

Multi-Agent Deep Reinforcement Learning for Cryptocurrency Trading



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2022
Copyright of Chulalongkorn University

การเรียนรู้แบบเสริมกำลังเชิงลึกแบบหลายตัวกระทำสำหรับการซื้อขายคริปโทเคอร์เรนซี



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title	Multi-Agent Deep Reinforcement Learning for Cryptocurrency Trading
By	Mr. Kittiwit Kumlungmak
Field of Study	Computer Science
Thesis Advisor	Associate Professor PEERAPON VATEEKUL, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

..... Dean of the FACULTY OF ENGINEERING
(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

THEESIS COMMITTEE

..... Chairman
(Professor BOONSERM KIJSIRIKUL, Ph.D.)

..... Thesis Advisor
(Associate Professor PEERAPON VATEEKUL, Ph.D.)

..... Examiner
(Punnarai Siricharoen, Ph.D.)

..... External Examiner
(Thanapat Kangkachit, Ph.D.)



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

กิตติวิรินทร์ กำลังมาก : การเรียนรู้แบบเสริมกำลังเชิงลึกแบบหลายตัวกระทำสำหรับการซื้อขายคริปโต
 เคอร์เรนซี. (Multi-Agent Deep Reinforcement Learning for Cryptocurrency Trading) อ.ที่
 ปริญญาหลัก : รศ. ดร.พีรพล เวทีกุล

การเรียนรู้แบบเสริมกำลัง (Reinforcement learning) เป็นวิธีการที่ถูกนำมาใช้ในการเพิ่มผลกำไรใน
 การซื้อขายคริปโตเคอร์เรนซี (cryptocurrency) อย่างไรก็ตาม ความผันผวนของตลาด โดยเฉพาะในช่วงเวลาที่
 ตลาดเป็นลักษณะตลาดขาลง (Bearish) กลายเป็นอุปสรรคที่สำคัญของด้านนี้ งานวิจัยที่มีอยู่ในปัจจุบัน มีความ
 พยายามที่จะแก้ปัญหานี้โดยการใช้เทคนิค Deep Q-Network (DQN), Advantage Actor-Critic (A2C), และ
 Proximal Policy Optimization (PPO) หรือการผสมผสานกันของเทคนิคดังกล่าว (Ensemble) แต่อย่างไรก็
 ตาม กลไกที่นำมาใช้เพื่อลดความเสียหายในช่วงตลาดขาลงสำหรับคริปโตเคอร์เรนซียังไม่มีประสิทธิภาพ
 เท่าที่ควร ดังนั้นประสิทธิภาพของวิธีการเรียนรู้แบบเสริมกำลังสำหรับการซื้อขายคริปโตเคอร์เรนซียังถูกจำกัด
 เพื่อเอาชนะข้อจำกัดนี้ เรานำเสนอเทคนิคใหม่สำหรับการซื้อขายคริปโตเคอร์เรนซี โดยการใช้การเรียนรู้แบบหลาย
 ตัวกระทำ (Multi-Agent) และฟังก์ชันรางวัลร่วม (Local-Global Reward Function) เพื่อปรับปรุง
 ประสิทธิภาพในการทำงานร่วมกันของตัวกระทำทุกตัว รวมถึงการทำงานของตัวกระทำแต่ละตัวไปพร้อมกันด้วย
 นอกจากนี้ เรายังใช้เทคนิคการปรับปรุงเป้าหมายหลายวัตถุประสงค์ (Multi-Objective Optimization
 Technique) และการทำโทษเมื่อมีการสูญเสียแบบต่อเนื่อง ซึ่งเราเรียกว่า Multi-Scale Continuous Loss
 (MSCL) Reward ที่เราดัดแปลงมาจากการลงโทษแบบเพิ่มเติม (Progressive Penalty) เพื่อป้องกันความ
 สูญเสียต่อเนื่องของมูลค่าพอร์ตการลงทุน ในการประเมินผลของวิธีการที่เรานำเสนอ เราได้ทำการเปรียบเทียบ
 กับเทคนิคอื่น ๆ ที่เป็นที่ยอมรับ และพบว่าผลตอบแทนสะสม (cumulative return) ของเทคนิคของเรามีค่าสูงกว่า
 เทคนิคดังกล่าว โดยเฉพาะในช่วงตลาดขาลง มีเพียงวิธีการของเราเท่านั้นที่สามารถให้ผลกำไรได้ ซึ่งวิธีการของ
 เราสร้างผลตอบแทนสะสมได้ถึง 2.36% ในขณะที่วิธีการอื่น ๆ ที่เรานำมาเปรียบเทียบเกิดการขาดทุนทั้งหมด และ
 เมื่อเปรียบเทียบกับ FinRL-Ensemble ซึ่งเป็นวิธีการที่ใช้การเรียนรู้แบบเสริมกำลัง เราพบว่าวิธีการของเราได้รับ
 ผลตอบแทนสะสมที่สูงกว่าถึง 46.05% ในช่วงตลาดขาขึ้น (Bullish)

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์
 ปีการศึกษา 2565

ลายมือชื่อนิสิต
 ลายมือชื่อ อ.ที่ปรึกษาหลัก

6470140221 : MAJOR COMPUTER SCIENCE

KEYWORD: Cryptocurrency trading, multi-agent reinforcement learning, portfolio management

Kittiwin Kumlungmak : Multi-Agent Deep Reinforcement Learning for Cryptocurrency Trading. Advisor: Assoc. Prof. PEERAPON VATEEKUL, Ph.D.

Reinforcement learning has emerged as a promising approach for enhancing profitability in cryptocurrency trading. However, the inherent volatility of the market, especially during bearish periods, poses significant challenges in this domain. Existing literature addresses this issue through the adoption of single-agent techniques such as deep Q-network (DQN), advantage actor-critic (A2C), and proximal policy optimization (PPO), or their ensembles. Despite these efforts, the mechanisms employed to mitigate losses during bearish market conditions within the cryptocurrency context lack robustness. Consequently, the performance of reinforcement learning methods for cryptocurrency trading remains constrained within the current literature. To overcome this limitation, we present a novel cryptocurrency trading method, leveraging multi-agent proximal policy optimization (MAPPO). Our approach incorporates a collaborative multi-agent scheme and a local-global reward function to optimize both individual and collective agent performance. Employing a multi-objective optimization technique and a multi-scale continuous loss (MSCL) reward, we train the agents using a progressive penalty mechanism to prevent consecutive losses of portfolio value. In evaluating our method, we compare it against multiple baselines, revealing superior cumulative returns compared to baseline methods. Notably, the strength of our method is further exemplified through the results obtained from the bearish test set, where only our approach demonstrates the ability to yield a profit. Specifically, our method achieves an impressive cumulative return of 2.36%, while the baseline methods result in negative cumulative returns. In comparison to FinRL-Ensemble, a reinforcement learning-based method, our approach exhibits a remarkable 46.05% greater cumulative return in the bullish test set.

Field of Study: Computer Science

Student's Signature

Academic Year: 2022

Advisor's Signature

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deepest gratitude and appreciation to the individuals who have played a significant role in the completion of my thesis.

First and foremost, I am indebted to Associate Professor Peerapon Vateekul, my esteemed advisor. His expert guidance, extensive knowledge, and unwavering dedication to my academic growth have been invaluable throughout my time in graduate school. Professor Vateekul's mentorship has not only shaped the direction of my research but has also inspired me to strive for excellence in every aspect of my academic pursuits.

I would also like to extend my heartfelt thanks to Miss Manassakarn, a Ph.D. candidate at the time, whose valuable advice on reinforcement learning has greatly contributed to the development and refinement of my research. Her expertise and willingness to share her knowledge have been pivotal in shaping the quality and depth of my work.

Additionally, I would like to express my sincere gratitude to DataMind lab for their invaluable contribution in enabling me to carry out my work. Their provision of robust computational resources for training and testing has been instrumental in the successful completion of my project.

Furthermore, I would like to express my deep appreciation to my family for their unwavering support and encouragement. Their financial assistance has enabled me to pursue my studies and focus wholeheartedly on my research, without which this achievement would not have been possible.

Lastly, I am grateful to my dear friends, Mr. Sothornin, Mr. Passakorn, Mr. Passin, Mr. Tanatorn, Mr. Natch, and Miss Yanisa, for their constant encouragement, motivation, and understanding throughout this challenging journey. Their unwavering belief in my abilities and their willingness to lend a helping hand during times of difficulty has been a constant source of inspiration.

Kittiwin Kumlungmak

TABLE OF CONTENTS

	Page
ABSTRACT (THAI)	iii
ABSTRACT (ENGLISH)	iv
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER I INTRODUCTION.....	1
1.1 Aims and objectives.....	3
1.2 Scope of work.....	3
1.3 Benefits.....	4
1.4 Publication.....	4
CHAPTER II BACKGROUND	5
2.1 Cryptocurrency market data.....	5
2.2 Financial technical indicator.....	6
2.2.1 Moving Average Convergence Divergence (MACD).....	6
2.2.2 Relative Strength Index (RSI).....	6
2.2.3 Commodity Channel Index (CCI)	7
2.2.4 Average Directional Index (ADX)	7
2.2.5 Cryptocurrency Volatility Index (CVIX).....	8
2.5.6 Social Media Sentiment.....	8
2.5.7 Fear and Greed Index	9

2.3 Portfolio performance measurement	9
2.3.1 Cumulative return.....	9
2.3.2 Sharpe ratio.....	10
2.3.3 Calmar ratio.....	10
2.3.4 Volatility	11
2.3.5 Maximum drawdown	11
CHAPTER III RELATED WORKS	12
3.1 Reinforcement learning.....	12
3.1.1 Valued-based method.....	13
3.1.2 Policy-based method.....	14
3.1.3 Actor-critic method.....	15
3.2 Reinforcement learning for cryptocurrency trading	17
CHAPTER IV CONCEPT AND RESEARCH METHODOLOGY	20
4.1 Data preparation.....	21
4.2 Multi-agent policy optimization for cryptocurrency trading (MAPPO).....	22
4.2.1. State and observation.....	22
4.2.2. Agent	24
4.2.3. Action	27
4.2.4. Reward	27
4.3 Simulated cryptocurrency market environment	33
4.3.1. Action manipulation.....	33
4.3.2. Trade.....	34
4.3.3. State transition	34
4.3.4. Reward calculation.....	35

CHAPTER V EXPERIMENTS AND RESULTS	36
5.1 Dataset.....	36
5.2 Implementation detail	38
5.3 Results	38
5.3.1 Comparing multi-agent to single-agent PPO	39
5.3.2 Comparing combinations of reward terms.....	40
5.3.3 Comparing to baseline methods.....	42
5.3.4 Comparing network architectures	45
5.4 Discussion.....	46
5.4.1 Bearish test result.....	47
5.4.2 Overall test result.....	49
5.4.3 Trade count and transaction cost	49
CHAPTER VI CONCLUSION.....	51
CHAPTER VII Appendices.....	52
7.1 Reward Hyperparameter Tuning.....	52
REFERENCES	53
VITA.....	59

LIST OF TABLES

	Page
Table 1 Cryptocurrency k-line datasets.....	36
Table 2 Comparison between MAPPO and SAPPO. Boldfaces refer to the winners. Gray numbers refer to uninterpretable results	39
Table 3 Comparing the combinations of reward terms assigned to MAPPO including risk-sensitive reward, cost-sensitive reward, and MSCL reward. Undoubtedly, the inclusion of the MSCL reward, our suggested reward term, has the capability to enhance the performance of all combinations. Boldfaces refer to the winners. Gray numbers refer to uninterpretable results.....	41
Table 4 Comparing our MAPPO with risk-sensitive reward, cost-sensitive reward, and MSCL reward to four baseline methods including UBAH, UCRP, Bitcoin, and FinRL-Ensemble. Boldfaces refer to the winners. Gray numbers refer to uninterpretable results.....	43
Table 5 Comparing network architectures. Boldfaces refer to the winners. Gray numbers refer to uninterpretable results.....	46
Table 6 Number of trades and transaction cost of our method compared to the baselines.....	50
Table 7 Validation results of reward hyperparameters tuning. The grey row refers to the best hyperparameter for the reward function.	52

LIST OF FIGURES

	Page
Figure 1 Schema of reinforcement learning.	13
Figure 2 System overview: (a) data preparation, (b) multi-agent proximal policy optimization (MAPPO), and (c) simulated cryptocurrency market environment.....	21
Figure 3 Schema of observation normalization.	24
Figure 4 The network architecture of the agent consists of two components: the actor network on the left-hand side and the critic network on the right-hand side. Each network comprises two layers of Multilayer Perceptron (MLP), commonly referred to as dense layers. Notably, the normalized market data and the portfolio vector are simultaneously provided as observations to both networks.	25
Figure 5 The normalized market data undergoes a feature extraction process after passing through either an LSTM or a GRU layer, followed by a dense layer. Subsequently, it is concatenated with the portfolio vector and forwarded to both the actor and critic networks.....	26
Figure 6 The normalized market data undergoes processing through two blocks of either ResNet or Res2Net, followed by a flatten layer and a dense layer. Subsequently, it is concatenated with the portfolio vector and directed towards both the actor and critic networks. The specific details of the ResNet block and the Res2Net block are depicted in (a) and (b) respectively.	26
Figure 7 Schema presenting the various terms employed in our reward function, with particular emphasis on our proposed reward term, MSCL reward. The green boxes represent reward terms that agents should strive to maximize, while the red boxes indicate penalty terms that agents should aim to minimize.	28
Figure 8 Algorithm of multi-scale continuous loss (MSCL) reward	32

Figure 9 In the simulated cryptocurrency market environment, agents submit actions, and in return, they receive a normalized observation at time $t + 1$ along with corresponding rewards.	35
Figure 10 Normalized close price of each dataset: (a) Cryptocurrency dataset encompasses all the historical data utilized in this paper and is subsequently divided into (b) Training set, (c) Validation set, and (d) Overall test set. The overall test set is further partitioned into (e) Bullish test set, (f) Bearish test set, (g) Up-down test set, and (h) Sideways test set.....	37
Figure 11 Result of the Bearish test set.....	47
Figure 12 Trade signals from the bearish test set. (a) Relative Strength Index (RSI). (b) Average Directional Index (ADX).	48
Figure 13 Multi-Scale Continuous Loss (MSCL) Reward during the bearish test set ...	48
Figure 14 The results of the overall test set.....	49

CHAPTER I

INTRODUCTION

By leveraging blockchain as a foundational technology, cryptocurrencies enable direct transactions between users without the need for intermediaries. As a result, the value of cryptocurrencies remains immune to interference or manipulation by governments or organizations. This unique attribute allows cryptocurrencies to acquire value without relying on physical representation.

Since the introduction of Bitcoin in 2009, a multitude of cryptocurrencies have surfaced, leading to the establishment of various alternative utility tokens such as Ethereum and Ripple. According to coinmarketcap.com, the platform that provides cryptocurrency market data, there are currently over 8,000 listed cryptocurrencies. In November 2021, the combined market capitalization of these cryptocurrencies reached a peak of \$2.97 trillion [1].

As a result of their significant value growth, cryptocurrencies have emerged as highly sought-after alternative assets for investment. This popularity has led to the establishment of over two hundred cryptocurrency exchanges by the end of 2022. A notable example is Binance, a prominent cryptocurrency exchange that boasts a daily trading volume surpassing \$15 trillion. This substantial trading volume serves as a testament to the potential profitability associated with trading on cryptocurrency exchanges [2].

In parallel with the surge of cryptocurrencies, the field of machine learning has witnessed rapid advancements and widespread adoption within the financial industry. In particular, researchers have been actively exploring various machine-

learning techniques to predict the price movements of cryptocurrencies. Notably, Khedr et al. [3] conducted a study and discovered a consistent increase in the number of publications focused on cryptocurrency price prediction since 2017. They found that approximately 54.03% of these publications incorporate machine-learning methodologies for prediction purposes. This trend underscores the growing interest in leveraging machine learning algorithms to forecast cryptocurrency price dynamics.

Supervised learning and reinforcement learning are extensively explored machine learning approaches that hold potential in the domain of cryptocurrency trading. Firstly, supervised learning involves constructing models that can predict future outcomes based on historical data. This approach requires the availability of features and corresponding labels. While models developed through supervised learning techniques can effectively forecast future cryptocurrency prices, the ability to generate profits from trading is a different matter [4-6]. Merely having knowledge of future prices does not guarantee profitable trades. Factors such as transaction fees and market volatility also need to be carefully considered to ensure effective trading and profitability.

In contrast, reinforcement learning offers a framework for training agents to make optimal decisions in response to a dynamic environment, such as the cryptocurrency market. Agents, based on the current state, select actions (e.g., buy, sell, or hold tokens) that maximize a designated reward function. Additionally, the reward function can be defined in a multi-objective manner, incorporating factors like return, transaction cost, and volatility. Consequently, reinforcement learning proves to be a more suitable approach for cryptocurrency trading.

This study presents a novel cryptocurrency trading strategy, contributing in the following ways:

- Adoption of collaborative multi-agent proximal policy optimization (MAPPO), where each agent specializes in trading a specific token from the portfolio. The strategy incorporates a local-global reward function that simultaneously optimizes the individual and collective performance of the agents.
- Implementation of a multi-objective optimization technique that includes return, volatility, and transaction cost within the reward function. Moreover, the study employs a multi-scale continuous loss (MSCL) reward mechanism to train agents effectively and prevent continuous loss of portfolio value, thereby achieving superior trading performance.

1.1 Aims and objectives

- To propose a cryptocurrency trading strategy based on multi-agent deep reinforcement learning, aiming to optimize both returns and portfolio risk reduction concurrently.
- To assess the effectiveness of the multi-agent deep reinforcement learning approach in cryptocurrency trading, specifically under various market conditions, including bullish, bearish, and sideways markets.

1.2 Scope of work

- Evaluate the proposed multi-agent deep reinforcement learning method, covering the following scopes:

- Conducting experiments using hourly k-line data obtained through the Binance API.
- Investigating the performance of 5 prominent tokens relative to Tether (USDT): Cardano (ADAUSDT), Binance Coin (BNBUSDT), Bitcoin (BTCUSDT), Ethereum (ETHUSDT), and Ripple (XRPUSDT).
- Employing data spanning from April 1, 2021, to August 31, 2022.
- Evaluate the effectiveness of the proposed multi-agent deep reinforcement learning method by analyzing portfolio performance measurements, including cumulative return, Sharpe ratio, Calmar ratio, volatility, and maximum drawdown (MDD).

1.3 Benefits

- Enable the autonomous execution of trades that are better aligned with cryptocurrency market conditions.
- Facilitate simultaneous monitoring and management of multiple tokens.
- Optimize profit while effectively managing volatility and transaction costs.

1.4 Publication

- K. Kumlungmak and P. Vateekul, "Multi-Agent Deep Reinforcement Learning With Progressive Negative Reward for Cryptocurrency Trading," in IEEE Access, doi: 10.1109/ACCESS.2023.3289844.
 - IEEE Access, Institute of Electrical and Electronics Engineers Inc.
 - Q1, Tier 1
 - Impact Factor = 3.476.

CHAPTER II

BACKGROUND

Within this chapter, an exposition of the contextual understanding relevant to the dissertation is presented. It encompasses an elucidation of the data description, algorithmic approach, problem formulation, experimental scenario, and evaluation metrics.

2.1 Cryptocurrency market data

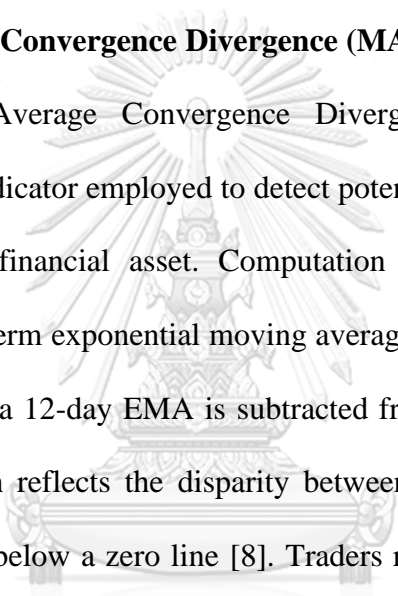
Within the domain of the cryptocurrency market, the price volatility of a token during a specified timeframe is conventionally depicted using k-line data. A standard collection of k-line data encompasses the opening price, highest price, lowest price, and closing price, as enumerated subsequently:

- The "open price" denotes the initial valuation of a token at the commencement of the period.
- The "high price" corresponds to the maximum price attained by the token during the specified period.
- The "low price" signifies the minimum price reached by the token within the given timeframe.
- The "close price" represents the final price of the token at the culmination of the period.

2.2 Financial technical indicator

Within the realm of finance, technical indicators serve as signals employed by traders to assess an asset's characteristics based on historical data encompassing price and volume. These indicators provide guidance to traders regarding optimal moments for buying and selling [7].

2.2.1 Moving Average Convergence Divergence (MACD)

The Moving Average Convergence Divergence (MACD) is a widely recognized technical indicator employed to detect potential buy and sell signals within the price trend of a financial asset. Computation of the MACD line involves subtracting the longer-term exponential moving average (EMA) from the shorter-term EMA. Conventionally, a 12-day EMA is subtracted from a 26-day EMA to generate the MACD line, which reflects the disparity between the two EMAs and displays oscillations above and below a zero line [8]. Traders rely on the MACD indicator to determine optimal entry and exit points, validate trends, and assess the intensity of price movements.  CHULALONGKORN UNIVERSITY

2.2.2 Relative Strength Index (RSI)

The relative strength index (RSI) is a momentum indicator widely utilized in technical analysis. Its purpose is to assess the velocity and magnitude of recent price changes in a security, aiming to identify potentially overvalued or undervalued conditions. The RSI not only indicates overbought and oversold securities but also provides insights into potential trend reversals or corrective pullbacks in price. By analyzing the RSI, traders can determine opportune moments for buying and selling.

In conventional practice, an RSI reading of 70 or above signifies an overbought situation, while a reading of 30 or below indicates an oversold condition [9].

2.2.3 Commodity Channel Index (CCI)

The Commodity Channel Index (CCI) is a momentum-based oscillator created by Donald Lambert that serves as a valuable tool for assessing the overbought or oversold condition of an investment vehicle. This technical indicator evaluates both the direction and strength of price trends, empowering traders to make informed decisions regarding trade entry or exit, trade avoidance, or the addition of positions. By monitoring the behavior of the CCI, traders can receive valuable trade signals that guide their actions in the market [10].

2.2.4 Average Directional Index (ADX)

The Average Directional Index (ADX) is a widely used technical indicator that measures the strength of a trend. By calculating the moving average of price range expansion over a specific time period, typically set to 14 bars but adjustable to other timeframes, ADX provides valuable insights into the intensity of a trend. Represented as a single line, ADX values range from zero to 100. Traders often consider ADX readings above 25 as an indication of a strong trend suitable for trend-trading strategies. Conversely, when ADX falls below 25, many traders prefer to avoid trend-trading strategies. It is important to note that ADX is a non-directional indicator, meaning it assesses trend strength regardless of whether the price is moving upwards or downwards. This versatile tool can be effectively utilized across various trading vehicles, including stocks, mutual funds, exchange-traded funds (ETFs), and futures [11].

2.2.5 Cryptocurrency Volatility Index (CVIX)

The Cryptocurrency Volatility Index (CVIX) functions as a quantitative metric that quantifies the anticipated level of price fluctuations within the overall cryptocurrency market over a 30-day period. Utilizing the Black-Scholes option pricing model, the CVIX is specifically designed to provide insightful information regarding market volatility. Ranging from 0 to 200, the CVIX assigns a value of 200 to indicate the highest level of implied volatility, while a value of zero signifies the lowest volatility. This index serves as a valuable instrument for investors, empowering them to adapt their trading strategies based on varying CVIX values and effectively manage potential risks. It is important to note that a higher CVIX value implies increased risks, but it also presents the potential for higher returns [12].

2.5.6 Social Media Sentiment

With the rise of artificial intelligence and machine learning, sentiment analysis tools have become increasingly valuable in assessing social media sentiment and its potential influence on the cryptocurrency market. Pant et al. [4] utilized sentiment analysis on Twitter to forecast the unpredictable price fluctuations of Bitcoin, achieving an impressive accuracy of 81.39% in classifying tweets as positive or negative. Additionally, by employing a Recurrent Neural Network (RNN), they attained a commendable accuracy of 77.62% in predicting Bitcoin's price. Similarly, Vo et al. [6] harnessed the power of sentiment analysis on news data to precisely anticipate the price direction of Ethereum, providing users with valuable insights for making informed decisions about whether to buy, sell, or hold.

2.5.7 Fear and Greed Index

The Fear and Greed Index in the cryptocurrency domain is a sentiment-based indicator that evaluates the prevailing emotional state of market participants. It incorporates multiple data points, including market volatility, trading volume, social media activity, surveys, and price movements, to generate a sentiment reading on a scale of 0 to 100, ranging from extreme fear to extreme greed. This index offers valuable insights into market sentiment and potential market conditions, whereby lower values indicate fear and bearish sentiment, while higher values indicate greed and bullish sentiment. It is important to note that the Fear and Greed Index should be used in conjunction with other factors when making investment decisions, as it can serve as a contrarian indicator [13].

In our approach, we derive technical indicators from historical k-line data and utilize them as features following the study done by Yang et al. [14]. Notably, we have utilized a publicly available library named Stock-stats [15] to compute the subsequent technical indicators

2.3 Portfolio performance measurement

In order to assess the effectiveness of our approach and make comparisons with baseline strategies, we utilize the following metrics.

2.3.1 Cumulative return

A cumulative return serves as an indicator of a trading strategy's profitability. This metric quantifies the overall gain or loss incurred during the trading period relative to the initial capital, expressed as a percentage. The calculation of a cumulative return is as follows:

$$\text{Cumulative Return} = \frac{v_{terminal} - v_{initial}}{v_{initial}} \quad (1)$$

where $v_{terminal}$ represents the portfolio value at the conclusion of the trading period, and $v_{initial}$ corresponds to the initial portfolio value [16].

2.3.2 Sharpe ratio

The Sharpe ratio is a risk-adjusted measure of the return on an investment strategy, utilized to facilitate comparisons among investments with varying levels of risk and return. This metric is mathematically defined as:

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p} \quad (2)$$

where R_p is the average return of an interested frequency, R_f is the risk-free rate, and σ_p is the standard deviation of the returns of an interested frequency. A higher Sharpe ratio signifies that the strategy attains a superior return relative to the associated risk [17]. In this study, the Sharpe ratio is computed using hourly returns. Additionally, for simplicity, the risk-free rate is assumed to be zero.

2.3.3 Calmar ratio

The Calmar ratio is an additional risk-adjusted measure that contrasts the portfolio return with its maximum drawdown (MDD) observed throughout the trading period. A higher Sharpe ratio implies a greater return per unit of maximum drawdown (MDD). This ratio can be mathematically defined as:

$$\text{Calmar Ratio} = \frac{R_p - R_f}{MDD} \quad (3)$$

where MDD denotes the maximum drawdown experienced throughout the entirety of the trading period [18].

Similar to the Sharpe ratio, we simplify the analysis by assuming a risk-free rate of zero. However, in our experiment, the average portfolio return obtained at a one-hour frequency is comparatively small when compared to the magnitude of the maximum drawdown (MDD) observed throughout the trading period. Consequently, this discrepancy poses challenges in interpretation. As a resolution, we employ a cumulative return instead of the portfolio return, albeit with a slight deviation from its original definition. Nonetheless, the interpretation of the results remains unchanged.

2.3.4 Volatility

Volatility, which represents the variance of a portfolio and signifies the comprehensive risk of an investment, can be defined as:

$$Volatility = \frac{\sigma(R_p)}{\sqrt{T}} \quad (4)$$

where T represents the number of periods within the desired time horizon [19]. In our experiment, we consider $T = 365 \cdot 24 = 8,760$ periods to obtain the annualized volatility.

2.3.5 Maximum drawdown

Maximum drawdown (MDD) quantifies the potential downside risk during a trading period. MDD captures the largest decline in portfolio values from a peak to a subsequent trough, prior to a new peak, and can be expressed as:

$$MDD = \frac{v_{trough} - v_{peak}}{v_{peak}} \quad (5)$$

where v_{trough} represents the portfolio value at the trough, while v_{peak} corresponds to the portfolio value at the peak. Consequently, MDD can be regarded as the most substantial loss encountered by a portfolio throughout the trading period [20].

CHAPTER III

RELATED WORKS

This chapter focuses on the exploration of reinforcement learning methods relevant to the present study. Furthermore, an assessment of recent research articles concerning reinforcement learning for cryptocurrency is conducted.

3.1 Reinforcement learning

Figure 1 provides an illustration of the underlying concept of reinforcement learning. In this framework, an agent perceives the current state or observation s_t given by the environment and subsequently responds to the environment with an action a_t . The agent's policy $\pi(a_t|s_t)$, a function mapping states to actions, determines the chosen action a_t based on the observed state s_t . Following the action, the environment provides feedback to the agent in the form of a new state s_{t+1} , and a reward signal r_{t+1} , which reflects the agent's performance in accomplishing its objective. Utilizing this reward signal, the agent employs a reinforcement learning algorithm to enhance its policy, enabling continual improvement in its decision-making capabilities. The interaction between the agent and the environment persists until a terminal state is reached. The ultimate aim is to maximize the expected reward at time t , which is computed as the sum of all future rewards discounted by a factor $\gamma \in (0,1]$ written as:

$$R_t = \sum_t^T \gamma^{t-T} r_t \quad (6)$$

The objective of solving the reinforcement learning problem is to determine the optimal policy for the agent to pursue, ultimately attaining maximum reward upon reaching the terminal state. The subsequent section elucidates the strategies employed to address reinforcement learning problems.

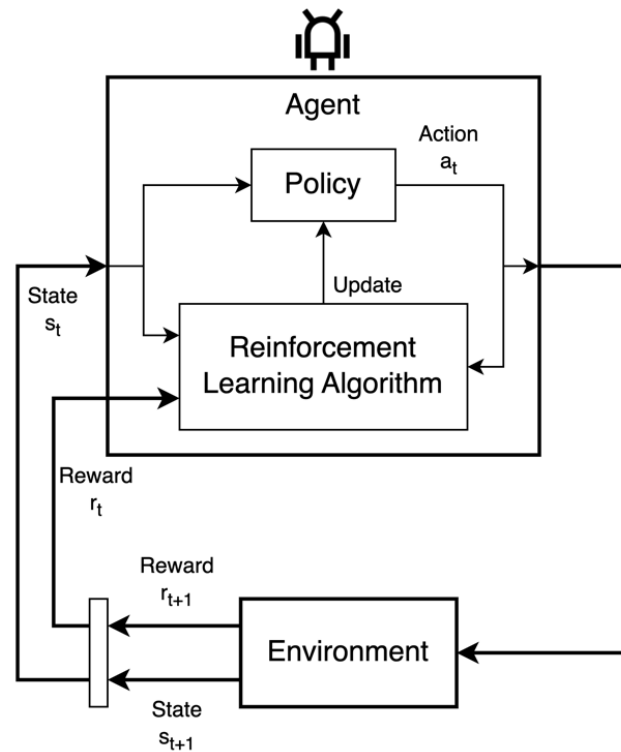


Figure 1 Schema of reinforcement learning.

3.1.1 Valued-based method

A valued-based method focuses on optimizing a value function, which enables the discovery of an optimal policy π^* that leads to the attainment of the maximum expected reward. Two types of value functions exist: the state-value function and the action-value function.

A state-value function, represented as $V_{\pi}(s)$, approximates the anticipated discounted reward an agent would obtain when commencing in a particular state and adhering to a specific policy. The state-value function can be expressed as:

$$V_{\pi}(s) = \mathbb{E}(R_t | s_t = s) \quad (7)$$

The action-value function, denoted as $Q(s, a)$, predicts the anticipated reward an agent would receive by choosing a specific action based on a policy within a given state. The action-value function can be expressed as:

$$Q_{\pi}(s, a) = \mathbb{E}(R_t | s_t = s, a_t = a) \quad (8)$$

The optimal action-value function, denoted as $Q^*(s, a)$, yields the highest action-value among all feasible actions for a given state. The expression for the optimal action-value function is:

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a) \quad (9)$$

Through the optimization of one of the value functions, an optimal policy is revealed, as depicted in the following relationship:

$$\pi^*(s) = \operatorname{argmax}_a Q^*(s, a) \quad (10)$$

where $\pi^*(s)$ represents an optimal policy that determines the optimal action for a given state.

3.1.2 Policy-based method

In contrast to the value-based method that indirectly determines an optimal policy through the training of a value function, a policy-based method directly optimizes the policy itself. Initially, the policy is parameterized using a set of parameters denoted as θ , which can take the form of a neural network. Consequently, the policy outputs a probability distribution over actions, which can be mathematically represented as:

$$\pi_{\theta} = \mathbb{P}(a|s; \theta) \quad (11)$$

Subsequently, the gradient ascent method is employed to update the parameter set θ to maximize an objective function denoted as $J(\theta)$ which represent the expected reward. Consequently, the objective function can be expressed as:

$$J(\theta) = \mathbb{E}(R_t) \quad (12)$$

REINFORCE serves as a prominent example of a policy gradient method, where the gradient utilized to update the parameter set θ is defined as:

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} \log \pi(a_t | s_t; \theta) R_t \quad (13)$$

where $\log \pi(a_t | s_t; \theta)$ represents the log probability of selecting action a_t given state s_t [21].

Nevertheless, the training of a policy-based method becomes challenging due to the substantial variance of the gradient, leading to training instability.

3.1.3 Actor-critic method

To address the issue of large gradients in the policy-based method, an actor-critic method introduces a baseline b_t by subtracting the expected reward term. An exemplary implementation of the actor-critic method is the advantage actor-critic (A2C), where the estimated state-value $V(s_t)$ serves as the baseline, subtracted from the expected reward R_t . The outcome of this subtraction is referred to as the advantage and can be mathematically represented as:

$$A(s_t, a_t) = R_t - b_t = Q(s_t, a_t) - V(s_t) \quad (14)$$

where $A(s_t, a_t)$ represents the advantage of action a_t at state s_t . It is important to mention that calculating the advantage requires an additional set of parameters to estimate the state-value function. Consequently, this approach is referred to as the actor-critic method, which leverages state-value estimation to enhance policy optimization.

As per the definition, the advantage quantifies the superiority or inferiority of an action compared to the average of all possible actions [22]. Consequently, the gradient used for updating the policy can be expressed as:

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} \log \pi(a_t | s_t; \theta) A(s_t, a_t) \quad (15)$$

However, the A2C method encounters instability during training due to the significant policy changes that occur after updates.

Proximal policy optimization (PPO) is specifically designed to address this limitation by constraining the gradient update during each training step using a clipped surrogate loss function, which is defined as:

$$J_{clip}(\theta) = \mathbb{E}[\min(q_t(\theta)A(s_t, r_t), \text{clip}(q_t(\theta), 1 - \epsilon, 1 + \epsilon)A(s_t, a_t))] \quad (16)$$

where $q_t(\theta)$ represents the probability ratio between the new policy and the old policy, precisely defined as:

$$q_t(\theta) = \left[\frac{\pi_{\theta}(a_t, s_t)}{\pi_{\theta_{old}}(a_t, s_t)} \right] \quad (17)$$

According to the surrogate loss function, the ratio $q_t(\theta)$ is clipped between $[1 - \epsilon, 1 + \epsilon]$. This new objective function prevents the drastic change of the policy during training. As a result, PPO is more stable than A2C [23].

By utilizing the surrogate loss function, the ratio $q_t(\theta)$ is bounded within the range $[1 - \epsilon, 1 + \epsilon]$. This modified objective function effectively mitigates the extreme policy changes during training. Consequently, PPO exhibits greater stability compared to A2C.

Given its capability to regulate gradient updates, PPO proves to be an appropriate reinforcement learning approach for handling the inherent noise present in

environments such as the cryptocurrency market. As a result, PPO has been chosen as the preferred method for our proposed approach.

3.2 Reinforcement learning for cryptocurrency trading

The finance industry was among the pioneers in adopting machine learning to enhance its operations, particularly trading. The advent of machine learning has expedited advancements in trading strategies. Among various machine learning techniques, reinforcement learning stands out as the most suitable approach for algorithmic trading. It excels at learning to interact with the environment and optimizing actions to achieve predefined objectives. In the context of algorithmic trading, reinforcement learning specifically learns when to sell, buy, or hold assets with the aim of maximizing profit.

Consequently, the application of reinforcement learning to financial trading has garnered significant interest among researchers. Taghian et al. [24] proposed the utilization of deep Q learning (DQN) to acquire asset-specific trading rules, commencing with a simpler task of trading one asset at a time. T. Théate and D. Ernst [25] introduced the trading deep Q-network strategy (TDQN). In a separate study, Tsai et al. [26] explored the use of candlestick patterns as learning features. While these methods have yielded intriguing results, it is worth noting that the suggested approaches do not effectively handle the simultaneous management of multiple assets.

Jiang et al. [27] proposed the ensemble of identical independent evaluators (EIIIE) for trading multiple assets, specifically focusing on portfolio management through a deterministic policy gradient algorithm. Weng et al. [28] [29] introduced deep reinforcement learning with a multidimensional attention-gating mechanism for

portfolio management. C. Betancourt and W.-H. Chen [29] adopted a self-attention network combined with A2C for portfolio selection. Additionally, various deep neural networks were employed. Sun et al. [30] utilized a deep residual shrinkage neural network to improve learning capability and address the issue of vanishing gradients. Yao et al. [31] implemented an inception module with a convolution block of different filter sizes and a bottleneck attention module to mitigate challenges associated with uniformly stacked networks, such as overfitting and computational overflow. However, these approaches did not incorporate risk management into their methodologies.

Reduction of unsystematic risk poses a formidable yet indispensable challenge within the realm of finance. An established approach to mitigate risk involves implementing a rule-based cut loss strategy, wherein trading agents adhere to predefined instructions upon the fulfillment of certain conditions. Notably, Gort et al. [32] relied on the cryptocurrency volatility index (CVIX) as a means of risk management, opting to sell all tokens and halt purchases once CVIX surpassed a predetermined threshold. In similar fashion, Yang et al. [14] opted to monitor a turbulence index instead of CVIX for their stock trading endeavors. Although these methodologies have demonstrated risk reduction capabilities, they are not universally optimal. This is due to the fact that financial markets exhibit dynamic behavior, influenced by a multitude of factors. Consequently, a predefined threshold may prove effective during certain time periods but fail to be efficacious during others.

A more sophisticated strategy involves training a reinforcement learning agent to optimize both risk reduction and return maximization. Sattarov et al. [33] implemented a conditional negative reward to discourage the agent from repeatedly

selecting a hold action. Another approach suggested by Betancourt and W.-H. Chen [34] is to incorporate risk considerations into the reward function by using a differential Sharpe ratio. Similarly, Bisht and Kumar [35] enhanced their reward function by including the standard deviation of returns (volatility) along with the average return.

Transaction costs have a notable impact on portfolio returns in algorithmic trading strategies. Excessive buying and selling can result in overtrading and high transaction costs. Zhang et al. [36] tackled this challenge by introducing cost-sensitive terms into their reward function. Their experiments showed that a trained agent can successfully manage transaction costs. In summary, our proposed method incorporates the objective of minimizing volatility and transaction costs into our reward function.

CHAPTER IV

CONCEPT AND RESEARCH METHODOLOGY

In this chapter, we present our proposed approach, which consists of three essential components: data preparation, multi-agent proximal policy optimization (MAPPO), and a simulated cryptocurrency market environment. Figure 2 offers a visual overview of the method. During the data preparation phase, historical k-line data is obtained from the Binance API, with a focus on open, high, low, and close prices. These prices are utilized to calculate technical indicators: i.e., MACD, RSI, CCI, and ADX. In the simulated cryptocurrency market environment, only the technical indicators and the close price are considered as data inputs. At each timestep, identical observations, comprising the processed data, are provided to all agents. These observations serve as inputs for the actor and critic networks of the agents. The actor network generates actions, while the critic network estimates expected rewards or state-values. The actions are then used to interact with the environment, triggering trade execution and transitioning to a new state. Based on the resulting state, rewards are computed for each agent and communicated to them. The agents learn from the received rewards using the PPO algorithm and generate new actions based on the most recent observation.

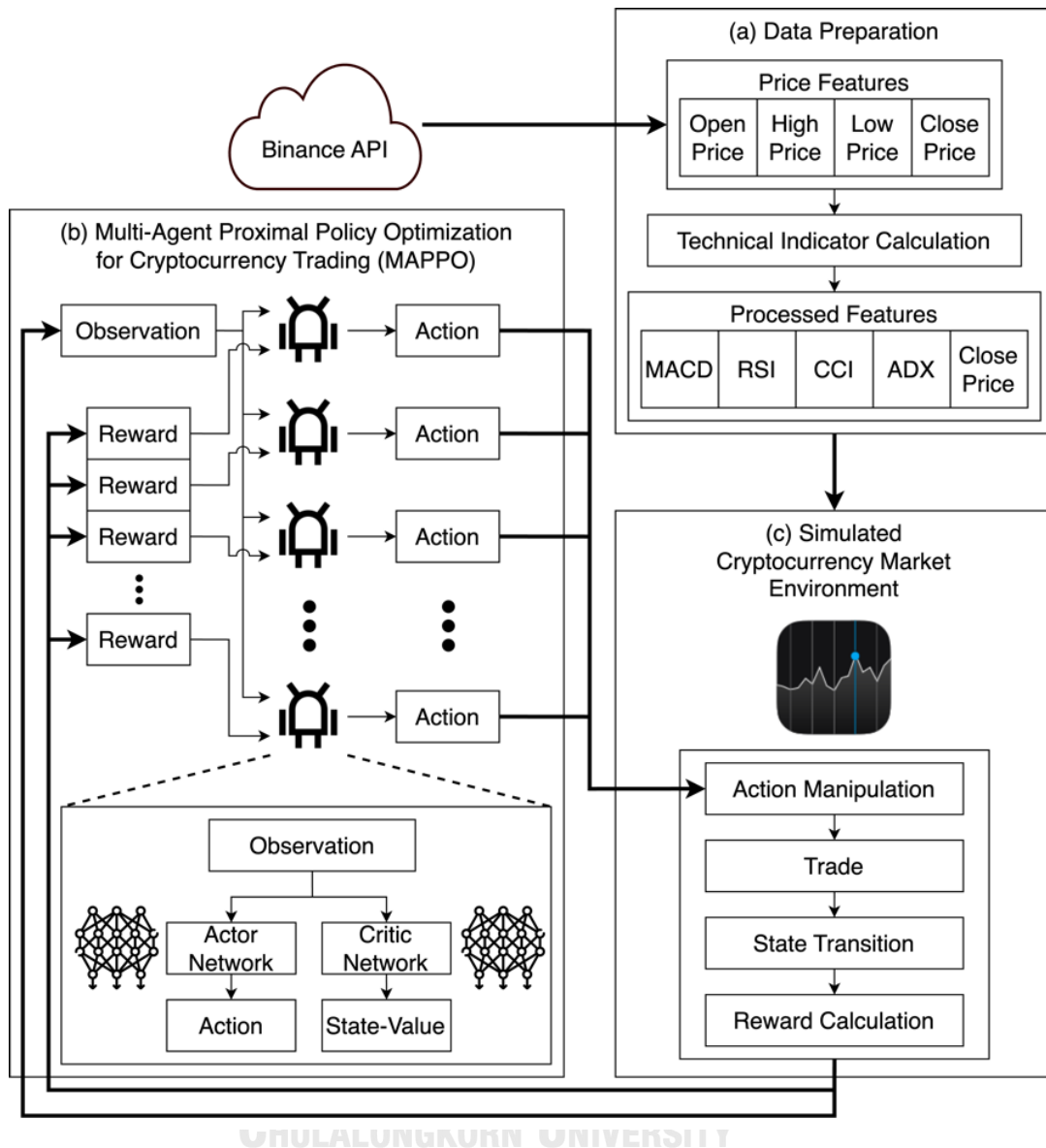


Figure 2 System overview: (a) data preparation, (b) multi-agent proximal policy optimization (MAPPO), and (c) simulated cryptocurrency market environment.

4.1 Data preparation

This study focuses on analyzing the historical k-line data at an hourly frequency for five popular token pairs: Cardano (ADAUSDT), Binance Coin (BNBUSDT), Bitcoin (BTCUSDT), Ethereum (ETHUSDT), and Ripple

(XRPUSDT). The dataset used in this research was obtained through the Binance API [37] and covers the period from April 1, 2021, to August 31, 2022.

The data retrieved from Binance API includes twelve fields for each token in the hourly historical k-line data. However, for the purpose of this study, only the price features, namely the open price, high price, low price, and close price, are utilized. Any unused features have been eliminated from the dataset.

In the subsequent stage, the dataset is enriched with informative features by calculating four financial technical indicators: moving average convergence divergence (MACD), relative strength index (RSI), commodity channel index (CCI), and average directional index (ADX). These indicators are widely utilized by traders in financial markets and have demonstrated their utility. Furthermore, as the technical indicators are derived from the price features, all price features except the close price are excluded from the dataset. Consequently, the dataset for each token consists of five features: close price, MACD, RSI, CCI, and ADX.

4.2 Multi-agent policy optimization for cryptocurrency trading (MAPPO)

This section outlines the fundamental components of our proposed method, namely: state, agent, action, reward, and the simulated cryptocurrency market environment.

4.2.1. State and observation

The state in our method represents the current status of the agents and the environment and is composed of two components. The first component is the portfolio data, which is represented by a "token value vector" (X_t). This vector contains the values of each token in the portfolio at time t , and in our study, it consists

of six elements representing the number of tokens, including USDT. The second component is the market data, which encompasses the close price, MACD, RSI, CCI, and ADX vectors (P_t , M_t , I_t , C_t , and D_t) for each token at time t . Each vector contains five elements, resulting in a total of twenty-five values stored in the market data at each timestep. Therefore, the state vector at each timestep encompasses thirty-one values.

Observation refers to the information accessible to agents at a given time within the simulated environment. Agents have the ability to retrieve both current and historical data within this environment. In this context, the observation is equivalent to the state. Each agent is able to observe not only the data related to its allocated token but also the data pertaining to all other tokens. This enables agents to identify correlations between the tokens they are trading, aiding them in determining the optimal actions to take.

Ensuring the normalization of observations is essential to maintain consistency in the range of values for each feature. Consequently, the transformation of a token value vector into a portfolio vector can be observed through the following equation:

$$W_t = \frac{X_t}{v_t} \quad (18)$$

where W_t represents the weight assigned to each token, contributing to the portfolio value at time t . Similarly, X_t denotes the token value vector at time t , while v_t represents the portfolio value at that specific time.

To normalize the market data, min-max scaling is employed, resulting in the transformation of the market data vector into P'_t , M'_t , I'_t , C'_t , and D'_t . Figure 3 provides a comprehensive overview of the observation normalization process.

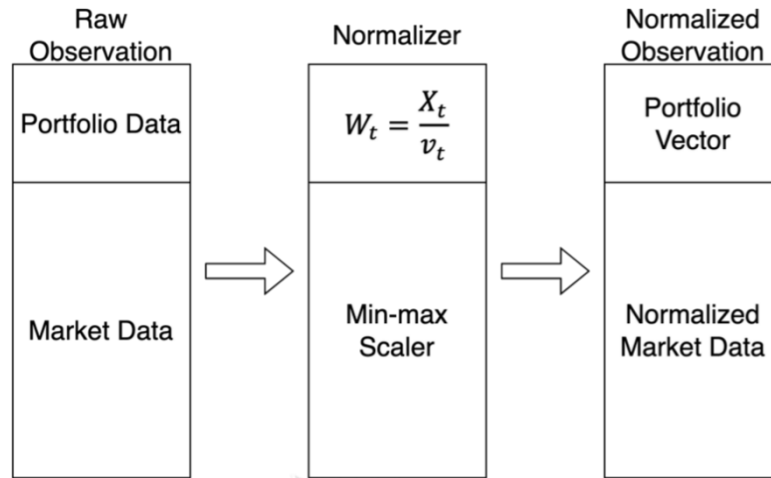


Figure 3 Schema of observation normalization.

Therefore, the normalized observation vector at timestep t , represented as S'_t , is provided to the agents as input. It can be expressed as:

$$S'_t = [W_t, P'_t, M'_t, I'_t, C'_t, D'_t] \quad (19)$$

4.2.2. Agent

In our multi-agent approach, each agent is assigned the task of trading a specific token, resulting in a collective effort of five agents during the trading process. Each deep reinforcement learning agent consists of two crucial components: the learning method and the policy network.

The learning algorithm chosen for our method is PPO (Proximal Policy Optimization) from the RLlib framework. PPO is specifically selected due to its capability to restrict gradients during each training step, resulting in improved training stability [38]. Given the inherent noisiness of cryptocurrency data, which often leads to significant changes in gradient values and gradient explosions during training, the ability of PPO to control these explosions is highly advantageous for cryptocurrency trading.

Each agent in our system consists of two neural networks: an actor network and a critic network. The actor network is responsible for generating actions, while the critic network predicts the expected reward or state-value $V(s_t)$. The reward obtained is then used to estimate the advantage $A(s_t, a_t)$, which plays a crucial role in the gradient update process as discussed previously.

Figure 4 showcases the network architecture utilized in our approach, which employs a multilayer perceptron (MLP). Additionally, we explore the effectiveness of different feature extractors, namely long-short time memory (LSTM) [39], gated recurrent unit (GRU) [40], ResNet [41], and Res2Net [42], to enhance our method. These feature extractors are fed with 6-hour observations, spanning from $t - 5$ to t , serving as historical data. Our hypothesis is that by leveraging a feature extractor, agents can extract valuable insights from historical data, ultimately leading to improved trading performance. Figure 5 and Figure 6 provide a visual representation of the architectures employed in this study.

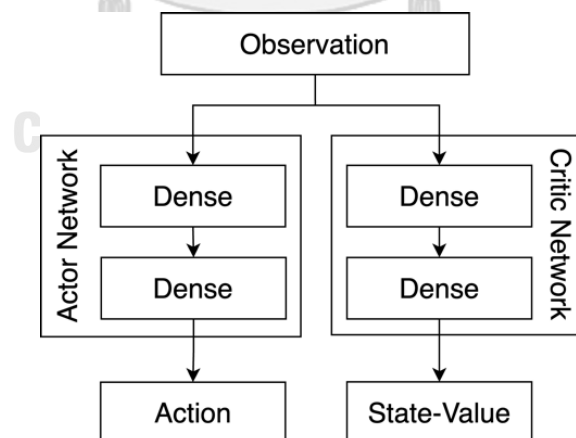


Figure 4 The network architecture of the agent consists of two components: the actor network on the left-hand side and the critic network on the right-hand side. Each network comprises two layers of Multilayer Perceptron (MLP), commonly referred to as dense layers. Notably, the normalized market data and the portfolio vector are simultaneously provided as observations to both networks.

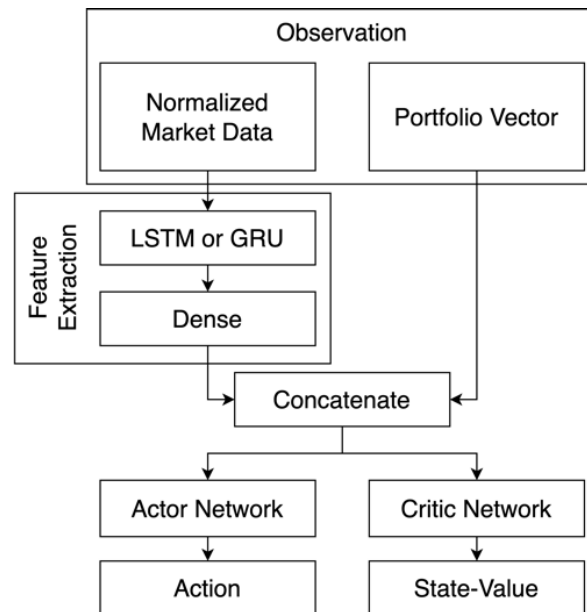


Figure 5 The normalized market data undergoes a feature extraction process after passing through either an LSTM or a GRU layer, followed by a dense layer. Subsequently, it is concatenated with the portfolio vector and forwarded to both the actor and critic networks.

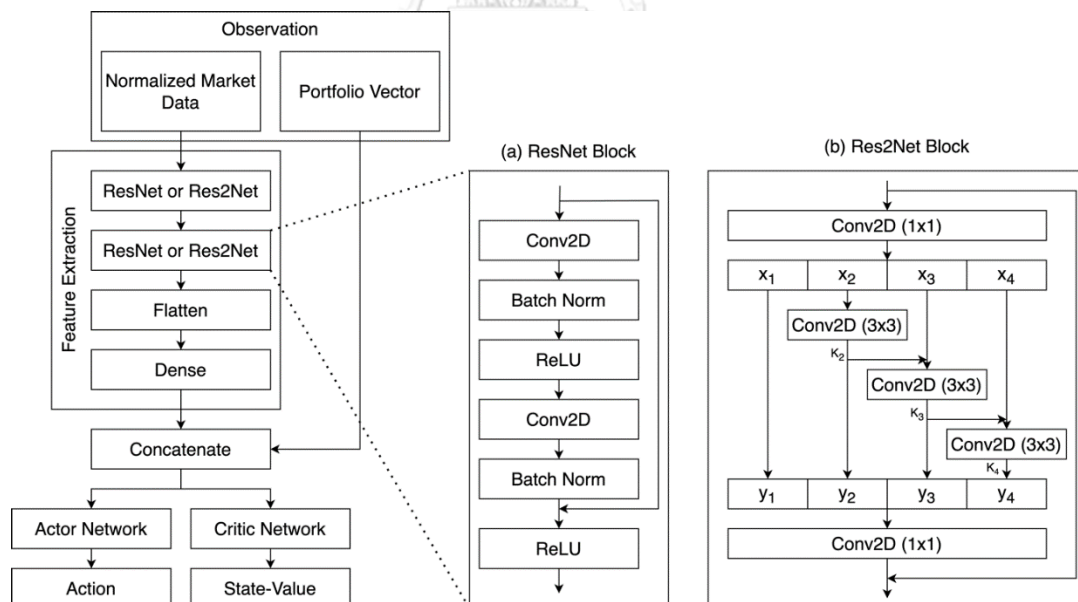


Figure 6 The normalized market data undergoes processing through two blocks of either ResNet or Res2Net, followed by a flatten layer and a dense layer. Subsequently, it is concatenated with the portfolio vector and directed towards both the actor and critic networks. The specific details of the ResNet block and the Res2Net block are depicted in (a) and (b) respectively.

4.2.3. Action

Upon receiving the observation, each agent generates an action that specifies the desired amount of tokens to be bought or sold. Consequently, the total number of actions at each timestep corresponds to the number of tokens being traded.

In this research, the action space for each agent is defined within the range of -1 to 1. The sign of each action represents the type of action, where a zero value corresponds to holding, a negative value indicates selling and a positive value signifies buying. The magnitude of a non-zero action determines the transaction size, expressed as a percentage of the allowable transaction size, which is fixed at 100,000 USDT.

$$A'_t(S_t) = A_t \cdot 10^5 \quad (20)$$

where A_t represents the vector of raw actions at timestep t , and A'_t represents the scaled action vector at timestep t . Consequently, the maximum magnitude of each buying or selling transaction is constrained to 100,000 units.

4.2.4. Reward

The reward function plays a crucial role in the success of reinforcement learning. In this study, we introduce a local-global reward function specifically designed to optimize the collaborative multi-agent deep reinforcement learning technique. Figure 7 provides a visual representation of the different components integrated into our reward function.

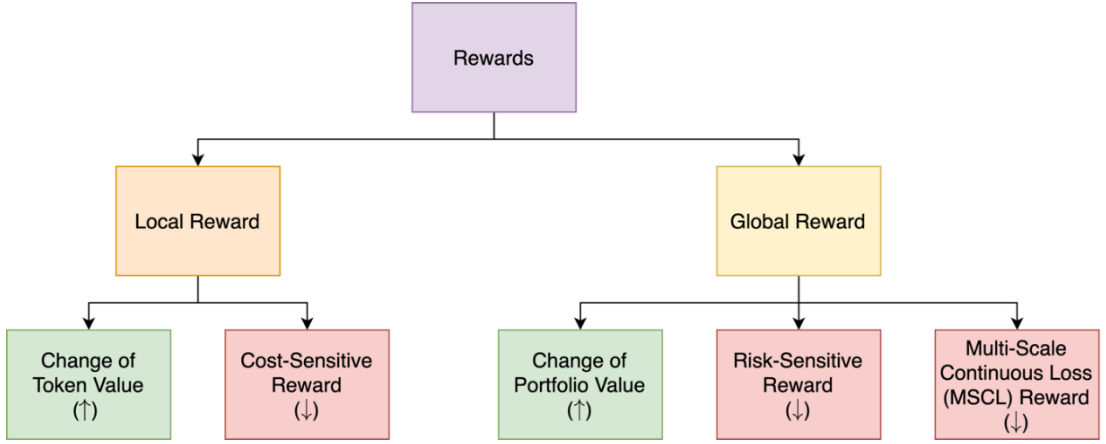


Figure 7 Schema presenting the various terms employed in our reward function, with particular emphasis on our proposed reward term, MSCL reward. The green boxes represent reward terms that agents should strive to maximize, while the red boxes indicate penalty terms that agents should aim to minimize.

The local reward assesses the performance of each individual agent by evaluating their trading actions on their assigned token, resulting in a unique reward value for each agent at every timestep in the environment. On the other hand, the global reward provides the same reward value to all agents, measuring their overall performance. By incorporating both local and global rewards, we aim to optimize the individual and collective performance of the agents simultaneously. The subsequent section provides a comprehensive explanation of these two reward mechanisms.

To begin with, the local reward comprises two sub-components, which serve to measure the fluctuations in token value and account for the sensitivity to transaction costs:

$$R_{local}(S_t, A_t, S_{t+1}) = R_{token}(S_t, A_t, S_{t+1}) + R_{cost}(S_t, A_t, S_{t+1}) \quad (21)$$

where $R_{local}(S_t, A_t, S_{t+1})$ represents the local reward vector obtained from the action vector A_t at state S_t , resulting in the transition to a new state S_{t+1} . Additionally, $R_{token}(S_t, A_t, S_{t+1})$ corresponds to the vector capturing the changes in the token value

vector caused by the action vector A_t at state S_t , leading to the new state S_{t+1} . Lastly, $R_{cost}(S_t, A_t, S_{t+1})$ signifies the cost-sensitive reward vector resulting from the action vector A_t at state S_t , leading to the new state S_{t+1} .

The change in token value plays a crucial role in enabling the agent to understand the patterns and dynamics of the token it is responsible for. By analyzing these changes, the agent gains insights into when it should execute buying, selling, or holding actions for the token. Mathematically, this term can be represented as follows:

$$R_{token}(S_t, A_t, S_{t+1}) = \frac{X_{t+1}}{X_t} - 1 \quad (22)$$

where X_t represents the token value vector after the trading activity at time t , and X_{t+1} represents the token value vector at time $t + 1$, following the transition of the state.

Regarding the cost-sensitive reward, agents need to take into account the impact of transaction costs on each transaction. This consideration prevents the agent from engaging in excessive trading and incurring excessive losses due to transaction costs. The formulation of this reward is as follows:

$$R_{cost}(S_t, A_t, S_{t+1}) = -\phi K_t \quad (23)$$

where, $\phi \geq 0$ represents the cost-sensitive reward coefficient, which regulates the impact of this reward term. Additionally, K_t denotes the vector encompassing transaction costs:

$$K_t = k|H_t \cdot P_t| \quad (24)$$

where k represents the transaction cost, which is assumed to be 0.1% for the purposes of this study. Additionally, H_t is a vector that contains the quantity of each token involved in the trading process.

The global reward encompasses three key elements: the change in portfolio value, the risk-sensitive reward, and the multi-scale continuous loss (MSCL) reward.

Therefore, the global reward can be formulated as follows:

$$r_{global}(S_t, A_t, S_{t+1}) = r_{port}(S_t, A_t, S_{t+1}) + r_{risk}(S_t, A_t, S_{t+1}) + r_{MSCL}(S_t, A_t, S_{t+1}) \quad (25)$$

where $r_{global}(S_t, A_t, S_{t+1})$ represents the value of the global reward resulting from the action vector A_t at state S_t , leading to a new state S_{t+1} , $r_{port}(S_t, A_t, S_{t+1})$ signifies the change in portfolio value caused by the action vector A_t at state S_t , resulting in a new state S_{t+1} , $r_{risk}(S_t, A_t, S_{t+1})$ represents the value of the risk-sensitive reward resulting from the action vector A_t at state S_t , leading to a new state S_{t+1} , and $r_{MSCL}(S_t, A_t, S_{t+1})$ denotes the value of the MSCL reward resulting from the action vector A_t at state S_t , leading to a new state S_{t+1} .

The initial term, $r_{port}(S_t, A_t, S_{t+1})$, represents the change in portfolio value resulting from holding, selling, or buying tokens as the environment transitions from time 1 to $t + 1$, given by the following equation:

$$r_{port}(S_t, A_t, S_{t+1}) = \frac{v_{t+1}}{v_t} - 1 \quad (26)$$

The reward $r_{port}(S_t, A_t, S_{t+1})$ holds utmost significance as it is the primary objective for agents to optimize because the ultimate aim of cryptocurrency trading is to maximize the portfolio's value.

The subsequent term in the global reward is the risk-sensitive reward, which represents the variance of all previous $r_{port}(S_t, A_t, S_{t+1})$ values since the start of the episode. It can be expressed as:

$$r_{risk}(S_t, A_t, S_{t+1}) = -\rho \frac{1}{T} \sigma^2 \quad (27)$$

$$\sigma^2 = \text{var} \left(r_{\text{portfolio}}(S_t, A_t, S_{t+1}) \right) \quad (28)$$

where, $\rho \geq 0$ represents the risk-sensitive reward coefficient, which governs the impact of this reward term. T denotes the total number of timesteps since the episode's inception, while σ^2 signifies the portfolio return's variance. Theoretically, this reward term allows the agents to reduce the volatility of the portfolio.

Moreover, in the event of a continuous decline in portfolio value, agents are subjected to the MSCL reward as a form of punishment. Drawing inspiration from the principle of progressive discipline, where penalties escalate with repeated mistakes by an employee, agents receive a more substantial negative reward if the following conditions are satisfied:

1. In a specified period τ , if the portfolio value at time $t - \tau$ ($v_{t-\tau}$) is higher than the portfolio value at time t (v_t), where t represents the current timestep, it indicates a persistent decline in the overall portfolio value throughout the given period.
2. During the time period from $t - \tau$ to t , if the number of positive r_{port} values (indicating portfolio gains) denoted by n_{win} is smaller than the number of negative r_{port} values (indicating portfolio losses) denoted by n_{loss} , it signifies that the agents' portfolio is experiencing more frequent decreases in value compared to increases.
3. The portfolio value at time t , represented as $r_{\text{port},t}$, exhibits a negative value.

The aforementioned conditions undergo multiple iterations with varying values of τ . Initially, τ is set at twelve timesteps for the first test. In the event that the three conditions are not fulfilled at $\tau = 12$, subsequent tests are conducted with

smaller τ values, such as eleven, and so forth. This iterative process persists until either the conditions are satisfied at a specific τ value or τ reaches its lower threshold. Should the conditions be met prior to τ reaching the lower bound, the agents incur a negative reward, as delineated below:

$$r_{MSCL}(S_t, A_t, S_{t+1}) = \mu \cdot |n_{loss} - n_{win}| \cdot r_{port,t} \quad (29)$$

where $\mu \geq 0$ represents the MSCL reward coefficient, which governs the impact of the MSCL reward. Conversely, if τ reaches the lower bound, the MSCL reward becomes zero. It is worth noting that $|n_{loss} - n_{win}|$ is always positive and $r_{port,t}$ is always negative. Consequently, $r_{MSCL,t}$ is always negative. Through a comprehensive analysis of the aforementioned conditions at different time intervals, the agent can enhance their ability to effectively minimize the continuous decline in portfolio value.

To enhance comprehension, the algorithm for the MSCL reward is elucidated through a pseudo-code representation in Figure 8.

Algorithm 1: Multi-Scale Continuous Loss (MSCL) Reward

Input: historical portfolio value and $r_{portfolio}$
Output: $r_{MSCL,t}$

- 1 **for** $\tau = \tau_{max}, \tau_{max} - 1, \tau_{max} - 2, \dots, \tau_{min}$ **do**:
- 2 **if** $(v_{t-\tau} > v_t)$ and $(n_{loss} > n_{win})$ and $(r_{portfolio,t} < 0)$:
- 3 **return** $r_{MSCL}(S_t, A_t, S_{t+1}) = \mu \cdot |n_{loss} - n_{win}| \cdot r_{portfolio,t}$
- 4 **return** $r_{MSCL,t} = 0$

Figure 8 Algorithm of multi-scale continuous loss (MSCL) reward

Ultimately, the overall reward given to the agent can be represented as follows:

$$R(S_t, A_t, S_{t+1}) = R_{local}(S_t, A_t, S_{t+1}) + r_{global}(S_t, A_t, S_{t+1}) \quad (30)$$

where $R(S_t, A_t, S_{t+1})$ signifies the reward vector containing distinct reward value for each agent, obtained through their interactions with the environment at state S_t , leading to a subsequent state S_{t+1} . Consequently, our reinforcement learning agents are optimized by aligning their goals with this reward function.

4.3 Simulated cryptocurrency market environment

We have created a simulated environment of a cryptocurrency market to facilitate the training and testing of our method. This section provides an explanation of the dynamics of the environment, including action manipulation, trade, state transition, and reward calculation.

4.3.1. Action manipulation

The initial phase in the environment involves action manipulation. Upon receiving scaled actions A'_t from the agents, denoting the amount of tokens to be bought or sold in USDT, each scaled action is transformed into its corresponding number of tokens based on the prevailing price. This conversion is executed as follows:

$$H_t = \frac{A'_t}{P_t} \quad (31)$$

where, H_t represents the converted action vector at time step t , indicating the number of tokens in which the actions are expressed.

Prior to proceeding with the trading step, the value of each token in the portfolio at time t , expressed in USDT, is transformed into its corresponding number of tokens. This conversion process is illustrated in Equation 32.

$$Q_t = \frac{X_t}{P_t} \quad (32)$$

where Q_t represents the vector denoting the number of tokens at time t .

4.3.2. Trade

With the completion of action manipulation, the trading process is initiated. During this step, tokens are exchanged based on the actions of the agents. The resulting number of each token after trading is represented as:

$$Q'_t = Q_t + H_t \quad (33)$$

where Q'_t represents the vector that signifies the number of tokens after the trading process.

As a result, the cash value denoted as USDT after the trading process, represented by b_{t+1} , is calculated as follows:

$$b_{t+1} = b_t - (H_t \cdot P_t) - K_t \quad (34)$$

During the trading process, it is important to highlight that in the practical implementation, selling actions are executed prior to buying actions. This sequential order ensures that the portfolio receives cash from selling tokens, which then is spent for purchasing other tokens.

4.3.3. State transition

After the completion of trading, the market data undergoes a transition from time t to $t + 1$. Consequently, the state vector at time $t + 1$ can be represented as $[X_{t+1}, P_{t+1}, M_{t+1}, I_{t+1}, C_{t+1}, D_{t+1}]$. This transition brings about a new token value vector, denoted by X_{t+1} , which encapsulates the values of the tokens.

$$X_{t+1} = Q'_t \cdot P_{t+1} \quad (35)$$

and the updated portfolio value, denoted as V_{t+1} , signifies the total value of tokens at time $t + 1$:

$$v_{t+1} = \sum_i^N x_{i,t+1} \quad (36)$$

where, N represents the total number of six tokens, which includes the cash: USDT.

4.3.4. Reward calculation

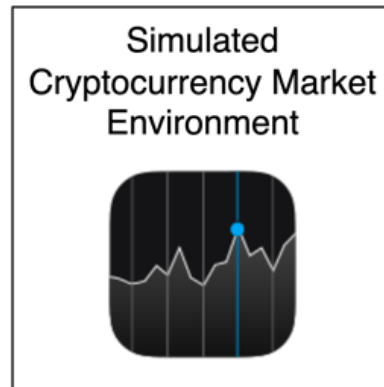
Following the completion of trade and state transition, the total reward $R(S_t, A_t, S_{t+1})$ is computed. The process involves the normalization of the state vector, resulting in the following outcome:

$$S'_{t+1} = [X_{t+1}, P'_{t+1}, M'_{t+1}, I'_{t+1}, C'_{t+1}, D'_{t+1}] \quad (37)$$

Figure 9 illustrates that the reward $R(S_t, A_t, S_{t+1})$ and the normalized state vector are sent to the agents.

Scaled Actions
(USDT)

ADA: -50,000
BNB: 80,000
BTC: -20,000
ETH: 7,000
XRP: -35,000



Normalized Observation (t+1)
Rewards

Figure 9 In the simulated cryptocurrency market environment, agents submit actions, and in return, they receive a normalized observation at time $t + 1$ along with corresponding rewards.

CHAPTER V

EXPERIMENTS AND RESULTS

This chapter provides a comprehensive overview of the data management approach employed in this study. Subsequently, the detailed implementation, encompassing the settings of hyper-parameters, is presented. Lastly, the evaluation metrics utilized for assessing the performance of the various methods are elaborated upon.

5.1 Dataset

In the preceding chapter, the processed data was partitioned into a training set, validation set, and test set. Moreover, we have purposefully chosen four supplementary test sets from the overall test set to thoroughly evaluate the performance of our methods under various market conditions, encompassing bullish, bearish, and sideways scenarios. This careful selection empowers us to comprehensively assess the effectiveness of our methods across these diverse market dynamics. Detailed information regarding each dataset is provided in Table 1 and Figure 10.

Table 1 Cryptocurrency k-line datasets

Dataset	Date
Training set	2021/04/01 - 2021/12/31
Validation set	2022/02/01 - 2022/02/28
Overall test set	2022/03/01 - 2022/08/31
Bullish test set	2022/07/01 - 2022/07/31
Bearish test set	2022/04/01 - 2022/04/30
Up-down test set	2022/03/19 - 2022/04/18
Side-way test set	2022/05/12 - 2022/06/11

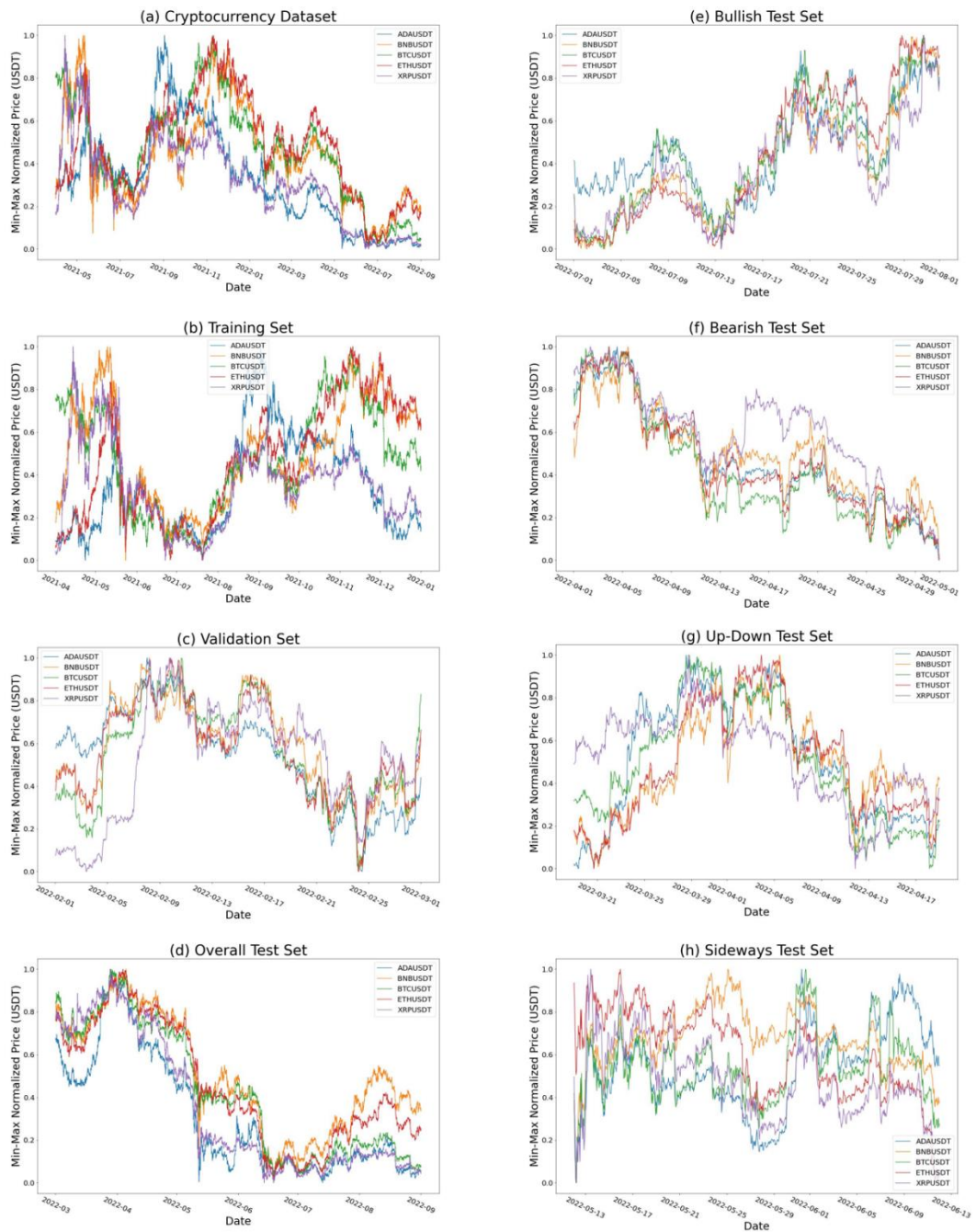


Figure 10 Normalized close price of each dataset: (a) Cryptocurrency dataset encompasses all the historical data utilized in this paper and is subsequently divided into (b) Training set, (c) Validation set, and (d) Overall test set. The overall test set is further partitioned into (e) Bullish test set, (f) Bearish test set, (g) Up-down test set, and (h) Sideways test set.

5.2 Implementation detail

Throughout all subsequent experiments, the experiments were conducted under the following standardized set of assumptions and conditions:

- The initial capital is 1,000,000 USDT.
- The initial portfolio only consists of USDT as cash.
- Each token has a maximum allowable transaction size of 100,000 USDT for both buying and selling.
- Transaction costs amount to 0.1% of the transaction size.
- Market liquidity is assumed to be abundant, ensuring that all transactions are promptly executed at the prevailing closing price. Furthermore, it is assumed that the trading activities of the agent do not have any impact on the overall cryptocurrency market.

All the methods underwent training on an identical training set, spanning 1,000 episodes. The agent exhibiting the highest cumulative return on the validation set was chosen for testing.

5.3 Results

In this section, we present the empirical results obtained through an extensive comparative analysis of various approaches across five distinct test sets, following the methodology employed in prior studies such as references [24], [26], and [33]. These references serve as examples where multiple methods were thoroughly evaluated using comprehensive test datasets. Firstly, Table 2 provides a comparison between multi-agent PPO (MAPPO) with local-global reward and single-agent PPO (SAPPO).

Subsequently, we examine the effects of introducing risk-sensitive, cost-sensitive, and MSCL rewards to MAPPO. Moreover, we demonstrate the superiority of our proposed method by comparing it to baseline methods. Finally, to further enhance the performance of our method, we investigate the outcomes of employing different network architectures.

5.3.1 Comparing multi-agent to single-agent PPO

Within this section, we conduct a comparative analysis of the performance between MAPPO with a local-global reward and SAPPO. It is important to emphasize that this experiment focuses exclusively on the changes observed in token values and portfolio values (two green boxes illustrated in Figure 7), specifically excluding the consideration of risk-sensitive, cost-sensitive, and MSCL rewards.

Table 2 Comparison between MAPPO and SAPPO. Boldfaces refer to the winners. Gray numbers refer to uninterpretable results

Method	Cumulative Return (percent) ↑	Sharpe Ratio ↑	Calmar Ratio ↑	Volatility (percent) ↓	MDD (percent) ↓
Overall					
SAPPO	-57.79	-1.59	-0.85	85.02	-67.67
MAPPO	-48.32	-1.31	-0.71	77.49	-68.33
Bullish					
SAPPO	19.32	3.24	1.31	72.35	-14.74
MAPPO	22.79	3.71	1.54	72.18	-14.84
Bearish					
SAPPO	-28.69	-6.46	-0.92	60.87	-31.09
MAPPO	-19.57	-4.89	-0.78	51.58	-25.08
Up-Down					
SAPPO	-4.11	-0.52	-0.17	60.14	-24.63
MAPPO	-7.87	-1.66	-0.38	50.61	-20.98
Sideways					
SAPPO	-5.15	-0.32	-0.24	84.04	-21.09
MAPPO	-2.45	-0.05	-0.19	81.99	-13.12

Table 2 presents the comparative performance of MAPPO and SAPPO across multiple test sets. MAPPO demonstrated superior cumulative return results in all test

sets, except for the Up-Down test set, where SAPPO only incurred a 4.11% loss of its initial portfolio value, compared to MAPPO's 7.87% loss. In the Bullish test set, the performance of MAPPO was further evaluated using the Sharpe ratio and Calmar ratio, both of which yielded higher values compared to SAPPO (3.71 and 1.54, respectively). MAPPO also exhibited lower volatility values across all test sets, indicating its better risk profile. Additionally, MAPPO outperformed SAPPO in terms of maximum drawdown (MDD) in three out of the five test sets, except for the Overall and Bullish test sets where SAPPO had marginally larger MDD values. In conclusion, based on the experimental results, MAPPO demonstrated superior performance over SAPPO.

5.3.2 Comparing combinations of reward terms

Within this section, we expand the scope of our investigation by incorporating three additional reward terms, represented by red boxes in Figure 7, alongside the ones utilized in the previous section. These new reward terms include: (1) risk-sensitive reward, (2) cost-sensitive reward, and (3) multi-scale continuous loss (MSCL) reward, with MSCL being a novel reward term proposed by our research. The primary objective of this section is to examine how different combinations of these reward terms affect the performance of MAPPO.

According to the findings presented in Table 3, a comparison between regular MAPPO and MAPPO with a risk-sensitive reward in its reward function demonstrates that the integration of a risk-sensitive reward term contributes to a decrease in portfolio volatility across all test sets.

Table 3 Comparing the combinations of reward terms assigned to MAPPO including risk-sensitive reward, cost-sensitive reward, and MSCL reward. Undoubtedly, the inclusion of the MSCL reward, our suggested reward term, has the capability to enhance the performance of all combinations. Boldfaces refer to the winners. Gray numbers refer to uninterpretable results

Method	Cumulative Return (percent) ↑	Sharpe Ratio ↑	Calmar Ratio ↑	Volatility (percent) ↓	MDD (percent) ↓	Total Transaction Cost (USDT) ↓
Overall						
MAPPO	-48.32	-1.30	-0.71	77.49	-68.33	2638
MAPPO + MSCL	-44.37	-1.14	-0.65	76.30	-67.95	1918
MAPPO + Risk	-47.93	-1.43	-0.74	72.21	-64.80	1004
MAPPO + Risk + MSCL	-42.45	-0.75	-0.57	90.99	-74.49	3173
MAPPO + Cost	-51.93	-1.31	-0.77	83.89	-67.56	999
MAPPO + Cost + MSCL	-11.99	-0.07	-0.27	78.66	-43.81	8132
MAPPO + Risk + Cost	-43.91	-0.85	-0.66	88.63	-66.50	1017
MAPPO + Risk + Cost + MSCL	-17.47	0.12	-0.35	75.69	-49.81	8196
Bullish						
MAPPO	22.79	3.71	1.54	72.18	-14.84	1422
MAPPO + MSCL	35.60	5.23	2.53	73.85	-14.06	1011
MAPPO + Risk	28.32	4.69	2.05	67.57	-13.85	999
MAPPO + Risk + MSCL	36.97	5.21	2.45	76.81	-15.07	999
MAPPO + Cost	29.03	4.48	1.99	72.99	-14.59	999
MAPPO + Cost + MSCL	32.59	4.98	2.33	72.05	-13.96	1000
MAPPO + Risk + Cost	26.97	4.12	1.75	75.09	-15.44	1029
MAPPO + Risk + Cost + MSCL	28.26	4.64	2.35	68.25	-12.02	1242
Bearish						
MAPPO	-19.57	-4.89	-0.78	51.58	-25.08	5251
MAPPO + MSCL	-6.28	-1.86	-0.50	38.55	-12.66	1492
MAPPO + Risk	-20.86	-5.95	-0.94	46.07	-22.23	1000
MAPPO + Risk + MSCL	1.30	2.74	1.44	5.81	-0.91	2329
MAPPO + Cost	-24.24	-5.84	-0.84	55.25	-28.74	999
MAPPO + Cost + MSCL	2.05	3.94	2.91	6.31	-0.70	1888
MAPPO + Risk + Cost	-14.62	-3.91	-0.76	46.45	-19.24	1096
MAPPO + Risk + Cost + MSCL	2.36	2.83	1.14	10.26	-2.07	2625
Up-Down						
MAPPO	-7.87	-1.66	-0.38	50.61	-20.98	5386
MAPPO + MSCL	3.52	1.88	0.49	23.14	-7.15	1534
MAPPO + Risk	-0.40	-0.13	-0.02	46.13	-17.35	1001
MAPPO + Risk + MSCL	-8.54	-2.02	-0.46	46.64	-18.40	1189
MAPPO + Cost	1.98	0.69	0.09	57.68	-21.21	999
MAPPO + Cost + MSCL	1.94	3.69	2.76	6.21	-0.70	1776
MAPPO + Risk + Cost	1.57	0.77	0.17	29.57	-9.26	1099
MAPPO + Risk + Cost + MSCL	2.35	2.76	1.13	10.09	-2.07	2578
Sideways						
MAPPO	-2.45	-0.05	-0.19	81.99	-13.12	2439
MAPPO + MSCL	-1.14	-0.18	-0.08	78.08	-18.30	1266
MAPPO + Risk	-9.65	-1.64	-0.62	61.41	-15.61	1000
MAPPO + Risk + MSCL	0.70	0.50	0.04	78.89	-17.88	1567
MAPPO + Cost	0.21	0.49	0.01	92.21	-18.17	999
MAPPO + Cost + MSCL	8.08	1.44	0.46	95.02	-17.55	999
MAPPO + Risk + Cost	7.35	1.32	0.37	104.42	-20.02	999
MAPPO + Risk + Cost + MSCL	1.21	0.62	0.07	93.29	-17.59	999

Throughout the testing phase, a comparative analysis is conducted between MAPPO and MAPPO with a cost-sensitive reward to examine the influence of this reward term on the overall transaction cost. As predicted, the inclusion of the cost-sensitive reward leads to a reduction in the total transaction cost. In certain combinations, the total transaction cost reaches approximately 1,000 USDT, which precisely corresponds to 0.1% of the initial capital of 1,000,000 USDT. This specific transaction amount is observed because the agents solely perform buy actions utilizing their entire cash holdings at the beginning of each episode, without engaging in any token selling transactions.

By comparing MAPPO to MAPPO with the MSCL reward, the influence of the MSCL reward on performance is examined. The findings demonstrate that the utilization of the MSCL reward leads to greater cumulative returns for the portfolio across all test sets and reduces the severity of maximum drawdown, except in the case of the Sideways test set, where the market exhibits highly volatile conditions.

5.3.3 Comparing to baseline methods

In this section, we compare our method to four baseline approaches:

- Uniform buy and hold (UBAH) strategy involves purchasing each token with the same quantity of USDT at the initial timestep and holding them without any subsequent selling.
- Uniform constant rebalanced portfolio (UCRP) strategy entails purchasing each token with an equal amount of USDT at the initial timestep and engaging in trading activities to ensure that the tokens maintain equal value proportions.
- Buy and hold Bitcoin (Bitcoin) strategy involves utilizing all the available USDT to purchase Bitcoin and refraining from any selling actions thereafter.

- Ensemble strategy (FinRL-Ensemble) proposed by Yang et al. [14].

In this experiment, MAPPO incorporating the MSCL reward, risk-sensitive reward, and cost-sensitive reward was selected for comparison with the baseline methods. This choice was based on its outstanding performance in the validation set.

Table 4 Comparing our MAPPO with risk-sensitive reward, cost-sensitive reward, and MSCL reward to four baseline methods including UBAH, UCRP, Bitcoin, and FinRL-Ensemble. Boldfaces refer to the winners. Gray numbers refer to uninterpretable results.

Method	Cumulative Return (percent) ↑	Sharpe Ratio ↑	Calmar Ratio ↑	Volatility (percent) ↓	MDD (percent) ↓
Overall					
UBAH	-48.35	-1.31	-0.74	77.41	-65.27
UCRP	-47.98	-1.27	-0.74	78.16	-65.12
Bitcoin	-53.67	-1.87	-0.85	68.92	-62.88
FinRL-Ensemble	-52.75	-1.38	-0.85	82.86	-61.82
MAPPO + MSCL + Risk + Cost (our)	-17.47	-0.12	-0.35	75.69	-49.81
Bullish					
UBAH	24.56	3.95	1.71	72.22	-14.36
UCRP	24.06	3.9	1.68	71.69	-14.34
Bitcoin	14.38	2.65	1.01	68.59	-14.24
FinRL-Ensemble	19.35	3.08	1.31	77.59	-14.79
MAPPO + MSCL + Risk + Cost (our)	28.26	4.64	2.35	68.25	-12.02
Bearish					
UBAH	-20.88	-5.3	-0.83	51.11	-25.29
UCRP	-21.27	-5.39	-0.83	51.58	-25.64
Bitcoin	-16.21	-4.44	-0.81	46.13	-19.9
FinRL-Ensemble	-25.42	-6.14	-0.83	55.67	-30.79
MAPPO + MSCL + Risk + Cost (our)	2.36	2.83	1.14	10.26	-2.07
Up-Down					
UBAH	2.08	0.73	0.11	52.11	-18.79
UCRP	2.19	0.75	0.12	51.52	-18.44
Bitcoin	-2.11	-0.33	-0.11	45.29	-19.08
FinRL-Ensemble	3.23	1.12	0.22	57.89	-19.52
MAPPO + MSCL + Risk + Cost (our)	2.35	2.76	1.13	10.09	-2.07
Sideways					
UBAH	-6.58	-0.45	-0.41	89.39	-16.12
UCRP	-6.77	-0.48	-0.42	89.14	-16.16
Bitcoin	-2.56	-0.11	-0.22	67.73	-11.67
FinRL-Ensemble	-0.3	-0.14	-0.04	90.52	-21.37
MAPPO + MSCL + Risk + Cost (our)	1.21	0.62	0.07	93.29	-17.59

Table 4 presents the results, showing that our proposed method outperformed all other methods in terms of cumulative return, except for the Up-Down test set. Notably, in the bullish test set, our method achieved a remarkable 46.05% higher cumulative return compared to FinRL-Ensemble. Additionally, our method demonstrated superior performance in terms of the Sharpe and Calmar ratios, indicating its ability to effectively balance both profit and risk in the portfolio. These findings highlight the superiority of our method over the baseline methods.

It is important to acknowledge that FinRL-Ensemble utilizes a sliding window procedure for training, evaluating, and testing purposes. This approach entails a dynamic change in the training and evaluation data with each adjustment of the window. Consequently, the outcomes of FinRL-Ensemble may not be directly comparable to other results, as it operates on distinct sets of data for training and evaluation.

5.3.4 Comparing network architectures

In this section, we investigate the performance of different network architectures (LSTM, GRU, ResNet, and Res2Net) as feature extractors. The goal is to identify the most appropriate architecture that can effectively extract valuable information from observations, thereby enhancing profitability and risk management. Furthermore, each network architecture was trained using a reward function that incorporates all the proposed terms: risk-sensitive, cost-sensitive, and MSCL reward. Alongside monitoring the changes in token value and portfolio value, we evaluate the performance of each architecture based on these combined reward terms.

The results presented in Table 5 demonstrate interesting findings. Among the tested network architectures, only MLP achieved positive cumulative return, Sharpe ratio, and Calmar ratio in the Bearish test set. Although LSTM exhibited a -44.29% cumulative return in the Overall test set, it surprisingly generated a significant 52.61% cumulative return in the Bullish test set. Res2Net emerged as the top-performing architecture in the Sideways test set, securing the second-highest cumulative return of 34.18% in the Bullish test set. On the other hand, ResNet delivered poor cumulative returns in both the Overall and Bearish test sets. Overall, while MLP did not claim the top position in this experiment, it demonstrated the most consistent performance across all the test sets, making it the most versatile and reliable choice.

Table 5 Comparing network architectures. Boldfaces refer to the winners. Gray numbers refer to uninterpretable results.

Method	Cumulative Return (percent) ↑	Sharpe Ratio ↑	Calmar Ratio ↑	Volatility (percent) ↓	MDD (percent) ↓
Overall					
LSTM	-44.29	-0.85	-0.63	89.41	-70.25
GRU	-54.13	-1.53	-0.84	79.93	-64.08
ResNet	-57.88	-1.59	-0.86	84.90	-67.67
Res2Net	-37.91	-0.62	-0.59	89.12	-64.68
MLP	-17.47	-0.12	-0.35	75.69	-49.81
Bullish					
LSTM	52.61	5.9	2.97	91.63	-17.71
GRU	31.36	4.98	2.36	69.44	-13.31
ResNet	23.7	4.03	1.75	67.96	-13.57
Res2Net	34.18	4.55	2.00	83.93	-17.06
MLP	28.26	4.64	2.35	68.25	-12.02
Bearish					
LSTM	-3.86	-1.99	-0.47	22.74	-8.23
GRU	-25.56	-6.25	-0.98	55.08	-26.02
ResNet	-29.56	-6.8	-0.96	60.08	-30.89
Res2Net	-11.21	-2.92	-0.71	45.99	-15.87
MLP	2.36	2.83	1.14	10.26	-2.07
Up-Down					
LSTM	1.59	1.47	0.38	13.22	-4.18
GRU	1.43	0.58	0.08	53.14	-17.24
ResNet	-5.61	-0.84	-0.23	59.83	-24.63
Res2Net	1.15	0.6	0.12	30.14	-9.31
MLP	2.35	2.76	1.13	10.09	-2.07
Sideways					
LSTM	-3.25	-0.11	-0.20	78.28	-16.01
GRU	-6.66	-0.55	-0.27	83.94	-24.44
ResNet	-4.01	-0.19	-0.21	80.94	-19.36
Res2Net	6.52	1.22	0.29	110.9	-22.19
MLP	1.21	0.62	0.07	93.29	-17.59

5.4 Discussion

This section provides a detailed discussion of the bearish test result, the overall test result, and an analysis of trade count and transaction cost.

5.4.1 Bearish test result

The findings presented in Table 4 highlight a noteworthy observation: among the baseline methods, only our method managed to generate profits in the Bearish test set, achieving a notable gain of 2.36% in portfolio value. This promising result prompted us to delve deeper into the analysis of the Bearish test set. Visual evidence in Figure 11 further supports the effectiveness of our method, as it accurately predicted trade signals by strategically buying tokens during value increases and selling them prior to value declines, resulting in a considerable increase in portfolio value.

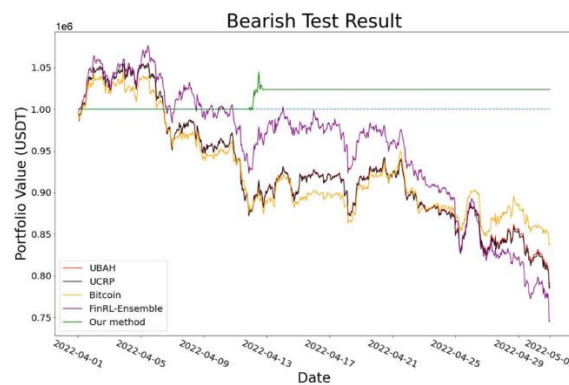


Figure 11 Result of the Bearish test set.

According to Figure 12, it was observed that at 13:00 on April 11, 2022, the ADX values exceeded the threshold of 60, indicating a highly robust market trend. Subsequently, at 14:00 on the same day, the RSI values dipped below 30, indicating an oversold market condition. As a result, starting from 20:00 on the same day, the agents initiated the purchase of all tokens except BTCUSDT, which continued until 1:00 on April 12, 2022. At 8:00 on April 12, 2022, the ADX values exhibited a decline below 30, indicating a market trend characterized by weakness and volatility.

Similarly, at 12:00 on the same day, the RSI values surpassed 50 and continued to rise, indicating that the tokens were nearing an overbought condition.

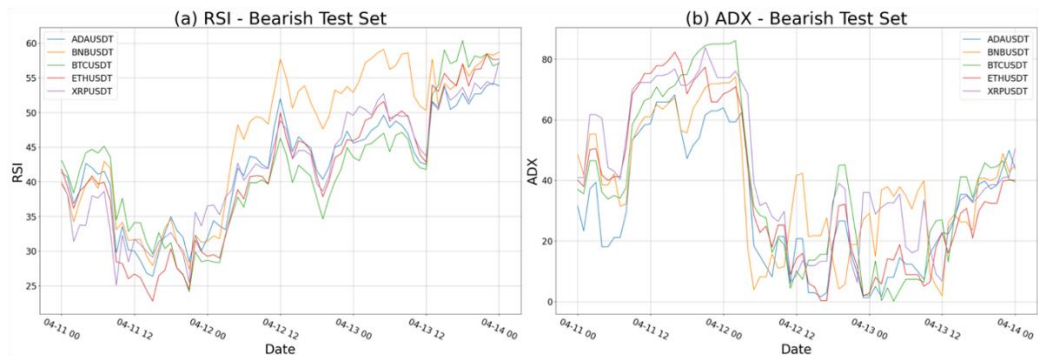


Figure 12 Trade signals from the bearish test set. (a) Relative Strength Index (RSI). (b) Average Directional Index (ADX).

Furthermore, Figure 13 shows that the agents incurred a penalty in the form of the MSCL reward, signaling a decrease in the portfolio value during the subsequent hour. Consequently, at 14:00 on April 12, 2022, the agents commenced the selling process of the tokens in the portfolio, culminating in the complete liquidation of all tokens by 22:00.

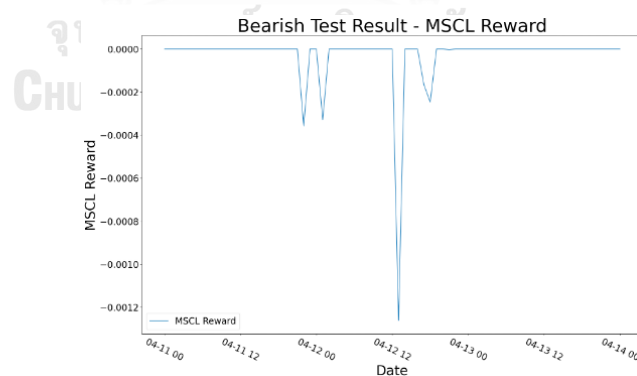


Figure 13 Multi-Scale Continuous Loss (MSCL) Reward during the bearish test set

5.4.2 Overall test result

While the outcomes presented in Table 4 highlight the profitability of our approach across bullish, bearish, up-down, and sideways test sets, it is noteworthy that our method yielded a negative cumulative return in the overall test set.

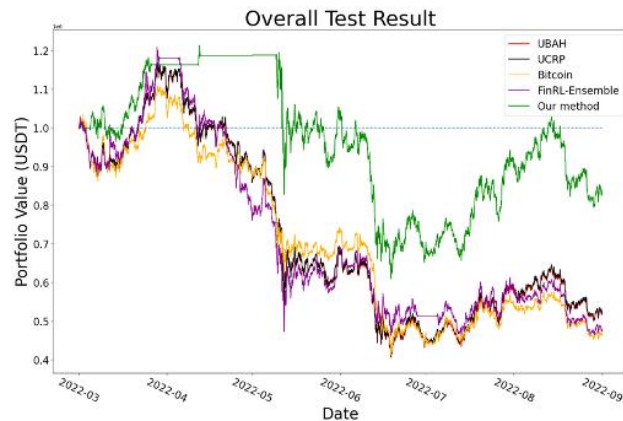


Figure 14 The results of the overall test set

Figure 14 depicts significant drawdown periods experienced by our method. The first substantial drawdown occurred from May 9, 2022, to May 12, 2022, where the portfolio value declined from 1.19 million USDT to 0.83 million USDT, reflecting a 30% drawdown. Another substantial drawdown took place from Jun 6, 2022, to Jun 18, 2022, as the portfolio value decreased from 1.02 million USDT to 0.61 million USDT, marking a 40% drawdown. Despite these two major drawdowns, our method exhibited lower losses than the baselines in the overall test set.

5.4.3 Trade count and transaction cost

Table 6 presents a comparison between our method and the baseline methods in terms of the frequency of trading occurrences and associated transaction costs. The findings reveal that our method executed fewer trades and incurred lower transaction costs compared to FinRL-Ensemble. Moreover, when considering the profit-making

capabilities highlighted in Table 4, it can be concluded that our method demonstrates more optimal trade execution than FinRL-Ensemble.

Table 6 Number of trades and transaction cost of our method compared to the baselines.

Method	Buy Count	Sell Count	Total Trade Count	Transaction Cost (USDT)
Overall Test Set				
UBAH	5	0	5	999.00
UCRP	8278	8137	16415	8458.08
Bitcoin	1	0	1	999.00
FinRL-Ensemble	1215	920	2135	94901.93
MAPPO + MSCL + Risk + Cost (our)	129	107	236	8195.97
Bullish Test Set				
UBAH	5	0	5	999.00
UCRP	1634	1480	3114	3118.41
Bitcoin	1	0	1	999.00
FinRL-Ensemble	178	111	289	22468.51
MAPPO + MSCL + Risk + Cost (our)	24	9	33	1241.51
Bearish Test Set				
UBAH	5	0	5	999.00
UCRP	1241	1374	2615	2131.90
Bitcoin	1	0	1	999.00
FinRL-Ensemble	102	64	166	2960.81
MAPPO + MSCL + Risk + Cost (our)	35	27	62	2625.14
Up-Down Test Set				
UBAH	5	0	5	999.00
UCRP	1511	1487	2998	2536.55
Bitcoin	1	0	1	999.00
FinRL-Ensemble	630	463	1093	32458.88
MAPPO + MSCL + Risk + Cost (our)	32	26	58	2578.13
Sideways Test Set				
UBAH	5	0	5	999.00
UCRP	1573	1591	3164	3198.00
Bitcoin	1	0	1	999.00
FinRL-Ensemble	259	151	410	26928.45
MAPPO + MSCL + Risk + Cost (our)	0	14	410	999.00

CHAPTER VI

CONCLUSION

In this study, we have proposed a novel approach for trading multiple cryptocurrency tokens using multi-agent deep reinforcement learning. Our method stands out from existing approaches by incorporating a local-global reward function that aims to optimize the performance of individual agents as well as collective behavior. Additionally, we have introduced a distinctive multi-scale continuous loss (MSCL) reward, which effectively prevents further declines in portfolio value, leading to improved cumulative returns. Comparative analysis against baseline methods clearly demonstrates the superior performance of our approach. Notably, our method exhibits the ability to generate profits, particularly in the challenging Bearish test set where other baseline methods experienced significant losses.

CHAPTER VII

Appendices

7.1 Reward Hyperparameter Tuning

In the quest for determining the optimal hyperparameters for the reward function, a thorough exploration of various values for the cost-sensitive reward coefficients (ϕ), risk-sensitive reward coefficients (ρ), and multi-scale continuous (MSCL) reward coefficients (μ) was undertaken. It should be acknowledged that a comprehensive grid search was not feasible due to the excessively long training duration. However, Table 7 demonstrates that our method achieved the highest cumulative return on the validation dataset when the following hyperparameters were employed:

- Cost-sensitive reward coefficient: $\phi = 10^{-6}$
- Risk-sensitive reward coefficient: $\rho = 10^3$
- MSCL reward coefficient: $\mu = 0.1$

As a result, this set of hyperparameters was utilized in our experiments.

Table 7 Validation results of reward hyperparameters tuning. The grey row refers to the best hyperparameter for the reward function.

	Cost-Sensitive Reward Coefficient (ϕ)	Risk-sensitive Reward Coefficient (ρ)	MSCL Reward Coefficient (μ)	Validation Cumulative Return (percent) \uparrow
1	10^{-6}	10^2	0.1	1.06
2	10^{-6}	10^3	0.1	25.05
3	10^{-6}	10^4	0.1	19.69
4	10^{-6}	10^3	0.2	23.59

REFERENCES

- [1] CoinMarketCap. "Global Cryptocurrency Chart."
<https://coinmarketcap.com/charts/> (accessed Jan. 13, 2023).
- [2] CoinMarketCap. "Top Cryptocurrency Spot Exchanges."
<https://coinmarketcap.com/rankings/exchanges/> (accessed Jan. 13, 2023).
- [3] A. M. Khedr, I. Arif, P. R. P V, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan, "Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey," *Intelligent Systems in Accounting, Finance and Management*, vol. 28, no. 1, pp. 3-34, 2021, doi: 10.1002/isaf.1488.
- [4] D. R. Pant, P. Neupane, A. Poudel, A. K. Pokhrel, and B. K. Lama, "Recurrent Neural Network Based Bitcoin Price Prediction by Twitter Sentiment Analysis," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, 25-27 Oct. 2018 2018, pp. 128-132, doi: 10.1109/CCCS.2018.8586824.
- [5] T. Shintate and L. Pichl, "Trend Prediction Classification for High Frequency Bitcoin Time Series with Deep Learning," *Journal of Risk and Financial Management*, vol. 12, no. 1, p. 17, 2019. [Online]. Available: <https://www.mdpi.com/1911-8074/12/1/17>.
- [6] A.-D. Vo, "Sentiment Analysis of News for Effective Cryptocurrency Price Prediction," *International Journal of Knowledge Engineering*, vol. 5, no. 2, pp. 47-52, 2019, doi: 10.18178/ijke.2019.5.2.116.
- [7] JAMES CHEN. "Technical Indicator: Definition, Analyst Uses, Types and Examples." Investopedia.
<https://www.investopedia.com/terms/t/technicalindicator.asp> (accessed Jan. 16, 2023).
- [8] BRIAN DOLAN. "MACD Indicator Explained, with Formula, Examples, and Limitations." <https://www.investopedia.com/terms/m/macd.asp> (accessed Jan. 16, 2023).
- [9] JASON FERNANDO. "Relative Strength Index (RSI) Indicator Explained With

- Formula." <https://www.investopedia.com/terms/r/rsi.asp> (accessed Jan. 16, 2023).
- [10] CORY MITCHELL. "What Is the Commodity Channel Index (CCI)? How To Calculate." <https://www.investopedia.com/terms/c/commoditychannelindex.asp> (accessed Jan. 16, 2023).
- [11] CORY MITCHELL. "Average Directional Index (ADX): Definition and Formula." <https://www.investopedia.com/terms/a/adx.asp> (accessed Jan. 16, 2023).
- [12] A. P. N. Nguyen, M. Crane, and M. Bezbradica, "Cryptocurrency Volatility Index: An Efficient Way to Predict the Future CVI," in *Artificial Intelligence and Cognitive Science*, Cham, L. Longo and R. O'Reilly, Eds., 2023// 2023: Springer Nature Switzerland, pp. 355-367.
- [13] Alternative. "Crypto Fear & Greed Index." <https://alternative.me/crypto/fear-and-greed-index/> (accessed May. 12, 2023).
- [14] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid, "Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy," presented at the International Conference on AI in Finance, Sep. 11, 2020.
- [15] Cedric Zhuang. stockstats (0.5.1) [Online] Available: <https://github.com/jealous/stockstats>
- [16] JAMES CHEN. "Cumulative Return: Definition, Calculation, and Example." <https://www.investopedia.com/terms/c/cumulativereturn.asp> (accessed Jan. 16, 2023).
- [17] JASON FERNANDO. "Sharpe Ratio Formula and Definition With Examples " <https://www.investopedia.com/terms/s/sharperatio.asp> (accessed Jan. 16, 2023).
- [18] WILL KENTON. "What Is the Calmar Ratio, Its Strengths & Weaknesses?" <https://www.investopedia.com/terms/c/calmarratio.asp> (accessed Jan.16, 2023).
- [19] ADAM HAYES. "Volatility: Meaning In Finance and How it Works with Stocks." <https://www.investopedia.com/terms/v/volatility.asp> (accessed Jan. 16, 2023).
- [20] ADAM HAYES. "Maximum Drawdown (MDD) Defined, With Formular for Calculation." <https://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp> (accessed Jan. 16, 2023).

- [21] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229-256, 1992/05/01 1992, doi: 10.1007/BF00992696.
- [22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [24] M. Taghian, A. Asadi, and R. Safabakhsh, "Learning financial asset-specific trading rules via deep reinforcement learning," *Expert Systems with Applications*, vol. 195, p. 116523, 2022/06/01/ 2022, doi: <https://doi.org/10.1016/j.eswa.2022.116523>.
- [25] T. Théate and D. Ernst, "An application of deep reinforcement learning to algorithmic trading," *Expert Systems with Applications*, vol. 173, p. 114632, 2021/07/01/ 2021, doi: <https://doi.org/10.1016/j.eswa.2021.114632>.
- [26] Y.-C. Tsai, F.-M. Szu, J.-H. Chen, and S. Y.-C. Chen, "Financial Vision-Based Reinforcement Learning Trading Strategy," *Analytics*, vol. 1, no. 1, pp. 35-53, 2022. [Online]. Available: <https://www.mdpi.com/2813-2203/1/1/4>.
- [27] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," *arXiv preprint arXiv:1706.10059*, 2017.
- [28] L. Weng, X. Sun, M. Xia, J. Liu, and Y. Xu, "Portfolio trading system of digital currencies: A deep reinforcement learning with multidimensional attention gating mechanism," *Neurocomputing*, vol. 402, pp. 171-182, 2020/08/18/ 2020, doi: <https://doi.org/10.1016/j.neucom.2020.04.004>.
- [29] C. Betancourt and W.-H. Chen, "Reinforcement Learning with Self-Attention Networks for Cryptocurrency Trading," *Applied Sciences*, vol. 11, no. 16, p. 7377, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/16/7377>.
- [30] R. Sun, Z. Jiang, and J. Su, "A Deep Residual Shrinkage Neural Network-based Deep Reinforcement Learning Strategy in Financial Portfolio Management," in *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, 5-8 March 2021 2021, pp. 76-86, doi: 10.1109/ICBDA51983.2021.9403210.
- [31] W. Yao, X. Ren, and J. Su, "An Inception Network with Bottleneck Attention

- Module for Deep Reinforcement Learning Framework in Financial Portfolio Management," in *2022 7th International Conference on Big Data Analytics (ICBDA)*, 4-6 March 2022 2022, pp. 310-316, doi: 10.1109/ICBDA55095.2022.9760343.
- [32] B. J. D. Gort, X.-Y. Liu, X. Sun, J. Gao, S. Chen, and C. D. Wang, "Deep reinforcement learning for cryptocurrency trading: Practical approach to address backtest overfitting," *arXiv preprint arXiv:2209.05559*, 2022.
- [33] O. Sattarov *et al.*, "Recommending cryptocurrency trading points with deep reinforcement learning approach," *Applied Sciences*, vol. 10, no. 4, p. 1506, 2020.
- [34] C. Betancourt and W.-H. Chen, "Deep reinforcement learning for portfolio management of markets with a dynamic number of assets," *Expert Systems with Applications*, vol. 164, p. 114002, 2021/02/01/ 2021, doi: <https://doi.org/10.1016/j.eswa.2020.114002>.
- [35] K. Bisht and A. Kumar, "Deep Reinforcement Learning based Multi-Objective Systems for Financial Trading," in *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 1-3 Dec. 2020 2020, pp. 1-6, doi: 10.1109/ICRAIE51050.2020.9358319.
- [36] Y. Zhang, P. Zhao, Q. Wu, B. Li, J. Huang, and M. Tan, "Cost-Sensitive Portfolio Selection via Deep Reinforcement Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 236-248, 2022, doi: 10.1109/TKDE.2020.2979700.
- [37] Binance. "Binance API." <https://binance-docs.github.io/apidocs/spot/en/> (accessed 2022).
- [38] E. Liang *et al.*, "RLlib: Abstractions for distributed reinforcement learning," in *International Conference on Machine Learning*, 2018: PMLR, pp. 3053-3062.
- [39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [40] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image

recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

- [42] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp. 652-662, 2019.





จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

VITA

NAME Kittiwin Kumlungmak

DATE OF BIRTH 14 August 1996

PLACE OF BIRTH Phitsanulok, Thailand

INSTITUTIONS ATTENDED Florida Institute of Technology

HOME ADDRESS 11/710 The Privacy Tha Phra Interchange, Wat Tha Phra, Bangkok Yai, Bangkok, Thailand, 10600

PUBLICATION K. Kumlungmak and P. Vateekul, "Multi-Agent Deep Reinforcement Learning With Progressive Negative Reward for Cryptocurrency Trading," in IEEE Access, doi: 10.1109/ACCESS.2023.3289844.