Portfolio Optimization using Reinforcement Learning

Krish Khadria (Roll No: 22112057), Kundan Kumar (Roll No: 22125018)
DA-202 Course Project
Mehta Family School Of DSAI, IIT Roorkee

Abstract

Portfolio optimization is considered to be a central problem in finance, where one seeks to maximize returns while keeping the risk below the minimal limit. Traditional approaches to portfolio optimization rest on mathematical models and historical data, which may, however, not reflect the dynamics of the financial market. We propose a novel approach to portfolio optimization using deep reinforcement learning (DRL).By formulating portfolio optimization as a problem of sequential decision-making, we apply DRL methods to directly learn the optimal investment strategies from raw market data. Our study has demonstrated the adaptability of DRL in adjusting to changing market conditions, surpassing state-of-the-art portfolio management tasks. We have evaluated our work empirically on real-world financial data to show our approach outperforming the state-of-the-art methods in terms of risk-adjusted returns and portfolio diversification performance. Overall, this research advances the state-of-the-art in portfolio optimization and brings out the potential of DRL for enhancing investment decision-making in financial markets.

Contribution: Krish~50% , Kundan~50%

I. INTRODUCTION

The efficient allocation of assets forms an essential prerequisite for achieving goals for investment and risk management in the financial market. Portfolio optimization, or the process of constructing investment portfolios that balance risk and return, forms the nucleus of modern portfolio theory. The traditional methods of portfolio optimization, such as mean-variance optimization and Markowitz's portfolio theory, have been based on mathematical models and historical data for the construction of the optimal portfolio. However, these are often subject to several limitations in terms of sensitivity to input parameters, lack of robustness to market dynamics, and inability to capture nonlinear relationships among assets.

In this paper, we explore portfolio optimization with deep reinforcement learning and provide benchmarking for many DRL algorithms applied in this domain. Our research investigates how effectively popular DRL algorithms such as Proximal Policy Optimization, Advantage Actor-Critic, and Soft Actor-Critic tune investment portfolios. Each of the DRL algorithms possesses certain advantages and characteristics that make them suitable for different portfolio optimization aspects.

The unified framework used in our experiments embeds these DRL algorithms to optimize a portfolio in a way that maximizes returns but keeps the risk of a diversified portfolio of financial assets under control. We turn portfolio optimization into a sequential decision-making problem and thus take advantage of DRL algorithms' flexibility and adaptability to learn investment strategies directly from raw market data.

II. DRL FRAMEWORK

In the context of DRL frameworks, portfolio management can be visualized as a problem of optimizing risk-adjusted returns for a set of assets.

To understand the results better, we make a couple of assumptions:
1.We can trade whenever we want in the market.
2.Trades made by the agent won't really affect the prices of stocks in the actual market.

Within this DRL framework, several elements interact to achieve the desired outcome:
Agent: The main element of the framework is a deep neural network acting as the agent. Its objective is to learn an optimal function, either value or policy based, that maximizes rewards by determining the most effective portfolio allocations.
Actions: These are the outputs that the agent makes for a given period. They represent the final weights assigned to each stock in the portfolio, indicating the percentage allocation of each.
State: It shows the present market scenario via financial metrics.It include the asset value, price of assets and the number of stocks of each asset. Represented as an array of length 2*n+1, where n is the number of different assets our portfolio includes.
Environment: The whole trading market comprising of every agent trading within the market forms the environment. Here, it takes asset weights from the agent and provides reward based on the new portfolio value.
Reward: The reward signal simply reflects how much our portfolio has changed from before to now, telling us if we are making more or less money across time. In our setup, an agent learns and adapts its moves—how much to invest in what assets—via stochastic gradient descent. It observes what's going on at this moment in time and how much money it's making, net of costs, and updates its strategy accordingly. We'll represent this relationship between components as done in the DRL framework diagram below.

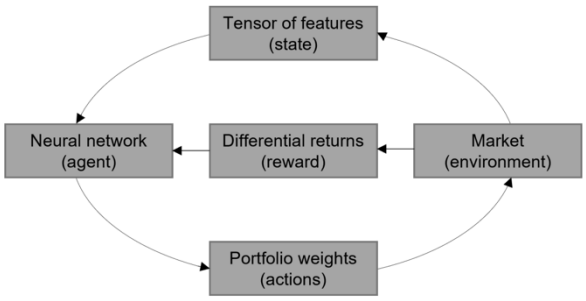

Figure 1:- DRL Framework

III. ALGORITHMS

Our study utilizes these DRL algorithms in a common framework for portfolio optimization:-

Actor-Critic (AC): Initially put forward by Konda and Tsitsiklis in 1999, AC is a two-body architecture: the actor picks actions and the critic criticizes. By improving both structures in an iterative manner, an actor-critic system is expected to achieve better performance than systems based on only one structure.

Proximal Policy Optimization (PPO): A simplified form of TRPO, introduced in 2017 by Schulman et al., PPO is using first-order optimization and bounds the policy updates within a smaller interval around 1.

Advantage Actor-Critic (A2C): A variant of AC, proposed by Mnih et al. in 2016, A2C uses advantage function estimates for bootstrapping. Doing this reduces policy variance and makes reinforcement learning algorithms more stable.

Soft Actor-Critic (SAC): An off-policy algorithm proposed by Haarnoja et al. in 2018, SAC maximizes an objective that includes an entropy term to encourage exploration, hence making the policy more robust.

IV. EXPERIMENTS

In this study, we focus on asset allocation, which entails determining how to distribute investments within a portfolio of assets. We perform a comparison between the different optimization techniques, including those based on DRL, to establish how well they work. We take a step further in their thorough evaluation under different market conditions, like during a bull and a bear market. We examine the behaviour of the methods when tuning investment allocations at different frequencies—specifically, daily.

The last traded price when the exchange closes is termed the close price, but it's not that reliable since it might not reflect the true value. Hence, it is modefied to accommodate various events like new

issues, rights offerings, stock dividends, spin-offs and mergers, or stock splits.
For the bullish trends, we use the adjusted price data from 8 different stocks. We have used a data of 10 years from 2010-2020. And for the bearish trends, we use the adjusted closing price data from 9 different companies. We use 20% of the data for testing purpose and the rest for the training purpose.

Performance of 3 different algorithms- A2C, PPO and SAC is then evaluated for both the bullish and bearish trends.. These algorithms, based on the Actor-Critic algorithm, involve stochastic weight initialization. We run each experiment independently 10 times to capture uncertainty boundaries.

We set up a stock trading RL environment and set the initial account balance and set the maximum number of shares that can be bought or sold in a single transaction to 10. Define action space normalization and shape as number of stocks. Defining the State Space with dimension: 1 + 2 * number of stocks: [[Current Balance] + [Prices] + [Owned Shares]]. We then initialise an action memory and asset memory for keeping track of all actions performed and the asset value. We then define sell, buy and step functions to bring about the required transactions and actions. Model is then trained on the given algorithm and training time data series. We check the performance of the algorithm for stocks showing bullish and bearish trend.

Bullish Market Trend:
The bullish trend stocks here are those of 3M Company, represented by MMM, AAPL for Apple, MSFT for Microsoft, GE for General Electric, JPM for JPMorgan Chase, NKE for Nike, NVDA for Nvidia, and VOD for Vodafone Group. The prices being adjusted close prices for the period between April 1, 2010, and April 1, 2020.

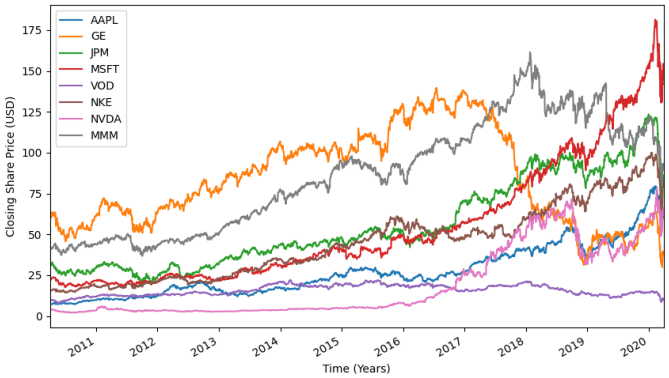

Figure 2: Bullish Market Trend of 8 stocks between 1 April 2010 and 1 April 2020

Bearish Market Trend:
A bearish trend is one in which price is declining or is expected to decline. Here we use 9 stocks which shows bearish trend. These stocks are Consol Investments Ltd. (CONSOFINVT.NS), West Coast Paper Mills Ltd. (WSTCSTPAPR.NS), Indo Borax & Chemicals Ltd. (INDOBORAX.NS), Dhunseri Ventures Ltd. (DVL.NS), DCW Ltd. (DCW.NS), Bannari Amman Spinning Mills Ltd. (BEPL.NS), Agarwal Industrial Corporation Ltd. (AGARIND.NS), Datamatics Global Services Ltd. (DATAMATICS.NS), and Banswara Syntex Ltd. (BANSWRAS.NS). We analyze their adjusted close prices spanning from April 1, 2013, to April 1, 2024.

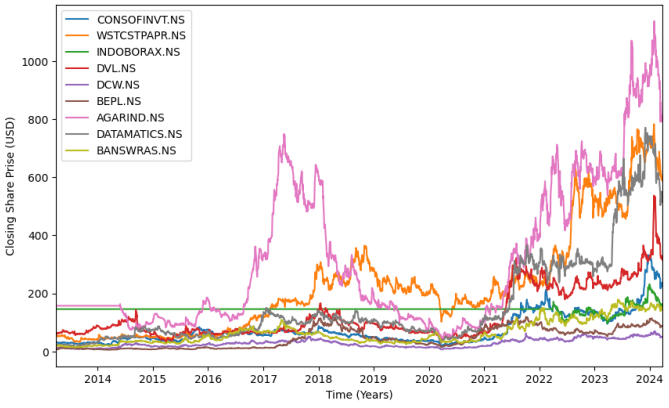

Figure 3: Bearish Market Trend of 9 stocks between 1 April 2013 and 1 April 2024

V. RESULTS

Results for Bullish Trend:

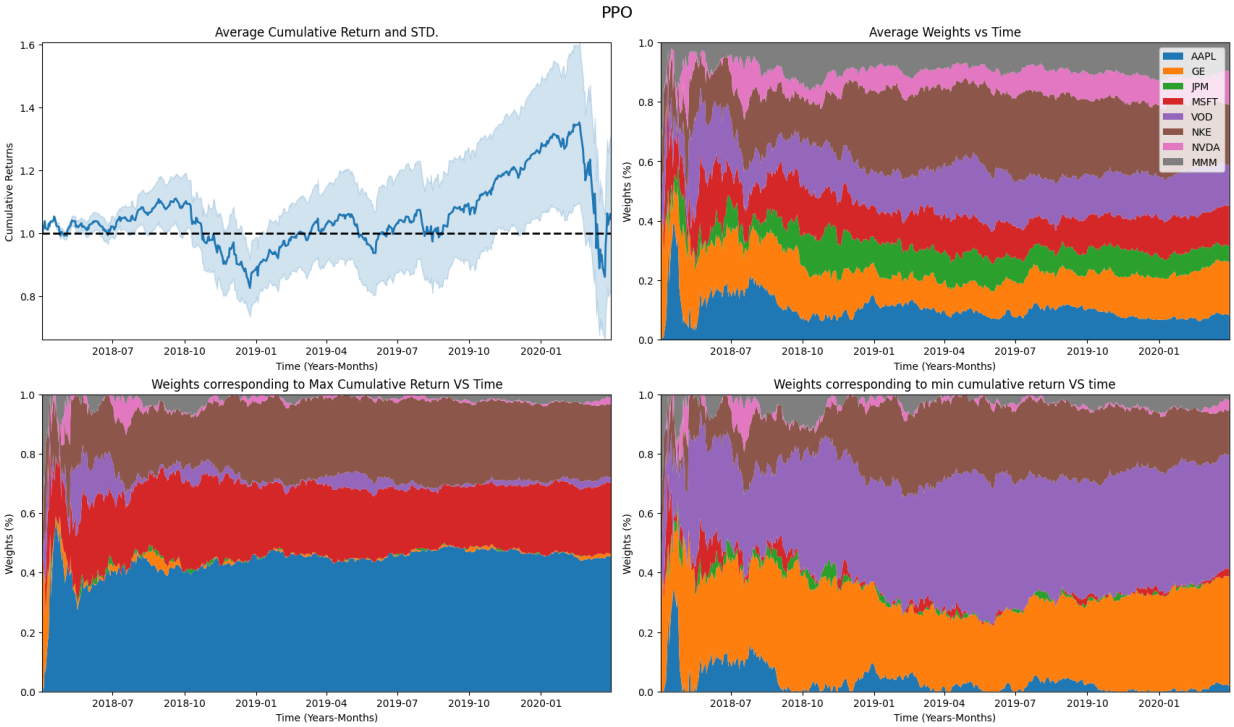

Figure 4: Bullish Trend Results on PPO Algorithm

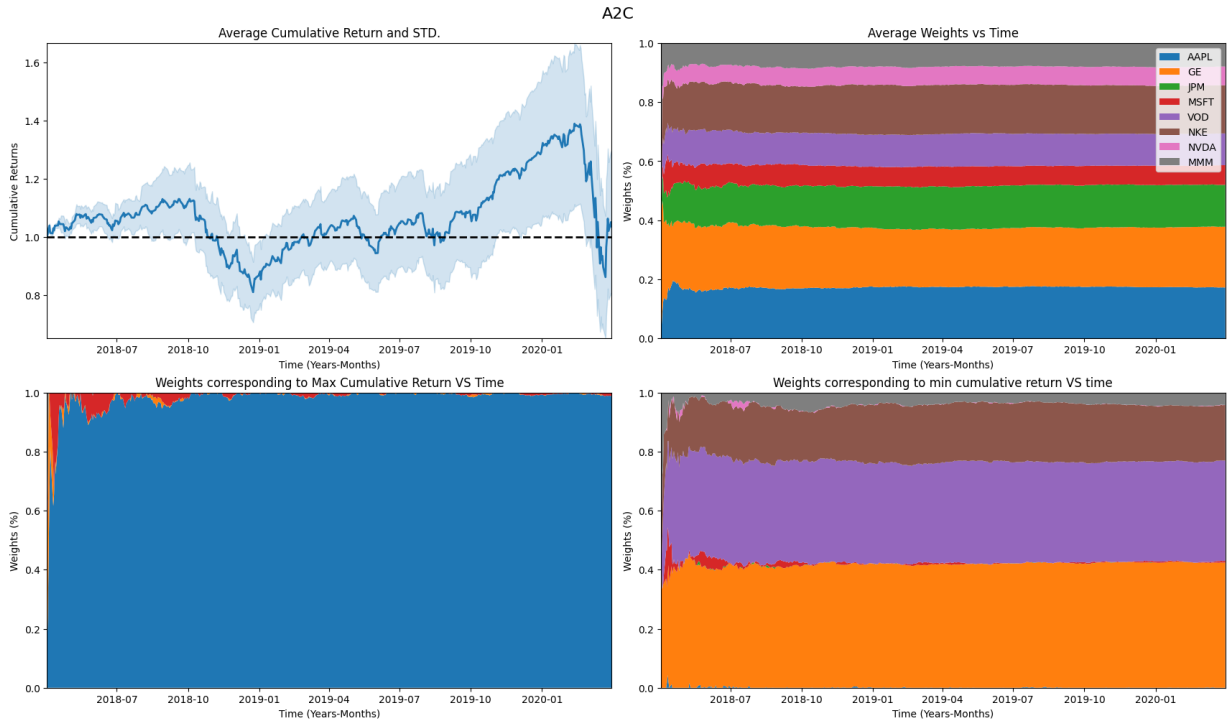

Figure 5:Bullish Trend Results on A2C Algorithm

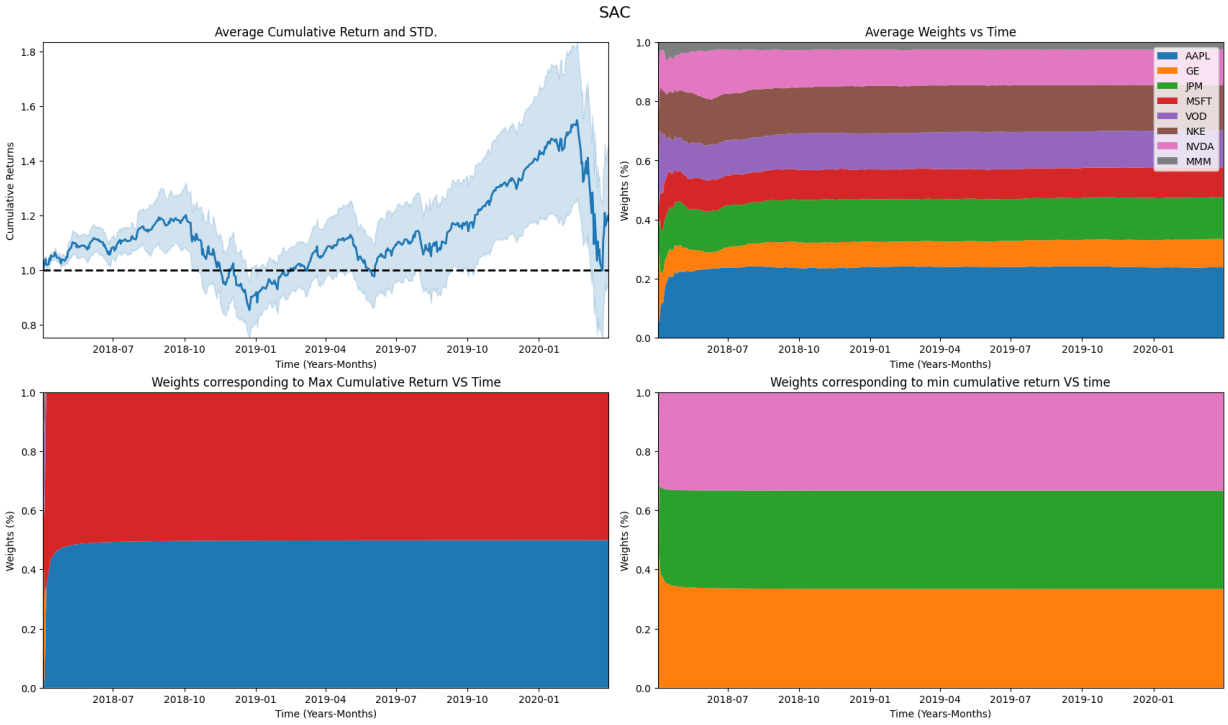

Figure 6:Bullish Trend Results on SAC Algorithm

Results on Bearish Trend:

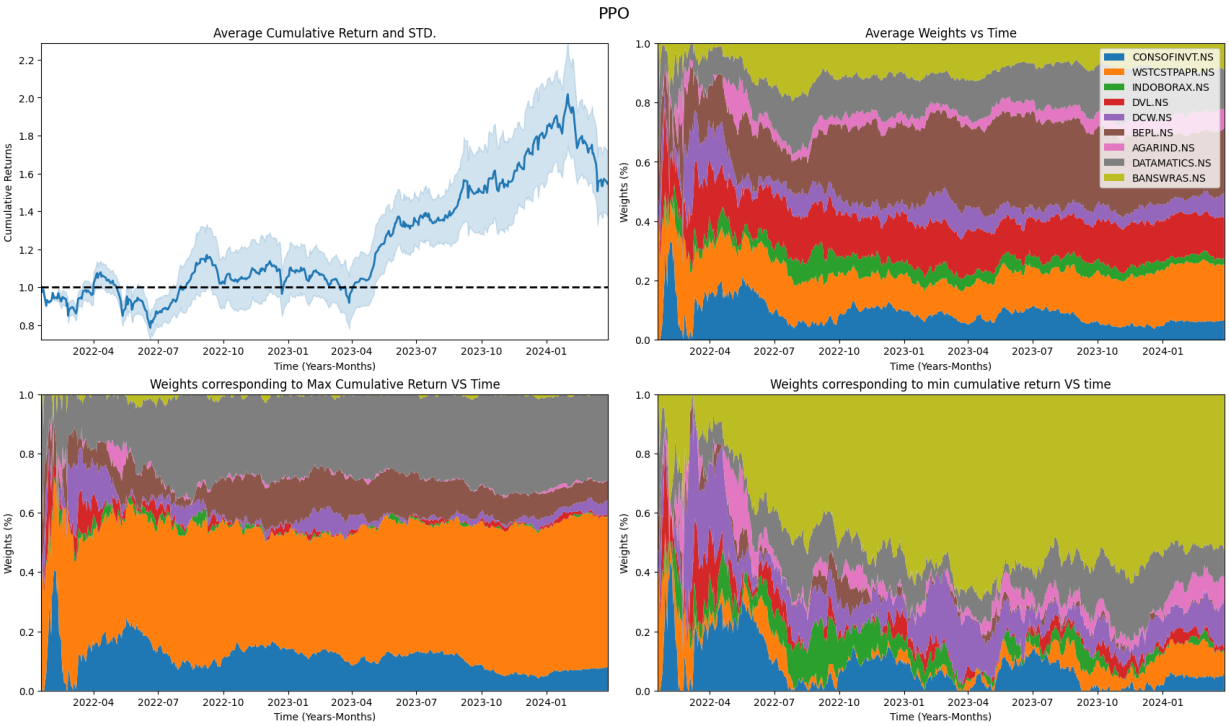

Figure 7: Bearish Trend Results on PPO Algorithm

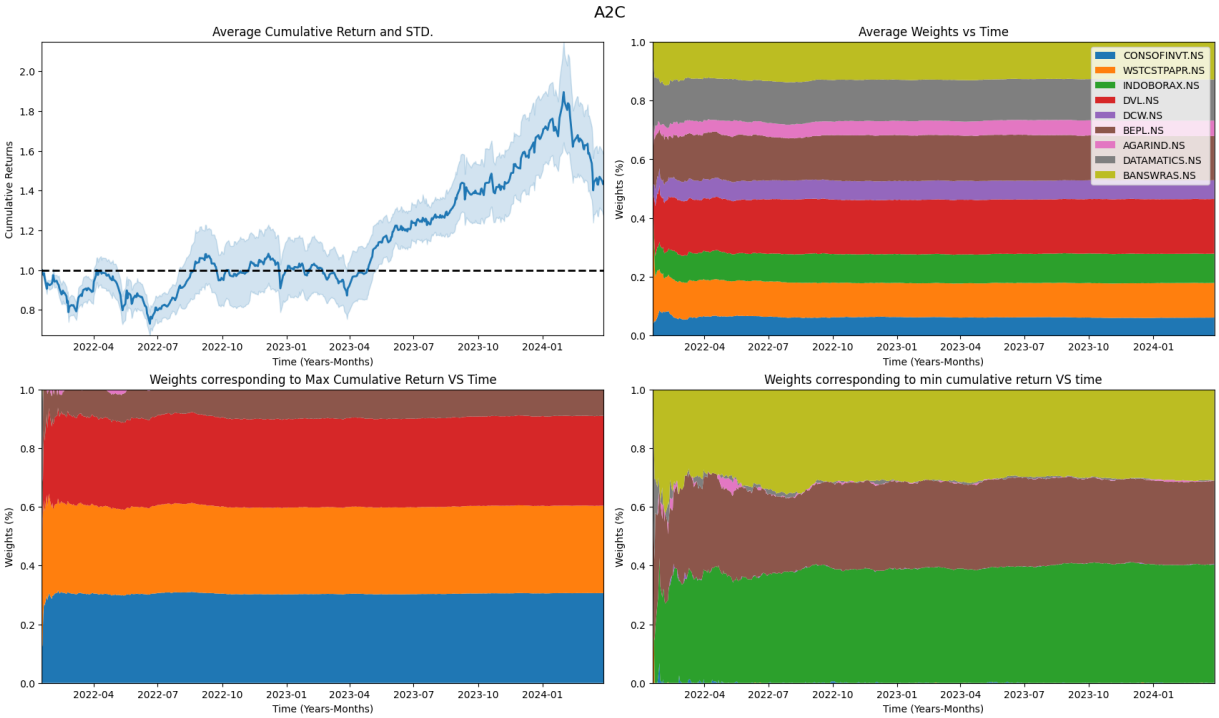

Figure 8: Bearish Trend Results on A2C Algorithm

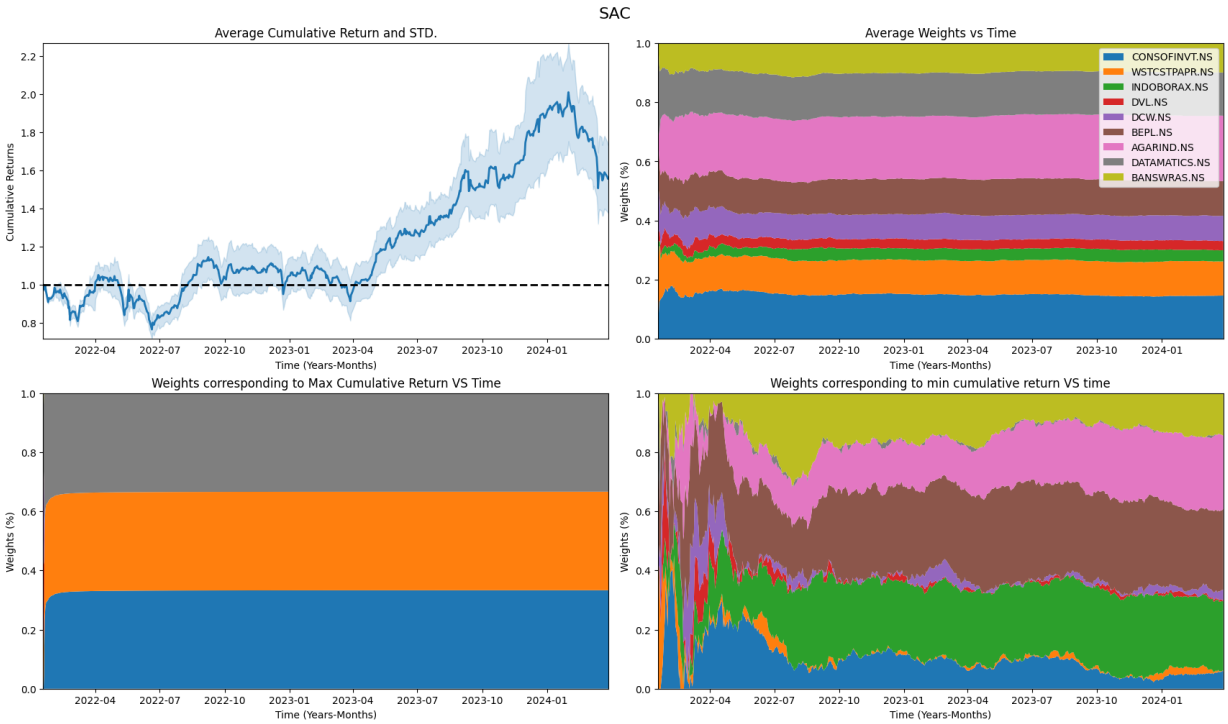

Figure 9: Bearish Trend Results on SAC Algorithm

VI. CONCLUSION

In this study, we explore how the effectiveness of optimization algorithms applies to asset allocation tasks, specifically A2C, PPO, and SAC. Through a comprehensive benchmarking process across various market conditions—bullish and bearish tendencies—the reliability and performance are quantified with the aim of ascertaining them.

Interpretability of the performance of DRL models is complex. Although DRL methods in some cases excel above traditional methods, their performance is not consistent over all runs. For instance, algorithms like PPO and SAC will likely perform well in rising and falling markets, respectively. Yet, there are cases when these highly advanced algorithms underperform and even perform worse than a simple equal-weight strategy. This non-determinism comes from the inherent probabilistic nature of the DRL training process. The agent optimization through trial and error may lead to highly efficient and less-than-optimal solutions in different experiments.

To reduce these irregularities, leveraging the flexibility inherent in neural networks by using a broader set of technical indicators or increasing the size of the input data window could further improve performance and robustness. One can also expand the dataset. A larger dataset can encourage better convergence and make the training phase more stable.

VII. REFERENCES

[1] Kumar Yashaswi, "Deep Reinforcement Learning for Portfolio Optimization using Latent Feature State Space (LFSS) Module," Quantitative Finance, arXiv:2102.06233.

[2] Ruoyu Sun et al., "Combining Transformer based Deep Reinforcement Learning with Black-Litterman Model for Portfolio Optimization," Quantitative Finance, arXiv:2402.16609.

[3] "Evaluation of Deep Reinforcement Learning Algorithms for Portfolio Optimization," arXiv:2307.07694.

[4] "Deep Reinforcement Learning and Mean-Variance Portfolio Optimization," Papers with Code, arXiv:2102.06233.

[5] J. Moody and M. Saffell, "Learning to trade via direct reinforcement," IEEE Transactions on Neural Networks, vol. 12, no. 4, pp. 875-889, 2001.

[6] Y. Deng et al., "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.

[7] X. Li and J. Wang, "A deep deterministic policy gradient-based algorithm for continuous futures portfolio trading," arXiv:1807.02787, 2018.

[8] T. Chakraborty et al., "Multi-agent deep reinforcement learning for liquidation strategy analysis," arXiv:1801.09719, 2018.

[9] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," Quantitative Finance, vol. 1, no. 2, pp. 223-236, 2001.

[10] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in Advances in Neural Information Processing Systems, 1992, pp. 950-957.

[11] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction," MIT Press, 1998.

[12] Y. Li and J. Schmid Huber, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," PLOS ONE, vol. 12, no. 7, e0180944, 2017.

[13] L. C. MacLean et al., "The Kelly criterion in blackjack sports betting, and the stock market," in Handbook of Asset and Liability Management, vol. 1, pp. 385-428, 2006.

[14] M. L. Lobo et al., "Portfolio selection with robust estimation," Operations Research, vol. 53, no. 3, pp. 504-516, 2005.

[15] S. J. Brown and W. N. Goetzmann, "Performance persistence," The Journal of Finance, vol. 50, no. 2, pp. 679-698, 1995.

[16] D. Bertsimas and A. W. Lo, "Optimal control of execution costs," Journal of Financial Markets, vol. 1, no. 1, pp. 1-50, 1998.