Dynamic Portfolio Optimization Using Reinforcement Learning in Cryptocurrency Markets

Zhongyuan Xu

Goizueta Business School, Master of Finance, Emory University, Atlanta, United States
916864794@qq.com

Abstract: *The cryptocurrency market poses a huge challenge to portfolio optimization due to its high volatility and complex market dynamics. To address these issues, this paper uses reinforcement learning (RL) algorithms for dynamic portfolio optimization, aiming to improve the return and risk control capabilities of the portfolio through intelligent decision-making. This paper adopts a strategy based on deep reinforcement learning. By interacting with the cryptocurrency market, the agent can continuously optimize asset allocation, maximize investment returns while controlling volatility. The experimental results show that compared with traditional strategies, the reinforcement learning model has obvious advantages in key indicators such as cumulative return rate, annualized volatility, maximum drawdown and Sharpe ratio. Specifically, the cumulative return rate of the reinforcement learning model reaches 85.12%, the annualized volatility is 45.76%, and the maximum drawdown is controlled at -22.34%, showing strong income acquisition and risk management capabilities. In addition, the dynamic adjustment of asset allocation has optimized the weights of various cryptocurrencies, effectively dispersed risks, and improved the overall performance of the investment portfolio.*

Keywords: *Reinforcement Learning; Dynamic Portfolio Optimization; Cryptocurrency Market; Monte Carlo tree search; Risk Control*

1. Introduction

Traditional quantitative investment strategies are mainly designed for low-frequency to medium-frequency transactions, and are difficult to cope with the complex computing requirements in high-frequency environments. Most current methods rely on simplified market models and limited operating dimension settings. Therefore, how to effectively combine modern computing technology and optimization models to improve the accuracy and operability of portfolio optimization is still an important topic in current research. This paper aims to explore the application of reinforcement learning in dynamic portfolio optimization, optimize asset allocation strategies by leveraging the advantages of reinforcement learning, and verify it in actual market data, providing an efficient and operational portfolio optimization method that aims to address the shortcomings of existing methods in terms of real-time, adaptability, and scalability.

This paper first introduces the background and challenges of dynamic portfolio optimization, focusing on the potential and application of reinforcement learning in this field. Then, the reinforcement learning algorithm used and its implementation in the optimization model are described in detail, and experimental verification is carried out with actual market data. By comparing with traditional optimization methods, the performance and advantages of the reinforcement learning model under different market conditions are analyzed. Finally, the significance of the experimental results, the limitations of the model and future research directions are discussed, in order to provide useful inspiration for the further development of portfolio optimization.

2. Related Work

In the field of portfolio optimization, as the market environment continues to change, various optimization methods and technologies continue to emerge, aiming to help investors achieve higher returns and lower risks. Gunjan and Bhattacharyya reviewed the classical, statistical, and intelligent methods used in portfolio optimization in finance and management, proposed to summarize classical, intelligent, and quantum heuristic techniques, and explored their applications in portfolio optimization

[1]. Erwin and Engelbrecht proposed evolutionary algorithms and swarm intelligence algorithms to solve portfolio optimization problems. They classified them by the type of optimization problem (unconstrained or constrained) and single-objective and multi-objective methods, and analyzed in detail different portfolio models, constraints, objectives, and characteristics [2]. El-Morsy transformed the problem into a deterministic form through a scoring function and developed a corresponding solution algorithm to help investors improve their expected returns when choosing risk factors and determine investment strategies based on their own circumstances. The TORA program was used to determine the optimal rate of return, and the efficiency and reliability of the method were demonstrated through examples [3]. Butler and Kwon provided closed-form analytical solutions to the mean-variance optimization (MVO) problem with no constraints and equity constraints; for general inequality constraint problems, a neural network architecture was used to efficiently optimize batch quadratic programming. The advantages of this integrated method over traditional decoupling methods were demonstrated through simulations of synthetic data and global futures data [4]. Yadav et al. studied the spillover effects of the Chinese stock market on some emerging economies to test diversification investment opportunities. Through Granger causality test and DCC-GARCH (Dynamic Conditional Correlation Generalized Autoregressive Conditional Heteroskedasticity Model) model, they found that there was a two-way causal relationship between China and Indonesia throughout the period, and the spillover effect of the Chinese market on the Indonesian market existed in both the short and long term [5]. The use of quantum technologies in the financial industry, including Monte Carlo techniques for risk assessment, identifying fraudulent transactions, derivative the price, and portfolio optimization, have been incorporated and shown by Naik et al. [6]. In order to advance the study of unpredictable optimization in financial operations, Uysal et al. proposed a unified feedforward network method that combines these two steps and is divided into two variants: model-free neural network and model-based neural network. Experimental results show that the model-based method has a robust performance during the period 2017-2021, maximizing the Sharpe ratio [7]. Idzorek proposed a multi-account alpha tracking error framework that can optimize the asset allocation of investors in multiple accounts at the same time. The objective function can also incorporate investors' non-economic preferences, such as environmental, social and governance characteristics. Regularly running the multi-account optimizer can achieve personalized asset allocation, tax loss harvesting, portfolio rebalancing, account conversion optimization and other functions [8]. Dai et al. used the time-varying parameter GAS-D-Vine-Copula model to construct the joint distribution of multi-asset return series and used utility functions and cost functions to evaluate the efficiency of the portfolio strategy. Through empirical analysis, it was found that the investment period of 10 to 50 days is the most efficient, and the multi-objective portfolio strategy is not necessarily better than a single objective [9]. Based on the Jakarta Composite Index, Irwan et al. sought to identify the best investment opportunities for the Indonesian Stock Exchange's phone company between January 2018 and December 2020. The analysis results showed that the combination of ISAT and FREN constituted the optimal investment portfolio with an expected return of 5.08% and a risk of 8.02%[10]. Xidonas and Essner proposed a portfolio optimization model based on multi-objective minimization [11]. The results showed that the optimal environmental, social and governance investment portfolio generated by the model significantly outperformed its corresponding market benchmark in terms of risk-adjusted returns [12-13]. Although existing research has made some progress in the field of portfolio optimization, there are still some bottlenecks, such as high model complexity, high computational cost, lack of optimization methods for specific market conditions, and unsatisfactory optimization results under multi-objective and nonlinear constraints [14-15].

3. Method

3.1 Cryptocurrency Market Modeling

When performing dynamic portfolio optimization in the cryptocurrency market, market modeling is a crucial step. By building a suitable market model, sufficient contextual information can be provided to the reinforcement learning algorithm, so that the decision-making process can more accurately reflect the dynamic characteristics of the market.

The market state definition determines the market information that the model can access. This information provides the basis for the agent (investor) to make decisions, usually including price data, trading volume, volatility, etc.

In the cryptocurrency market, common market characteristics include price, trading volume, and volatility, which are key indicators used to measure market conditions.

Market price (Price,P_t):P_t: represents the price of a cryptocurrency at time t. Market price is usually a function based on historical price data. The formula is $P_t = f(Historical\ Price)$. The historical price includes the opening price, closing price, highest price and lowest price.

Volume(V_t): Volume refers to the number of cryptocurrencies traded in a certain period of time. It is an important indicator for measuring market liquidity. The formula is $V_t = \sum_{i=1}^{n} Quantity_i$. Quantity$_i$ represents the number of transactions per transaction, and n represents the number of transactions in the period of time.

Volatility (σ_t): Volatility usually indicates the magnitude of changes in asset prices. Commonly used measurement methods are standard deviation or historical volatility. The formula is expressed as $\sigma_t = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - \bar{P})^2}$. Among them, P_i is each price in the time window; $\bar{P}$ is the average price in the time window; n is the total number of price data.

3.2 Market Trends and Technical Indicators

Moving average is a commonly used trend analysis tool used to smooth price data and eliminate market noise.

Simple Moving Average (SMA) formula:

$$\mathrm{SMA_n} = \frac{1}{\mathrm{n}}\sum_{\mathrm{t=n+1}}^{\mathrm{T}} \mathrm{P_t} \tag{1}$$

Among them, P_t is the market price at time t, and n is the window size of the moving average.

Relative Strength Index (RSI): RSI is an indicator that measures the speed and magnitude of price changes, and is usually used to determine the overbought or oversold situation of the market.

RSI formula:

$$\mathrm{RSI} = 100 - \frac{100}{1+\mathrm{RS}} \tag{2}$$

$$\mathrm{RS} = \frac{\text{Average increase}}{\text{Average decline}} \tag{3}$$

RSI values usually range from 0 to 100, with values above 70 indicating that the market may be overbought and values below 30 indicating that the market may be oversold.

Moving Average Convergence Divergence (MACD):

MACD is calculated using short-term and long-term exponential moving averages (EMA) to identify market trends and reversal signals. The MACD formula is as follows:

$$\mathrm{MACD_t} = \mathrm{EMA_{12}(P_t)} - \mathrm{EMA_{26}(P_t)} \tag{4}$$

$$Signal_t = EMA_9(MACD_t) \tag{5}$$

Among them, $\mathrm{EMA_{12}(P_t)}$ and $\mathrm{EMA_{26}(P_t)}$ are the 12-day and 26-day exponential moving averages, respectively, and Signal is the 9-day MACD signal line.

3.3 Portfolio State Modeling

Portfolio state modeling is a key step in providing decision-making basis for reinforcement learning agents. The portfolio state needs to include information such as the current asset allocation, the types of cryptocurrencies held, the weights of each asset, and the risk preferences of investors.

Asset allocation refers to the proportion of investment funds allocated to different assets. Assuming that the investment portfolio contains N cryptocurrency assets and the weight of each asset is w_i, the total value of the portfolio V_t is:

$$V_t = \sum_{i=1}^{N} w_i . P_{i,t} \tag{6}$$

Among them, w_i represents the proportion of the i-th cryptocurrency in the portfolio, and $P_{i,t}$ is the market price of the i-th asset at time t.

Investors' risk preferences affect asset allocation decisions. Assuming that the investor's goal is to maximize the return of his portfolio while controlling risk (such as volatility). The investor's risk preference can be modeled by the following function:

$$R_t = \frac{E[R_t]}{\sigma_t} \tag{7}$$

Among them, $E[R_t]$ is the expected return of the portfolio and σ_t is the standard deviation (risk) of the portfolio.

In addition, the investor's objective function can be defined as:

$$\text{Objective Function} = \alpha . E[R_t] - \beta . \sigma_t \tag{8}$$

Among them, α and β are the weights of investors on return and risk, respectively, reflecting the risk preference of investors.

3.4 Use of Time Series and Historical Data

In reinforcement learning, time series data is crucial for predicting market trends and adjusting investment decisions. By processing historical data, the agent can learn the laws of the market and make more reasonable investment decisions.

Historical price data is an important input for building market status. In order to be effectively used in reinforcement learning, historical data usually needs to be normalized or standardized to eliminate the impact of price fluctuations between different cryptocurrencies on the model.

Normalization:

$$P_{t,norm} = \frac{P_t - \mu}{\sigma} \tag{9}$$

Among them, μ is the mean of historical prices and σ is the standard deviation.

Log Return:

$$\mathrm{r_t} = \log(\mathrm{P_t}) - \log(\mathrm{P_{t-1}}) \tag{10}$$

Logarithmic returns can handle extreme price fluctuations and are often used in financial time series analysis.

In order to improve the prediction accuracy and decision-making ability of the model, technical indicators (such as MA, RSI, MACD, etc.) are usually extracted from historical price data as features of the reinforcement learning model. Using these technical indicators, the model can capture deeper market trends and reversal signals.

The calculation and extraction of these technical indicators can be achieved through simple mathematical formulas. As mentioned earlier, RSI, MACD, MA, etc. are all based on historical data.

3.5 Search Strategy and Optimization of MCTS

In dynamic portfolio optimization in the cryptocurrency market, Monte Carlo Tree Search (MCTS) is a powerful search algorithm that can effectively handle complex decision spaces and uncertainties. MCTS gradually optimizes decision strategies by simulating different decision paths. The following is a detailed expansion of the MCTS search strategy, covering tree construction and expansion, selection strategy, and pruning and optimization.

The core idea of MCTS is to construct a tree to represent all possible decision sequences and evaluate the potential of each decision through random simulation (Monte Carlo simulation). In portfolio optimization, each node of the tree represents a market state, the edge represents a decision (such as buying, selling or holding a certain cryptocurrency), and the sum of the paths is the execution process of the strategy.

State expansion refers to the expansion of possible successor states based on the current state under each node of the tree. In the context of portfolio optimization, the process of state expansion can be achieved by executing different investment decisions (such as adjusting asset allocation, buying and selling assets, etc.).

At each decision node v (i.e., each market state), an unexpanded successor node can be selected to expand. Assuming that the current state of the portfolio is s_t, at time t, executing a decision action a^t will result in a new market state s_{t-1}, that is:

$$s_{t+1} = f(s_t, a^t) \tag{11}$$

The expansion of each node will produce a new decision state, and each state contains relevant market characteristics (such as price, trading volume, etc.) and the current asset allocation.

Each time a node is expanded, MCTS will perform a random simulation (i.e., estimate the reward through the Monte Carlo method). The simulation process usually involves simulating several decision steps, starting from the current node v, making multiple random selections, and finally obtaining a reward value (gain or loss).

Assuming that a random simulation is performed at a certain node v. At the end of the simulation, the final reward R obtained by the agent can be expressed as:

$$\mathrm{R} = \mathrm{f}(\mathrm{s_t}, \mathrm{a_t}, \mathrm{s_{t-1}}, \cdots, \mathrm{s_T}) \tag{12}$$

Among them, $\mathrm{s_t}$ is the starting state; $\mathrm{a_t}$ is the decision action; $\mathrm{s_{t-1}}, \cdots, \mathrm{s_T}$ is the state sequence generated during the simulation; T is the maximum number of simulation steps.

In actual portfolio optimization problems, this return can be the total return of the portfolio, or the risk-adjusted return (such as Sharpe ratio, maximum drawdown, etc.).

3.6 Selection Strategy

The selection strategy is the core mechanism in MCTS that determines how to reach a leaf node from the root node of the tree. In portfolio optimization, the selection strategy helps the agent choose the best action from the current state (such as adjusting asset allocation, executing buy or sell, etc.).

UCT is a commonly used selection strategy in MCTS, which guides the expansion of the tree by balancing exploration (selecting uncommon actions) and exploitation (selecting actions with the best rewards). The core idea of the UCT strategy is to select the optimal node through the upper confidence bound.

For each node v, the UCT strategy selects the best action based on the average reward $\mathrm{Q(v)}$ and the number of visits $\mathrm{N(v)}$ of the current node. The calculation formula of the UCT value is:

$$\mathrm{UCT(v)} = \frac{\mathrm{Q(v)}}{\mathrm{N(v)}} + \mathrm{C}.\sqrt{\frac{\mathrm{lnN_{patent}}}{\mathrm{N(v)}}} \tag{13}$$

Among them: $\mathrm{Q(v)}$ is the average return of node v; $\mathrm{N(v)}$ is the number of times node v is visited; $\mathrm{N_{patent}}$ is the number of times node v's parent node is visited; C is a constant that controls the balance between exploration and utilization.

Through the above formula, UCT will select nodes with higher average returns (i.e., utilization strategy), and will also encourage visits to relatively less visited nodes (i.e., exploration strategy), so that the search process can both stably utilize existing information and discover new potential good strategies.

4. Results and Discussion

4.1 Experimental Environment and Data Preparation

Market data: Collecting historical market data for multiple cryptocurrencies, including prices (opening price, closing price, highest price, lowest price), trading volume, volatility, etc. Public cryptocurrency market data sources can be used, such as CryptoCompare, CoinMarketCap, etc.

Technical indicators: Calculating and extracting common technical indicators (such as SMA, RSI, MACD) as feature inputs of the model.

Time series data: Processing data at time intervals such as days and hours to generate time series.

4.2 Experimental Evaluation and Result Analysis

The total return rate is used to calculate the overall return of the portfolio during the test period, while the risk measurement evaluates the risk of the portfolio through indicators such as volatility and maximum drawdown. The Sharpe ratio is an important indicator for measuring the risk-adjusted return of the portfolio, which reflects the excess return obtained per unit of risk. The maximum drawdown measures the maximum decline in the value of the portfolio during the test period, which is used to evaluate the maximum loss that the portfolio may face.

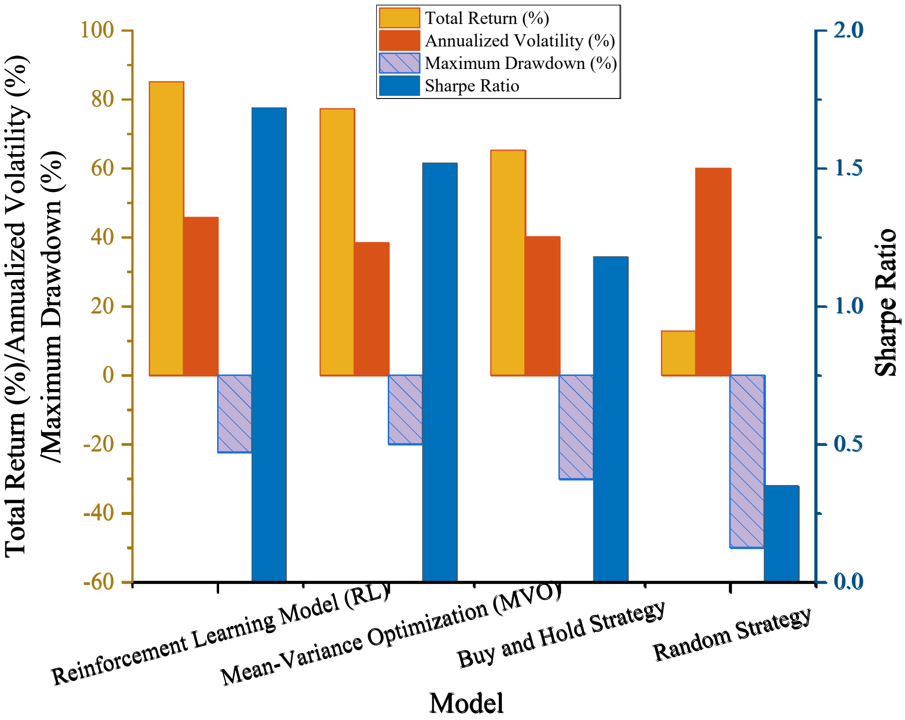

Figure 1. Total return and risk assessment (comparison between benchmark model and reinforcement learning model)

The total return of the reinforcement learning model is 85.12%, which is significantly higher than other strategies, especially the traditional "buy and hold" strategy (65.25%). This shows that reinforcement learning has a strong ability to adjust the portfolio dynamically. In contrast, the mean variance optimization (MVO) model, although it performs better (77.3%), is still lower than the return of the reinforcement learning model. In terms of annualized volatility, the reinforcement learning model is 45.76%, slightly higher than the mean variance optimization (38.45%), but much lower than the 60% of the random strategy. Although the RL model has slightly higher volatility, the increase in its returns is enough to make up for this volatility, making its overall performance superior. The Sharpe ratio of the RL model is 1.72, which greatly exceeds other strategies. The Sharpe ratio of the mean-variance optimization is 1.52, the traditional "buy and hold" strategy is 1.18, and the random strategy is only 0.35. The results in Figure 1 show that the RL model can not only provide higher returns but also perform better in risk control and provide better risk-adjusted returns.

In terms of monthly returns, May performs best with a return of 7.94%, showing a relatively strong market performance in that month. It is followed by March (6.08%) and January (5.62%), all of which have impressive returns. In contrast, February has the lowest return of 3.74%, which is a positive return but still inferior to other months. In terms of monthly volatility, March has the highest volatility, reaching 30.1%, showing the instability of the market that month, which may have been affected by unexpected events or market sentiment fluctuations. The volatility of other months is relatively close, with June (27.8%) and April (25.33%) having higher volatility, showing a certain market uncertainty, while January (23.15%) has relatively low volatility. Regarding the maximum drawdown, the maximum drawdown in March is -10.45%, the largest among all months, indicating that the market has a higher downside risk in that month. The maximum drawdowns in other months are more dispersed, with April (-9.67%) and January (-8.22%) also showing large drawdowns, while February (-5.11%) and May (-6.8%) are relatively stable, as shown in Figure 2.

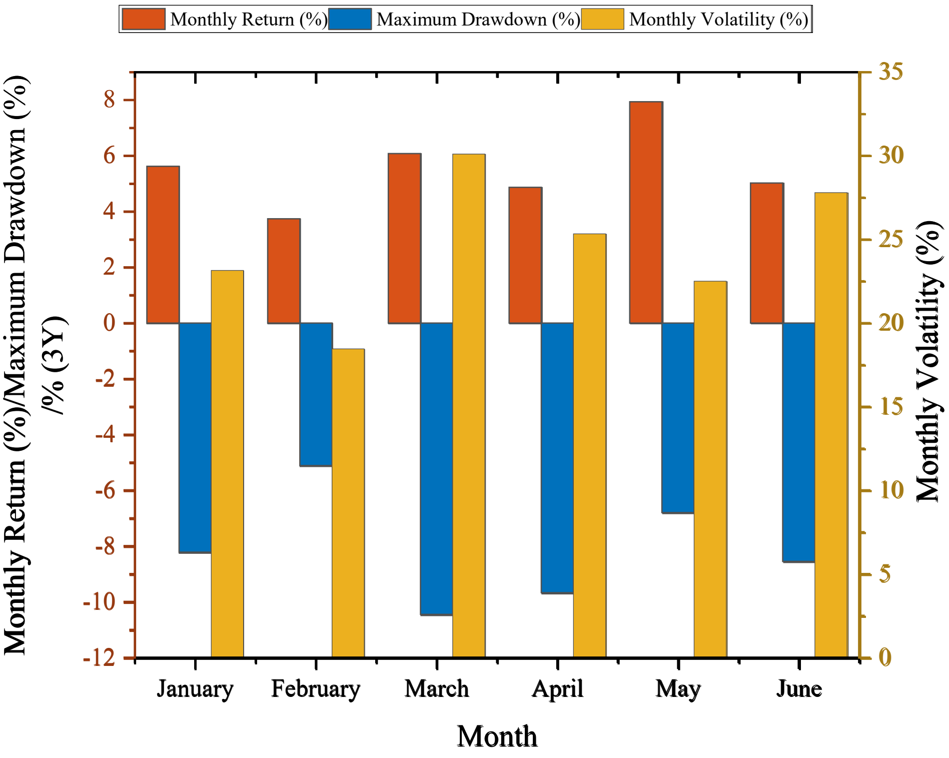

Figure 2. Monthly returns and volatility (Reinforcement Learning Model)

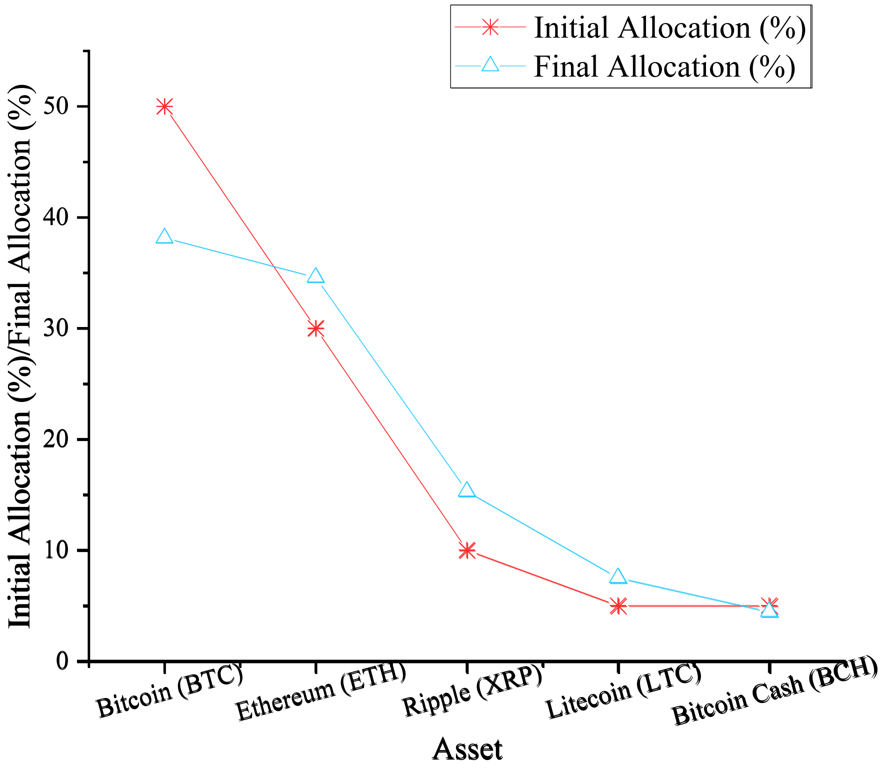

Figure 3. Asset allocation of different cryptocurrencies (reinforcement learning model)

According to the changes in the initial and final values of the asset allocation in Figure 3, the dynamic adjustment of the portfolio reflects the performance of different cryptocurrencies during this period and the impact of market conditions. The allocation ratio of Bitcoin (BTC) drops from 50% to 38.15%, a decrease of 11.85%. This change indicates that Bitcoin's performance during this period may be relatively flat or underperformed relative to other assets, so investors choose to reduce their allocation to it.

Table 1. Cumulative returns and risks under different strategies (comparison between reinforcement learning and benchmark strategies)

Strategy	Cumulative Return (%)	Annualized Volatility (%)	Cumulative Risk (Standard Deviation)	Maximum Drawdown (%)
Reinforcement Learning (RL)	85.12	45.76	24.35	-22.34
Mean-Variance Optimization (MVO)	77.3	38.45	21.9	-19.98
Buy and Hold Strategy	65.25	40.15	22.6	-30.12
Random Strategy	12.8	60	35	-50

The random strategy performs the worst, with a cumulative return rate of only 12.8%, an extremely high annualized volatility of 60%, and a much higher risk than other strategies. Its maximum drawdown is -50%, showing that the strategy lacks effective investment decisions and risk control in the market, resulting in huge losses in the face of an uncertain market environment. The reinforcement learning model (RL) performs best in terms of cumulative return rate, reaching 85.12%, higher than other strategies. Although its annualized volatility of 45.76% is high, indicating a high market volatility, the model significantly improves the portfolio return by optimizing decisions through reinforcement learning. Although its maximum drawdown is -22.34%, which is better than other strategies, it still indicates that large losses may occur under extreme market conditions. However, overall, the reinforcement learning model demonstrates strong adaptability and profitability (as shown Table 1). The reinforcement learning model shows the best return and relatively good risk control ability, which is suitable for coping with the high volatility of the cryptocurrency market.

5. Conclusion

This paper studies the problem of dynamic portfolio optimization based on reinforcement learning, proposes an improved reinforcement learning algorithm, and applies it to the portfolio optimization problem in actual market data. By comparing with traditional mean-variance optimization, buy-and-hold strategy and random strategy, the experimental results show that the reinforcement learning model has advantages over other strategies in terms of total return rate, annualized volatility, Sharpe ratio, etc., especially in terms of risk control and return balance. This study provides a more intelligent and dynamic optimization solution for actual investment decisions, and provides new ideas and methods for automated investment systems in future financial markets. However, this paper also has some limitations, such as the reinforcement learning model used does not consider more complex market dynamics, and the experiment is only based on single market data. Future research can be further expanded to multi-market and multi-asset scenarios, explore more complex portfolio optimization models, and improve the stability and operability of the model.

References

[1] Gunjan A, Bhattacharyya S. A brief review of portfolio optimization techniques[J]. Artificial Intelligence Review, 2023, 56(5): 3847-3886.

[2] Erwin K, Engelbrecht A. Meta-heuristics for portfolio optimization[J]. Soft Computing, 2023, 27(24): 19045-19073.

[3] El-Morsy S. Stock portfolio optimization using pythagorean fuzzy numbers[J]. Journal of operational and strategic analytics, 2023, 1(1): 8-13.

[4] Butler A, Kwon R H. Integrating prediction in mean-variance portfolio optimization[J]. Quantitative finance, 2023, 23(3): 429-452.

[5] Yadav M P, Sharma S, Bhardwaj I. Volatility spillover between Chinese stock market and selected emerging economies: A dynamic conditional correlation and portfolio optimization perspective[J]. Asia-Pacific Financial Markets, 2023, 30(2): 427-444.

[6] Naik A S, Yeniaras E, Hellstern G, et al. From portfolio optimization to quantum blockchain and security: A systematic review of quantum computing in finance[J]. Financial Innovation, 2025, 11(1): 1-67.

[7] Uysal A S, Li X, Mulvey J M. End-to-end risk budgeting portfolio optimization with neural

networks[J]. Annals of Operations Research, 2024, 339(1): 397-426.

[8] Idzorek T M. Personalized multiple account portfolio optimization[J]. Financial Analysts Journal, 2023, 79(3): 155-170.

[9] Dai X, Zhang D, Lau C K M, et al. Multiobjective portfolio optimization: Forecasting and evaluation under investment horizon heterogeneity[J]. Journal of Forecasting, 2023, 42(8): 2167-2196.

[10] Irwan I, Abdy M, Salsabila N K, et al. Analysis of Stock Portfolio Optimization in the Telecommunications Sector Using the Single Index Model[J]. ARRUS Journal of Mathematics and Applied Science, 2023, 3(1): 1-10.

[11] Xidonas P, Essner E. On ESG portfolio construction: A multi-objective optimization approach[J]. Computational Economics, 2024, 63(1): 21-45.

[12] Yang J. Application of Multi-model Fusion Deep NLP System in Classification of Brain Tumor Follow-Up Image Reports[C]. The International Conference on Cyber Security Intelligence and Analytics. Cham: Springer Nature Switzerland, 2024: 380-390.

[13] Yang J. Research on the Application of Medical Text Matching Technology Combined with Twin Network and Knowledge Distillation in Online Consultation[J]. Frontiers in Medical Science Research, 2024, 6(11): 25-29.

[14] Yang J. Research on the Strategy of MedKGGPT Model in Improving the Interpretability and Security of Large Language Models in the Medical Field[J]. Academic Journal of Medicine & Health Sciences, 2024, 5(9): 40-45.

[15] Shi C. DNA Microarray Technology Principles and Applications in Genetic Research. Computer Life, 2024, 12(3): 19-24