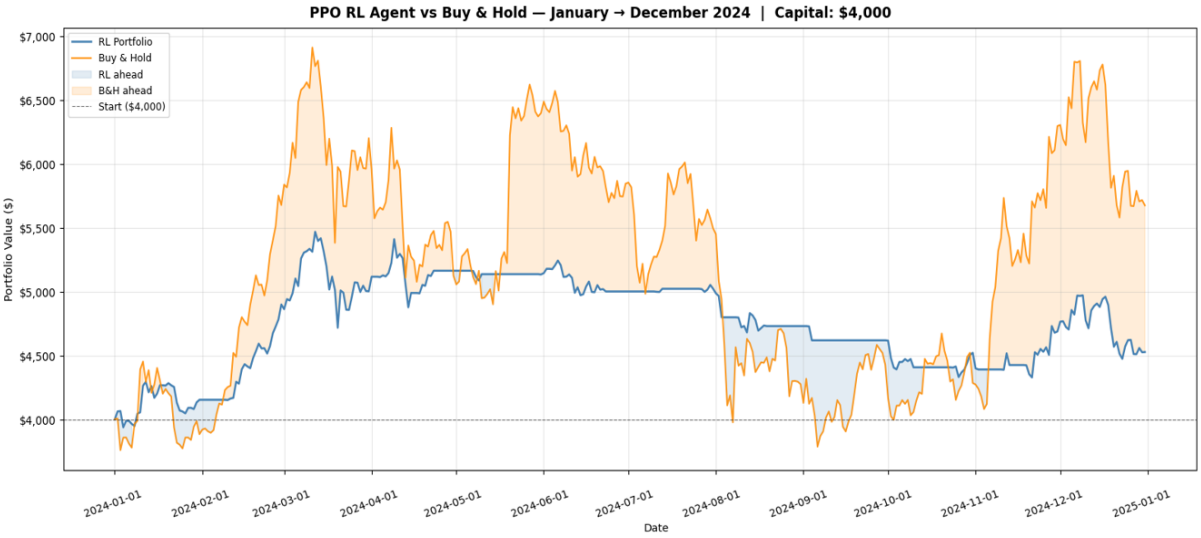


Figure 4.13: PPO Performance vs Buy & Hold on ETH over 12 month

Table 4.3: ETH Model Performance by Horizon

Model	Return (%)					worst DD (%)	Best Sharpe
	10D	1M	3M	6M	12M		
RF	+12.44	-3.71	+3.89	-7.87	+9.70	-21.46	7.24
XGBoost	+12.44	-7.64	+2.96	-5.69	+9.82	-20.51	7.24
LSTM	+4.91	-0.85	-5.10	-12.48	+12.45	-21.57	1.23
GRU	+4.81	-1.59	-1.22	-8.29	+14.38	-17.86	1.18
DQN	+4.56	+7.41	+16.20	-19.18	-3.35	-31.05	1.60
PPO	+3.36	+0.68	+4.46	-11.15	+13.17	-21.32	5.53
B&H	+14.79	+7.89	+11.35	-25.62	+45.40	—	—

Summary Finding

In summary, RF and XGBoost achieve the highest Sharpe ratio in the entire study (7.24) at 10 days but trail B&H (+14.79%) even at that short window. DQN is the only model to beat B&H at 3 months (+16.20% vs +11.35%), adding value through active trading decisions and dynamic position sizing, but records the deepest average drawdown on ETH (−31.05%). At 6 months, all algorithmic models recorded better returns than B&H which collapsed to −25.62%, confirming that algorithmic strategies offer better capital protection during adverse market conditions on ETH.

4.4.1.3 Cardano (ADA)

Table 4.4 shows that Cardano is the most bearish asset that experienced a continuous decline during 2024. This situation is reflected by the performance of the B&H strategy: B&H declined to (−9.55%) at 1 month, (−37.80%) at 3 months, (−51.98%) at 6 months before recovering to (+35.73%) at 12 months. Given such a strong bearish situation, Cardano became the most difficult asset for all models, especially supervised models.

The results of RF were poor at every horizon, having negative returns at each of them, e.g., (−22.29%) at 6 months and (−3.32%) at 12 months. With the worst drawdown of (−29.63%) and a negative Sharpe ratio (−0.24), which proves that RF failed to add any value in ADA forecasting. At the same time, XGBoost performed relatively well with a (+13.24%) return at 12 months and a worst drawdown of (−19.98%), however, it suffered from a structural problem of supervised models where it continuously generated purchase signals even when the market started falling, as shown in Figure 4.14.

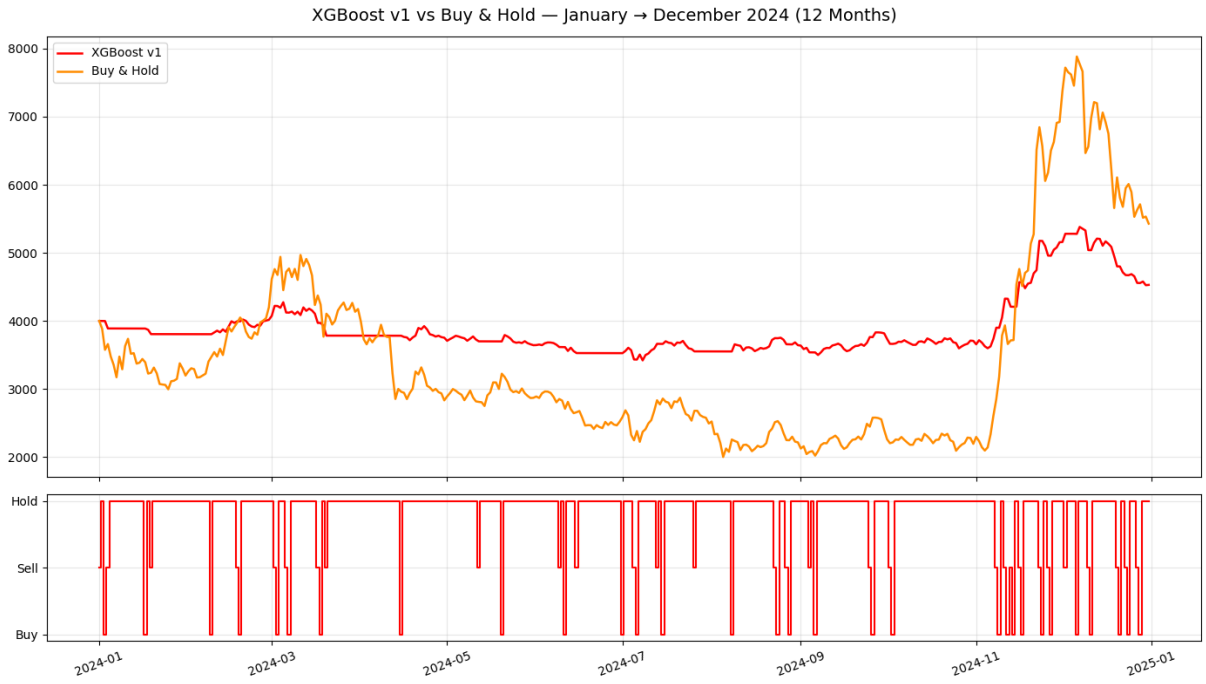


Figure 4.14: XGBoost Performance vs Buy & Hold on ADA over 12 month

LSTM produced negative returns at 10 days up to 6 months, with its worst performance at 6 months (−19.60%). However, at 12 months LSTM managed to recover a little bit, achieving (+8.03%), although, with average drawdown of (−27.82%) and nearly zero Sharpe (0.02), as shown in Figure 4.15. GRU was the only supervised learning model capable of producing a positive return at a 10-day horizon (+4.82%), however, like LSTM and XGBoost, GRU generated negative returns at later points in time: (−19.77%) at 6 months and (−3.08%) at 12 months. With worst drawdown of (−27.67%) and Sharpe of 0.76, GRU demonstrates minor improvements over LSTM.

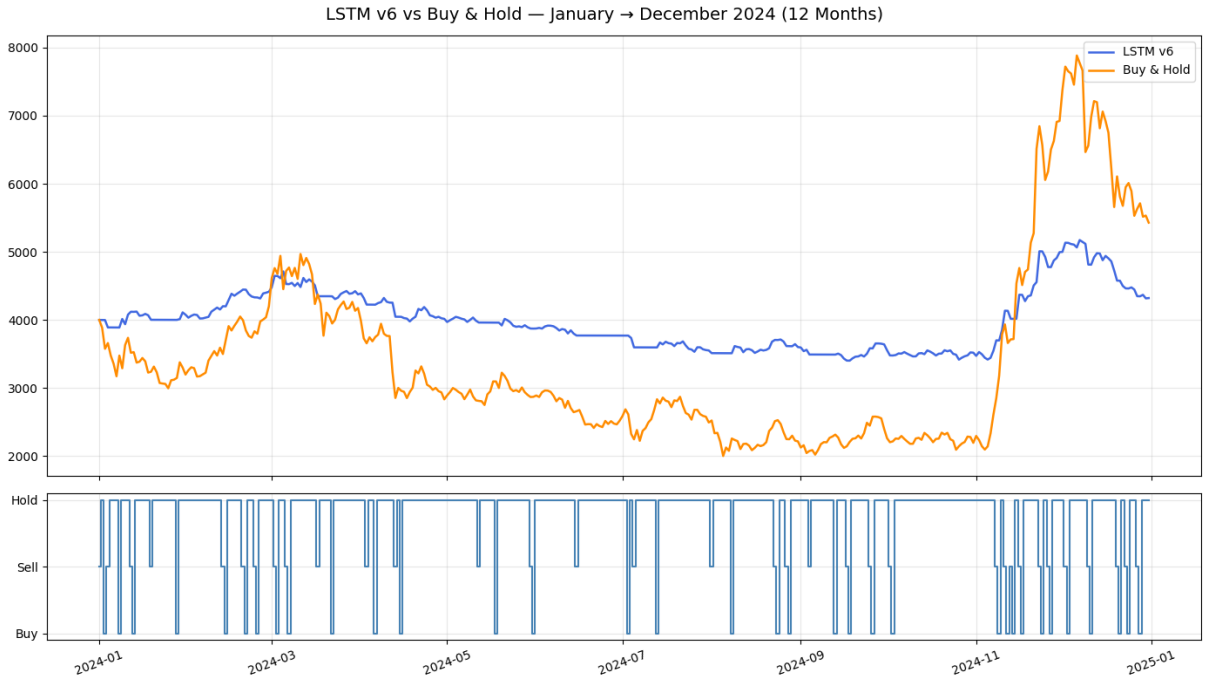


Figure 4.15: LSTM Performance vs Buy & Hold on ADA over 12 month

On Cardano (ADA), DQN demonstrated highly irregular performance. The model generated near-

zero result at 10 days (−0.03%) and a positive result of (+2.23%) at 1 month; however, at 3 months, it fell dramatically to (−18.10%), which is the worst 3 month result on all assets. After 6 months, DQN recovered and generated (+10.77%) while achieving (+58.03%) at 12 months outperforming B&H (+35.73%) , as shown in Figure 4.16. This pattern demonstrates that this model needed more time to learn how to behave in a bearish market. As a consequence, the worst drawdown of DQN (−40.04%) is the highest.

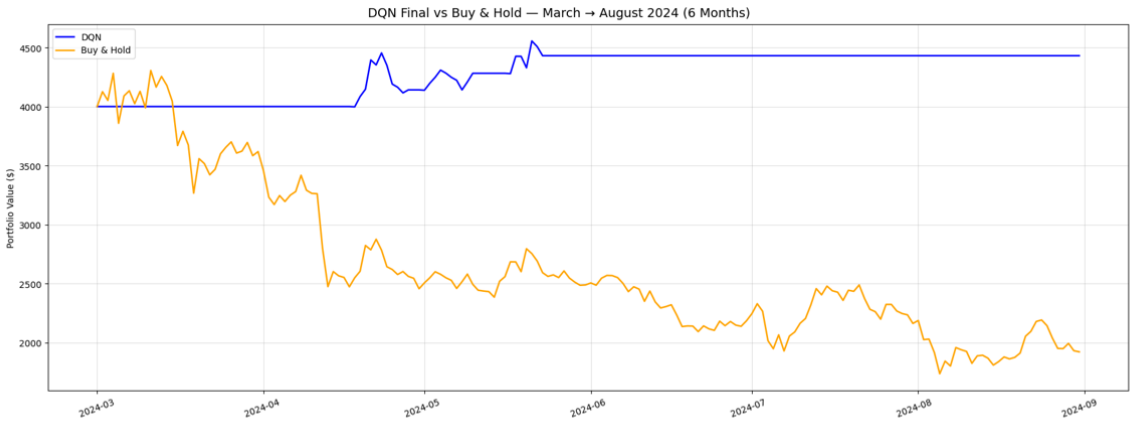


Figure 4.16: DQN Performance vs Buy & Hold on ADA over 6 month

Among the evaluated algorithms, PPO produced the best performance on Cardano (ADA). It demonstrated the following results: (+2.83%) at 10 days, (+1.99%) at 1 month, and the highest return (+74.60%) at 12 months outperforming B&H (+35.73%), as shown in Figure 4.17. In addition, the model has the best Sharpe ratio (3.06) and the best worst drawdown (−24.33%) on across models. Such strong performance of PPO on the bearish asset can be explained by the clipping mechanism, which allows for more stable and gradual policy updates, making it easier for it to master the optimal bearish approach faster and more consistently than DQN.

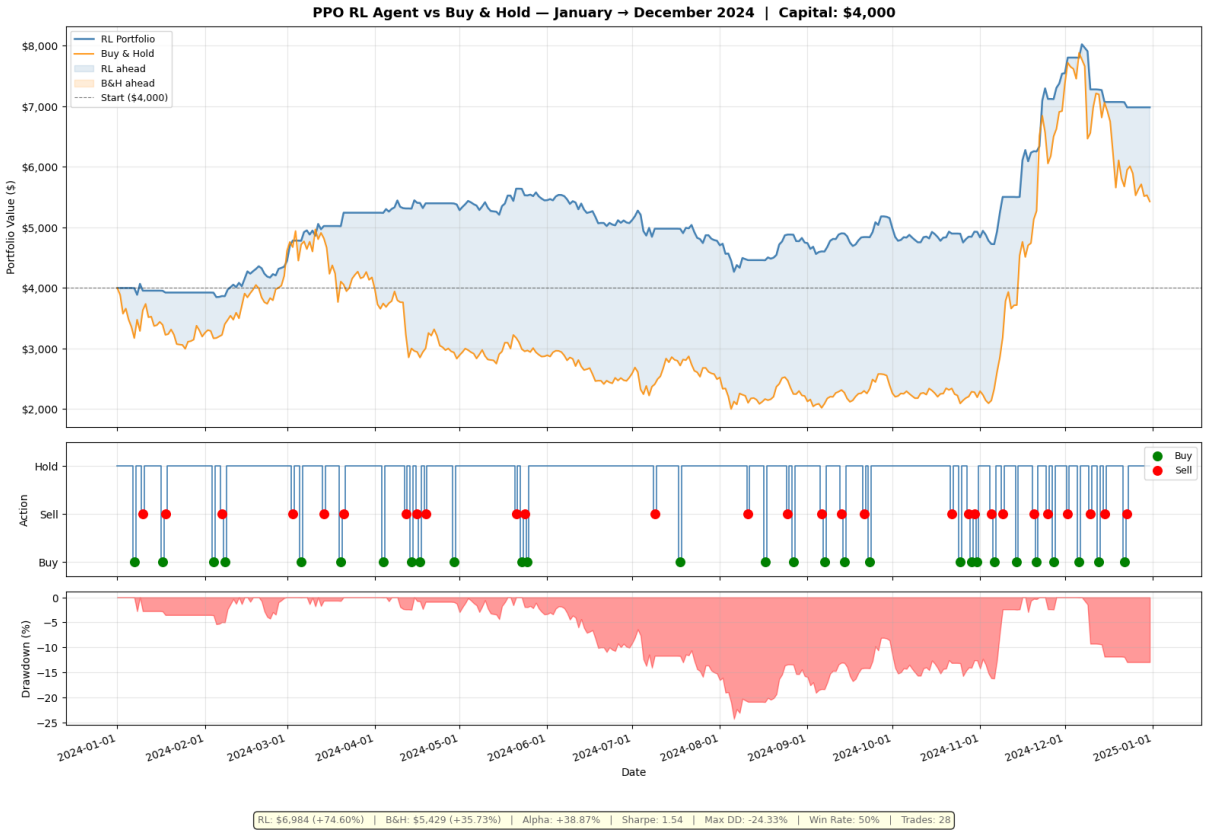


Figure 4.17: PPO Performance vs Buy & Hold on ADA over 12 month

Table 4.4: ADA Model Performance by Horizon

Model	Return (%)					Worst DD (%)	Best Sharpe
	10D	1M	3M	6M	12M		
RF	-1.74	-7.85	-15.71	-22.29	-3.32	-29.63	-0.24
XGBoost	-1.74	-7.03	-10.48	-10.48	+13.24	-19.98	0.50
LSTM	-2.67	-2.33	-13.54	-19.60	+8.03	-27.82	0.02
GRU	+4.82	-2.71	-14.10	-19.77	-3.08	-27.67	0.76
DQN	-0.03	+2.23	-18.10	+10.77	+58.03	-40.04	1.10
PPO	+2.83	+1.99	-4.01	-3.38	+74.60	-24.33	3.06
B&H	+6.65	-9.55	-37.80	-51.98	+35.73	—	—

Summary Finding

In summary, All algorithmic strategies beat the B&H at both 3 and 6 months time horizons. The PPO is the best performer with the highest individual return (+74.60%) at the 12-month time horizon, beating the B&H strategy by (+38.87%). The DQN strategy also gains (+58.03%) at the 12-month time horizon. Both reinforcement learning models perform much better than all supervised learning models, proving that RL models learn to stay out of assets during bearish periods, unlike supervised learning models that generate losing buy signals.

4.4.1.4 Solana (SOL)

The table 4.5 below shows that Solana proves to be the best-performing asset for all models in the research since its price demonstrated a robust and prolonged uptrend during most of the test period. The directionality of the asset during the year 2024 served as a perfect scenario for testing both supervised trend-following models and reinforcement learning algorithms, leading to consistently positive results in almost all cases.

RF and XGBoost deliver good results for the SOL asset, with RF providing a positive return of (+32.82%) at 12 months and XGBoost returning (+28.75%) at the same period, as shown in Figure 4.18. The two models gain from the fact that SOL continued to move in one clear direction, unlike their unfavorable results in ADA. For smaller periods, RF provides a return of (+11.05%) at 1M and (+10.57%) at 3M, while XGBoost gives (+9.59%) and (+7.48%) respectively, suggesting both models do a solid job reading the trend, just not an exceptional one.

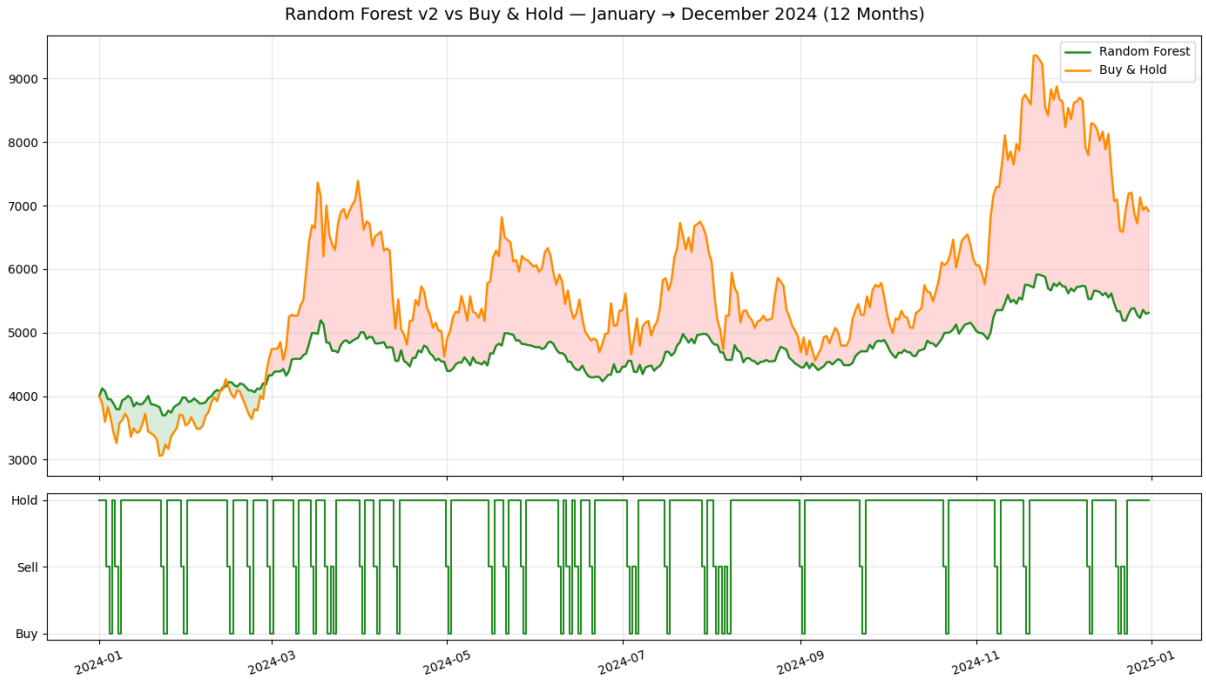


Figure 4.18: RF Performance vs Buy & Hold on SOL over 12 month

GRU model performs especially well on the medium-term trend of SOL, standing out on the 3 months (+15.83%) and 6 months (+23.25%) horizons, as shown in Figure 4.19. With excellent results on these timeframes in combination with minimal worst maximum drawdowns of (−10.62%), it appears that the GRU model can accurately detect multi-week momentum trends in the SOL price movements. Its highest return on the 6M horizon is (+23.25%), and its total Sharpe Ratio of 2.96 indicates that it was well risk-controlled.

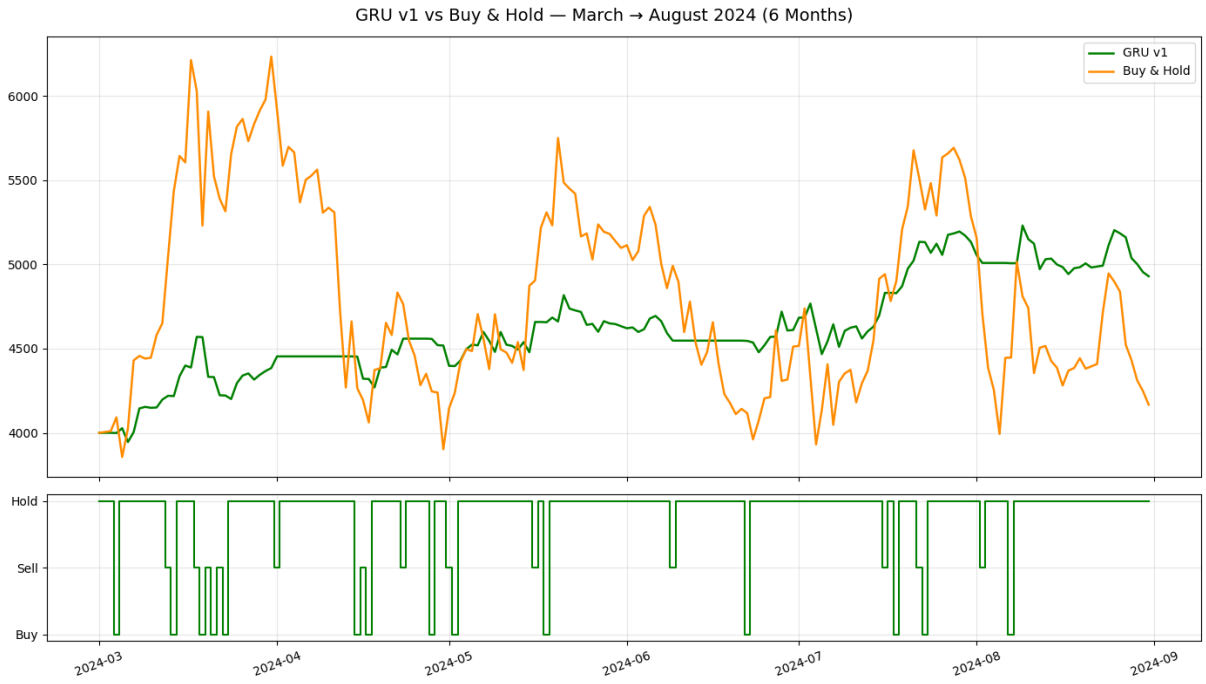


Figure 4.19: GRU Performance vs Buy & Hold on SOL over 6 month

The LSTM algorithm shows the most conservative profile in SOL, as shown in Figure 4.20. Since its average maximum drawdown of (-10.42%) is the smallest among all models for this asset and positive returns on all horizons, this suggests that its memory cell preserves signal stability at the expense of return growth, making it an attractive choice for risk-sensitive strategies even in strongly trending environments.

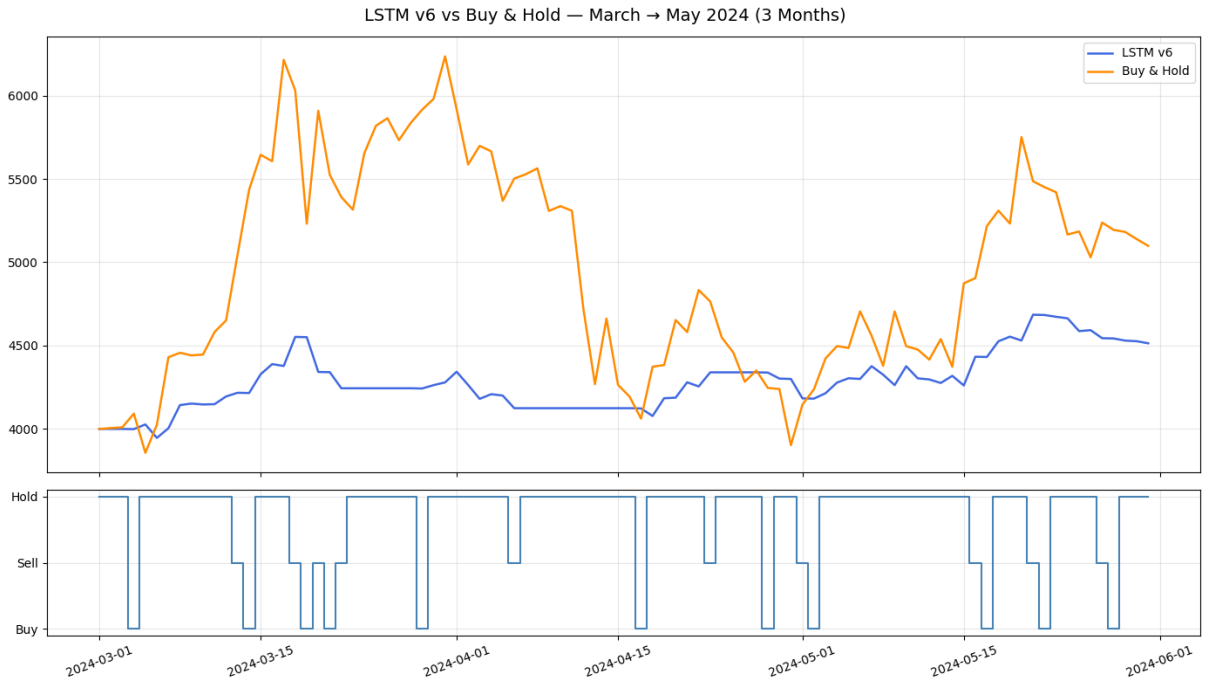


Figure 4.20: LSTM Performance vs Buy & Hold on SOL over 3 month

The DQN algorithm produces its best performance on Solana at a 1-month horizon level: (+36.59%) with a Sharpe Ratio equaling 4.88, as shown in Figure 4.21. This is because of the Buy 30%/70% dynamic position sizing policy, where more capital is allocated when the agent has higher confidence, enabling the strategy to earn more profits during strong signals. This validates the recommendation by Yang & Malik

(2024) to use position sizing as an action to be learned rather than a fixed rule. On the other hand, DQN also produces (+63.56%) at 12 month horizon level, indicating that the allocation policy works well for a sustained trend.

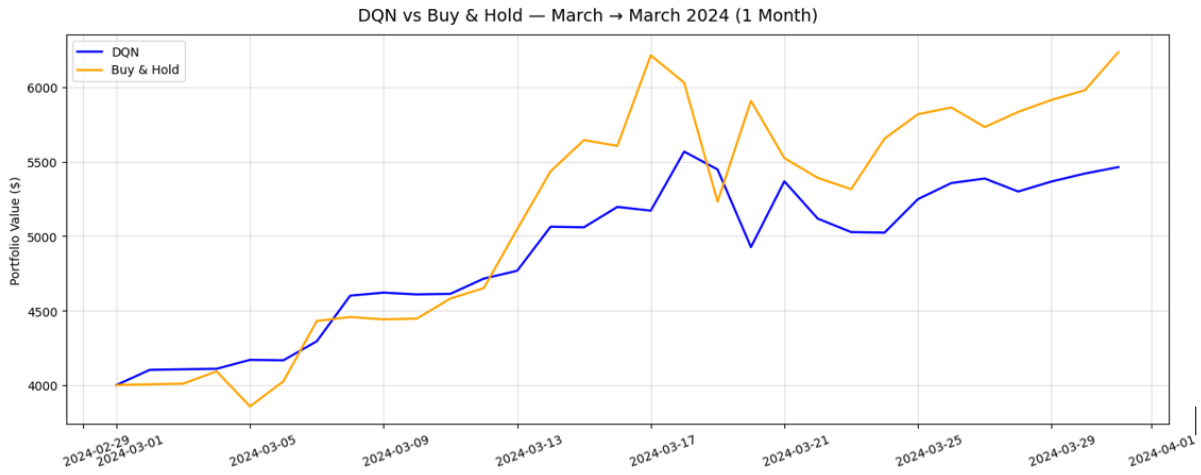


Figure 4.21: DQN Performance vs Buy & Hold on SOL over 1 month

PPO suffered losses for medium term intervals of (-7.14%) and (-3.29%) after 3 months and 6 months, respectively, but then recovered to a profit of (23.47%) after 12 months, as shown in Figure 4.22. The early losses indicate that PPO is slow to pick up on momentum due to its cautious policy update approach. However, its final one-year return demonstrates that given enough time, the model does align well with SOL's sustained uptrend.

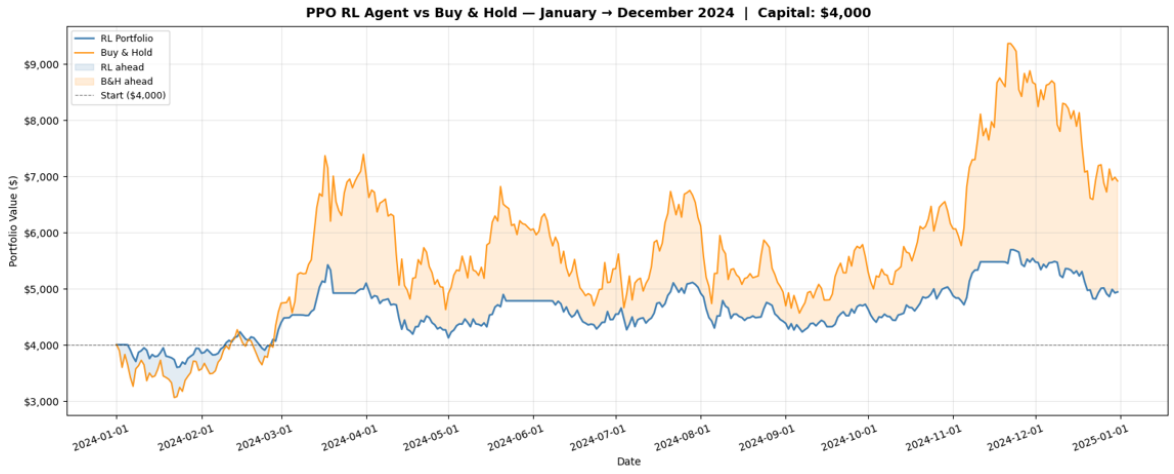


Figure 4.22: PPO Performance vs Buy & Hold on SOL over 12 month

Table 4.5: SOL Model Performance by Horizon

Model	Return (%)					worst DD (%)	Best Sharpe
	10D	1M	3M	6M	12M		
RF	+3.37%	+11.05%	+10.57%	+3.76%	+32.82%	-18.51%	2.97
XGBoost	+3.37%	+9.59%	+7.48%	+0.97%	+28.75%	-16.91%	2.96
LSTM	+4.16%	+6.98%	+12.84%	+11.26%	+10.56%	-10.42%	2.42
GRU	-1.60%	+9.62%	+15.83%	+23.25%	+31.75%	-10.62%	2.9623
DQN	+1.86%	+36.59%	+11.16%	+14.96%	+63.56%	-28.87%	4.88
PPO	+2.82%	+6.74%	-7.14%	-3.29%	+23.47%	-23.96%	4.12
B&H	+8.69%	+55.88%	+27.45%	+4.18%	+72.84%	—	—

Summary Finding

In summary, The DQN delivers the maximum result among assets evaluated to date(+36.59%)at 1 month with Sharpe 4.88 which clearly verifies the effectiveness of the dynamic position sizing strategy. The GRU is the leader at both 3 month (+15.83%) and 6 month (+23.25%) periods. The lowest drawdown (-10.42%) among all models on SOL is recorded by the LSTM indicating that it is possible to achieve both low drawdowns and positive profits in a trending market. The PPO returns to (+23.47%) at 12M although having negative results at 3M and 6M periods, while RF and XGBoost deliver solid sustained performance peaking at (+32.82%) and (+28.75%) respectively at 12M.

4.4.1.5 Binance Coin (BNB)

The table 4.6 shows that Binance coin emerges as the most favorable candidate for the supervised learning techniques in the experiment. Due to the consistently rising trend of the BNB token within the evaluation period, the machine learning and deep learning models are able to benefit from this positive directionality and show the best performance across all assets tested.

XGBoost shows optimal sustained performance for BNB: (22.38%, 30.32%, and 55.43%) returns at the 3 month, 6 month, and 12 month timeframes respectively. Of these, (+55.43%) 12 month return is the best individual performance shown by XGBoost and constitutes the highest return achieved by a supervised machine learning algorithm among all assets in the 12 month timeframe, as shown in Figure 4.23. This kind of performance demonstrates the capacity of gradient boosting architecture to leverage the presence of an unchanging alignment between features and target, which is characteristic of the uptrend of BNB, since with continued price rises, the buy predictions made by the model continue to be on the right track.

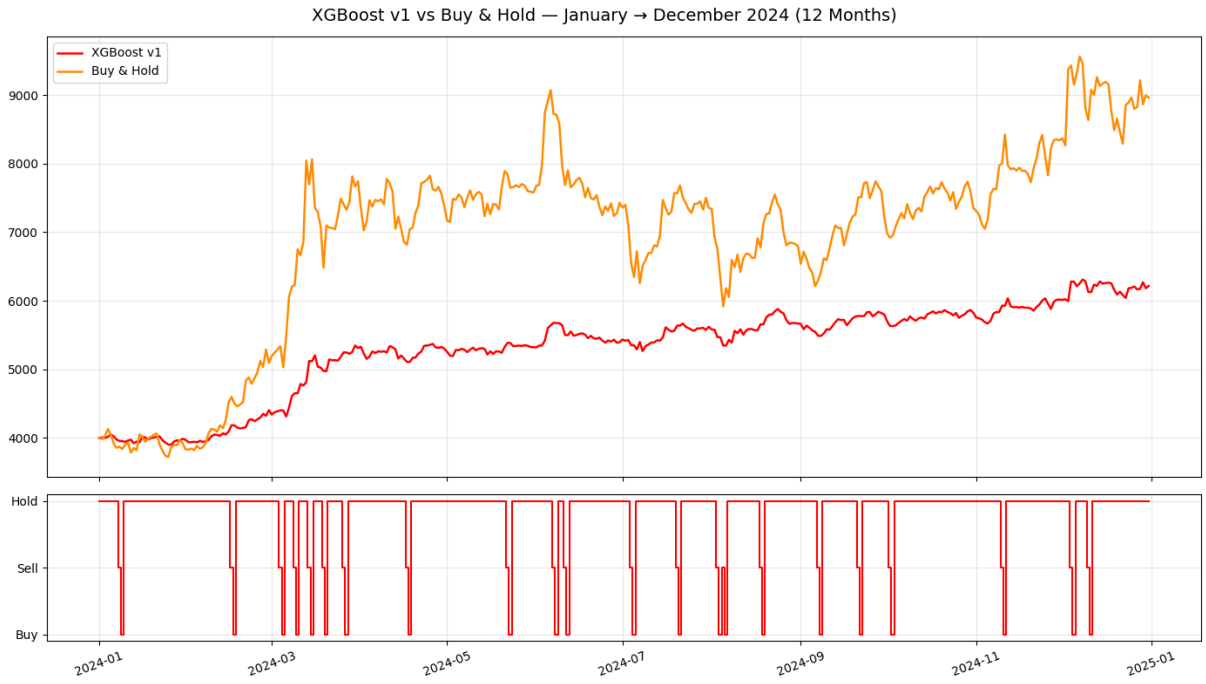


Figure 4.23: XGBoost Performance vs Buy & Hold on BNB over 12 month

The lowest maximum drawdown recorded by RF in this study is obtained in the case of BNB, where RF has the minimum maximum drawdown values of (-5.33%,−2.05%,-0.51%,-0.51% and −3.11%) for 10 days,1 month, 3 month, 6 month and 12month respectively. It validates again the capital preservation thesis proposed by Jay & Berlanga (2024), as shown in Figure 4.24.The low drawdown together with positive gains ensures that the profile of BNB is the most favorable asset for supervised machine learning models due to the stability of trends, low volatility, and less regime shifts. The Sharpe ratio of 5.59 at 1M for RF reinforces this claim.

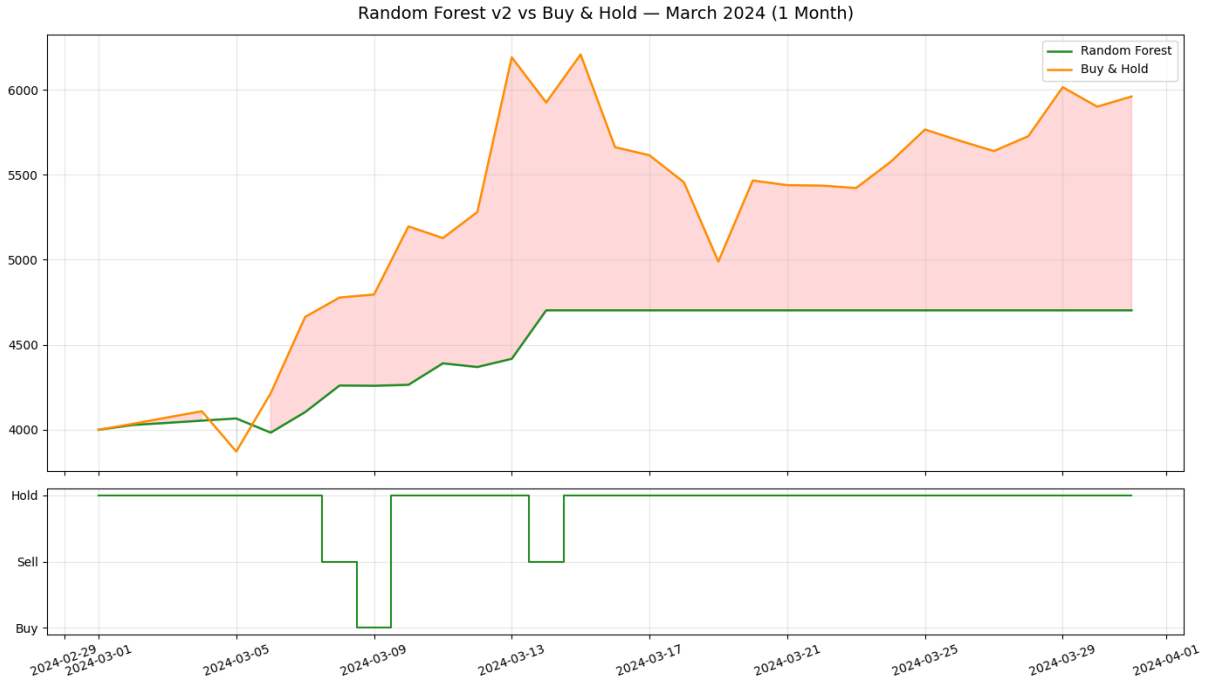


Figure 4.24: RF Performance vs Buy & Hold on BNB over 1 month

Both GRU and LSTM produce good and reliable performance in the BNB data set at all horizons

with small drawdowns (−10.49%, −7.17%), as shown in Figure 4.25. The results generated by GRU and LSTM on the BNB indicate that the recurrent architecture contributes positively to on trending assets with moderate volatility,in line with the empirical results reported in Adedigba et al. (2025) for sequence models on large-cap cryptocurrencies with clear directional bias.

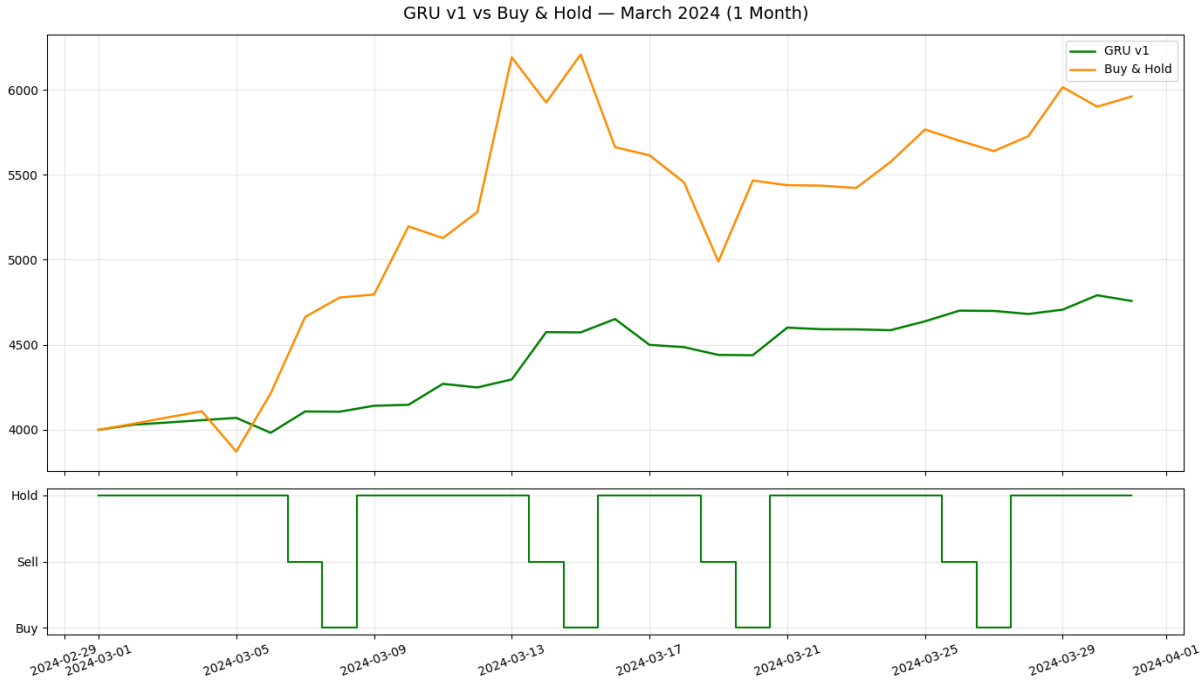


Figure 4.25: GRU Performance vs Buy and Hold on BNB over 1 month

The return generated by DQN in 3 month time horizon on BNB stands at (+49.52%), another outstanding performance attributable to the dynamic sizing strategy that exploits momentum in BNB, as shown in Figure 4.26. Nevertheless, the return for DQN on 6M horizon falls back to close to zero (-0.37%), while its maximum drawdown of (-28.49%) is the greatest observed in any of the tested models in BNB. The explosive returns accompanied by high drawdown is a hallmark of the explorative nature of DQN, whereby the agent makes bold bets during confident times then incurs losses when market conditions shift.

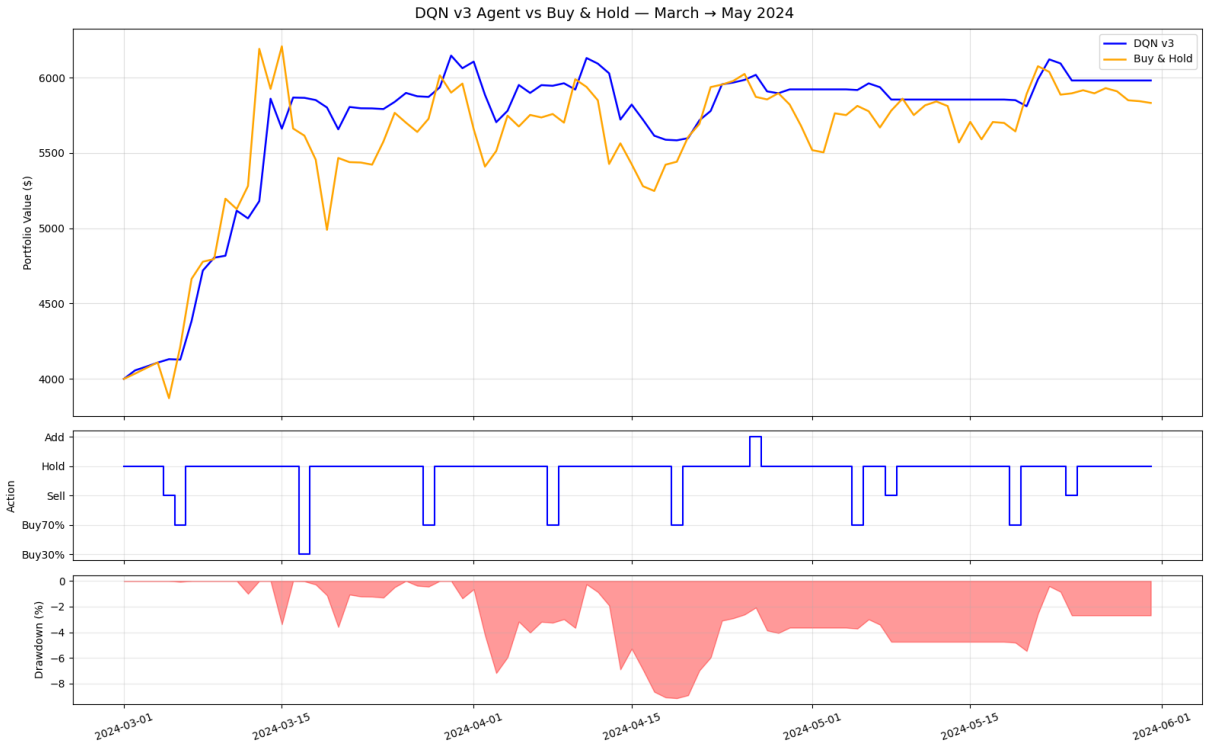


Figure 4.26: DQN Performance vs Buy & Hold on BNB over 3 month

The Sharpe ratio for PPO model is the best amongst all models 7.23 over the 10-day time horizon on BNB. In addition to that, PPO demonstrates the best result in 12M returns on BNB among all models at(+ 83.87%), which outperforms DQN with (+45.63%) returns and XGBoost with (+55.43%) returns at this horizon, as shown in Figure 4.27.The fact that PPO leads in both shortest and longest time horizons proves that the ability of this model to adaptively learn trading policies suits the stable trending nature of BNB best.

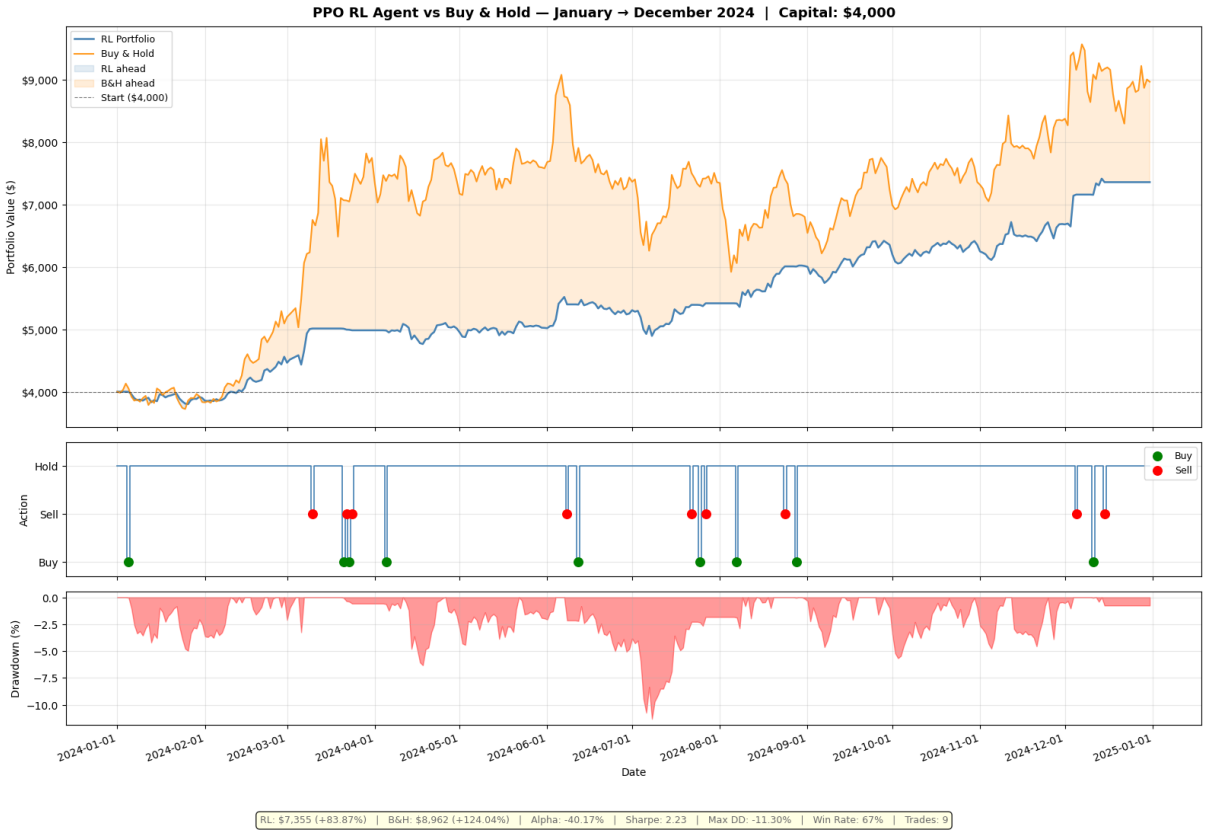


Figure 4.27: PPO Performance vs Buy & Hold on BNB over 12 month

Table 4.6: BNB Model Performance by Horizon

Model	Return (%)					Avg DD (%)	Best Sharpe
	10D	1M	3M	6M	12M		
RF	+0.14%	+17.56%	+11.76%	+11.76%	+17.35%	-5.33%	5.59
XGBoost	+0.14%	+18.67%	+22.38%	+30.32%	+55.43%	-7.27%	5.26
LSTM	+6.41%	+13.80%	+12.85%	+15.98%	+30.84%	-7.17%	4.39
GRU	+6.18%	+18.94%	+13.87%	+14.82%	+37.57%	-10.49%	5.27
DQN	-1.06%	+20.28%	+49.52%	-0.37%	+45.63%	-28.49%	2.86
PPO	+4.56%	+5.17%	+11.12%	+16.74%	+83.87%	-12.24%	7.23
B&H	+30.89%	+49%	+45.79%	+30.84	+124.04%	—	—

Summary Finding

In summary, supervised models, XGBoost gives the maximum return of (+55.43%) for the period of 12 month, while RF shows minimum drawdown values between (-2.05% and -3.11%). For RL, PPO shows maximum Sharpe ratio value of 7.23 at 10 days, while DQN gives (+49.52%) returns at 3 month. The stable and consistent uptrend of BNB is beneficial for supervised models' pattern recognition capabilities as well as for RL models' adaptive nature, making it the strongest overall asset in this study.

4.4.4.2 Cross Asset Comparative Analysis

In this section, performance of the models is studied across all five assets and five horizons in order to determine any possible patterns in their performance and also evaluate which models outperform other models on different horizons and how close to universal superiority any one model comes. Two viewpoints have been used for this purpose: a best-of approach by horizon and aggregating average returns of all assets.

4.4.2.1 Best Model per Horizon

The table 4.7 below shows that the horizon-by-horizon breakdown reveals a clear and consistent pattern: no single model dominates every time frame, but each model has a window where it is clearly the right tool.

Ensemble methods dominate at the 10 day horizon. The results indicate that RF and XGBoost exhibit the highest return on ETH (+12.44%), whereas PPO delivers the best risk-adjusted return with a Sharpe ratio of 7.23 on BNB. This is an example of one of the key benefits of tree based models: the ability to extract strong short-term signals from noisy price data without overfitting to transient patterns.

The dynamics are completely shifted to DQN at 1 and 3 month horizons. As the market is trending, DQN aggressively captures the momentum on trending assets like SOL (+36.59% at 1 month) and BNB (+49.52% at 3 months) by learning to change the position size based on signal confidence, instead of committing a fixed amount in every trade. No supervised model even approaches this horizon, confirming that flexibility in capital allocation, not signal direction, is a meaningful driver of returns.

At 6 months the picture turns more nuanced. XGBoost is in the lead on return for BNB (+30.32%), but GRU offers the best risk-adjusted and drawdown-controlled performance on SOL, indicating that recurrent architectures are better suited to capture the slower, smoother trends that emerge at this horizon. At 12 months the market regime is the decisive factor: PPO dominates on the bearish ADA asset (+74.60%) by learning to stay out of the market during sustained downtrends, while RF provides the tightest drawdown control on the bullish BNB (-3.11%).

Table 4.7: Best Performance by Horizon (across all assets)

Horizon	Best Return	Best Sharpe	Lowest Drawdown
10 Days	RF / XGBoost (ETH: +12.44%)	PPO (BNB: 7.23)	DQN (ADA: -1.00%)
1 Month	DQN (SOL: +36.59%)	RF (BNB: 5.59)	RF (BNB: -2.05%)
3 Months	DQN (BNB: +49.52%)	RF (BNB: 2.54)	XGBoost (BNB: -4.49%)
6 Months	XGBoost (BNB: +30.32%)	GRU (SOL: 1.42)	GRU (SOL: -8.08%)
12 Months	PPO (ADA: +74.60%)	XGBoost (BNB: 1.92)	RF (BNB: -3.11%)

4.4.2.2 Average Returns Across All Assets

Table 4.8 Computing the average performance over all five assets cancels out asset-specific advantages and reveals the structural tendencies of each model more clearly.

The most striking pattern is DQN, which outperforms the other models by more than twice at the 1-month (+16.68%) and 3-month (+11.08%) horizons. That's no coincidence — it's a direct consequence of its dynamic sizing mechanism, which lets it ramp up exposure when it's confident, and ramp it down when the signals are weak. This advantage is real and consistent enough to survive averaging across trending and non-trending assets.

Collectively, reinforcement learning models lead the way at the 12-month horizon. DQN is leading with (+40.72%), followed by PPO with (+31.48%) and XGBoost as the strongest supervised model with (+26.31 %). The RL advantage here is driven mainly by their ADA performance, with the ability to exit the market during a sustained bear run delivering returns that no static supervised model can match. Notably, without ADA, XGBoost would be much closer to DQN at this horizon, suggesting that the RL edge at 12 months is regime-specific, not universal.

At the short end, RF and XGBoost have the same top 10-day average of (+3.72%), showing they're reliable for brief holding periods. Yet, RF's average plummets to (-3.76%) after 6 months, which is the lowest of all the models. This big drop points out a key issue: RF's cautious, low-confident predictions do poorly when trends need to be sustained long-term. Meanwhile, GRU does great compared to other recurrent models. It keeps positive average returns from 1 month to 12 months, and it's got the best 1-month average for non-DQN models at (+6.91%).

Table 4.8: Average Return per Model by Horizon (across all assets)

Model	Avg 10D	Avg 1M	Avg 3M	Avg 6M	Avg 12M
RF	+3.72%	+5.46%	+2.11%	–3.76%	+12.90%
XGBoost	+3.72%	+4.75%	+5.92%	+3.13%	+26.31%
LSTM	+2.91%	+5.58%	+1.71%	–2.19%	+15.80%
GRU	+2.83%	+6.91%	+5.67%	+2.43%	+21.08%
DQN	+0.88%	+16.68%	+11.08%	+2.07%	+40.72%
PPO	+3.33%	+2.60%	+0.93%	–0.42%	+31.48%

4.5 Discussion

Our empirical results in Section 4.4 lead us to a practically and theoretically important conclusion: the performance of algorithmic trading models in cryptocurrency markets is not a fixed property of the model itself but a function of the interaction between the model's architecture, the asset's market regime (Hamilton, 1989; Pagan & Sossounov, 2003), and the investment horizon. This section synthesizes the evidence in five interrelated conclusions and places each finding in relation to the traditional Buy and Hold benchmark, to show where intelligent trading systems really add financial value and where passive exposure is the more competitive approach.

4.5.5.1 Dynamic Position Sizing Enhances DQN's Short-to-Medium Return

The most immediate and measurable advantage noted in this study is the ability of DQN to learn position size as a tradeable parameter rather than treat every buy signal as an equal commitment of capital. DQN produces outsized returns in trending markets that no fixed-allocation model can match by using a 30%/70% sizing mechanism, and taking a smaller position when confidence in the signal is moderate and a larger position when confidence is high. The numbers tell the story: (+36.59%) on SOL at 1 month horizon and (+49.52%) on BNB at 3 months horizon, with average returns across all assets more than double those of the second best model at both horizons.

The result is a direct answer to the criticism by Yang & Malik (2024)that the traditional binary buy/sell action space does not consider the capital allocation decisions of real traders. The evidence here shows that this is not a theoretical concern, but a measurable performance gap. A trading agent that knows how much to invest, not just if to invest, is a fundamentally more powerful trading agent.

DQN is significantly better than Buy and Hold, especially for short to medium horizons. Buy and Hold can only compete at the 12-month horizon for strongly trending assets like BNB, where passive exposure is able to benefit from the full duration of an uptrend. We show that no passive strategy can provide the kind of momentum-timed returns that DQN's dynamic sizing yields, at 1 and 3 month horizons. This is exactly where algorithmic trading adds real and irreplaceable financial value.

4.5.5.2 Reinforcement Learning Agents Outperform Supervised Models in Bearish Regimes

Where DQN is good at exploiting momentum, PPO is good at surviving its absence. The ADA results over 12 months are the clearest and most compelling evidence in the entire study of what reinforcement learning can do that supervised models fundamentally cannot.

During a long bearish phase when ADA lost more than half of its value in Buy and Hold (-51.98%), PPO returned (+74.60%) and DQN (+58.03%). The best supervised models, XGBoost at (+13.24%) and LSTM at (+8.03%), produced a fraction of those gains, and both RF and GRU both recorded losses. The difference in performance is not marginal, it is structural.

The difference is in how each model responds to a changing environment. Once trained on historical patterns, a supervised classifier is frozen. If the market regime changes, like what happened with ADA, the model keeps sending buy signals on patterns that no longer hold and losses accumulate. In contrast, a reinforcement learning agent is constantly evaluating the reward signal its actions generate. If the purchase is always rewarded negatively, the agent will not purchase and will keep the cash. It doesn't need to be retrained; it adapts inside the episode itself.

This behavioral difference is what Adedigba et al. (2025)predicted theoretically and this study confirms empirically in a live bearish market environment. The practical implication for portfolio management is

that RL agents provide a degree of downside protection in assets with uncertain or deteriorating trend direction, which supervised models are incapable of providing.

4.5.5.3 Machine learning Excel at Capital Preservation

While reinforcement learning agents dominate in terms of raw return in trending and bearish conditions, the evidence also shows that supervised ensemble methods, especially RF, have an equally important and complementary role: capital preservation.

RF gets the maximum drawdowns of just (-2.05%) on BNB at 1 month and (-3.11%) at 12 months which is the lowest values recorded by any model across the whole study. These are not random numbers. They are a testament to the structural conservatism of the Random Forest algorithm: the need for consensus across many decision trees before a trade signal is issued means false positives are rare and losses from mistimed trades are minimized. This conservatism, as predicted by Jay & Berlanga (2024), does not come without a tradeoff in return –with a 12 month average return of (+12.90%), RF has the lowest return of all the models. However, for strategies operating under strict drawdown constraints, this tradeoff is not a weakness, it is the entire point.

XGBoost is somewhere in the middle, arguably making it the most versatile supervised model in the study. Its ability to pick up deeper feature interactions helps it spot directional moves that RF misses, delivering a 12-month return of (+55.43%) on BNB, while drawdowns are still relatively well contained. XGBoost is the best option in the supervised world for investors who want to get the best return-to-risk ratio in trending markets.

4.5.5.4 Ethereum Is the Hardest Asset for Algorithmic Trading

ETH is unique amongst the other four assets in this study, not in that it is unprofitable at all horizons, but rather in that it is the only asset for which model performance declines as the horizon increases.

At 10 days, RF and XGBoost generate a (+12.44%) return on ETH, showing that short-term ensemble trading is able to find robust signals even in a highly volatile asset. But that window closes quickly. DQN reaches (-3.35%) on ETH with a maximum drawdown of (-31.05%) by 12 months, the highest drawdown on all five cryptocurrencies in the entire study. No model reliably generates positive returns on ETH beyond the short term horizon.

There are two reasons for this. First, ETH has high volatility but no sustained directional trend during the test period, which implies that supervised models generate frequent but unreliable buy signals, and RL agents struggle to find a stable policy to commit to. Second and more specifically, the feature set used in this study does not include ETH specific on-chain data – gas usage, DeFi total value locked, and staking yields – which are widely known to be significant drivers of ETH price behavior. A model trained without these features is effectively trading ETH blind to some of the most important price signals.

This result has a direct methodological implication: asset-specific feature engineering is not optional for ETH. General price and volume traits that work well for BNB and SOL are not enough for an asset whose price is as much impacted by network activity as market sentiment.

4.5.5.5 No Single Model Dominates Across All Assets and Horizons

The most significant point obtained with this study is not which model performs best : it is that no model consistently performs best. Every model in this study has a context where it excels and a context where it fails, and understanding that boundary is more valuable than any single performance metric, as shown in Table 4.9.

The prime example of this duality is DQN. It dominates at 1 and 3 month horizons with average returns more than double that of any competing model, but delivers near-zero or negative returns at the 10-day horizon and significantly underperforms supervised methods on BNB over 6 months. Its strength, the aggressive exploitation of momentum through dynamic position sizing, becomes a liability in the absence of a clear market trend or a holding period that is too short for momentum to develop.

RF is another profile. It is the most robust model for capital preservation, with maximum drawdowns as low as (-2.05 %) on BNB, and it is the best performing model at the 10-day horizon on ETH. However, it has the lowest 12 month average return of all models (+12.90%) and collapses to (-3.76%) average at 6 months. The very conservatism of RF that makes it such a strong risk management tool becomes a structural weakness when you need sustained trend following.

PPO occupies a different sort of role. It has the highest single asset return in the entire study (+74.60% on ADA at 12 months) and the best Sharpe ratio at the 10-day horizon on BNB (7.23), but consistently

underperforms at 3 and 6 month horizons across multiple assets. Its strength is regime adaptation — learning when not to trade — which pays off meaningfully at the extremes of the horizon spectrum.

This is not a failing of the models. It reflects how financial markets actually behave. Different assets, different regimes, different horizons demand fundamentally different approaches. A momentum optimised model will perform badly in consolidation. A capital preservation model will not have sustained trends. A model that works in bear markets may trade too much in calm ones.

The practical implication is clear and actionable: model selection should not be treated as a one-time fixed decision. The summary of Table 23 suggests that for 10-day horizons, under any market condition, ensemble methods are the proper way forward. for 1 to 3 month trending markets, DQN is the best way to follow. GRU provides the most reliable balance of return and drawdown control at 6 months, and at 12 months the market regime itself should determine the choice, RL methods for bearish or mixed conditions, XGBoost for confirmed bullish trends. Therefore, the most important direction for future research is not to build a better single model but to build a framework that makes this selection dynamically and automatically based on the detected market regime.

Taken together, these five conclusions provide a coherent and actionable picture. DQN is a suitable model when momentum is present and the horizon is short to medium. PPO is the right model for a bearish or regime shifting market with a long horizon. If your primary goal is to protect your capital, then RF is the right model. XGBoost is the model of choice when the asset is trending and a balance of return and risk is needed. GRU is the right model when consistency over medium horizons is more important than peak performance. But none of these statements is universally true. Each is conditionally true, and the condition is the market regime. This is the principal lesson of this study. It will be the basis for future work on a dynamic regime-aware model selection framework.

Table 4.9: Recommended Models by Investment Horizon and Market Condition

Investment Horizon	Recommended Model(s)	Market Condition
10 days	RF / XGBoost	All conditions
1 and 3 months	DQN	Trending markets
6 months	GRU	All conditions
12 months	PPO / DQN	Bearish / Mixed
12 months	XGBoost	Bullish / Trending

4.5.5.6 Implementation

Figure 4.28 shows The six models developed and evaluated in this research are deployed in Trademind IA, an AI-driven real-time trading dashboard available on both mobile and desktop interfaces. The platform integrates all six models — LSTM, GRU, XGBoost, Random Forest, PPO, and DQN — into a single unified application, consolidating the outputs of each model into one actionable trading signal: Buy, Sell, or Hold. Rather than requiring the user to interpret raw model outputs or manage multiple tools separately, Trademind IA abstracts the full model pipeline into a clean decision interface. The platform is designed to bridge the gap between the backtesting framework developed in Chapter 3 and practical real-world deployment, allowing users to select an asset, choose an investment horizon, and receive a model-driven signal in real time.

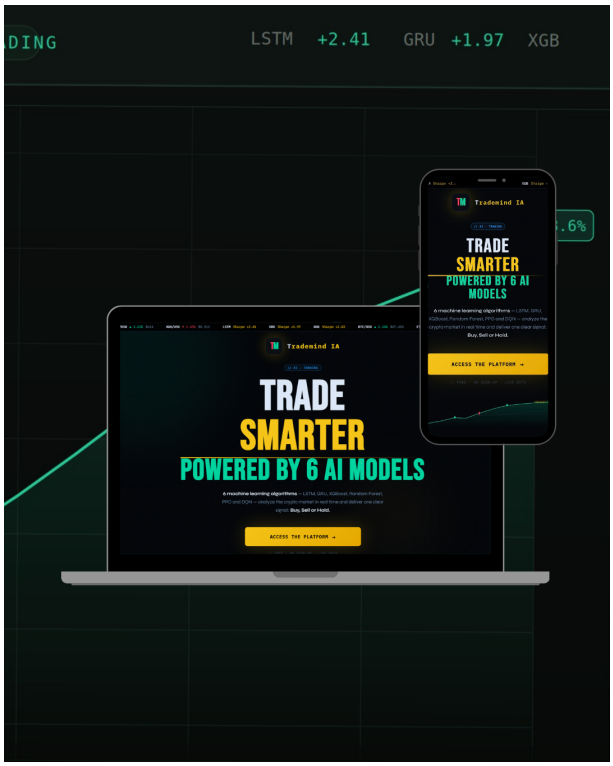


Figure 4.28: Trademind IA landing page on mobile

4.5.6.1 Signal Configuration Panel

User can choose asset (BTC, ETH, ADA, SOL, BNB), investment horizon (from 10 days to 12 months), and model mode: Auto Best (automatic selection of the model with the highest Sharpe ratio) or Manual (direct selection of the model). Once configured the platform performs inference for all the loaded models and outputs the signal of the best performing model, as shown in Figure 4.29.

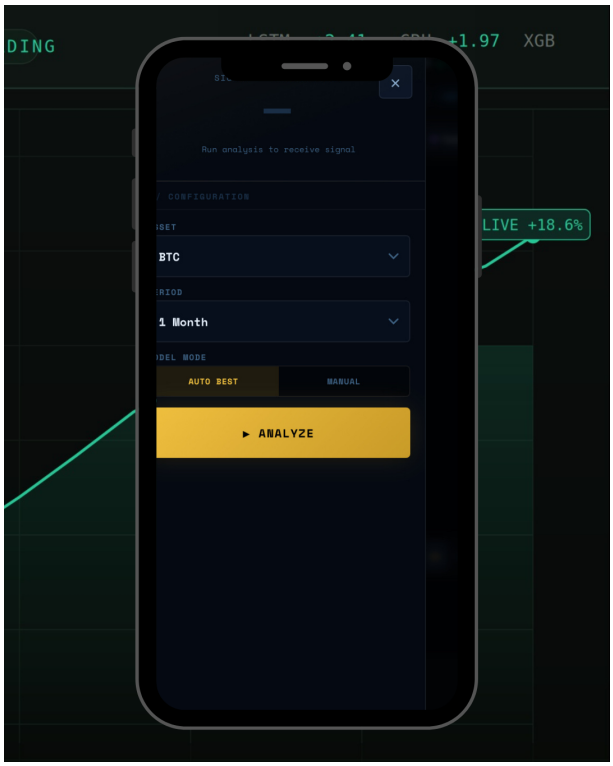


Figure 4.29: Signal configuration panel on mobile

4.5.6.2 Performance Dashboard and Chart

The main dashboard shows the backtest equity curve, drawdown profile and live OHLCV candlestick chart with Buy/Sell signal overlays. On desktop all the panels are visible at once, on mobile are available via dedicated tabs (Config, Chart, Signal, Trade). Key metrics (return, Sharpe ratio and maximum drawdown) update in real-time, as shown in Figure 4.30.

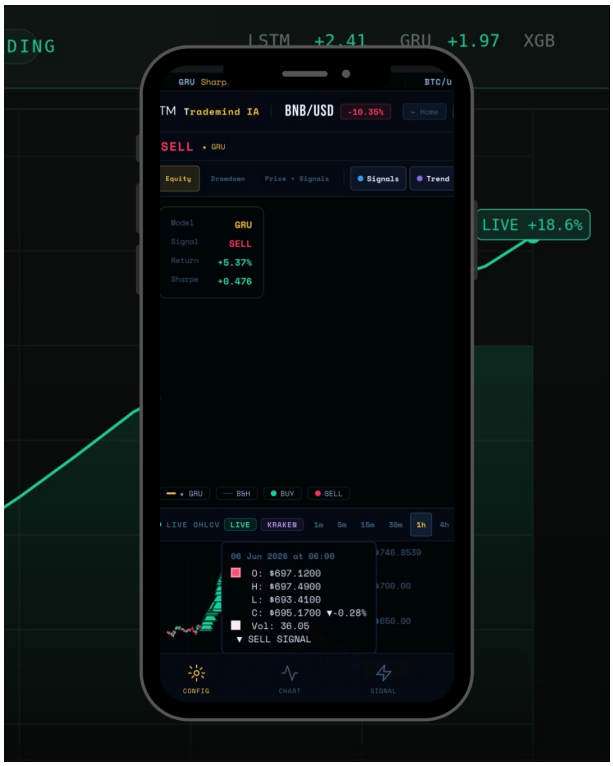


Figure 4.30: Main trading dashboard on mobile

4.5.6.3 Trade Execution Panel

There is a simulated trade execution interface on the platform, available for both mobile and desktop. The panel shows the current balance, asset price and quantity input, and the key statistics of the active model, namely signal direction, Sharpe ratio, return, maximum drawdown and number of trades, providing the complete context of the decision at the time of execution, as shown in Figure 4.31.

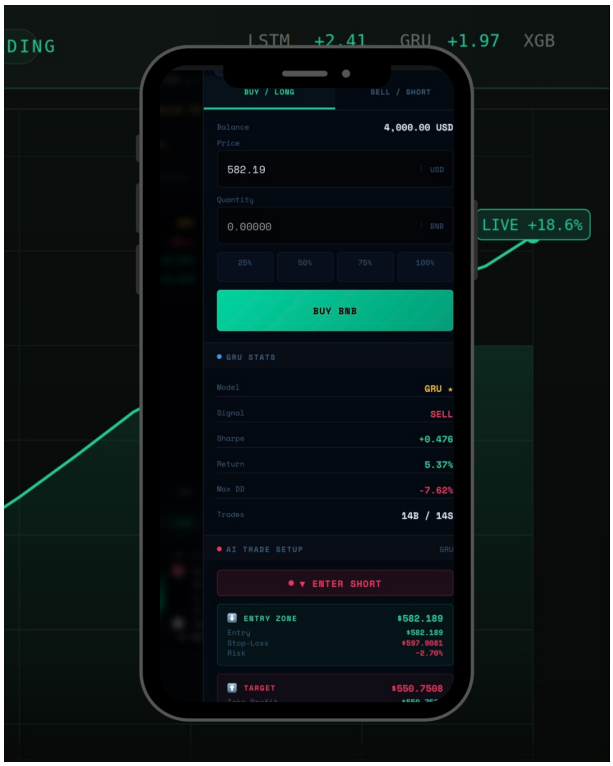


Figure 4.31: Simulated trade execution panel on mobile

4.5.5.7 Conclusion and Future Work

This chapter evaluated six models — DQN, PPO, LSTM, GRU, Random Forest, and XGBoost — across five investment horizons (10 days, 1 month, 3 months, 6 months, and 12 months) on five cryptocurrency assets (BTC, ETH, ADA, SOL, BNB) using 2024 out-of-sample market data.

The results confirm that no single model dominates across all assets and horizons, which is consistent with the findings of Jay & Berlanga (2024) and Adedigba et al. (2025). DQN demonstrates clear superiority at short to medium horizons, a direct consequence of its dynamic position sizing mechanism that allocates capital proportionally to signal confidence (Yang & Malik, 2024). PPO proves to be the most effective model in bearish and regime-shifting conditions, achieving the highest individual return in the entire study (+74.60% on ADA at 12 months). Among supervised models, XGBoost delivers the strongest performance on trending assets, while Random Forest provides the tightest drawdown control, recording maximum drawdowns as low as −2.05% on BNB at 1 month — validating the capital preservation thesis proposed by Jay & Berlanga (2024). GRU emerges as the most consistent recurrent model across medium horizons, outperforming LSTM on four of the five assets at the 6-month horizon.

These results open several natural directions for future research. First, the robustness of the model rankings should be tested by extending the evaluation beyond the single year of 2024 to cover multiple market cycles, including bull, bear, and consolidation regimes. Second, incorporating asset-specific on-chain data — such as gas usage, DeFi total value locked, and staking yields — would likely improve model performance on Ethereum, which proved the most challenging asset in this study. Third, and most importantly, the development of a dynamic regime-aware model selection framework — one that automatically identifies the current market regime and assigns the optimal model to the right asset at the right horizon — represents the most promising and impactful direction for future research in algorithmic cryptocurrency trading.

Bibliography

[1] Weber, M., Candia, B., Coibion, O., and Gorodnichenko, Y. (2023). *Do You Even Crypto, Bro? Cryptocurrencies in Household Finance*. NBER Working Paper No. 31284. https://www.nber.org/papers/w31284

[2] Delfabbro, P., King, D., and Williams, J. (2021). *The Psychology of Cryptocurrency Trading: Risk and Protective Factors*. Journal of Behavioral Addictions, 10(2), 201–213. https://researchnow.flinders.edu.au/files/47324207/Delfabbro_Psychology_P2021.pdf

[3] Adedigba, D., Agbolade, D., and Hasan, R. (2025). *A Comparative Analysis of Predictive Machine Learning Models for Cryptocurrency Profit Maximization*. https://link.springer.com/article/10.1007/s44163-025-00519-y

[4] Coulling, A. (2013). *A Complete Guide to Volume Price Analysis*. Amazon Publishing.

[5] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf

[6] Buterin, V. (2015). *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. https://ethereum.org/en/whitepaper/

[7] BNB SmartChain (2022). *BNB Smart Chain: Proof of Staked Authority*. https://docs.bnbchain.org/bnb-smart-chain/introduction/?h=proof+staked

[8] Hoskinson, C. (2017). *Cardano Settlement Layer Documentation*. IOHK. https://docs.cardano.org

[9] Kiayias, A., Russell, A., David, B., and Oliynykov, R. (2017). *Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol*. https://eprint.iacr.org/2016/889.pdf

[10] Yakovenko, A. (2018). *Solana: A new architecture for a high performance blockchain*. https://solana.com/solana-whitepaper.pdf

[11] Solana Foundation (2020). *Proof of History: A Clock for Blockchain*. https://solana.com/news/proof-of-history

[12] Murphy, J. J. (1999). *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. New York Institute of Finance, Prentice Hall Press. https://archive.org/details/technicalanalysi0000murp

[13] Fama, E. F. (1970). *Efficient Capital Markets: A Review of Theory and Empirical Work*. The Journal of Finance. https://doi.org/10.1111/j.1540-6261.1970.tb00518.x

[14] Covel, M. W. (2009). *Trend Following: Learn to Make Millions in Up or Down Markets*. FT Press. https://archive.org/details/trendfollowingle0000cove

[15] Poterba, J. M. and Summers, L. H. (1988). *Mean Reversion in Stock Prices: Evidence and Implications*. Journal of Financial Economics, 27–59. https://www.nber.org/papers/w2343

[16] Jegadeesh, N. and Titman, S. (1993). *Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency*. The Journal of Finance. https://www.bauer.uh.edu/rsusmel/phd/jegadeesh-titman93.pdf

[17] Aldridge, I. (2013). *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*. Wiley.

[18] Chen, T. and Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. https://arxiv.org/pdf/1603.02754

[19] Breiman, L. (2001). *Random Forests*. https://doi.org/10.1023/A:1010933404324

[20] Hochreiter, S. and Schmidhuber, J. (1997). *Long Short-Term Memory*. https://doi.org/10.1162/neco.1997.9.8.1735

[21] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. https://arxiv.org/pdf/1406.1078

[22] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. https://arxiv.org/pdf/1707.06347

[23] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). *Human-Level Control through Deep Reinforcement Learning*. https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf

[24] Jay, P. and Berlanga, A. (2024). *Smart Trading Using Technical Analysis on the Crypto Market*. SSRN Working Paper. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4987237

[25] Tay, J. and Lim, K. (2024). *Deep Reinforcement Learning in Cryptocurrency Trading*. Journal of Telecommunications and the Digital Economy. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4987237

[26] Yang, H. and Malik, S. (2024). *Reinforcement Learning Pair Trading: A Dynamic Scaling Approach*. MDPI Risks and Financial Innovation. https://doi.org/10.3390/jrfm17120555

[27] Seabe, P. L., Moutsinga, C. R. B., and Pindza, E. (2023). *Forecasting Cryptocurrency Prices Using Deep Learning: Integrating Financial, Technological, and Text Data*. Fractal and Fractional. https://doi.org/10.3390/fractalfract7020203

[28] Brockman, G., Cheung, V., Pettersson, L., et al. (2016). *OpenAI Gym*. https://arxiv.org/pdf/1606.01540

[29] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). *Stable-Baselines3: Reliable Reinforcement Learning Implementations*. Journal of Machine Learning Research. https://jmlr.org/papers/v22/20-1364.html

[30] Paszke, A., Gross, S., Massa, F., et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Advances in Neural Information Processing Systems. https://arxiv.org/pdf/1912.01703

[31] Buterin, V., et al. (2021). *EIP-1559: Fee Market Change for ETH 1.0 Chain*. Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-1559

[32] Schär, F. (2021). *Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets*. Federal Reserve Bank of St. Louis Review, 103(2). https://doi.org/10.20955/r.103.153-74

[33] Ethereum Foundation (2022). *Ethereum Proof-of-Stake: Staking and Validator Rewards*. https://ethereum.org/en/staking/

[34] Hamilton, J. D. (1989). *A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle*. Econometrica. https://www.jstor.org/stable/1912559

[35] Pagan, A. R. and Sossounov, K. A. (2003). *A Simple Framework for Analysing Bull and Bear Markets*. Journal of Applied Econometrics. https://www.jstor.org/stable/30035186

[36] Baur, D. G. and Dimpfl, T. (2018). *Asymmetric Volatility in Cryptocurrencies*. Economics Letters. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3347617

[37] Chu, B. (2025). *Technical Analysis with Machine Learning Classification Algorithms: Can It Still 'Beat' the Buy-and-Hold Strategy?* Computational Economics. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4765615

[38] Brock, W., Lakonishok, J., and LeBaron, B. (1992). *Simple Technical Trading Rules and the Stochastic Properties of Stock Returns.* The Journal of Finance. https://www.jstor.org/stable/2328994

[39] Chong, T. T. L. and Ng, W. K. (2014). *Revisiting the Performance of MACD and RSI Oscillators.* Journal of Risk and Financial Management. https://www.mdpi.com/1911-8074/7/1/1

[40] Lento, C. (2007). *Tests of Technical Trading Rules in the Asian-Pacific Equity Markets: A Bootstrap Approach.* Academy of Financial and Quantitative Analysis. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1111879

[41] Detzel, A., Liu, H., Strauss, J., Zhou, G., and Zhu, Y. (2018). *Bitcoin: Learning, Predictability and Profitability via Technical Analysis.* Working Paper, Washington University in St. Louis. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3115846

[42] Ahmad, I., Ahmad, M. O., Alqarni, M. A., and Almazroi, A. A. (2021). *Using Algorithmic Trading to Analyze Short Term Profitability of Bitcoin.* PeerJ Computer Science. https://peerj.com/articles/cs-337/

[43] Hamayel, M. J. and Owda, A. Y. (2021). *A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms.* https://www.mdpi.com/2673-2688/2/4/30

[44] Konda, V. R. and Tsitsiklis, J. N. (2003). *Actor-Critic Algorithms.* Neural Information Processing Systems. https://proceedings.neurips.cc/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[45] Liu, R. and Zou, J. (2017). *The Effects of Memory Replay in Reinforcement Learning.* https://arxiv.org/abs/1710.06574

[46] Yang, H., Lim, X. Y., Zhong, S., and Walid, A. (2020). *Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy.* ACM International Conference on AI in Finance. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3690996

[47] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press. http://incompleteideas.net/book/the-book-2nd.html

[48] Ng, A. Y., Harada, D., and Russell, S. J. (1999). *Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping.* In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999).* Morgan Kaufmann. https://dl.acm.org/doi/10.5555/645528.657613

[49] Wilder, J. W. (1978). *New Concepts in Technical Trading Systems.* Trend Research. https://archive.org/details/newconceptsintec00wild

[50] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). *Focal Loss for Dense Object Detection.* Proceedings of the IEEE International Conference on Computer Vision (ICCV). https://arxiv.org/abs/1708.02002

[51] Bollinger, J. (2001). *Bollinger on Bollinger Bands.* McGraw-Hill.

[52] Hayes, A. (2022). *Dollar-Cost Averaging (DCA).* https://www.investopedia.com/terms/d/dollarcostaveraging.asp

[53] Beers, B. (2020). *Buy and Hold.* https://www.investopedia.com/terms/b/buyandhold.asp

[54] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research. https://jmlr.org/papers/v12/pedregosa11a.html

[55] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. (2020). *Array Programming with NumPy.* Nature. https://doi.org/10.1038/s41586-020-2649-2

[56] McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference (SciPy 2010). https://doi.org/10.25080/Majora-92bf1922-00a

[57] Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

[58] Kingma, D. P. and Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1412.6980

[59] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. Journal of Machine Learning Research, 15(1), 1929–1958. https://jmlr.org/papers/v15/srivastava14a.html

[60] Prechelt, L. (1998). *Early Stopping — But When?* In Orr, G. B. and Müller, K.-R. (Eds.), *Neural Networks: Tricks of the Trade*. Springer. https://link.springer.com/chapter/10.1007/3-540-49430-8_3

[61] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. https://www.deeplearningbook.org

[62] Sattarov, O., Khujamatov, E. H., Ismanov, K., and Mallaev, O. U. (2026). *Optimizing Crypto-Trading Performance: A Comparative Analysis of Innovative Reward Functions in Reinforcement Learning Models*. Mathematics. https://www.mdpi.com/2227-7390/14/5/794

[63] Elsayed, N., Maida, A. S., and Bayoumi, M. (2019). *Gated Recurrent Units (GRUs) and the Absence of a Separate Memory Cell*. In IEEE Big Data 2019.

[64] Appel, G. (1979). *The Moving Average Convergence-Divergence Trading Method*. Reprinted archival edition.

[65] Aspray, T. (1986). *MACD Histogram/Momentum*. Technical Analysis of Stocks & Commodities.

[66] Gatev, E., Goetzmann, W. N., and Rouwenhorst, K. G. (2006). *Pairs Trading: Performance of a Relative-Value Arbitrage Rule*. The Review of Financial Studies. http://stat.wharton.upenn.edu/~steele/Courses/434/434Context/PairsTrading/PairsTradingGGR.pdf

[67] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research. https://arxiv.org/pdf/1106.1813

[68] Sharpe, W. F. (1994). *The Sharpe Ratio*. The Journal of Portfolio Management. https://web.stanford.edu/~wfsharpe/art/sr/SR.htm

[69] Park, H. and Seo, Y. (2022). *A Deep Learning-Based Action Recommendation Model for Cryptocurrency Profit Maximization*. Electronics. https://www.mdpi.com/2079-9292/11/9/1466

[70] Magdon-Ismail, M. and Atiya, A. F. (2004). *Maximum Drawdown*. Risk Magazine. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=874069

[71]Capital Research Brasil (2020). *Title of the Report or Article*. Available at: https://www.capitalresearch.com.br (Accessed: date).