Portfolio Optimization using Reinforcement Learning

A Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

by

Ishwar Govind
(111901024)

and

Jerry John Thomas
(111901055)

COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

CERTIFICATE

*This is to certify that the work contained in the project entitled "**Portfolio Optimization using Reinforcement Learning**" is a bonafide work of **Ishwar Govind (Roll No. 111901024)** and **Jerry John Thomas (Roll No. 111901055)**, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Palakkad under my guidance and that it has not been submitted elsewhere for a degree.*

Dr. Chandrashekar Lakshminarayan

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Madras

Acknowledgements

We would like to express our sincere gratitude to our mentor Dr. Chandrashekar Lakshminarayan for providing his invaluable guidance, comments and suggestions throughout the course of the project. We would also like to thank Dr. Albert Sunny for his advice and support.

Contents

Chapter 1

Introduction

1.1 Background

With the help of artificial intelligence, banking and finance industries have new ways to satisfy client needs for smarter, safer, and more practical ways to access, spend, save, and invest money. Today Machine Learning is used extensively in the field of finance for algorithmic trading, fraud detection and prevention, credit score calculation and many other use cases. With the breakthroughs in AI, these approaches have a better performance than traditionally employed methods.

We are exploring the problem of Portfolio Optimisation which involves dynamic allocation of funds into different stocks assets optimally to get the best financial outcome. To maximize the returns on the stock we have to estimate the returns on the stock, often this comes out to be quite a challenge in a complex and dynamic market where over reliance on the predictive models could lead to bad outcomes. Errors in this estimation could lead to traditional naive approaches outperforming ML optimized strategies[4]. Feature extraction and trading strategy design are other important aspects in this problem.

Although many approaches have been proposed in this field, most of them have too many assumptions that prevent these methods from being used in the real market. Some of these unrealistic assumptions are around the financial signals' second-order and higher-

order statistical moments[5]. Models are usually limited to discrete action spaces[6]. Some models also do not consider the transaction and other costs involved in trading. Methods with reduced assumptions tend to have better performance in the real market and we try to tackle this.

Portfolio optimization can be modeled as a Markov Decision Process (MDP) and Reinforcement Learning can be used on it. Due to the large state spaces involved in dealing with the stock market, dynamic programming cannot be used because it has limited scalability, hence RL is a better suited candidate for tackling the unpredictive market and dynamically changing the strategy.

1.2 Problem Definition

This is an optimisation problem where we try to maximize the portfolio returns (risk adjusted) for our portfolio. We have an initial capital that we could invest into the market having m assets. After the initial allocation we wait for the market's response and reallocate the portfolio to have a better prospect of obtaining the maximum returns. We continue this process till the end of a fixed time period. We are also allowed to have cash (wealth not in stocks) also in this problem. The strategy needs to optimize for both bear and bull market scenarios.

We have some assumptions about the financial market [3][**?**]

1. Zero Slippage (orders executed at the expected price)
2. Possible to trade at the market at any time
3. Our transactions do not to affect the market price of the assets

This problem can be modeled as a Markov Decision Process (MDP) and RL could be employed to solve this.

State s consists of $[b_t, p_t, h_t,$ Market Indicators$_t]$. B_t refers to the balance at time t. P_t refers to the closing prices (adjusted) of each stock at time t. h_t refers to the shares owned

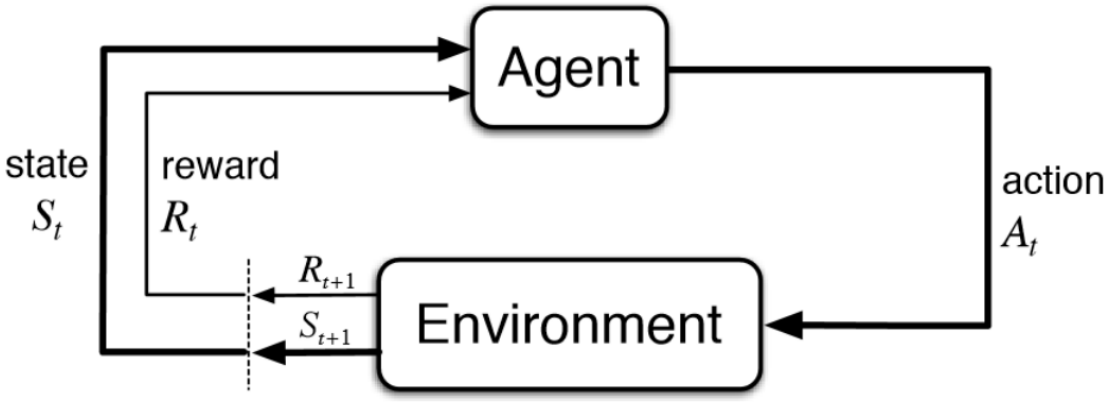

Fig. 1.1 Components of a RL system [7]

of each stock. Market Indicators[3] include the Relative Strength Index (RSI), Commodity Channel Index (CCI), Average Directional Index (ADX) etc. All these indicators could be calculated with all or combinations of high, low and close price over a period of a time.

Action a consists of a vector of actions over each stock as in how much to sell or buy each stock. Not selling nor buying results in holding of the stock

Reward r aims to quantise the change in portfolio value's when we do an action a from state s to reach a new state s'. Log returns (weighted sum of log-returns for the portfolio) and differential Sharpe (instantaneous risk-adjusted Sharpe ratio) are two methods used for estimating rewards.

1.3 Aim and Objectives

1. Compare the performance of different Reinforcement Learning and traditional algorithms in a financial market.

2. The goal is to design a trading strategy that maximizes the positive cumulative change of the portfolio value in the dynamic environment

Chapter 2

History of Portfolio Optimisation

In this section, we investigate the origin and evolution of the Portfolio theory. Portfolio refers to a collection of financial assets. This allocation should consider that any of these stocks could result in a negative return. There is always a risk associated with a stock, the risk-return tradeoff [8] is an investment principle that indicates that the higher the risk, the higher the potential reward [7]. Portfolio Optimisation refers to constructing and optimizing an objective function taking into consideration the investor's preferences [6]. One naive method of allocation would be to always do an equal weight allocation.

2.1 Markowitz Model

The Markowitz model (1952) [9] is one of the first attempts to come up with a strategy. We have a portfolio vector w and a market of M assets. At each point, the Markowitz model gives us the optimal portfolio vector w* which minimizes the risk for a given return level.

$$\sum_{i=1}^{M} w_{*,i} = 1, w_x \in R$$

It assumes that the investor is risk averse and thus wants to have the minimum risk for the given return or maximize the returns given some risk. This is a one time investment

i.e. we are only able to distribute our wealth into different stocks only at the start of the investment and cannot distribute it later on.

2.2 Modern Portfolio Theory

MPT is a mathematical framework based on the Markowitz model. This maximizes the expected return for a given risk. Diversification of the portfolio helps us to minimize the risk even if some of the stocks result in negative returns it will be counterbalanced by positive returns from the other stocks and the net effect could be positive. This is a better approach than having all the wealth invested in one stock which could go either way.

Another Principle is in how the risk is calculated. The risk is not calculated for each stock separately rather the risk for the whole portfolio is calculated. It uses the standard deviation of all returns to assess the risk of a specific portfolio. Efficient Frontier is another concept introduced here. In the risk toleration vs expected return plot, there is an optimal curve named the efficient frontier, any other point is suboptimal in the risk-reward framework i.e. risk is more for a given reward or reward is less for a given risk.

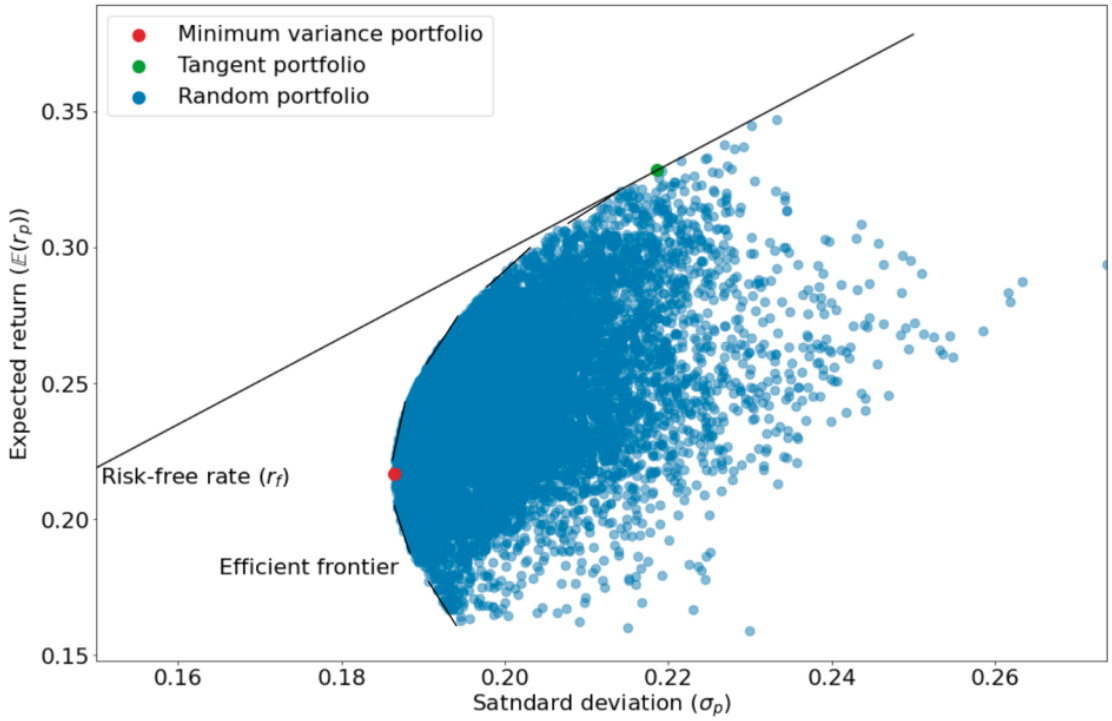

Fig. 2.1 different types of portfolios based on Markowitz model[3]

2.3 Post-modern Portfolio Theory

Post-modern Portfolio Theory is an extension to (Modern Portfolio Theory) MPT based on the downside risk of returns rather than the expected returns. MPT is based on symmetrical risk i.e. the amount of profit will be more or less similar to the amount of loss. PMPT is based on asymmetrical risk i.e. the profit and loss are not similar to one another. The downside risk is quantified by target semi-deviation. The key point here is that it also takes into consideration negative returns, something which the investors are very anxious about.

Chapter 3

Machine Learning in Finance

Not only has machine learning taken over Portfolio Optimization, but also several other fields in finance. Different models have been employed for different classes of problems. Some of the major areas where ML is used extensively in finance are listed below.

For tasks such as fraud detection, risk assessment, credit score prediction, policy monitoring, tax evasion monitoring.

Chatbots and robot advisors in customer service, algorithmic trading, price prediction, market sentiment analysis, portfolio management, exchange rate prediction, market behavior prediction.

In data extraction, information exchange, document translation, and financial text mining.

Since finance has a lot of data, these algorithms can analyse millions of data points, provide useful insights, and prevent human errors in the process. The finance sector, including banks, trading companies, and fin-tech companies, is rapidly implementing machine algorithms to automate tedious, time-consuming tasks and provide a much more streamlined and customised consumer experience.

Chapter 4

Reinforcement Learning in Finance

4.1 Reinforcement Learning

Reinforcement learning (RL) is a branch of machine learning that studies how agents should behave in a given environment in order to maximize the concept of cumulative reward. Here agents are virtual beings capable of making decisions based on observations made from the environment. The agent is not pre-programmed and learns based on previous experiences. Experience from the reward received based on action made in the environment.

4.2 Why Reinforcement Learning ?

The real world financial market is a very complex dynamic system. The system is stochastic in nature and keeps changing with each transaction that happens in the market. Many supervised deep machine learning algorithms have been tried out on the financial markets. Most of them are used to predict future stock price movements.

The performance of these algorithms are based on the accuracy of the price prediction. The prediction of stock prices are difficult. The actions of buying and selling a certain quantity of stock cannot be done using supervised learning. Although we could program a model to buy and sell based on price prediction. This does not make the model "intelligent" and

adaptive to market changes.[10]

Reinforcement Learning provides algorithms which can train agents capable of doing these actions by using "intelligence". Deep reinforcement learning can be used to create agents capable of beating the best human player in one of the most challenging classical board games, Go.[11]. Model-free deep RL was shown to be successful in its previous attempts at algorithmic trading.[12]. This is another reason why we are focusing more on Model-free RL techniques.

Reinforcement Learning algorithms can be divided into 2 sections

- Model Free RL
- Model Based RL

4.3 Model Based RL

Model based reinforcement learning involves the agent learning a model based on previous experiences from the environment. The main focus of the agent is on the model. Appropriate actions are chosen by searching and planning in the model.

Examples : World Models, I2A (Imagination-Augmented Agents) , (MBVE)Model-Based Value Estimation

4.4 Model Free RL

In Model free reinforcement learning, the agent uses experience rather than model to learn any of these functions (state function, action function, policy function). The state function determined by the policy gives the value of the expected total utility on choosing the state.

Examples : Q learning, DQN (Deep Q Network) ,PPO (Proximal Policy Optimization), Advantage Actor-Critic (A3C)

Model Based reinforcement learning has more sample efficiency than model-free. This is due to the fact that model-based learning tries to learn the environment rather than a policy or a utility function.

Some of the RL Algorithms we are planning to use are given below and explained briefly.

4.5 RL Algorithms

4.5.1 Q learning

The action-value function Q is learned in the method.

In the function $Q(s, a)$, 'Q' stands for quality. It is the expected future reward on taking action a at state s.

This is a model free RL algorithm. The standard q learning algorithm is limited to only discrete space and action spaces. The Q values are learned by choosing an action which gives the agent a reward, later used to update the Q learning table. The Q learning table contains all the $Q(s, a)$ values.

4.5.2 Policy Gradient

It is the most basic policy based reinforcement learning algorithm. The agent starts with a random initial policy.The agent makes actions based on the policy and the trajectory made from the policy action is stored. The rewards are computed based on the trajectory. The policy is updated to give higher probability to action that will lead to bigger rewards and lower probability to actions which will lead to lower rewards. The objective function is based on the policy and advantage function.

The advantage function $A(s, a)$ gives information regarding how much advantage of rewards the agent will receive if chosen action a than the expected reward for state s. The objective function is maximized by using gradient ascent.

4.5.3 Actor-Critic

Another type of Model free reinforcement learning method which uses policy gradient. It has 2 networks. The Actor network learns based on the policy gradient method. The critic network is used to evaluate the action produced by the actor using the value function predicted by the network. Both networks are simultaneously trained during each move the agent makes.

Chapter 5

Approach

5.1 Sharpe Ratio

Sharpe Ratio is an index used to measure the performance of a security or a portfolio by comparing it with a risk free asset. Sharpe Ratio is defined as :

$$S = \frac{R_p - R_f}{\sigma_p}$$

where, R_p is the return of the portfolio p, R_f is the return of Risk-free asset f and σ_p is the standard deviation of the portfolio p. The standard deviation gives us the statistical measure of a portfolio's market volatility.

Higher the Sharpe Ratio the more returns the security will provide relative to its risk. A Sharpe Ratio of 1 is considered to be acceptable and lesser than 1 are sub-optimal. Sharpe Ratios that are greater than 2 are considered to be good and superior.

5.1.1 Maximizing Sharpe Ratio

One of the best ways to make the best Portfolios are to choose weights in such a way that they maximize the Sharpe Ratio. The Sharpe Ratio of a Portfolio with n stocks are given

as :

$$S = \left(\frac{\sum_{i=1}^{n} w_i \cdot \mu_i - R_f}{\sqrt{\sum_i \sum_j w_i . w_j \cdot \sigma_{ij}}} \right)$$

Where μ_i is the mean return of the stock, and w_i is the weights given to each stocks and σ_{ij} is the covariance of stocks i and j.

One method was to find the weights that maximize Sharpe Ratio of the given portfolio during each time step and find the final cumulative return. The higher the cumulative return, the better our Agent (Sharpe MAX Agent) performed.

We tried to find the maximal Sharpe weights by converting it into a Lagrangian Equation.

$$L(w_1, w_2, ..., w_n, \lambda) = Sharpe(w_1, w_2, ..., w_n) + \lambda(\sum_{i=1}^{n} w_i - 1)$$

Using gradient descent we were able to find the weights that maximize the Sharpe ratio subject to the given weight constraints. The issue with Lagrangian equation here is that it was quasi-convex function.

Another method we tried for maximizing the Sharpe Function was to convert it into a convex minimization problem and solve it by minimizing the denominator as explained in *Optimization Methods in Finance*. [13]

5.2 Proximal Policy Optimization[1]

Proximal Policy Optimization is a policy gradient reinforcement learning method. PPO is an on policy algorithm (algorithms that learn policies which is used to select actions). PPO can also be used in environments with continuous action spaces.

Algorithm 1 PPO-Clip

Input: initial policy parameters θ_0, initial value function parameters ϕ_0

for $k = 0, 1, 2, \ldots$ **do**

Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.

Compute rewards-to-go $\hat{R}_t$.

Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.

Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \;\; g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

typically via stochastic gradient ascent with Adam.

Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

typically via some gradient descent algorithm.

end for

Fig. 5.1 PPO Algorithm [1]

For the initial method we created an environment where the reward was the return of assets after every Agent action. For the second we created an environment where the reward was Sharpe Ratio of the portfolio during each time step.

5.3 Environment

The initial setup of the environment had n stocks with maximum number of time steps from start date to end date (Provided by the user with the dataset). Currently the environment was setup only for daily stock price.(Only delivery trading). The environment has an initial balance which is set to 10000 USD. Maximum trade limit per stock which is set to 50. Transaction cost is 0.1%.

5.3.1 State Space

The state space has a dimension of $2n + 1$, where S_0 is the current balance available and S_1 till S_n are the prices of the stocks are the current time step. S_{n+1} till S_{2n} are the total number of shares owned for each stock.S_{n+j} is the number of shares owned for stock number j.

The values of each index range from $[0, \infty)$

5.3.2 Action Space

The Action Space has a dimension of n, where A_j is the action corresponding to stock j. The range of Action is $[-1, 1]$. Value of $x \in [-1, 1]$ indicates buy $x * MaxTradeLimit$ number of shares of a certain stock. If x is negative, it means to sell that much amount of certain stock.If the agent does not have enough money, then it will buy the maximum shares of stock it can. In the case of selling the agent can sell $\min(S_{n+j}, x * MaxTradeLimit)$ of stocks.

5.3.3 Reward

Basic Reward

$$R_t = Assets_t - Assets_{t-1}$$

$$Assets = S_0 + \sum_{i=1}^{n} S_i * S_{n+i}$$

We tried to use Sharpe Ratio of the portfolio as the reward for creating an agent that maximizes the Sharpe Ratio.

Chapter 6

Experiments and Observations

6.1 Market Analysis

6.1.1 Bull Market

When the economic conditions are favorable and the stock prices are rising, the market is said to be a bull market. Market conditions are also impacted by the Investor's attitudes and how they feel about the market trends. Bull Markets have a sustained period where the stock price rises. Investors feel safe when the market is showing booming trends. Economic conditions are strong, and employment is also high. This usually results in a buyer's market. [14]

6.1.2 Bear Market

When the market is on the decline, we say the market is in a bear market. We term it a bear market when it falls by more than 20% from the highs. Stock prices continuously fall, and unemployment rates increase. This also causes anxiety in the investors, contributing to the bear market. Since 1928, there have been 25 bear markets; fourteen (56%) have also been followed by recessions, while the other eleven (44%) have not. This results in a seller's market.[2]

6.1.3 Time Lines [2]

1. 2000 - 2002: The dot-com crash caused the SP 500 to crash by 36.8
2. 2007 - 2009: The economy goes into recession and enters the second-worst bear market
3. 2014 - 2016: Oil prices and other stocks were in a bear market
4. 2020: Due to the pandemic, the market entered a bear market.
5. 2022 - present: We are currently in a bear market.

6.2 Experiment

We intend to benchmark multiple allocation optimization strategies, including standard deterministic and Deep Reinforcement Learning methods. We conduct a thorough evaluation in which we examine the robustness and performance of such approaches for various market situations and different frequencies of reallocation.

6.2.1 Experiment Setting

When the exchange closes, it does so on the closing price, which may not be an accurate presentation of the stock value; hence we use the adjusted price, which considers the dividends, stock splits, and new stock issues. The dataset is split into training and testing with the 80-20 ratio split for the DRL methods. For traditional deterministic approaches, no split is used. We assume a small transaction cost for each transaction and we limit the total number of transactions on each stock. We try the following methods.

Traditional Methods

1. Max Sharpe
2. Minimum variance portfolio
3. Risk parity
4. Equal weight

Deep Reinforcement Learning Methods

1. A2C (Actor Critic method)
2. PPO (Proximal Policy Optimization)
3. DDPG (Deep Deterministic Policy Gradient)
4. SAC (Soft Actor Critic)
5. TD3 (Temporal Difference)

We do the DRL methods over many runs, and we report three types of results - mean of all the runs, maximum return, and minimum return of all the runs.

6.2.2 Experiment - 1

In this initial experiment we tried to test out our hypothesis on whether choosing the portfolio which maximises Sharpe Ratio show good results.

- iShares MSCI World ETF (URTH)
- Vanguard Total International Bond Index Fund ETF (BNDX)

The dataset we used for this was daily adjusted stock price data of BNDX and URTH from 06-2013 to 11-2022. Here BNDX is the risk-free asset and URTH the risky asset.

We compared the performance of the Sharpe Ratio maximisation method with 60:40 portfolio. Where 60% of the assets are invested in equities and 40% are invested in bonds. In our scenario we invested 60 in URTH and 40 in BNDX.

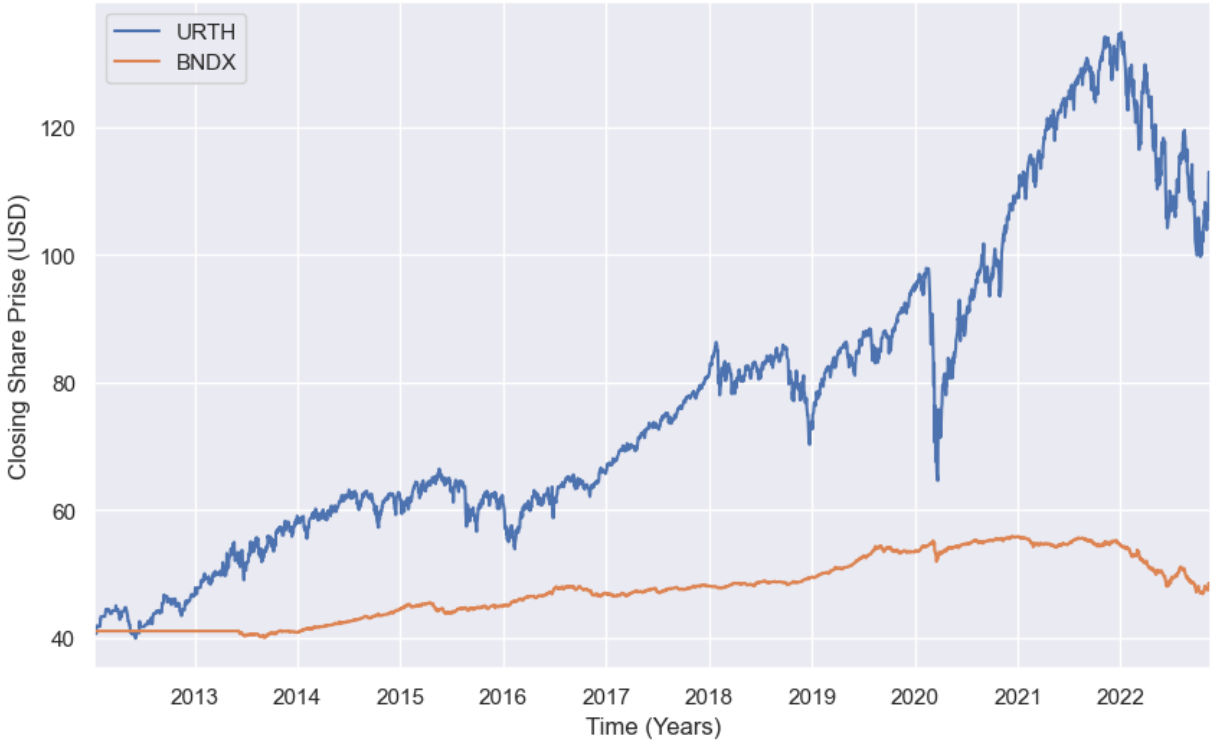

Fig. 6.1 Stock Price over the years

Lagrangian Multiplier

The initial method was maximizing the Lagrangian Equation with contraints. This was done using Tensorflow Contrained Optimization and the optimizer we used was AdaGrad. [15]

$$L(w_1, w_2, ..., w_n, \lambda) = Sharpe(w_1, w_2, ..., w_n) + \lambda(\sum_{i=1}^{n} w_i - 1)$$

Fig. 6.2 Performance of Max Sharpe v/s 60:40

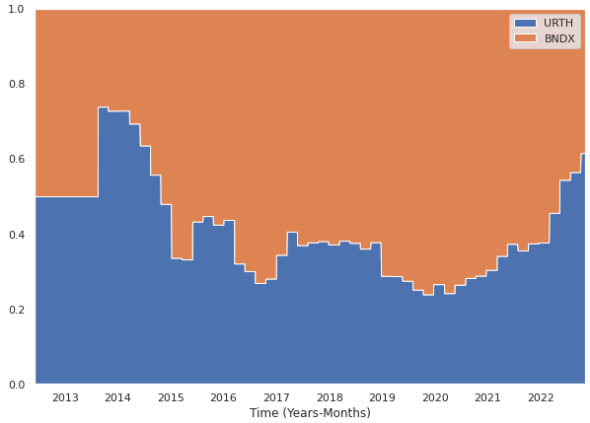

Fig. 6.3 Weights Distribution in Max Sharpe

Convex Optimization

Method 2 involved converting the the Sharpe Function into a convex minimization problem as mentioned in *Optimization Methods in Finance.*[13].

Fig. 6.4 Performance of Max Sharpe v/s 60:40

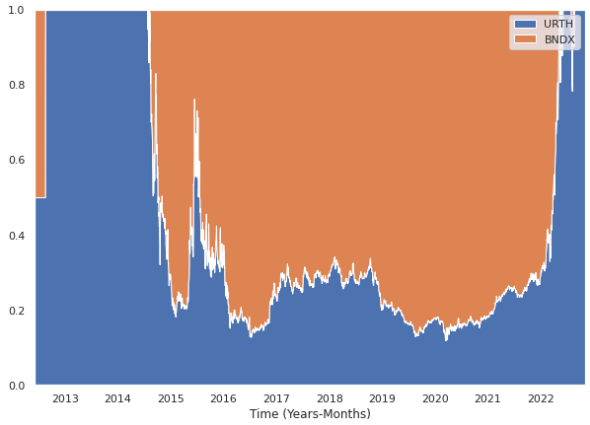

Fig. 6.5 Weights Distribution in Max Sharpe

PPO reward

In the 3rd method we used Sharpe Ratio of the Portfolio as the reward for Proximal Policy Optimization Agent.

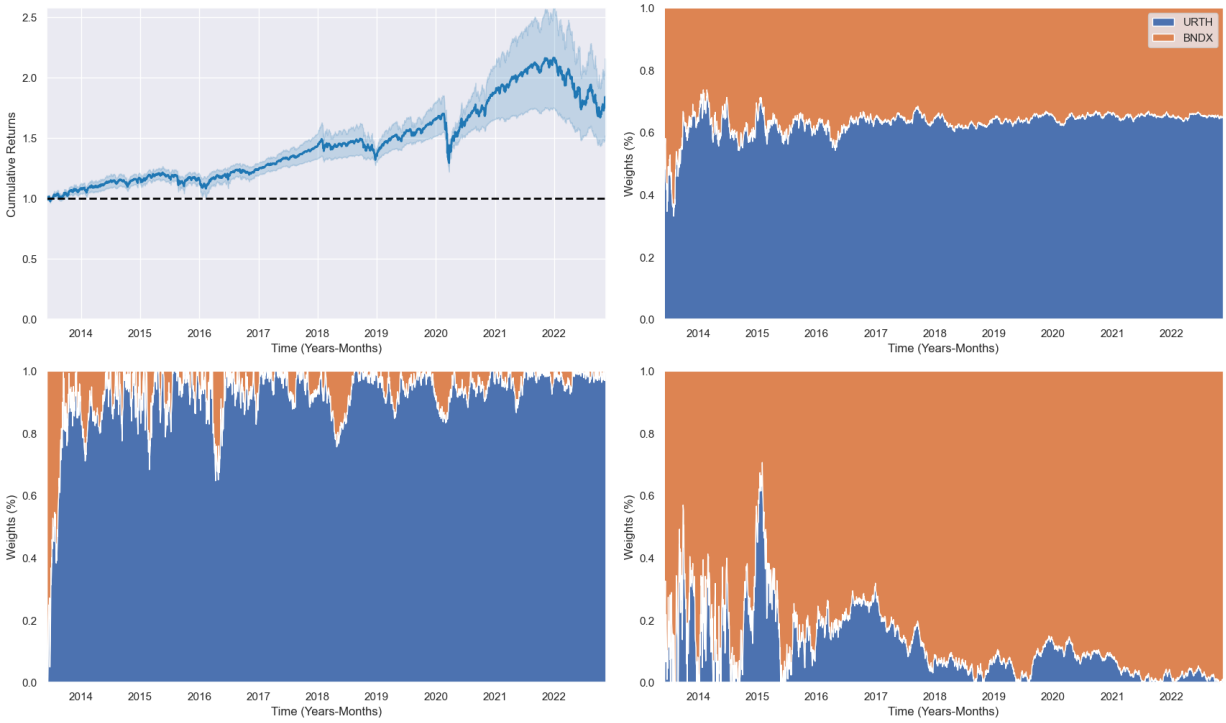

Fig. 6.6 Performance of PPO with reward as Sharpe Ratio.(i) The graph on the top right is weights distribution for median reward run of the Agent. (ii) The graph on bottom left is the weights distribution for best reward run of the Agent. (iii) The graph on bottom right is the weights distribution for worst reward run of the Agent.

6.2.3 Experiment - 2

We take the previous two stocks - BNDX and URTH. The training period is from 01-2019 to 03-2022.

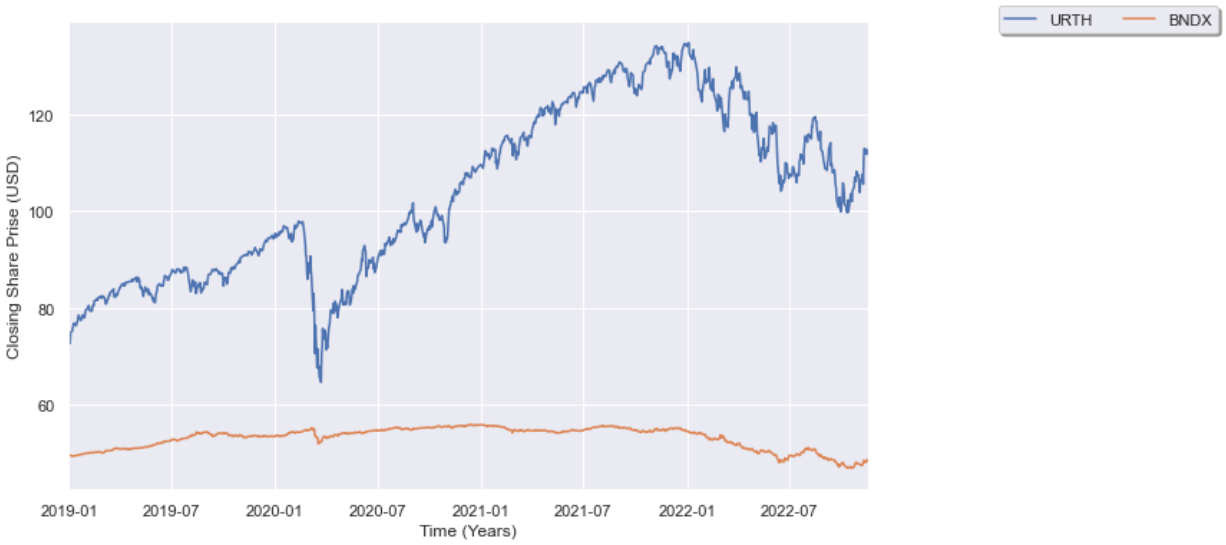

Fig. 6.7 Performance of Stock

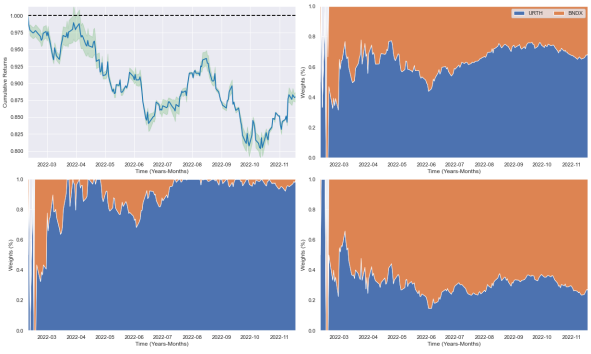

Fig. 6.8 PPO

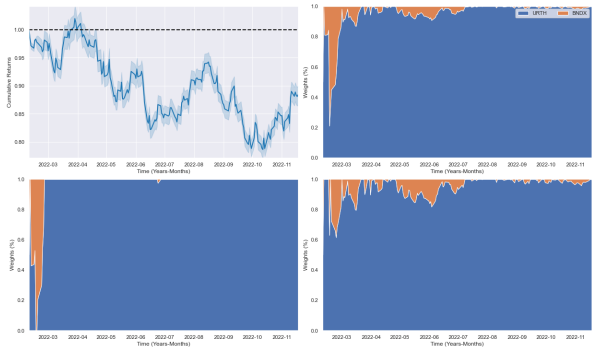

Fig. 6.9 A2C

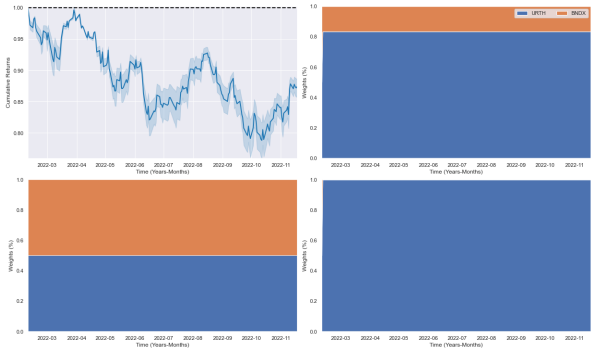

Fig. 6.10 DDPG

Fig. 6.11 SAC

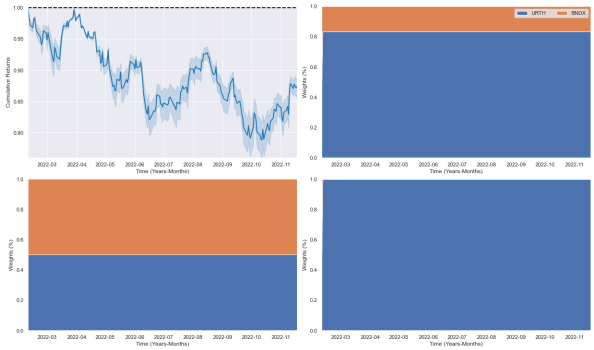

Fig. 6.12 TD3

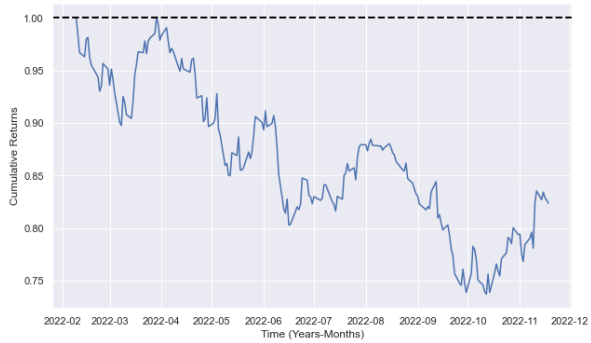

Fig. 6.13 Max Sharpe

Fig. 6.14 Risk Parity

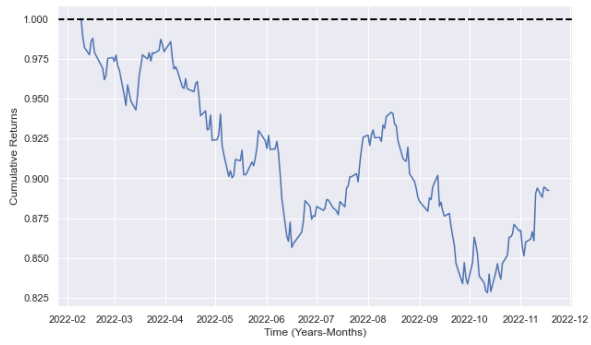

Fig. 6.15 Equal Weight

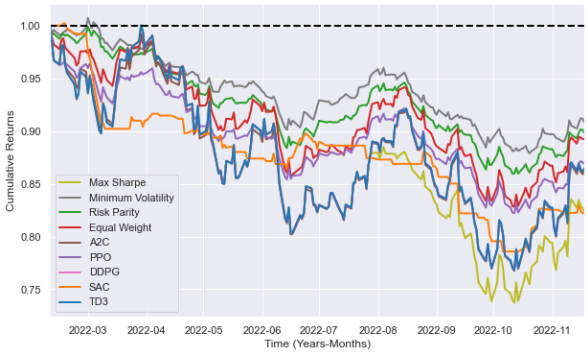

Fig. 6.16 Cumulative returns with Worst run

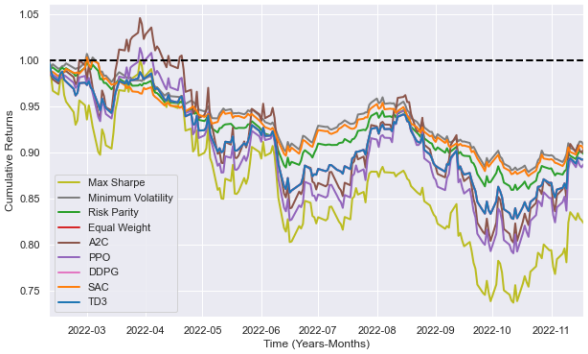

Fig. 6.17 Cumulative returns with Best run

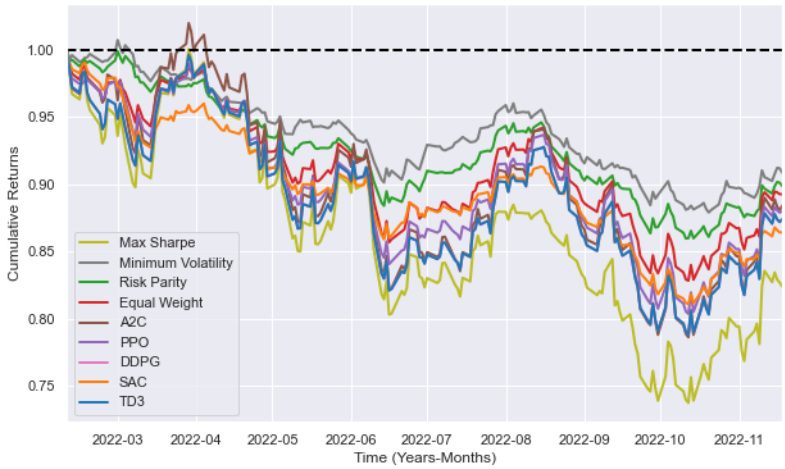

Fig. 6.18 Cumulative returns with Average runs

6.2.4 Experiment - 3[3]

We choose seven of the most prominent stocks in the SP 500 - Apple (APPL), General Electric (GE), JPMorgan Chase (JPM), Microsoft (MSFT), Nike (NKE), Nvidia (NVDA) and 3M (MMM). We also choose a risk free asset - Vodafone Group (VOD). The training period is from 01-2019 to 03-2022.

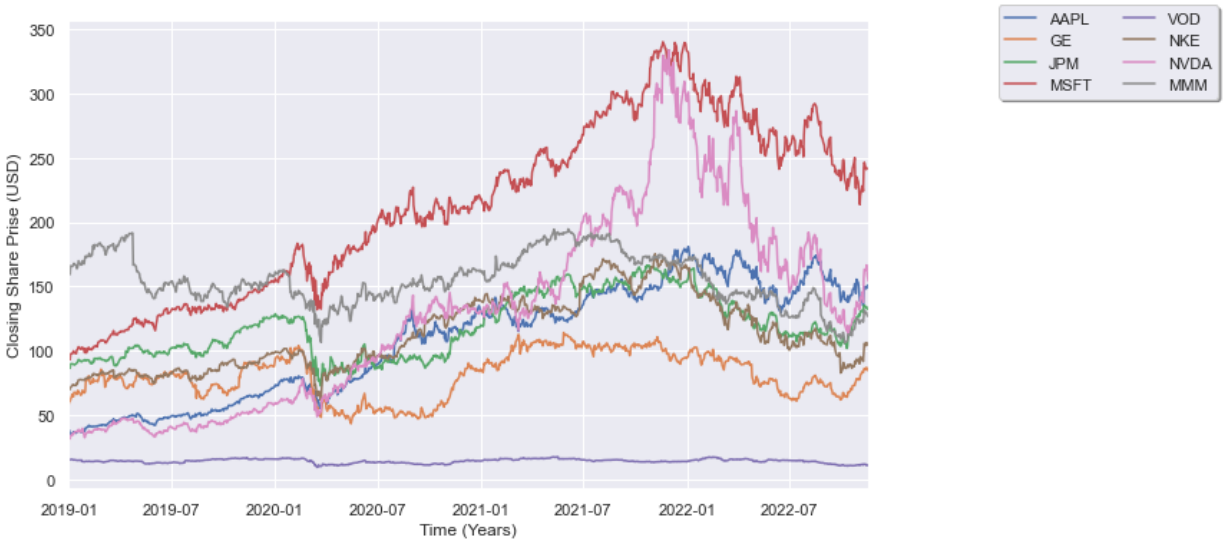

Fig. 6.19 Performance of Stock

Fig. 6.20 PPO

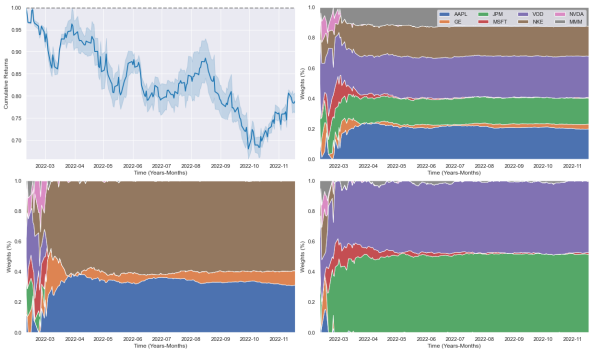

Fig. 6.21 A2C

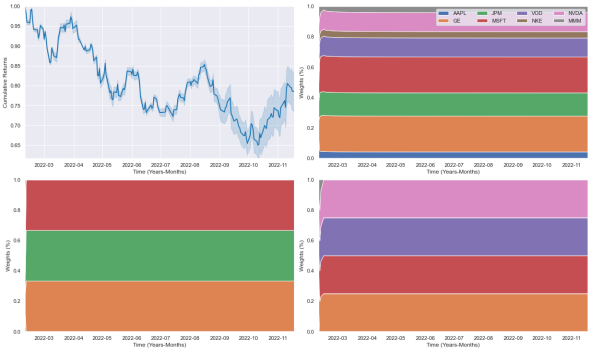

Fig. 6.22 DDPG

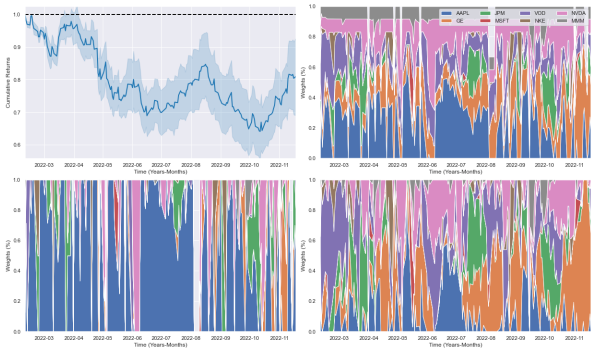

Fig. 6.23 SAC

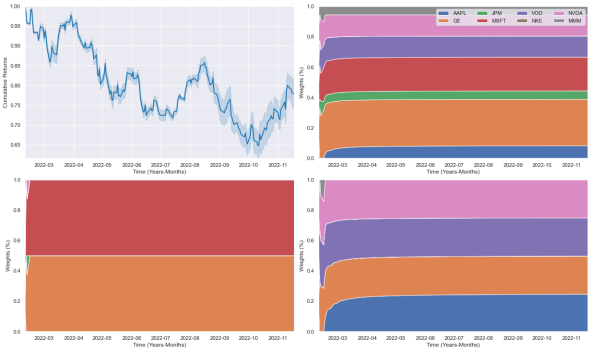

Fig. 6.24 TD3

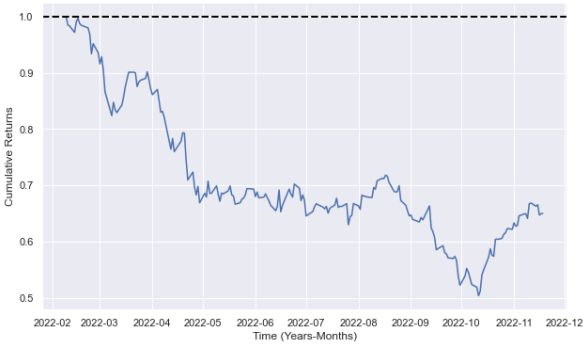

Fig. 6.25 Max Sharpe

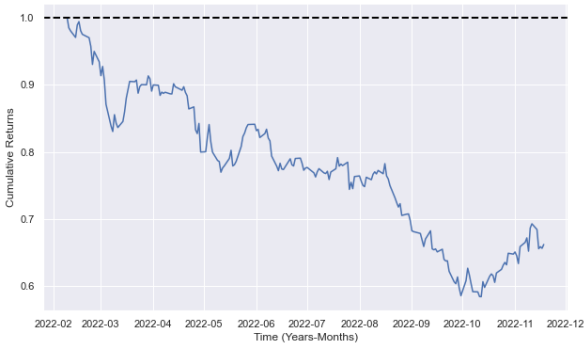

Fig. 6.26 Risk Parity

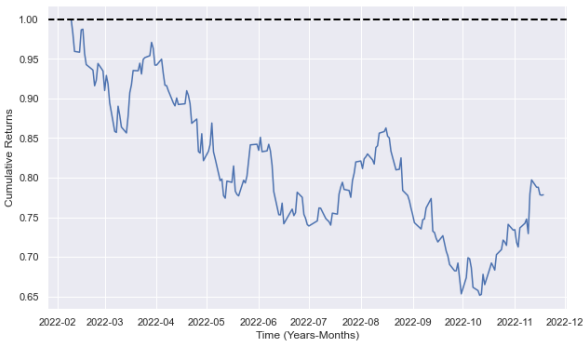

Fig. 6.27 Equal Weight

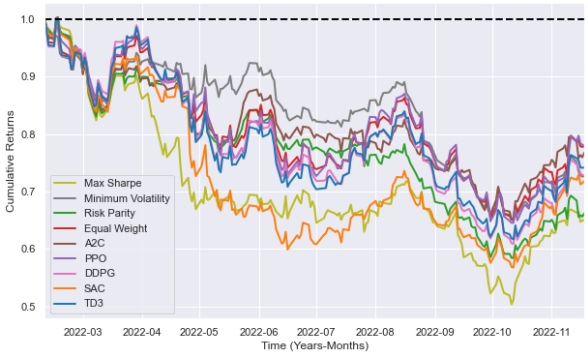

Fig. 6.28 Cumulative returns with Worst run

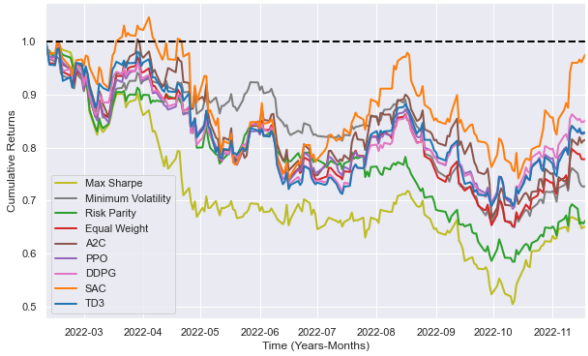

Fig. 6.29 Cumulative returns with Best run

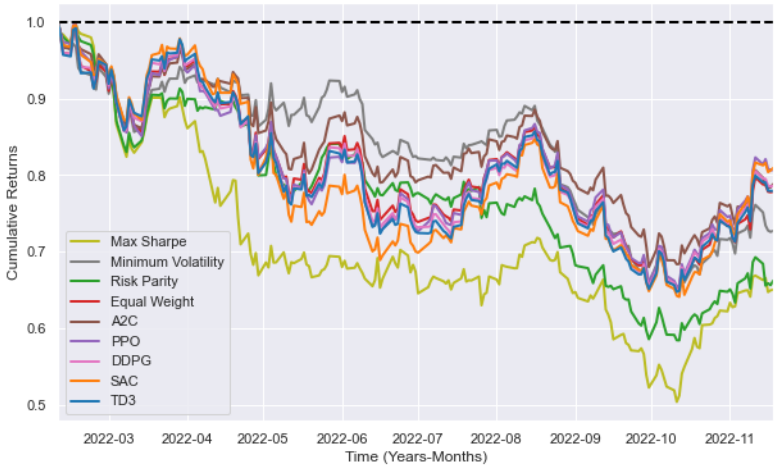

Fig. 6.30 Cumulative returns with Average runs

6.2.5 Observations

The first method used for maximising Sharpe was very expensive as it required optimising a Lagrangian equation during every time step. The method did not gave a global optimum solution for weights (quasi-convex) and did not display good performance.

The second method for maximising sharpe was more accurate at predicting the most optimal weights and also displayed better results in the early timesteps.

The Sharpe Ratio as PPO reward did perform better when compared to the other two methods in the during the recession period.

We observe that no method can outperform all others; different methods work differently. DRL methods are also unstable regarding results, whereas traditional approaches are more stable. Hybrid combinations between them could produce good results.

We observe the following in Experiment 2 with two stocks. In Cumulative returns with the worst run, minimum volatility has the best performance, closely followed by Risk parity. TD3 and A2C could have better performance in general. The worst point is achieved by the Max Sharpe method. In the Cumulative returns with the Best run, A2C has the highest returns and succeeds in producing profits for some time in a recession market, then dips towards the latter half. PPO also produces good results in the initial part and dips later. SAC produces a consistent return without much variation, and PPO achieves results very close to Minimum Volatility. In the Cumulative returns with Average runs, we can see that the most consistent output is achieved by minimum volatility, with risk parity and equal weight followed very closely by. A2C also produces profit in the initial part in this case. In the DRL algorithms A2C, PPO achieves comparable good results.

We observe the following in Experiment 3 with eight stocks. In Cumulative returns with the worst run, minimum volatility achieves the best output; DDPG also has good outputs, although it dips later. Max Sharpe and SAC achieve the poorest results. A2C also produces the best output towards the latter half and has good returns in the first half. In the Cumulative returns with the Best run, SAC has outdone all the other methods, almost producing a cumulative result of 1. A2C follows it. Here we can observe that many DRL methods - SAC, A2C, DDPG, TD3 all outperform the traditional deterministic approaches. In Cumulative returns with Average runs, we again see that minimum volatility achieves the best results, followed by A2C. SAC dips initially, but we see that it catches up. PPO also produces reasonably good results in comparison.

Chapter 7

Future Work and Scope

1. Understanding the Agent's actions to get insights into better trading strategies.
2. Biasing the Agent's reward.
3. Develop a better measure for understanding the Agent's performance regardless of the market scenario (Bullish/Bearish/Recessions).
4. Evaluate the performance of RL methods in Intraday Trading.

References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[2] "History of bear market," https://www.investopedia.com/a-history-of-bear-markets-4582652.

[3] R. Durall López, "Asset allocation: From markowitz to deep reinforcement learning," 07 2022.

[4] B. Clark, Z. Feinstein, and M. Simaan, "A machine learning efficient frontier," *Operations Research Letters*, vol. 48, no. 5, pp. 630–634, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637720301139

[5] Y. Gao, Z. Gao, Y. Hu, S. Song, Z. Jiang, and J. Su, "A framework of hierarchical deep q-network for portfolio management," in *ICAART*, 2021.

[6] A. Oshingbesan, E. Ajiboye, P. Kamashazi, and T. Mbaka, "Model-free reinforcement learning for asset allocation," 09 2022.

[7] 2016. [Online]. Available: https://deepmind.com/learning-resources/-introduction-reinforcement-learning-david-silver

[8] 2020. [Online]. Available: https://www.investopedia.com/terms/r/riskreturntradeoff.asp

[9] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952. [Online]. Available: http://www.jstor.org/stable/2975974

[10] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," 2017. [Online]. Available: https://arxiv.org/abs/1706.10059

[11] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 01 2016.

[12] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653–664, 2017.

[13] G. Cornuéjols, J. Peña, and R. Tütüncü, *Optimization Methods in Finance*. Cambridge University Press, 2018. [Online]. Available: https://books.google.co.in/books?id=Dq1jDwAAQBAJ

[14] "Bull and bear markets," https://www.investopedia.com/insights/digging-deeper-bull-and-bear-markets.

[15] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.