Asset Allocation: Financial Model VS Deep Reinforcement Learning

Xuanhe Chen
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
223015057@link.cuhk.edu.cn

Abstract

In this research, we evaluated the effectiveness of ten asset allocation algorithms, including traditional Markowitz-based methods (Tangency Portfolio, Minimum Variance Portfolio, Risk Parity, Equal Weight, Hierarchical Risk Parity) and advanced deep reinforcement learning (DRL) algorithms (A2C, PPO, DDPG, SAC, TD3). Our findings reveal that while traditional methods are stable in consistent market conditions, they are susceptible to sudden market shifts. In contrast, DRL algorithms demonstrated superior performance in handling time-series forecasting and portfolio allocation, efficiently managing risks and returns with only asset time series as input. Notably, the TD3 model achieved the highest annualized return, and the DDPG model attained the best Sharpe ratio among the DRL algorithms. This study underscores the potential of DRL in asset allocation and sets the stage for future research focusing on hyperparameter optimization, integration of diverse data sources, and exploration of various reward functions in complex market scenarios.

1 Introduction

Asset allocation is one of the most critical issues in finance, involving the calculation of optimal weights for assets within a given investment portfolio. The expected returns and risks during this period are essential components required by traditional asset allocation models. Since the advent of Modern Portfolio Theory by Markowitz [1952], numerous approaches to portfolio optimization have been developed within the framework of financial theory. A comprehensive overview can be found in Rubinstein [2002].

MPT presents several inherent limitations. A significant challenge arises from the high sensitivity of MPT to its model inputs, namely expected returns and the covariance matrix. To mitigate these issues, researchers have developed various methodologies for more precise estimation of expected returns and the covariance matrix. Among these, the Black and Litterman model stands out [Black and Litterman, 1992]. This model allows practitioners to incorporate their own projections of expected returns, differentiated from prevailing market views, along with a quantifiable confidence level in these alternative assumptions.

Beyond the Black and Litterman approach, the financial community has explored other methodologies focusing on risk management. These include the construction of equally weighted portfolios [Malladi and Fabozzi, 2017], the $1/N$ portfolio strategy [DeMiguel et al., 2009], and the equal volatility portfolio, which allocates identical levels of volatility to each asset [Stefanovits, 2010]. Furthermore, the Minimum Variance Portfolio focuses on minimizing overall portfolio volatility [Clarke et al., 2011]. Additionally, the concept of Risk Parity has gained traction, which ensures that each asset contributes equally to the portfolio's overall risk [Roncalli and Weisang, 2016]. Lastly, the Maximum Sharpe Ratio Portfolio is another key strategy, aimed at maximizing the return per unit of risk [Vinzelberg and Auer, 2022]. Each of these methodologies offers unique advantage.

Preprint. Under review.

The proliferation of artificial intelligence has precipitated a paradigm shift in deep-learning and reinforce learning methodologies, catalyzing unprecedented advancements and engendering state-of-the-art outcomes across a plethora of domains. Notably, this includes computer vision [Chai et al., 2021, Esteva et al., 2021], natural language processing [Sorin et al., 2020], and speech recognition [Zhang et al., 2018]. This surge in innovation has not gone unnoticed in the realm of finance research, where there has been a burgeoning interest. Consequently, recent years have witnessed a substantial escalation in the incorporation of deep-learning and reinforce learning techniques in asset allocation.

Reinforcement learning facilitates the resolution of dynamic optimization problems in the financial domain through an approach that is nearly model-free, thereby alleviating the reliance on assumptions typically requisite in classical financial methodologies. Kolm and Ritter [2020] elucidated the interconnection between these dynamic optimization issues and reinforcement learning, specifically addressing how to employ reinforcement learning techniques to formulate and solve problems of intertemporal utility maximization. Ye et al. [2020] proposed an innovative state-augmented reinforcement learning framework, which has been empirically tested in the Bitcoin market and Big Tech stocks, yielding performance that surpasses baselines. Betancourt and Chen [2021] developed a novel portfolio management approach, employing deep reinforcement learning in markets characterized by dynamic asset quantities. This methodology was rigorously tested on a dataset from one of the world's largest cryptocurrency markets, where it outperformed state-of-the-art methods, achieving an impressive daily average return exceeding 24%. In the realm of options trading, Buehler et al. [2019] introduced a framework for hedging derivative portfolios using cutting-edge deep reinforcement machine learning techniques under market frictions such as transaction costs, liquidity constraints, or risk limitations.

In this study, we conducted a comparative analysis of two distinct portfolio optimization approaches:

- Four traditional methods based on Markowitz's Portfolio Theory, namely Tangency Portfolio, Minimum Variance Portfolio, Risk Parity, Equal Weight, along with a Hierarchical Risk Parity approach that integrates risk parity with machine learning techniques.
- Deep reinforcement learning algorithms, including A2C, PPO, DDPG, SAC, and TD3.

Our findings reveal that the deep reinforcement learning algorithms significantly outperformed the traditional methods. A notable advantage of the reinforcement learning approach is its ability to learn based on a custom, domain-specific reward function. This contrasts with the standard metrics such as prediction accuracy used by supervised learning methods, which often overlook the intricacies of the problem at hand. For instance, reinforcement learning can control portfolio turnover or mitigate excessive trading prompted by noisy signals, which typically erode portfolio returns.

2 Methodology

2.1 Traditional Approach

The asset allocation conundrum seeks to refine the interplay between returns and risk. Markowitz's seminal concept of the efficient frontier is achieved through a dual objective of maximizing returns while concurrently minimizing risk, typically gauged as the portfolio's volatility. This volatility is mathematically represented by the standard deviation of the stochastic variable $w^T X$, where w denotes the vector of asset weights and X encapsulates the matrix of asset returns. The portfolio volatility is thus quantified by: $\sigma(w) = \sqrt{w^T \Sigma w}$, with Σ representing the covariance matrix of the asset returns.

First, the Minimum Variance Portfolio, as explored by [Clarke et al., 2006], which strives to attenuate variance by optimizing the weight vector w to minimize $w^T \Sigma w$.

$$\min_{\omega} \quad \frac{1}{2}\omega^{\mathrm{T}}\Sigma\omega \qquad s.t. \quad \omega^{\mathrm{T}} r = \mathbb{E}(r_p), \quad \omega^{\mathrm{T}} 1 = 1 \tag{1}$$

This equation embodies a critical financial principle: the minimization of the portfolio's variance, represented by $\frac{1}{2}\omega^{\mathrm{T}}\Sigma\omega$, subject to constraints ensuring that the portfolio's expected return, $\mathbb{E}\left(r_p\right)$, is met and the sum of the asset weights equals unity.

Second, the tangency portfolio optimizes asset allocation by maximizing the Sharpe ratio [Bailey and Lopez de Prado, 2012]. This metric quantifies the excess return per unit of volatility, or total risk, over the risk-free rate, offering investors a clearer perspective on their investment returns. The Sharpe ratio is expressed as:

$$Sharpe\,Ratio = \frac{\mathbb{E}(r_p) - r_f}{\sigma_p} \tag{2}$$

Here, r_f denotes the risk-free rate, typically represented by a theoretically secure investment. The optimization of the tangency portfolio is mathematically represented as:

$$\max_{\omega} \quad \frac{\omega^{\mathrm{T}} r - r_f}{\sqrt{\omega^{\mathrm{T}} \Sigma \omega}} \qquad s.t. \quad \omega^{\mathrm{T}} 1 = 1. \tag{3}$$

Third, risk parity posits that equalizing risk across asset allocations enhances the portfolio's Sharpe ratio, thereby fortifying it against market downturns [Maillard et al., 2010]. The crux of risk parity lies in balancing each asset's contribution to the total volatility of the portfolio.

The risk parity solution targets the resolution of a fixed point problem, expressed as $w_i = \frac{\sigma(w)^2}{(\Sigma w)_i N}$, where N represents the total number of assets in the portfolio. An alternative, yet equally rigorous formulation, seeks to identify the weight vector w that minimizes the following expression:

$$\sum_{i=1}^{N} \left[w_i - \frac{\sigma(w)^2}{\left(\sum w\right)_i N} \right]^2. \tag{4}$$

Forth, the hierarchical risk parity model represents a recent advancement within the traditional methodologies of portfolio optimization. In practice, the first, second, and third models traditionally addressed by Markowitz's Critical Line Algorithm (CLA) are subject to certain limitations [Niedermayer and Niedermayer, 2010]. These include the requirement for the inversion of the covariance matrix and the susceptibility of asset allocations to significant changes even with minor deviations in return predictions [Michaud and Michaud, 2007]. To address these challenges, De Prado [2016] introduced the Hierarchical Risk Parity model, employing clustering techniques from machine learning to uncover the hierarchical structure in asset correlations for more nuanced resource allocation. This model is methodically segmented into three stages: Hierarchical Tree Clustering, Matrix Seriation, and Recursive Bisection.

Step 1. Hierarchical Tree Clustering. This process begins with calculating the correlation matrix ρ among N assets, leading to a distance matrix D defined as

$$D(i,j) = \sqrt{0.5 \times (1 - \rho(i,j))}. \tag{5}$$

A modified distance matrix $\widehat{D}$ is then computed, where

$$\widehat{D}(i,j) = \sqrt{\sum_{k=1}^{N} (D(k,i) - D(k,j))^2}. \tag{6}$$

Asset clusters are formed by identifying the minimal distances in $\widehat{D}$, continuously merging assets and recalculating distances until a single cluster remains.

Step 2. Matrix Seriation: This step rearranges the data to position larger covariances closer to the matrix's diagonal, producing a quasi-diagonal covariance matrix, enhancing interpretability and simplifying subsequent analyses.

Step 3. Recursive Bisection: Assigns weights to assets based on the previously formed hierarchical structure. Initial weights are set uniformly $(w_i = 1)$, and for each cluster, weights are recalculated as

$$w_i = \frac{diag\,(V_i)^{-1}}{trace\left(diag\,(V_i)^{-1}\right)}. \tag{7}$$

Variances for each cluster node are computed $\left(\sigma_1 = w_1^T V_1 w_1, \sigma_2 = w_2^T V_2 w_2\right)$, followed by weight rescaling $(w_i = \alpha_i w_i)$ based on the calculated variances.

2.2 Deep Reinforcement Learning for Asset Allocation

The trading process can be elegantly conceptualized as a Reinforcement Learning (RL) process. See Figure 1. Within this framework, an agent iteratively interacts with an environment, basing its actions on observed environmental states and receiving rewards contingent on the states encountered and actions executed. In asset trading, the environmental state corresponds to the recent history of the assets. Actions comprise the trading decisions to divest from certain assets and acquire others, while rewards are quantified as scalar functions reflecting the agent's financial gains or losses resulting from these transactions. The portfolio, a vector detailing the assets held by the agent at any given time, thus becomes central to this process, also known as portfolio management.

Portfolio management can be adeptly modeled as a deep reinforcement learning (DRL) problem. Here, deep neural networks are employed to optimize the risk-adjusted returns of a portfolio comprising a specific set of assets. For effective convergence, these neural networks are tasked with deciphering market dynamics during asset reallocation. The DRL framework for portfolio management encompasses several components:

- Agent: A deep neural network, designed to discover an optimal function—policy- or value-based—that learns actions maximizing rewards.
- Actions: Outputs from the network for a specific period, representing the final portfolio weights, i.e., the proportion of each asset in the portfolio.
- State: Inputs for the neural network, encapsulating the current stock market scenario through various financial indicators. These include a tensor of features such as the current portfolio balance, stock prices, and owned assets.
- Environment: The stock market serves as the environment, responding to the agent's actions and providing feedback in the form of rewards.
- Reward: The reward is calculated as the differential returns, the difference between the current and previous portfolio values.

In DRL, the agent typically employs Stochastic Gradient Descent (SGD), based on a snapshot of the market (state) and the reward (cost-adjusted returns), to learn the action (portfolio weights) that optimizes portfolio allocation at a specific point in time. Figure 9 illustrates the interplay among the various elements of the DRL framework.

Our primary methodology encompasses the use of Actor-Critic Optimization in reinforcement learning [La and Ghavamzadeh, 2013]. This term denotes a dual-structured approach, amalgamating both policy-based and value-based strategies. The 'actor' in this context signifies a policy update mechanism, aptly named for its role in guiding actions in accordance with the evolving policy. Conversely, the 'critic' represents a value update phase, critically appraising the actor's performance by assessing the values associated with its actions. Furthermore, from a broader perspective, actor-critic methods are subsumed under the umbrella of policy gradient algorithms, highlighting their integral role in refining policy-based decision-making processes.

In our research, we employ a quintet of algorithms, namely A2C, DDPG, PPO, SAC, and TD3, each with its unique characteristics and applications. See Figure 2. The Proximal Policy Optimization (PPO) algorithm, currently a leading algorithm in deep reinforcement learning (DRL), has demonstrated remarkable success in both discrete and continuous control environments, notably in OpenAI Five [Schulman et al., 2017]. However, as an on-policy algorithm, PPO grapples with severe sample inefficiency, necessitating substantial data sampling for effective learning, a constraint impractical for real-world robotic training.

The Advantage Actor-Critic (A2C) algorithm represents a variation of the AC method, utilizing estimates of the advantage function for bootstrapping - updating values based on estimates rather than exact values [Babaeizadeh et al., 2016]. This algorithm introduces a baseline to diminish estimation variance, where the advantage function evaluates the relative merit of an action in comparison to the average action for a given state. Such an approach effectively reduces the variance between old and new policies, thereby enhancing the stability of the reinforcement learning algorithm.

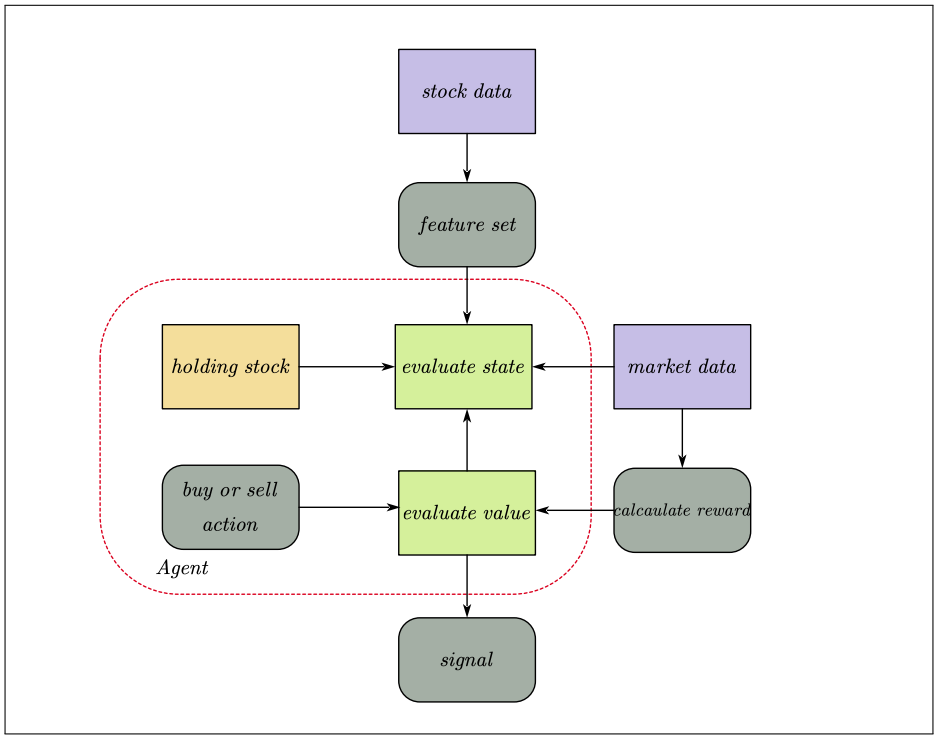

Figure 1: A Transaction Framework Based on Reinforcement Learning

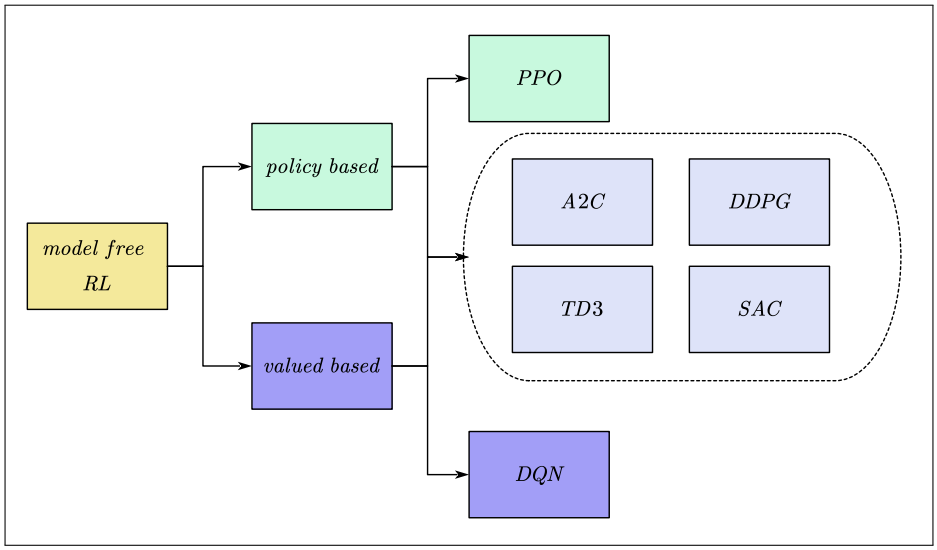

Figure 2: Taxonomy of the reinforcement learning algorithms related to this work

Deep Deterministic Policy Gradient (DDPG), developed by Lillicrap et al. [2015], merges insights from Deterministic Policy Gradient (DPG) and Deep Q-Network (DQN). It leverages a critic learning from a temporal loss and an actor learning via policy gradient, distinguishing itself as an off-policy method capable of sampling from extensive experience buffers for improved sample efficiency during training.

While DDPG can yield impressive results, it often exhibits sensitivity to hyperparameters and requires meticulous fine-tuning. A notable drawback of DDPG is its tendency to overestimate the q_π values of the critic network, leading to potential local optima entrapment or catastrophic forgetting. The Twin Delayed DDPG (TD3) algorithm, introduced by Fujimoto et al. [2018], addresses these challenges by implementing three innovative modifications: the use of dual critic networks, delayed actor updates, and the incorporation of noise to regulate the target action.

The Soft Actor Critic (SAC) algorithm integrates the maximization of entropy learning with the Actor-Critic framework in an off-policy reinforcement learning setting [Haarnoja et al., 2018a]. Traditional reinforcement learning algorithms often become increasingly deterministic during the learning process, weakening their exploratory capabilities and converging prematurely to local optima. SAC, however, aims not only to maximize environmental rewards but also to enhance the entropy of policies, thereby bolstering its exploratory efficiency and demonstrating superior performance across various tasks [Haarnoja et al., 2018b].

3 Experiments

3.1 Data

We utilized the constituent stocks of the Dow Jones Industrial Average, excluding three stocks that were either temporarily suspended or listed later, resulting in a total of 27 stocks under analysis. See Table 1. The research period spanned from January 1, 2010, to December 31, 2022. This timeframe was divided into two distinct sets: the training set, which encompassed data from January 1, 2010, to December 31, 2018, and the testing set, which included data from January 1, 2019, to December 31, 2022.

Table 1: List of Stocks Researched

AXP.N	BA.N	CAT.N	CRM.N	CVX.N
DIS.N	GS.N	HD.N	IBM.N	JNJ.N
JPM.N	KO.N	MCD.N	MMM.N	MRK.N
NKE.N	PG.N	TRV.N	UNH.N	V.N
VZ.N	WMT.N	AAPL.O	AMGN.O	CSCO.O
INTC.O	MSFT.O			

3.2 Settings

3.2.1 Financial Settings

It is important to note that for traditional methodologies, which do not involve a learning component, the division into training and testing sets is not applicable. In these cases, we employ a window size of 50, and no transaction costs are considered. Conversely, for the Deep Reinforcement Learning (DRL)-based methods, a window size of 1 is utilized, and transaction costs are set at 0.1% of the total value for each trade.

Our evaluation encompasses ten distinct algorithms: the Tangency Portfolio, Minimum Variance Portfolio, Risk Parity, Hierarchical Risk Parity, Equal Weight Portfolio, A2C, PPO, DDPG, SAC, and TD3. The first five strategies are deterministic in nature, whereas the latter five, grounded in the Actor-Critic algorithm framework, incorporate a stochastic element due to the weight initialization process. Therefore, to accurately capture performance variability and establish uncertainty boundaries, we report results from 10 independent runs for each of these stochastic approaches. This methodology ensures a comprehensive and nuanced assessment of each algorithm's performance under varying market conditions. The Hierarchical Risk Parity model necessitates the application of machine learning, with its hyperparameters outlined in Table 2.

Table 2: Model Hyperparameters of Hierarchical Risk Parity

Hyperparameter-Model	HRP
Risk measure	CDaR
Codependence	Pearson
Linkage	average
Max clusters	-
Optimal leaf order	True

3.2.2 Reinforcement Learning Framework Settings

In our empirical analysis, the closing price of a stock, recorded at the close of trading sessions, serves as a preliminary data point. However, this nominal closing price may not always accurately reflect the stock's intrinsic value, given its exclusion of critical market factors. Consequently, we utilize the 'adjusted close price', which incorporates adjustments for dividends, stock splits, and new stock issuances, to provide a more precise representation of the stock's value.

The agent, represented by a neural network with parameters varying according to the specific algorithm, acts based on the state as depicted by the input tensor. This approach is integral to our experimental setup.

Diverging from the conventional supervised learning paradigms, where reward functions are typically generic—such as prediction accuracy or mean square error—the reinforcement learning framework allows for a more comprehensive and contextually grounded reward function.

The focal point of our reward maximization strategy is the average logarithmic cumulative return. Mathematically, this is denoted as:

$$R^T = 1/T * \log\left(V^T/V^0\right) \tag{8}$$

Equivalently, this can be expressed as:

$$R^T = 1/T * \sum_{t=0}^{T} (r_t) \tag{9}$$

Here, R^T represents the reward function at time T, V^T denotes the portfolio value at time T, and r^t is the logarithmic return of the portfolio at time t. This reward function uniquely accounts for each episodic return r^t and remains agnostic to the duration T over which it is calculated, thereby ensuring an equitable weighting of each episode. This characteristic renders the reward function holistically inclusive, embracing both short-term and long-term returns in its scope.

The hyperparameters for the five algorithms are delineated in the Table 3, Table 4, Table 5, Table6, Table 7 below.

3.3 Results

3.3.1 Traditional Models

As depicted in Figure 3 and Table 8, the equal-weight portfolio exhibited superior performance on the test set, yielding the highest annualized return and the most favorable Sharpe ratio.

3.3.2 Deep Reinforce Learning Models

As illustrated in Figure 4 and Table 9, within the five reinforcement learning algorithm models, the TD3 model achieved the highest annualized return, while the DDPG model secured the most favorable Sharpe ratio.

Table 3: Hyperparameters for the PPO model

Hyperparameter	PPO
Learning rate	0.0003
Number of steps	10000
Batch size	256
Number of epochs	10
Discount factor (gamma)	0.99
GAE lambda	0.95
Clipping range	0.2
Clipping range for VF (if specified)	None
Normalize advantage	True
Entropy coefficient (ent_coef)	0.0
Value function coefficient (vf_coef)	0.5
Maximum gradient norm	0.5

Table 4: Hyperparameters for the A2C model

Hyperparameter	A2C
Learning rate	0.0003
Number of steps	10000
Discount factor (gamma)	0.99
GAE lambda	1.0
Entropy coefficient (ent_coef)	0.0
Value function coefficient (vf_coef)	0.5
Maximum gradient norm	0.5
RMSProp epsilon	1e-05
Use RMSProp	True
Use SDE	False
SDE sample frequency	-1
Normalize advantage	False

Table 5: Hyperparameters for the TD3 model

Hyperparameter	TD3
Learning rate	0.0003
Buffer size	1,000,000
Learning starts	100
Batch size	256
Tau	0.005
Discount factor (gamma)	0.99
Train frequency	(1, 'episode')
Gradient steps	-1
Action noise	None
Replay buffer class	None
Replay buffer kwargs	None
Optimize memory usage	False
Policy delay	2

Table 6: Hyperparameters for the SAC model

Hyperparameter	SAC
Learning rate	0.0003
Buffer size	1,000,000
Learning starts	100
Batch size	256
Tau	0.005
Discount factor (gamma)	0.99
Train frequency	1
Gradient steps	1
Action noise	None
Optimize memory usage	False
Target update interval	1
SDE sample frequency	-1
Use SDE at warmup	False

Table 7: Hyperparameters for the DDPG model

Hyperparameter	DDPG
Learning rate	0.0003
Buffer size	1,000,000
Learning starts	100
Batch size	256
Tau	0.005
Discount factor (gamma)	0.99
Train frequency	(1, 'episode')
Gradient steps	-1

Table 8: Performance Metrics of Traditional Models

Metric	min variance	min volatility	risk parity	equal weight	hierarchical risk parity
Annual return	2.70%	5.00%	9.20%	10.70%	10.30%
Cumulative returns	11.40%	21.30%	42.20%	50.30%	48.00%
Annual volatility	26.40%	16.80%	19.10%	22.10%	19.80%
Sharpe ratio	0.24	0.37	0.56	0.57	0.60
Calmar ratio	0.09	0.21	0.35	0.32	0.34
Stability	0.09	0.52	0.61	0.67	0.74
Max drawdown	-32.00%	-23.40%	-26.50%	-33.50%	-30.00%
Omega ratio	1.05	1.08	1.12	1.12	1.13
Sortino ratio	0.31	0.52	0.79	0.81	0.84
Skew	-2.45	-0.09	-0.36	-0.35	-0.35
Kurtosis	41.47	11.35	13.75	16.68	18.07
Tail ratio	0.99	0.94	0.88	0.89	0.86
Daily value at risk	-3.30%	-2.10%	-2.40%	-2.70%	-2.40%

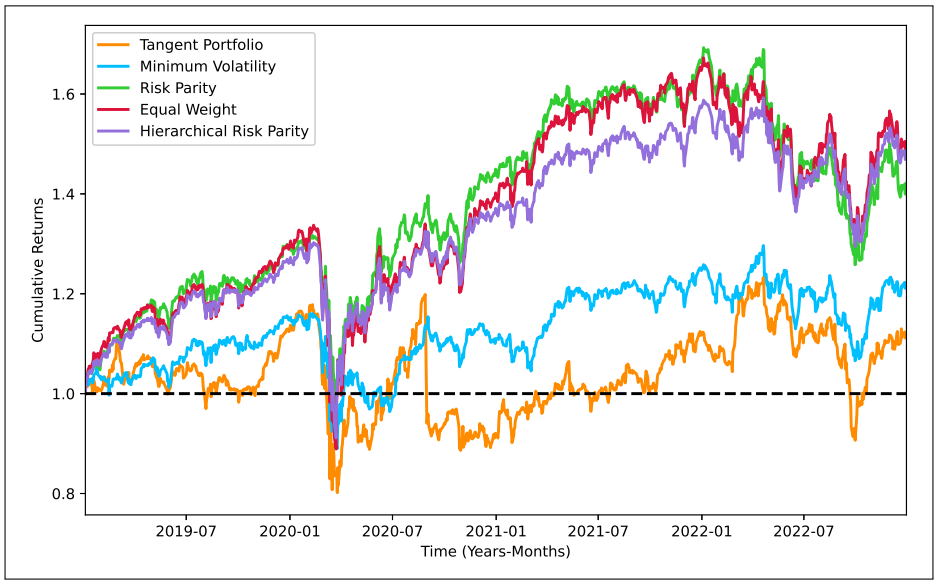

Figure 3: Returns of the traditional strategies

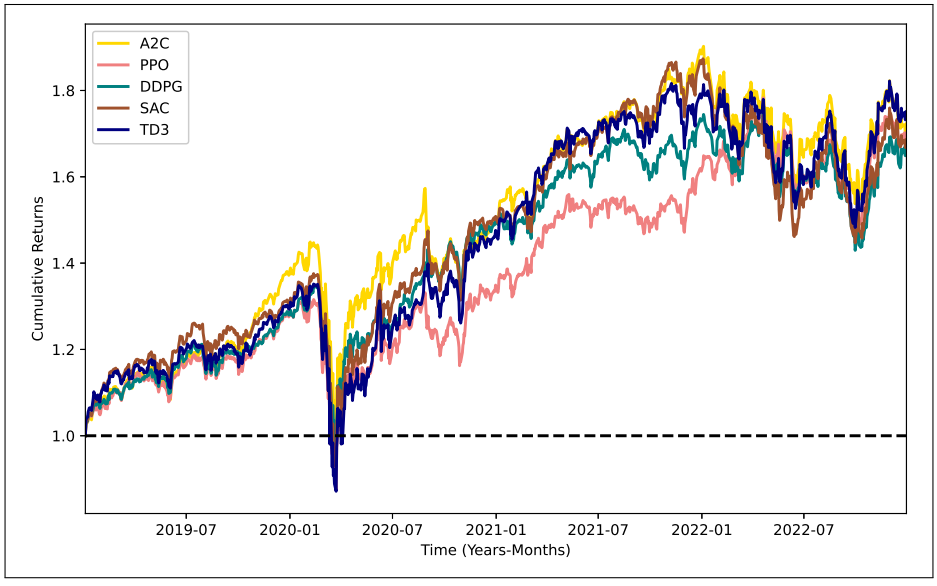

Figure 4: Returns of the DRL models

Table 9: Performance Metrics of DRL Models

Metric	PPO	A2C	DDPG	SAC	TD3
Annual return	14.10%	14.50%	13.50%	13.90%	15.00%
Cumulative returns	69.40%	72.00%	66.00%	68.40%	74.80%
Annual volatility	21.50%	22.30%	19.80%	22.90%	23.80%
Sharpe ratio	0.72	0.72	0.74	0.68	0.71
Calmar ratio	0.42	0.48	0.45	0.43	0.42
Stability	0.87	0.83	0.82	0.74	0.78
Max drawdown	-33.60%	-30.10%	-29.80%	-32.50%	-35.50%
Omega ratio	1.16	1.16	1.16	1.15	1.15
Sortino ratio	1.02	1.02	1.05	0.96	0.99
Skew	-0.15	-0.33	-0.3	-0.5	-0.52
Kurtosis	15.61	16.33	14.21	17.62	17.35
Tail ratio	0.9	0.9	0.93	0.98	0.94
Daily value at risk	-2.70%	-2.80%	-2.40%	-2.80%	-2.90%

4 Conclusion

In this research, we delved into the potential of leveraging optimization algorithms for asset allocation tasks. We conducted an extensive benchmark study encompassing ten varied algorithms: Tangency Portfolio, Minimum Variance Portfolio, Risk Parity, Equal Weight, Hierarchical Risk Parity, A2C, PPO, DDPG, SAC, and TD3. See Figure 5.

The four traditional methods based on Markowitz's Portfolio Theory—Tangency Portfolio, Minimum Variance Portfolio, Risk Parity, Equal Weight—require no fitting or optimization process (training) due to their non-reliance on learnable parameters. Hierarchical Risk Parity, however, integrates machine learning clustering. These models exhibited consistent outcomes in our experiments. Notably, the Equal Weight Portfolio consistently outperformed others by delivering the highest annual and cumulative returns, alongside optimal Sharpe and Calmar ratios. Despite their efficacy in stable market conditions, these traditional strategies are sensitive to outliers and abrupt market shifts.

The five deep reinforcement learning algorithms we tested markedly surpassed the traditional methods in performance. It can be asserted that the deep reinforcement learning framework adeptly handles time-series forecasting and portfolio allocation, inclusive of transaction costs. Remarkably, with only the time series of assets as input, deep reinforcement learning efficiently managed returns and risks without explicitly featuring risk or considering it within the reward function, demonstrating the flexibility of reinforcement and deep learning methods in handling nearly model-free tasks. These algorithms also adeptly navigated the complex dynamics of financial time series. Future research will explore diverse reward functions, including the impact of risk on these functions—a natural progression—along with market effects and the framework's capacity to learn diversified and heterogeneous risks.

Our focus will extend to the interpretability of different reinforcement learning approaches vis-à-vis traditional methods, particularly in utilizing exogenous factors within state spaces and time series.

Future endeavors should aim to optimize hyperparameters, including network architecture. A natural progression for the current system would be to incorporate an additional layer of decision-making and data aggregation to simultaneously manage multiple markets. The integration of additional data sources, such as social media sentiments and different price features.

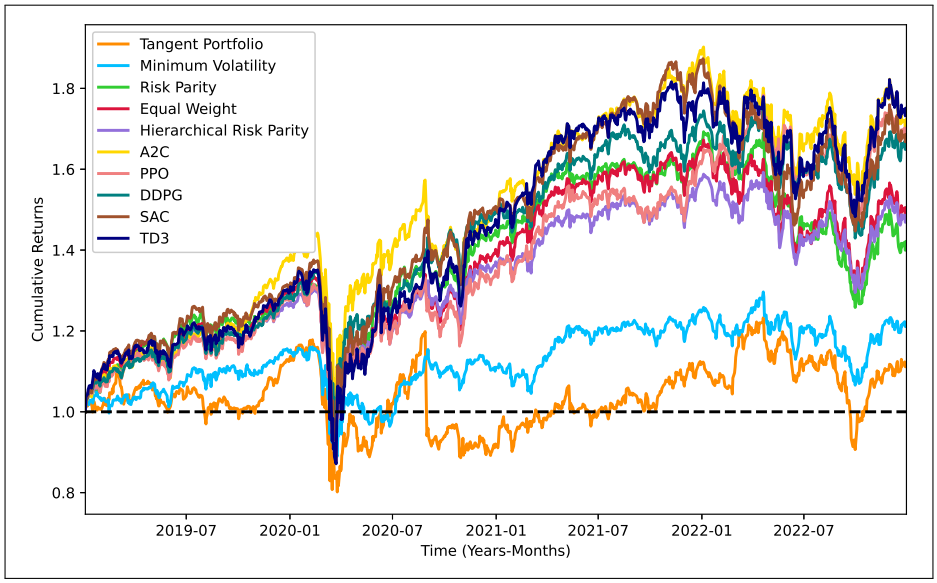

Figure 5: Return of 10 models

Reference

M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz. Reinforcement learning through asynchronous advantage actor-critic on a gpu. *arXiv preprint arXiv:1611.06256*, 2016.

D. H. Bailey and M. Lopez de Prado. The sharpe ratio efficient frontier. *Journal of Risk*, 15(2):13, 2012.

C. Betancourt and W.-H. Chen. Deep reinforcement learning for portfolio management of markets with a dynamic number of assets. *Expert Systems with Applications*, 164:114002, 2021.

F. Black and R. Litterman. Global portfolio optimization. *Financial analysts journal*, 48(5):28–43, 1992.

H. Buehler, L. Gonon, J. Teichmann, and B. Wood. Deep hedging. *Quantitative Finance*, 19(8): 1271–1291, 2019.

J. Chai, H. Zeng, A. Li, and E. W. Ngai. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134, 2021.

R. Clarke, H. De Silva, and S. Thorley. Minimum-variance portfolios in the us equity market. *Journal of Portfolio Management*, 33(1):10, 2006.

R. Clarke, H. De Silva, and S. Thorley. Minimum-variance portfolio composition. *The Journal of Portfolio Management*, 37(2):31–45, 2011.

M. L. De Prado. Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, 42(4):59–69, 2016.

V. DeMiguel, L. Garlappi, and R. Uppal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The review of Financial studies*, 22(5):1915–1953, 2009.

A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher. Deep learning-enabled medical computer vision. *NPJ digital medicine*, 4(1):5, 2021.

S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018a.

T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

P. N. Kolm and G. Ritter. Modern perspectives on reinforcement learning in finance. *Modern Perspectives on Reinforcement Learning in Finance (September 6, 2019). The Journal of Machine Learning in Finance*, 1(1), 2020.

P. La and M. Ghavamzadeh. Actor-critic algorithms for risk-sensitive mdps. *Advances in neural information processing systems*, 26, 2013.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

S. Maillard, T. Roncalli, and J. Teïletche. The properties of equally weighted risk contribution portfolios. *The Journal of Portfolio Management*, 36(4):60–70, 2010.

R. Malladi and F. J. Fabozzi. Equal-weighted strategy: Why it outperforms value-weighted strategies? theory and evidence. *Journal of Asset Management*, 18:188–208, 2017.

H. M. Markowitz. Portfolio selection. *The Journal of Finance*, 7:77–91, 1952.

R. O. Michaud and R. Michaud. Estimation error and portfolio optimization: a resampling solution. *Available at SSRN 2658657*, 2007.

A. Niedermayer and D. Niedermayer. Applying markowitz's critical line algorithm. In *Handbook of portfolio construction*, pages 383–400. Springer, 2010.

T. Roncalli and G. Weisang. Risk parity portfolios with risk factors. *Quantitative Finance*, 16(3): 377–388, 2016.

M. Rubinstein. Markowitz's" portfolio selection": A fifty-year retrospective. *The Journal of finance*, 57(3):1041–1045, 2002.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

V. Sorin, Y. Barash, E. Konen, and E. Klang. Deep learning for natural language processing in radiology—fundamentals and a systematic review. *Journal of the American College of Radiology*, 17(5):639–648, 2020.

D. Stefanovits. Equal contributions to risk and portfolio construction. *ETH Zurich, ethz. ch/content/dam/ethz/special-interest/math/risklab-dam/documents/walter-saxerpreis/ma-stefanovits. pdf.[Accessed April 11]*, 2010.

A. Vinzelberg and B. R. Auer. A comparison of minimum variance and maximum sharpe ratio portfolios for mainstream investors. *The Journal of Risk Finance*, 23(1):55–84, 2022.

Y. Ye, H. Pei, B. Wang, P.-Y. Chen, Y. Zhu, J. Xiao, and B. Li. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1112–1119, 2020.

Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller. Deep learning for environmentally robust speech recognition: An overview of recent developments. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(5):1–28, 2018.

Appendix

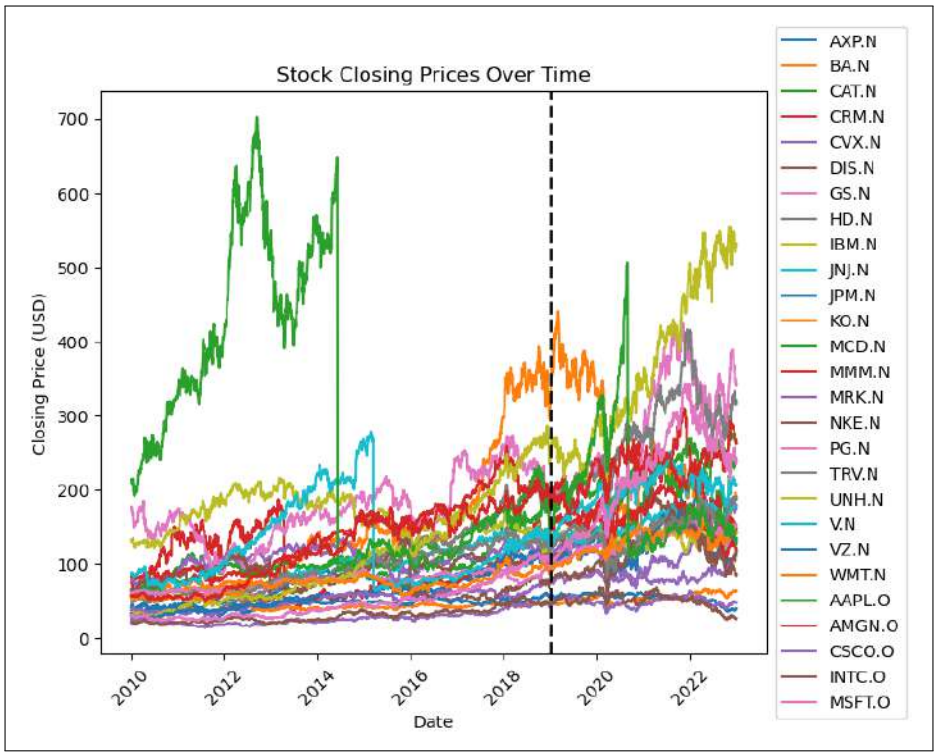

Figure 6: Data set

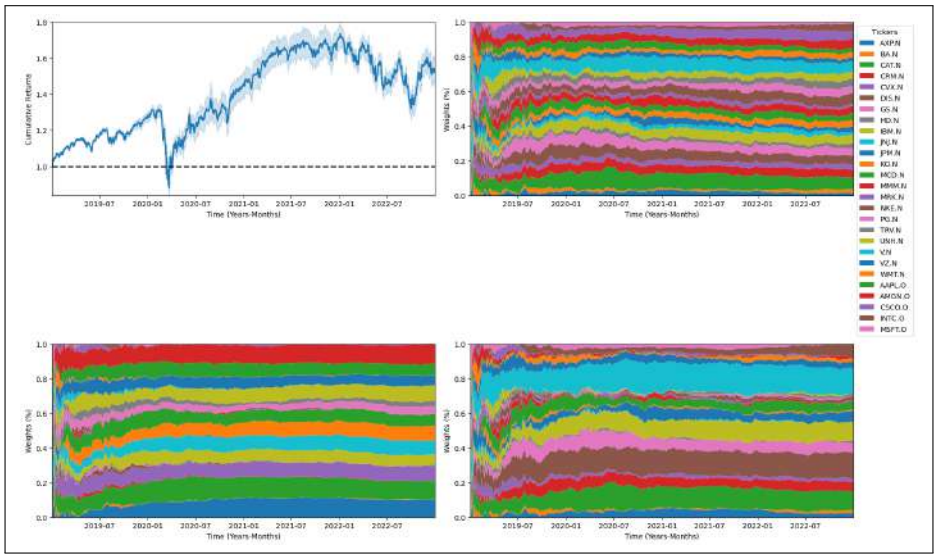

Figure 7: Result of PPO

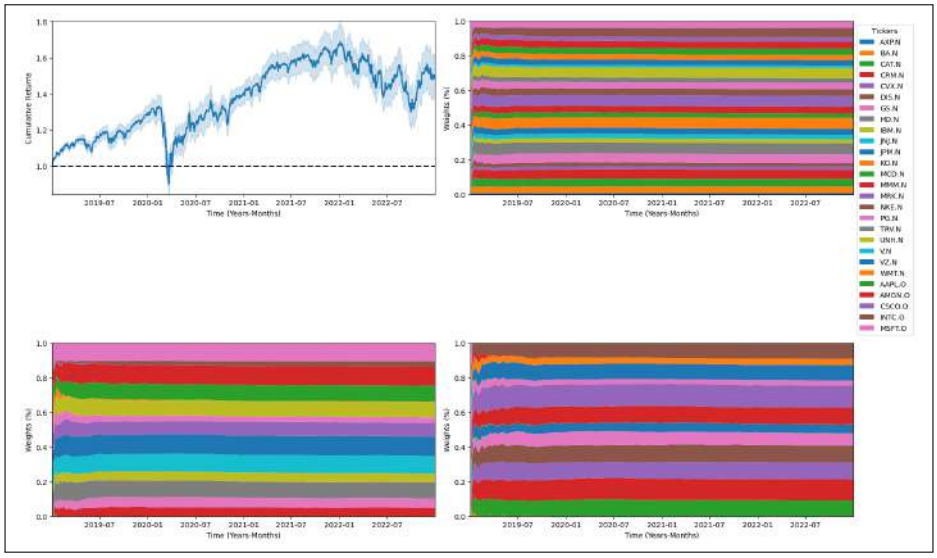

Figure 8: Result of A2C

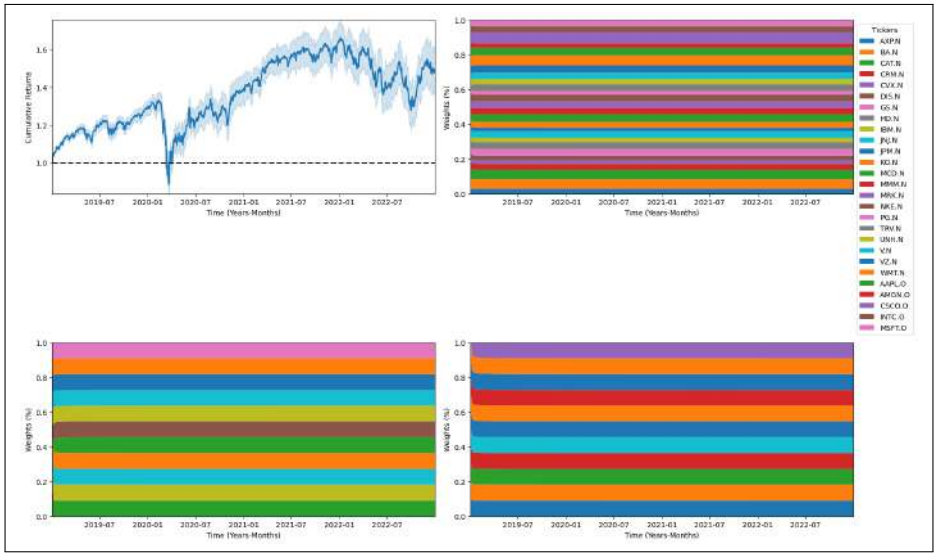

Figure 9: Result of DDPG

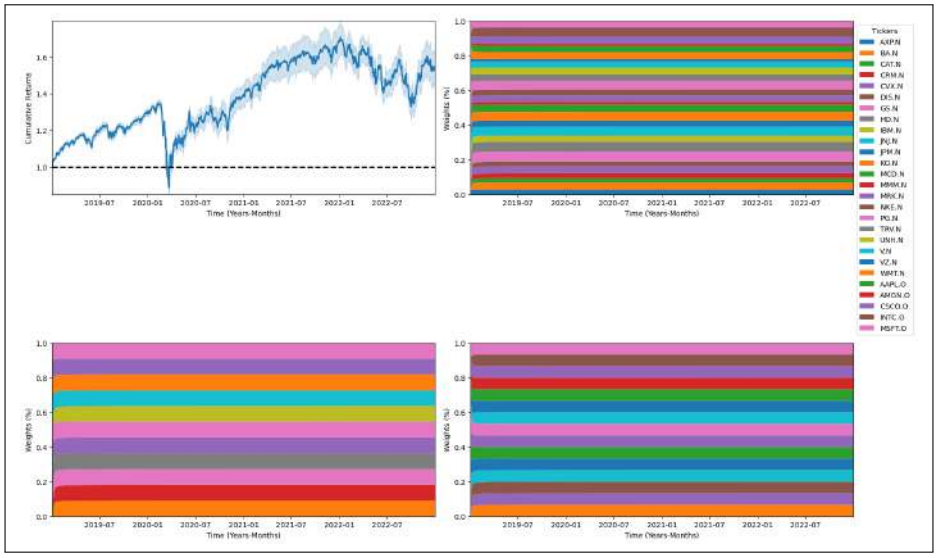

Figure 10: Result of SAC

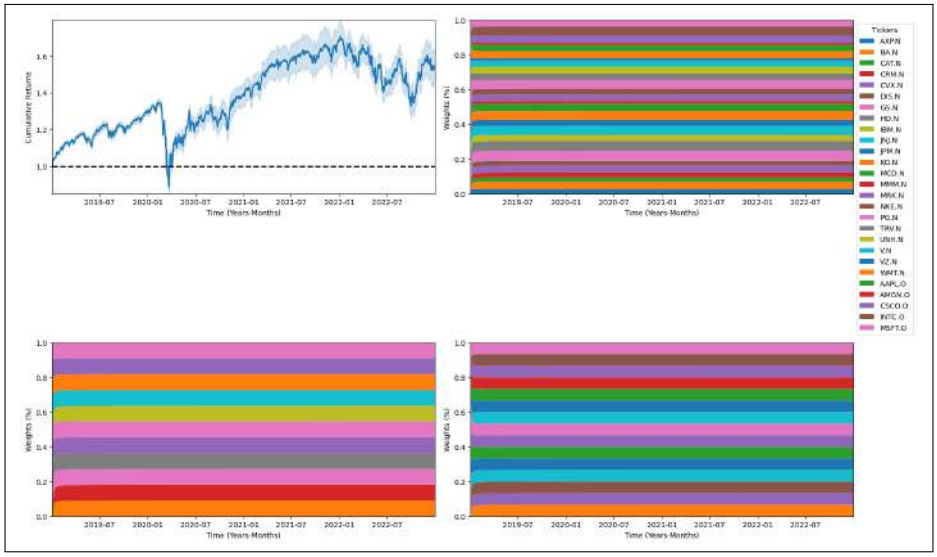

Figure 11: Result of TD3