Portfolio Optimization using Deep Reinforcement Learning models

A comparison between modern Actor-Critic networks and classical portfolio selection using Mean-Variance Optimization on S&P500 Stocks

LUND UNIVERSITY

Jesper Hedlund

Lund University School of Economics and Management

Supervisor: Andreas Johansson

Second year Master's thesis

August 2024

Abstract

Portfolio optimization involves selecting assets to maximize risk-adjusted returns, typically using linear methods like Mean-Variance Optimization (MVO). However, such approaches may not fully capture the complexities of financial markets. This study leverages recent advances in machine learning, specifically reinforcement learning with deep neural networks, to identify alternative methods that may improve upon MVO. Using data from the broad S&P 500, we compare the performance of five modern deep reinforcement learning (DRL) models against MVO, with a focus on risk-adjusted returns. Additionally, we assess whether incorporating a goal-oriented reward function, explicitly designed to maximize risk-adjusted returns, improves DRL performance. To account for the stochastic nature of DRL training, the mean performance across 10 independent runs was calculated and used to ensure result stability, with transaction costs included for increased real-world applicability.

Our findings indicate that DRL models generally outperformed the MVO benchmark, especially in the Maximum Sharpe and ETF portfolios, by achieving higher Sharpe ratios. However, no significant improvement was observed when using the goal-oriented reward function. A robustness test using a Minimum Variance portfolio revealed that DRL models did not clearly surpass MVO, suggesting that the effectiveness of DRL models may depend on the portfolio strategy employed. Despite these mixed results, DRL continues to show potential for enhanced portfolio optimization, though its practical applications warrant further exploration, especially considering factors such as model complexity and transaction costs.

Keywords: Mean-Variance Optimization, Deep Reinforcement Learning, Portfolio Optimization, Differential Sharpe Ratio, Machine Learning

Table of Contents

List of abbreviations

A2C	Advantage Actor-Critic
AI	Artificial Intelligence
ANN	Artificial Neural Network
DDPG	Deep Deterministic Policy Gradient
DRL	Deep Reinforcement Learning
DSR	Differential Sharpe Ratio
ETF	Exchange Traded Fund
MDP	Markov Decision Process
ML	Machine Learning
MLP	Multilayer Perceptron
MPT	Modern Portfolio Theory
MVO	Mean Variance Optimization
OHLC	Open, High, Low, Close
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
SAC	Soft Actor Critic
SR	Sharpe Ratio
TD3	Twin Delayed Deep Deterministic Policy Gradient

1 Introduction

In the field of finance, portfolio optimization is the process of selecting the best combination of assets to achieve specific investment goals, typically balancing risk and return. Effective portfolio optimization strategies allow investors to manage risk by diversifying across multiple assets, sectors, or asset classes, which helps mitigate the impact of negative movements in individual investments (Markowitz, 1952). Classical portfolio optimization is grounded in Modern Portfolio Theory (MPT), introduced by Markowitz in 1952, who later won a Nobel prize. MPT emphasizes diversification, asserting that by spreading investments across assets with low correlation, investors can reduce the overall risk of their portfolio without necessarily sacrificing returns. The practical implementation of MPT is Mean-Variance Optimization (MVO), a mathematical framework that aims to construct a portfolio that either maximizes expected return for a given level of risk or minimizes risk for a given level of expected return. Alternatively, instead of fixing one and maximizing the other, the trade-off between risk and return can be optimized by maximizing risk-adjusted returns, which can be measured by the Sharpe ratio (SR), introduced by Sharpe (1966).

However, MVO relies on several assumptions that can limit its effectiveness in practice. First, it assumes that asset returns follow a normal distribution and that the future can be predicted based on past performance (Markowitz, 1952). This means that MVO requires accurate estimates of expected returns, variances, and covariances between assets, all of which are typically based on historical data (Pástor & Stambaugh, 1999). Financial markets are however highly unpredictable, and relying solely on historical estimates can lead to inaccuracies. Furthermore, financial markets are inherently noisy environments where the signal-to-noise ratio is low, making it difficult to distinguish between meaningful trends and random fluctuations (Liu, Xia, Yang, Gao, Zha, Zhu, Wang, Wang & Guo, 2023).

Given these challenges, there is growing interest in applying machine learning techniques to portfolio optimization. Machine learning has already made significant contributions to other fields by processing large datasets and identifying complex patterns that traditional models struggle with. For example, supervised learning models have been used to predict stock prices based on historical data and unsupervised learning has been applied to classify assets based on shared characteristics (López de Prado, 2018). These advancements have made machine

learning particularly useful in fields where the environment is complex and constantly changing, such as finance.

A subset of machine learning, reinforcement learning (RL), offers an alternative approach to decision-making by training models to learn from their interactions with the environment. In Deep Reinforcement Learning (DRL), RL is combined with deep neural networks, enabling models to handle high-dimensional data and learn strategies in dynamic environments like financial markets. DRL has shown great promise in portfolio management due to its ability to adapt to changing market conditions and make sequential decisions, such as buying or selling assets based on past performance and future predictions (Jiang, Xu & Liang, 2017).

DRL models are particularly useful because they do not rely on the static assumptions that underlie MVO. Instead of optimizing a portfolio based on historical estimates of risk and return, DRL models can dynamically adjust asset allocations based on reward feedback (Sutton & Barto, 2018). These models typically use portfolio value as the reward function during training, which serves as the driving force behind the learning process (Jiang, Xu & Liang, 2017). That means that the model receives feedback on how well its actions have maximized the portfolio's value. While this approach simplifies the optimization process, it does not take volatility into account, completely neglecting the risk that could be associated with maximizing returns.

While DRL models often focus on maximizing portfolio value, they are typically evaluated based on risk-adjusted returns, measured by the SR during testing. This evaluation places them within the realm of portfolio optimization, even though they are not explicitly trained to optimize for these metrics. Despite this, recent studies have shown that DRL models can outperform traditional benchmarks like MVO in certain cases, particularly in dynamic and complex market environments where their adaptability can be advantageous (Benhamou, Saltiel, Ungari & Mukhopadhyay, 2020).

However, while these models can perform well on risk-adjusted returns during evaluation, there is potential to further improve alignment between the training objective and these performance metrics. One approach is to optimize directly for risk-adjusted returns during training, ensuring that the model's reward function explicitly reflects the investor’s goal of balancing risk and return.

One way to directly optimize risk-adjusted returns is by using the Differential Sharpe ratio (DSR) as the reward function in DRL models. The DSR allows the model to optimize for

returns relative to risk during training, providing a more targeted approach to portfolio optimization (Moody, Wu, Liao & Saffell, 1998). However, this method introduces additional complexity due to the higher dimensionality and interdependencies between assets.

This study is inspired by the work of Sood, Papasotiriou, Vaiciulis and Balch (2023), who compared DRL, guided by risk-adjusted returns using DSR, to an MVO benchmark. Although DRL models have shown potential in some studies, uncertainties remain regarding their consistent performance, particularly when applied to different datasets and market conditions. Financial markets are complex, and models like DRL may overfit to historical data, which raises concerns about how well these models generalize. Additionally, while the DSR adds theoretical appeal, its complexity may not always translate into consistently better performance compared to simpler approaches, such as using portfolio value as the reward function.

This study aims to explore these uncertainties by addressing two key research questions:

1. Can DRL models outperform MVO benchmarks in terms of risk-adjusted returns?
2. Does optimizing directly for risk-adjusted returns using the more complex DSR lead to better performance than using portfolio value?

To answer these questions, we evaluate DRL models using both portfolio value and DSR as reward functions. While influenced by Sood et al. (2023), our study expands the scope by evaluating five DRL models on three different portfolios using a comprehensive 20-year dataset. We employ a single extensive training period with multiple independent runs, incorporating transaction costs and providing insights into long-term model stability and real-world applicability. By testing on an ETF portfolio and a Minimum Variance portfolio, we assess whether DRL models can generalize their performance across diverse datasets and conditions.

Our findings indicate that DRL models generally outperformed the MVO benchmark in terms of SR, particularly in the Maximum Sharpe and ETF portfolios. This outperformance was observed in models optimized for portfolio value as well as those using the goal-oriented DSR as the reward function. However, models using DSR as a reward function showed mixed results, with added complexity not consistently improving performance. Robustness tests revealed that DRL models did not clearly surpass MVO in the Minimum Variance portfolio, suggesting that their effectiveness may depend on the portfolio strategy employed.

The inclusion of transaction costs highlighted practical constraints, affecting the frequent rebalancing characteristic of DRL strategies. These findings contribute to the literature by providing a practical assessment of DRL models in portfolio optimization, addressing both risk-adjusted return optimization and transaction costs. They suggest that while DRL holds potential as a powerful tool for dynamic financial environments, its success hinges on careful model selection, reward design, and consideration of real-world constraints tailored to specific investment goals and market conditions.

2 Related Work

Exploring machine learning models as preselection tools for portfolio formation, Ma, Han and Wang (2021) compared Random Forest and Support Vector Regression to deep learning models like Long-Short Term Memory, Deep Multilayer Perceptron's, and Convolutional Neural Networks. Their findings indicate that integrating these models for return prediction showed potential for enhancing portfolio strategies when compared to traditional time-series methods, such as Autoregressive integrated moving average (ARIMA). However, DRL has emerged as a next-generation technique, providing a more dynamic and flexible approach to portfolio optimization.

Research in portfolio optimization using DRL models has gained momentum in recent years. For instance, Aboussalah and Lee (2020) provided evidence for the adaptability of DRL techniques in dynamic financial markets by applying a Stacked Deep Dynamic Recurrent Reinforcement Learning model. Their study achieved superior risk-adjusted returns compared to traditional methods, indicating that their DRL approach was able to adjust dynamically to changing market conditions.

DRL models require guidance from a reward function. A common method involves using daily returns as the reward function, see for example Théate and Ernst (2021) and Jiang, Xu and Liang (2017). Typically, these studies focus on maximizing portfolio returns, with performance in terms of risk-adjusted returns like the Sharpe ratio (SR) being evaluated only during the testing period. Almahdi and Yang (2017) took a different approach by prioritizing the expected maximum drawdown in their reinforcement learning-based portfolio, emphasizing risk reduction over return maximization. Benhamou et al. (2020) similarly found that DRL models could outperform MVO and a Minimum Variance portfolio in terms of SR and Sortino ratio, though they observed a higher maximum drawdown.

In contrast, some research has aimed to align the training objective more directly with risk-adjusted performance metrics. For instance, Sood et al. (2023) addressed the limitations of traditional approaches by using implements of the Differential Sharpe ratio (DSR) as the reward signal in portfolio optimization. Unlike the standard SR, which is calculated over a period and is not well-suited for incremental learning in DRL agents, the DSR allows for more effective integration into the reward function, ensuring that the optimization process is directly focused on achieving the desired outcome. They used data on US equities market and

found that the DRL models performed better in terms of risk-adjusted returns than their benchmark model.

Not all studies on DSR have shown convincing results. Sadighian (2020) tested seven different reward functions on the cryptocurrency market and found that DSR performed inconsistent.

Yang, Liu, Zhong and Walid (2020) used data from 30 stocks on Dow Jones and compared three DRL models, PPO, DDPG and A2C, to the DJIA and a Minimum Variance portfolio. Additionally, to capture the strengths of each individual DRL model in dynamic markets, they introduced the use of ensemble strategy. It greedily picked the best-performing model in a 60-day period as the optimizer for the next period, maximizing portfolio value as the reward function. The ensemble strategy outperformed all the individual DRL models and the two benchmarks in terms of SR.

Other studies have used different benchmarks, Sood et al. (2023) and Benhamou et al. (2020) compared their DRL models to portfolios optimized using MVO. The datasets used in these studies also vary, reflecting the broad applicability of DRL in portfolio management. Contrary to Yang et al. (2020) who utilized data from the Dow Jones Index, Sood et al. (2023) focused on the U.S. Equities market while Jiang, Xu and Liang (2017) explored the use of DRL in the cryptocurrency market, showcasing the flexibility of these models across different asset classes.

The consideration of transaction costs and the robustness of back testing procedures are crucial aspects of these studies. for instance, conducted extensive back testing by running simulations 10 times and presenting average results, a method that adds robustness to their findings. Both Jang and Seong (2023) and Jiang, Xu and Liang (2017) included transaction costs in their evaluations, ensuring that the reported performance metrics more accurately reflect real-world trading conditions.

Regarding input features, there is variability in the complexity of the data used. Some studies, such as those by Yang et al. (2020), relied solely on basic stock price data (open, low, high, close), while others, like Jang and Seong (2023) and Gu, Du, Muntasir Rahman and Wang (2023), enriched their models by incorporating technical indicators, thereby enhancing the feature set available to the DRL agents. Gu et al. (2023) found that annualized returns were twice as good as the Dow Jones index benchmark, but it should be caveated that they had a leveraged portfolio using a margin trader.

3 Theoretical Framework

This chapter outlines the key theories and models applied in this study. We begin by discussing Modern Portfolio Theory (MPT) and Mean-Variance Optimization (MVO), which emphasize risk-return balance through diversification. These methods, however, may not fully account for the complexities of financial markets.

To address this, we explore Deep Reinforcement Learning (DRL) models, which adapt to dynamic market environments. Key concepts such as neural networks, activation functions and backpropagation will be discussed, as well as reward functions like Differential Sharpe ratio (DSR).

3.1 Modern Portfolio Theory and Mean-Variance Optimization

Modern Portfolio Theory, introduced by Harry Markowitz (1952), provides a mathematical framework for assembling a portfolio of assets that maximizes expected return for a given level of risk. The theory is based on the premise that investors are rational and risk-averse, preferring portfolios with lower risk for a specified level of expected return (Markowitz, 1952). MPT suggests that diversification can reduce overall portfolio risk by investing across multiple assets, thereby enhancing risk-adjusted returns.

At the core of MPT is the concept that a portfolio's risk and return can be quantified using its mean return and variance. For a portfolio containing multiple assets, the expected return $E(R_p)$ is calculated as the weighted sum of the expected returns of the individual assets:

$$E(R_p) = \sum_{i=1}^{n} (w_i * E(R_i))$$

where w_i is the weight of asset i and $E(R_i)$ is its expected return. The weights w_i are subject to the constraint $\sum_{i=1}^{N} w_i = 1$, ensuring that the total investment is fully allocated.

The portfolio variance σ_p^2, representing risk, is given by:

$$\sigma_p^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij},$$

where σ_{ij} is the covariance between the returns of assets i and j.

MVO is a central technique within MPT that determines the optimal allocation of assets in a portfolio to achieve specific investment objectives, such as maximizing expected return for a given level of risk or minimizing risk for a desired expected return. Risk is typically measured by the volatility of portfolio returns, and MVO requires estimates of asset expected returns, variances, and covariances. Since future returns are uncertain, these estimates are generally derived from historical data.

3.1.1 Efficient Frontier

The portfolio optimization problem in MVO is often framed as a convex optimization problem, which can be mathematically expressed as:

$$\min_{w} \frac{1}{2} \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}$$

subject to $\boldsymbol{w}^T \boldsymbol{\mu} \geq r_p$, $\sum_{i=1}^{N} w_i = 1$ and $w_i \geq 0$ for all i,

where $\boldsymbol{w}$ is the vector of asset weights, $\boldsymbol{\Sigma}$ is the covariance matrix of asset returns, $\boldsymbol{\mu}$ is the vector of expected asset returns, and r_p is the target portfolio return. This optimization seeks to minimize the portfolio's variance while achieving at least the target expected return r_p, under the constraints that the total weight sums to one and that no short selling occurs.

By solving the optimization problem for various levels of r_p, we can map out the efficient frontier. The efficient frontier represents the set of optimal portfolios that offer the highest expected return for each level of risk. Portfolios lying on this frontier are considered efficient because they provide the best possible expected return for a given level of risk. Conversely, portfolios below the frontier are sub-optimal, as they yield lower returns for the same level of risk.

3.1.2 Minimum Variance Portfolio

A special case within MVO is the Minimum Variance portfolio, which focuses solely on minimizing risk without considering expected return. The optimization problem for the Minimum Variance portfolio is formulated as:

$$\min_{w} \frac{1}{2} \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}$$

subject to $\sum_{i=1}^{N} w_i = 1$ and $w_i \geq 0$ for all i.

This portfolio occupies the leftmost point on the efficient frontier, representing the lowest possible risk achievable through diversification among the available assets.

3.1.3 Maximum Sharpe Portfolio

While traditional MVO requires specifying a target return or risk level, an alternative approach is to optimize the Sharpe Ratio (SR), which measures the risk-adjusted return of a portfolio. The SR, developed by Sharpe (1966), is defined as:

$$S_P = \frac{E[R_p - R_f]}{\sigma_P}$$

Where R_p is the portfolio return, R_f is the risk-free rate, and σ_P is the standard deviation of the portfolio's returns. Optimizing the SR involves finding the portfolio weights that maximize this ratio, effectively maximizing the expected return per unit of risk. This method simplifies the optimization process by eliminating the need to specify a particular target return or risk level.

The optimization problem for maximizing the SR can be expressed as:

$$\max_{w} \frac{\boldsymbol{w}^T(\boldsymbol{\mu} - R_f \mathbf{1})}{\sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w}}}$$

subject to $\sum_{i=1}^{N} w_i = 1$ and $w_i \geq 0$ for all i,

where $R_f \mathbf{1}$ is the vector of risk-free rates.

However, this problem is non-linear and non-convex due to the square root in the denominator. To make it tractable, it can be reformulated using a variable substitution or approximated using numerical methods (Cornuejols & Tütüncü, 2006). By introducing a scaling factor, the problem can be restated to convert it into a convex optimization that is more manageable using standard optimization techniques.

3.1.4 Limitations of MVO

Despite its foundational role in portfolio management, Mean-Variance Optimization has several limitations. It depends heavily on historical data for estimates of expected returns, variances, and covariances, which may not accurately predict future market behaviour (Michaud, 1989). The model assumes that asset returns are normally distributed and that investors are concerned only with the mean and variance of returns. In reality, asset returns can exhibit skewness and kurtosis, and investors may consider higher moments of the return distribution (Harvey & Siddique, 2000). Additionally, MVO is highly sensitive to the input

parameters. Small changes in the estimated returns or covariances can lead to significant alterations in the optimal portfolio, potentially making it unstable in practice. Furthermore, traditional MVO provides a static solution that does not adapt to changing market conditions over time, which can be a disadvantage in dynamic markets.

3.1.5 Dynamic Market Adaptation

Given these limitations, there is a need for more adaptive and robust portfolio optimization methods. DRL offers a promising alternative by enabling models to learn optimal strategies through interaction with the environment (Jiang, Xu & Liang, 2017). DRL can adapt to changing market conditions by continuously updating its policy based on new data, making it well-suited for real-time portfolio management. However, applying DRL to portfolio optimization necessitates appropriate reward functions that provide immediate feedback. While the SR is effective for evaluating the performance of static portfolios over a fixed period, it does not offer the real-time feedback required for DRL models (Moody et al. 1998). To address this, the DSR is introduced as a more suitable performance metric for DRL, providing continuous feedback that guides the learning process effectively.

By integrating advanced techniques like DRL with traditional portfolio optimization theories, this study aims to develop models that better capture the complexities of financial markets and enhance investment performance.

3.2 Reinforcement Learning

Machine learning (ML) encompasses a variety of techniques that allow models to learn from data. In this study, we focus on Reinforcement Learning (RL), a paradigm where an agent learns to make decisions by interacting with its environment and receiving feedback through rewards or penalties. Unlike supervised and unsupervised learning, which use static datasets to identify patterns, RL learns through active interaction with an environment, adjusting its strategies based on feedback in the form of rewards or penalties (Sutton & Barto, 2018). Notably, RL's adaptability has proven effective in complex decision-making environments, ever since it showed the first superhuman performance in Atari games (Mnih, Kavukcuoglu, Silver, Graves, Antonoglou, Wierstra & Riedmiller, 2013).

The agent's goal is to maximize cumulative rewards over time, which makes RL particularly useful for dynamic tasks like financial portfolio management (Sutton & Barto, 2018). A key challenge in RL is balancing exploration (trying new strategies) with exploitation (leveraging known successful strategies). In financial markets, where conditions change unpredictably,

exploration is crucial to discover new opportunities, but exploitation is necessary to optimize known strategies and achieve stable returns. This balance is vital for RL models to adapt to evolving market conditions (Sutton & Barto, 2018).

In this study, RL models are applied to portfolio management, where the agent interacts with market data, adjusting asset allocations to maximize risk-adjusted returns over time. By learning through trial and error, RL models can potentially outperform traditional methods like MVO by dynamically adapting to changing market environments.

3.2.1 Markov Decision Processes

The RL problem is formalized using a Markov Decision Process (MDP), which provides a mathematical framework for modelling decision-making in environments where outcomes are partly random and partly under the control of the decision-maker (Bellman, 1957). An MDP is defined by the tuple (S, A, P, R, γ), where S is the set of states, A is the set of actions, P is the state transition probability function, R is the reward function, and γ is the discount factor.

The agent interacts with the environment by observing the current state s_t, selecting an action a_t, and then receiving a reward $R(s_t, a_t)$ while transitioning to a new state s_{t+1} (Sutton & Barto, 2018). The transition probabilities and rewards depend only on the current state and action, embodying the Markov property.

3.2.2 Value Functions and Bellman Equations

To evaluate the quality of states and actions, RL utilizes value functions. The state-value function $V^{\pi}(s)$ represents the expected value when starting from state *s* and following policy π thereafter. It is defined as:

$$V^{\pi}(s) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_t = s]$$

Where E_{π} denotes the expected value under policy π and $\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$ represents the cumulative discounted reward from time $t = 0$ to infinity.

Similarly, the action-value function $Q^{\pi}(s, a)$ represents the expected return after taking action *a* in state *s* and thereafter following policy π:

$$Q^{\pi}(s, a) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_t = s, a_t = a]$$

The Bellman expectation equation expresses the relationship between the value of a state and the values of its possible successor states under a specific policy π:

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \left[R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) V^{\pi}(s') \right]$$

The Bellman optimality equation characterizes the optimal value function $V^*(s)$, representing the maximum expected return achievable from state *s*.

$$V^*(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) V^*(s') \right]$$

$V^*(s)$ is the optimal state-value function, representing the highest expected return from state *s*. The equation selects the action *a* that maximizes the expected return considering immediate rewards and future values. It forms the basis for finding the optimal policy π^* that maximizes the expected return from every state. The Bellman equation is a fundamental concept in RL. It provides a way of solving sequential decision-making processes by recursively breaking them down into smaller sub-problems (Bellman, 1957).

3.2.3 Neural Networks in Reinforcement Learning

DRL integrates neural networks into RL algorithms to handle high-dimensional inputs and approximate complex functions (Mnih, Badia, Mirza, Graves, Lillicrap, Harley, Silver & Kavukcuoglu, 2016). Artificial Neural Networks (ANNs) are used to approximate the policy and value functions, enabling the agent to learn optimal strategies in complex environments like financial markets.

3.2.4 Multilayer Perceptrons and Backpropagation

Multilayer Perceptrons (MLPs) are a class of feedforward neural networks consisting of an input layer, one or more hidden layers, and an output layer (Goodfellow, Bengio & Courville, 2016). Each layer is composed of neurons that apply a weighted sum of inputs followed by an activation function. The MLPs capture non-linear relationships in the data, which is essential for modelling the complexities of financial markets.

The learning process involves adjusting the weights of the network to minimize a loss function, typically using backpropagation and gradient descent (Goodfellow, Bengio & Courville, 2016). The backpropagation algorithm computes the gradient of the loss function with respect to each weight by applying the chain rule, allowing efficient updates to the network parameters.

3.2.5 Activation Functions

Activation functions introduce non-linearity into neural networks, enabling them to model complex patterns. Two common activation functions used in DRL networks are the hyperbolic tangent (tanh) function and the Rectified Linear Unit (ReLU).

The tanh activation function, as described by Goodfellow, Bengio and Courville (2016) maps output values to a range between -1 and 1, expressed as:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

This makes it useful for data centered around zero. However, they note that tanh is more prone to the vanishing gradient problem, slowing learning in deep networks. In contrast, the ReLU function, shown as:

$$ReLU(x) = \max(0, x)$$

is computationally efficient and helps mitigate the vanishing gradient problem but, as they explain, can result in inactive neurons for negative inputs which stop contributing to the learning process, known as the "dying ReLU" problem.

These functions allow the DRL models to capture complex non-linear relationships in the financial data, enhancing their ability to adjust portfolios dynamically.

3.2.6 Backpropagation in Neural Networks

Backpropagation is the algorithm used to train neural networks by minimizing the loss function (Goodfellow, Bengio & Courville, 2016). It involves the following steps:

1. **Forward Pass**: Compute the output of the network by passing the input data through each layer.
2. **Compute Loss**: Calculate the loss function L based on the difference between the predicted output and the actual target.
3. **Backward Pass**: Compute the gradients of the loss function with respect to each weight using the chain rule.
4. **Update Weights**: Adjust the weights using an optimization algorithm like stochastic gradient descent (SGD):

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial L}{\partial w_{ij}}$$

where η is the learning rate. This iterative process allows the network to learn complex patterns in the data, improving its performance over time.

By integrating RL with neural networks, the agent can learn optimal portfolio allocation strategies that adapt to changing market conditions. The mathematical foundations provided by MDPs, value functions, and Bellman equations enable the agent to make informed decisions that maximize cumulative rewards. Activation functions like ReLU and Tanh enhance the network's ability to capture non-linear relationships in financial data. Through backpropagation and appropriate reward functions, the agent continuously improves its performance, potentially outperforming traditional portfolio optimization methods.

3.2.7 Actor-Critic Networks

Actor-Critic networks are a RL architecture that integrates two key components: the actor, which determines the actions to take, and the critic, which evaluates those actions by estimating a value function (Konda & Tsitsiklis, 1999). The actor updates its policy based on feedback from the critic, who provides a measure of the expected cumulative reward for the chosen action. This approach effectively combines the strengths of value-based and policy-based methods, offering more stable learning and improved sample efficiency. By reducing the variance in policy gradient estimates, Actor-Critic networks are particularly useful in environments with continuous action spaces, such as in portfolio optimization (Sutton & Barto, 2018). Additionally, these networks enable better use of available data, which is crucial in complex state-action spaces (Mnih et al. 2016).

Figure 1 shows the flow of an Actor-Critic network. The actor that takes actions and the critic evaluates how those actions are rewarded. The state represents the market environment at time t, including features such as historical stock prices and current portfolio holdings. The actor, or policy network, takes the state as input and decides the action. The action is executed, influencing the market environment, leading to a new state S_{t+1} and generating a reward R_{t+1}. The Critic network evaluates the action by predicting its value, comparing the predicted value to the actual reward to calculate an advantage. In the training cycle, the Actor and Critic are continuously updated based on the advantage, refining the strategy to maximize cumulative returns over time.

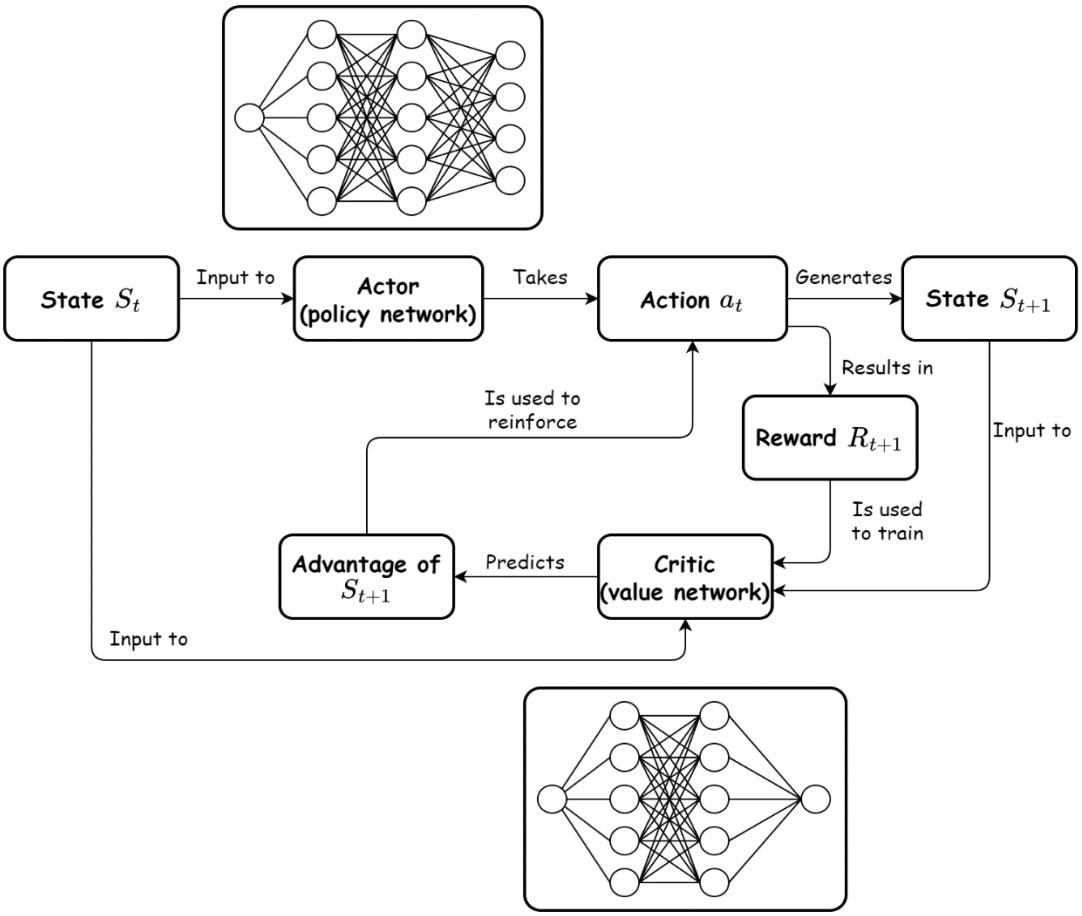

Figure 1. Illustration of components of a simple Actor-Critic network. Source (고민수, 2024)

3.2.8 Reward Functions

The choice of reward function in DRL models is important, as it defines the agent's objective (Sutton & Barto, 2018). A straightforward approach is to use the portfolio return as the reward function. The portfolio returns R_t at time t is defined as:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

where P_t is the portfolio value at time t. This reward function incentivizes the agent to increase the portfolio value by maximizing the returns from its investment decisions. However, in portfolio optimization it is not returns but instead risk-adjusted returns that is evaluated. The SR is a widely used metric in portfolio optimization because it incorporates both return and risk, measuring risk-adjusted returns. However, while effective for evaluating the risk-adjusted performance of static investment strategies, it is not well-suited for DRL

models. This is because the standard SR provides delayed feedback, as it is calculated over a fixed period, which prevents the agent from receiving timely updates necessary for adaptive learning. Additionally, its lack of differentiability with respect to the agent's actions complicates the optimization process, making it difficult for the model to learn effectively. To address these limitations, Moody et al. (1998) introduced the DSR, which is designed to be more dynamic and responsive to real-time changes in the market.

The DSR measures the incremental change in the SR, providing immediate feedback that is crucial for DRL models that learn and adapt continuously. It is calculated using exponential moving averages of the first and second moments of returns. The formula for the DSR is:

$$D_t \equiv \frac{\partial S_t}{\partial \eta} = \frac{B_{t-1}\Delta A_{t-1}\Delta B_t}{B_{t-1} - A_{t-1}^2}$$

Where A_t and B_t are the exponentially weighted moving averages of the returns R_t and the squared returns R_t^2, respectively. These moving averages are updated as follows:

$$A_t = A_{t-1} + \eta \Delta A_t = A_{t-1} + \eta(R_t - A_{t-1})$$

$$B_t = B_{t-1} + \eta \Delta B_t = B_{t-1} + \eta(R_t^2 - B_{t-1})$$

The smoothing factor η controls the rate at which these averages adapt to new data, with smaller values leading to slower adaptation and larger values leading to more rapid changes. This approach allows DRL agents to receive real-time, risk-adjusted feedback, enabling more effective learning and adaptation in non-stationary environments (Moody et al. 1998).

4 Data Collection and Preprocessing

The primary dataset consisted of historical stock prices from the S&P 500 index, covering the period from 2004-06-01 to 2024-05-31. The dataset included daily open, low, high, and close (OHLC) prices for all 500 constituent companies in the index. This dataset provides a broad representation of the U.S. stock market, making it a good foundation for portfolio optimization analysis.

For the robustness test, data was collected from the largest Exchange-Traded Funds (ETFs) by market capitalization within each sector of the S&P 500. This approach ensures that the robustness of the portfolio optimization model is tested across different sectors, which may exhibit varying market dynamics. Similar to the primary dataset, daily OHLC prices were used to maintain consistency in data frequency and structure.

In addition to the raw price data, technical indicators were derived from the closing prices of both the S&P 500 stocks and the sector-specific ETFs.

4.1 Technical Indicators

Technical analysis, which uses statistical properties in historical data to identify patterns that may provide insights into future price movements, is a common practice in trading. Since the DRL agent is not fed with data on earnings, news, sentiment, or other fundamental factors, it relies only on technical analysis for pattern recognition. Technical indicators are tools that help identify trends, potential reversals, and the strength of price movements. While the technical indicators are derived from the same stock price data (open, high, low, close) that the agent already uses, they transform this data into more interpretable signals which makes it easier for the agent to recognize patterns and thus learn faster.

Jang and Seong (2023) found that including a wide range of technical indicators in their models improved their performance, especially when the indicators represented different market characteristics. This supports the approach of including a diverse set of indicators to capture a more comprehensive view of the market, which is why we have included many different technical indicators. The technical indicators used as inputs in the DRL environments are MA30, MA60, Bollinger bands, Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Average Directional Index (ADX), Commodity Channel Index (CCI), and Turbulence Index (TI). These are explained more in Appendix C:

Technical Indicators. Figure 2 shows an example of technical indicators for one of the stocks used as input to the Actor-Critic networks in the DRL models. Exactly how the DRL models use the information from each of the technical indicator is difficult to get an understanding of since they search for patterns that may be non-linear and difficult to interpret.

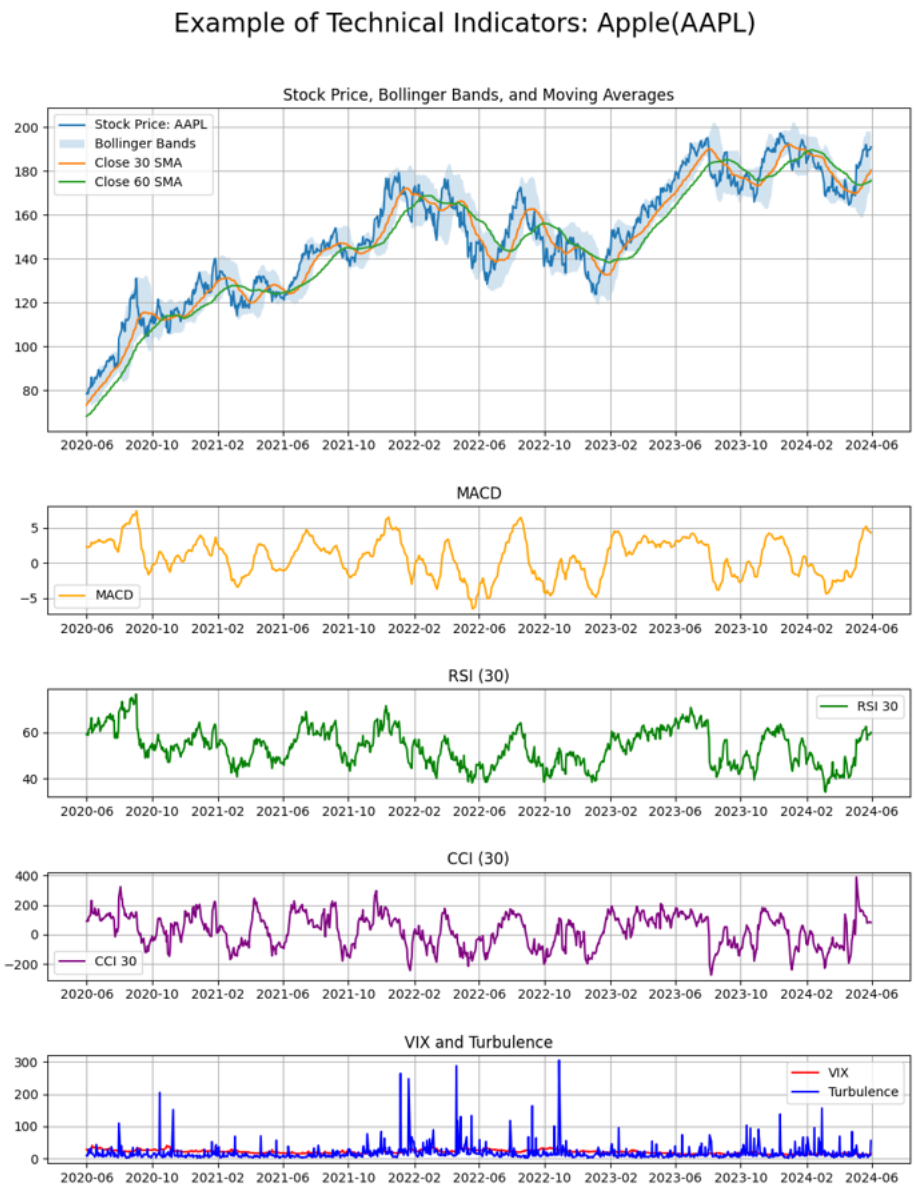

Figure 2. Technical Indicators for Apple during the testing period 2020-2024

5 Methodology

In this study, we aimed to construct a portfolio that maximizes risk-adjusted returns using RL models, with the S&P 500 as the initial asset pool. While the ideal scenario would involve selecting from all global assets to maximize diversification and risk-adjusted returns, practical constraints such as data availability, currency conversion, and increased transaction costs make it more feasible to focus on large-cap stocks from a well-established index like the S&P 500. We further narrowed our selection to 384 stocks that had complete data for the entire period from 2004 to 2024.

To implement the MVO model, we first needed to estimate the expected returns, volatility, and covariances of the 384 stocks. These estimates were derived from historical data within the training period. While this approach is widely used in portfolio optimization studies, it is one of several methods available for these estimations. Alternative methods, such as those proposed by Ledoit and Wolf (2004) for covariance matrix estimation, can be particularly useful in contexts where increased accuracy is required. However, for the purpose of this study which is to compare the relative performance of DRL models with a traditional portfolio optimization approach, the use of historical data is both practical and sufficiently appropriate. This approach ensures consistency across all models, as both the MVO and DRL models are trained on the same dataset, providing a robust basis for comparison.

Given the complexity of optimizing portfolios with a large number of assets, particularly in high-dimensional spaces where DRL models may struggle to perform effectively, we needed to reduce the set of assets to a manageable size while still approximating a portfolio that closely mirrors an MVO-constructed portfolio. To achieve this, we used the MVO model to determine the weights of each of the 384 S&P 500 stocks. The stocks with the highest MVO-assigned weights were selected for the final portfolio, under the assumption that these stocks would provide the closest approximation to a mean-variance optimal portfolio. We selected exactly 16 stocks because the weights dropped significantly between the 16th and 17th stock. Specifically, these 16 stocks constituted more than 99,9% of the portfolio's allocation. This significant concentration of weight in the top 16 stocks justified their selection for further optimization with the RL models. Table 1 shows the weights proposed by the Maximum Sharpe-selected portfolio and Table 2 shows tickers and company names.

Table 1. Selected stocks from MVO-optimized weights

Max Sharpe Selected Stocks			
Selected stocks and their weights			
Number	Location	Ticker	Weight
1	1	AAPL	0.1817578000
2	68	CHD	0.1501100000
3	234	MNST	0.1029559000
4	251	NFLX	0.0743018500
5	133	EW	0.0740286900
6	349	TYL	0.0732484600
7	170	HRL	0.0610751000
8	308	SBAC	0.0518724700
9	186	ISRG	0.0484451800
10	368	WEC	0.0446359800
11	175	HUM	0.0430507700
12	52	BKNG	0.0345118400
13	375	WST	0.0205963200
14	296	REGN	0.0198526000
15	39	AZO	0.0097602700
16	360	VRTX	0.0095238450
17	221	MCD	0.0002716028
18	249	NEE	0.0000005789
19	311	SHW	0.0000001933
20	105	DLTR	0.0000001904
21	213	LMT	0.0000001068
22	235	MO	0.0000000972
23	73	CLX	0.0000000965
24	26	AMZN	0.0000000500
25	86	CPRT	0.0000000487

Table 2. Stock tickers and company names

Max Sharpe selected stock Portfolio

Ticker	Full Name
AAPL	Apple Inc.
CHD	Church & Dwight Co., Inc.
MNST	Monster Beverage Corporation
NFLX	Netflix, Inc.
EW	Edwards Lifesciences Corporation
TYL	Tyler Technologies, Inc.
HRL	Hormel Foods Corporation
SBAC	SBA Communications Corporation
ISRG	Intuitive Surgical, Inc.
WEC	WEC Energy Group, Inc.
HUM	Humana Inc.
BKNG	Booking Holdings Inc.
WST	West Pharmaceutical Services, Inc.
REGN	Regeneron Pharmaceuticals, Inc.
AZO	AutoZone, Inc.
VRTX	Vertex Pharmaceuticals Incorporated

5.1 The MVO Benchmark

In the next step, the MVO model was also used to construct a portfolio with these 16 stocks, optimized to maximize the SR. Since we now only included 16 stocks in the portfolio, the portfolio's weights were normalized to 100%. This is the benchmark that we used to compare the performance of the DRL models against.

Figure 3 visualizes the share of the selected stocks according to the MVO at the start of the trading period. For the static MVO, the weights remained steady during the entire trading period.

The portfolio that the DRL models optimized for Sharpe ratio performance, also contained the same 16 stocks. This ensures a fair comparison since they could only choose from the same selection of stocks. The aim of the DRL models is thereby to find optimal weights that maximized performance using the same stocks. At the start of the training process, they started with equal weights in the portfolio.

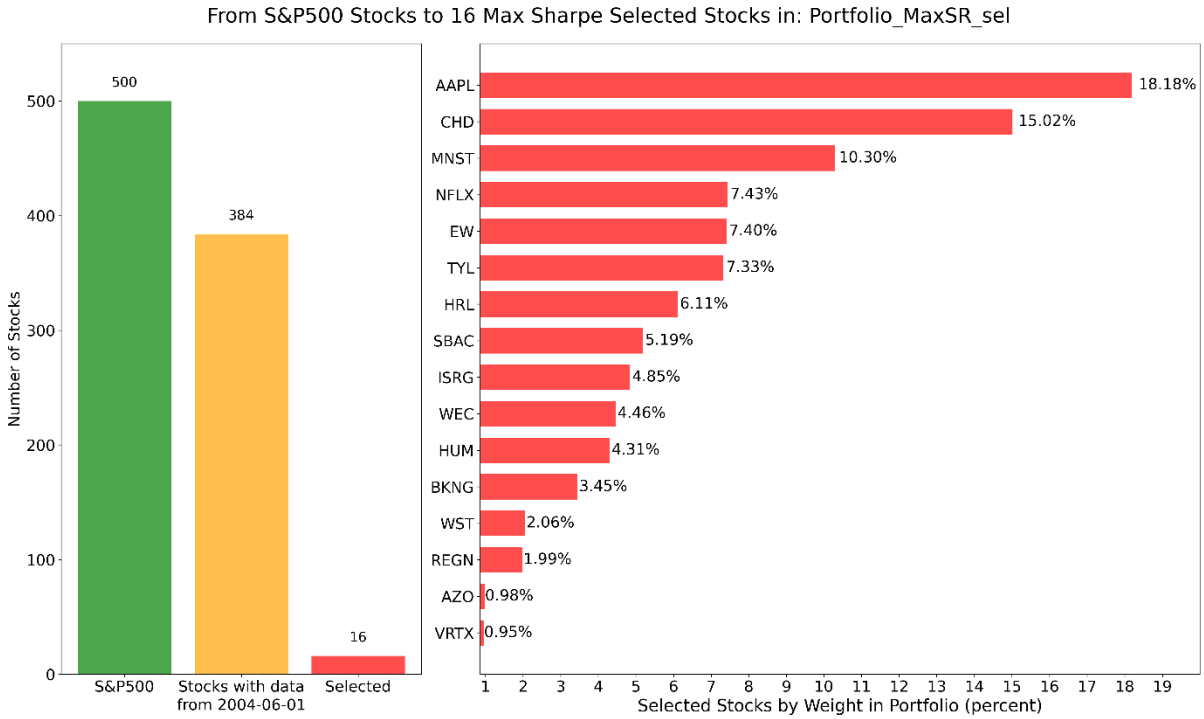

Figure 3. Selection of 16 stocks from S&P 500 based on Maximum Sharpe

For a robustness test, we also created another portfolio consisting of 11 ETFs, one for each sector in the S&P 500. The largest ETF in each sector in terms of market cap was selected. The list of ETFs can be found in Table 11 in Appendix A: ETF and Minimum Variance portfolios. Using assets with different risk profiles for the analysis shows if the DRL models can perform consistently with different data.

Another portfolio we created for the robustness test was a Minimum Variance portfolio, also starting from the 384 S&P 500 stocks with full data. The first step was now guided by selecting the 16 stocks which minimized the portfolio variance. This process was done in a similar way as the selecting process mentioned before. In Appendix A: ETF and Minimum Variance portfolios, a list of the selected stocks can be found in Table 12 and the weights in Figure 10.

Figure 4 shows two portfolios on the efficient frontier, one with Maximum SR and one with Minimum Variance. These are two of the portfolios that was used.

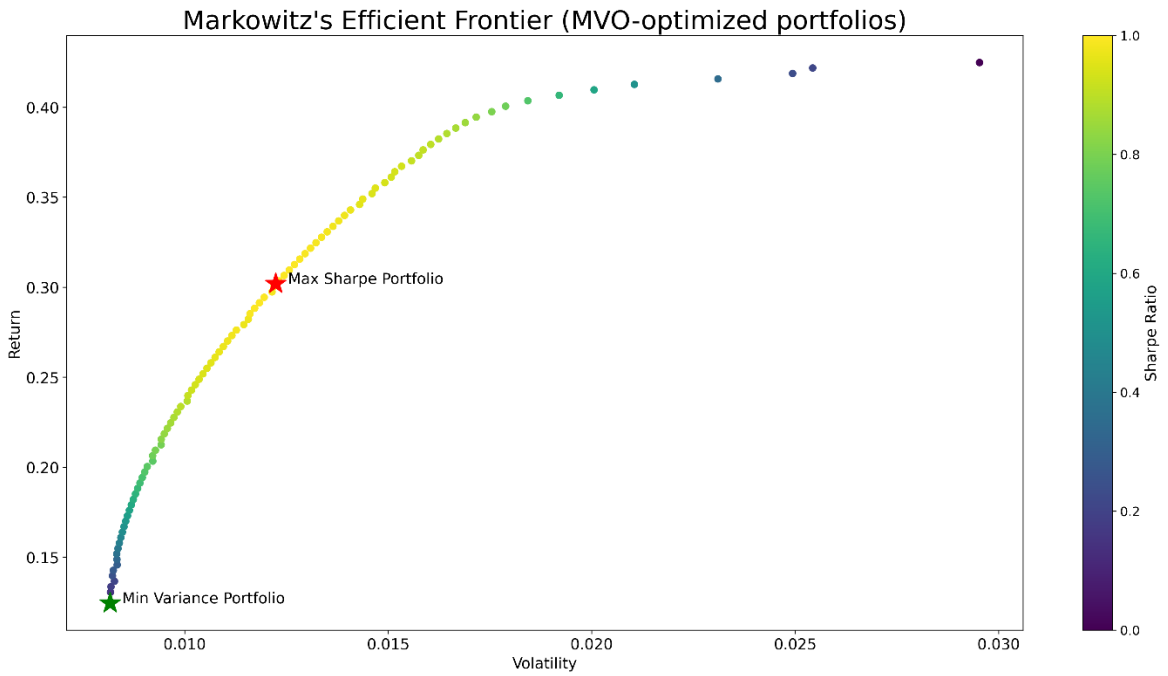

Figure 4. Portfolios on the efficient frontier based on optimizing S&P 500 stocks using historical data, highlighting our two chosen portfolios

To train the DRL agents, we first split our dataset into a training set, a validation set and a test set (as shown in Figure 5) that was hidden for later out-of-sample evaluation. Then the hyperparameter tuning was made on the training set and validated on the validation set. After the models were trained, we used the optimal hyperparameters and retrained the models on both the training set and validation set. In the last step, when the Actor-Critic networks had been trained, they were used on the out-of-sample data. The outcome was then evaluated against the benchmark performance.

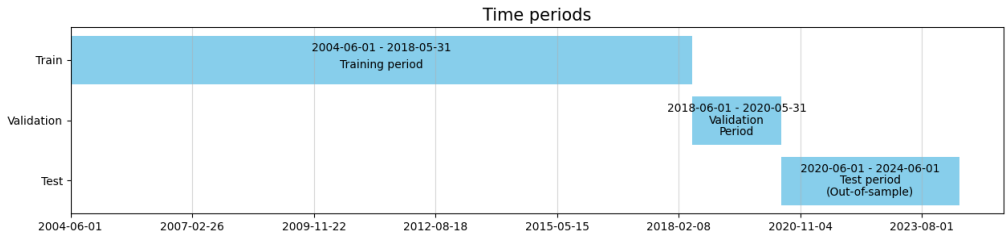

Figure 5. Data split in three time periods for the DRL models

Because of the stochastic behaviour of the DRL agents, we ran each experiment 10 times for each model and calculated the average values, like Jang and Seong (2023). Those averages were compared to the benchmark portfolio and presented in the results section.

5.2 Actor-Critic Networks Used in This Study

The 5 DRL models use Actor-Critic networks. A description of these in general is presented first, and then the architectures for each one specifically used in this study is shown.

5.2.1 A2C

The Asynchronous Advantage Actor-Critic (A3C) model introduced by Mnih et al. (2016) reduces policy gradient variance using an advantage function. A2C, a synchronous version, runs multiple parallel environment instances while using a single actor and critic network to improve training efficiency.

5.2.2 DDPG

Deep Deterministic Policy Gradient (DDPG) is an off-policy, model-free network that uses an experience replay to store transitions observed in the training set (Lillicrap et al. 2015). The actor network deterministically maps states to actions, with the states in this context being historical stock prices and technical indicators, and the actions representing the allocation of capital in each asset. The critic network evaluates the actions taken by the agent by estimating the Q-value.

5.2.3 PPO

Proximal Policy Optimization (PPO) is a policy gradient method by Schulman, Wolski, Dhariwal, Radford and Klimov (2017) designed for stable training by clipping the probability ratio in policy updates, preventing excessive deviations. In-high dimensional and continuous action spaces, such as those in portfolio optimization, PPO´s conservative approach helps reduce the risk of overfitting or forgetting learned strategies.

5.2.4 SAC

Soft Actor-Critic is an off-policy Actor-Critic model by Haarnoja, Zhou, Abbeel and Levine (2018) that maximizes both reward and entropy, thereby promoting exploration and improving sample efficiency. SAC performs well in tasks requiring a balance between exploration and exploitation, but excessive exploration can be computationally costly, necessitating careful control.

5.2.5 TD3

Twin Delayed Deep Deterministic Policy Gradient (TD3), introduced by Fujimoto, van Hoof and Meger (2018), builds on DDPG by using a double critic (Q)-network to mitigate overestimation bias, yielding more accurate value estimates. This feature helps avoid overly optimistic predictions that can lead to poor portfolio decisions.

5.2.6 Actor-Critic Architectures

Table 3 shows that the implemented DRL models have different characteristics. A2C and PPO share identical, lightweight architectures. In contrast, DDPG and TD3 employ more complex structures and larger hidden layers using ReLU-Tanh combinations.

Table 3. The 5 DRL models' architectures

Model Architectures	DRL Models				
	A2C	DDPG	PPO	SAC	TD3
Actor Network					
Number of Networks	1	1	1	1	1
Trainable parameters	14528	189916	14528	115488	189916
Input Layer	161x64	161x400	161x64	161x256	161x400
Hidden layer(s)	64x64	400x300	64x64	256x256	400x300
Activation function(s)	2xTanh	2xReLU+Tanh	2xTanh	2xReLU	2xReLU+Tanh
Output layer	- *(softmax)*	300x16	-	2x256x16	300x16
Critic Network(s)					
Number of Networks	1	1	1	2	2
Trainable parameters	14528	191801	14528	223234	383602
Input Layer	161x64	177x400	161x64	177x256	2x177x400
Hidden layer(s)	64x64	400x300	64x64	256x256	2x400x300
Activation function(s)	2xTanh	2xReLU	2xTanh	2xReLU	2x2xReLU
Output layer	64x1	300x1	64x1	2x256x1	2x300x1

5.3 Hyperparameter Tuning

When using DRL models with Actor-Critic networks, configuring hyperparameters is crucial for optimizing performance, as they dictate the model's learning dynamics and balance between short-term and long-term rewards (Sutton & Barto, 2018). Poorly chosen hyperparameters can result in slow convergence or ineffective policies, while well-tuned parameters enhance learning and performance. Although grid search is a common approach to hyperparameter tuning, it can be computationally expensive due to the exhaustive search process (Akiba, Sano, Yanase, Ohta & Koyama, 2019). Young, Hinkle, Kannan and Ramanathan (2020) explored Bayesian optimization as an advanced alternative, demonstrating its efficiency in optimizing complex models like DRL. This approach is especially relevant in financial contexts where resource constraints and sample inefficiency

are common. This study utilizes the Optuna framework for hyperparameter tuning, which leverages Bayesian optimization to explore the hyperparameter space intelligently. Additionally, Optuna employs pruning techniques to terminate unpromising trials early, thereby reducing computational cost and speeding up the optimization process.

We applied hyperparameter optimization to the DRL models. In Appendix D: Hyperparameter tuning, the hyperparameter tuning process and the improvement it yielded is shown for A2C as a representative example. The hyperparameter values after optimization with Optuna is presented for all the DRL models in Table 4. We used these values in all the portfolios, both the main test and the robustness test, to ensure a consistent setup for fair comparison. By keeping the hyperparameters constant, we can more accurately assess the impact of the portfolio composition on the models' performance without the confounding effects of varying hyperparameters.

Table 4. Hyperparameters after tuning for each DRL model

Hyper parameters	DRL Models				
	A2C	DDPG	PPO	SAC	TD3
Time steps	20000	30000	50000	30000	20000
Initial amount	1M	1M	1M	1M	1M
Turbulence threshold	70	70	70	70	70
hmax	100	100	100	100	100
Learning rate	8.7575E-01	3.0513E-03	1.9815E-05	3.8721E-05	2.8680E-05
gamma	0.99	0.999	0.99	0.999	0.98
n_steps	256	-	1024	-	-
ent_coef	1.1566E-08	-	1.7204E-05	5.0000E-02	-
buffer_size	-	10000	-	10000	-
batch_size	-	128	-	128	256
clip_range	-	-	0.4	-	-
Polyak coeff, tau	-	-	-	0.05	-

- **Time steps:** The number of interactions between the agent and the environment.
- **Initial amount:** The starting capital (in USD) at the beginning of the test period.
- **Turbulence threshold:** The threshold for selling all assets when the turbulence index is high, aiming to avoid potential market crashes.
- **hmax:** The maximum number of shares that can be traded in a single transaction.
- **Learning rate:** Regulates the speed of learning, significantly impacting performance.
- **Gamma (discount factor):** A parameter between 0 and 1 that determines the weight given to future rewards relative to immediate rewards. A higher gamma places greater emphasis on future rewards.

- **n_steps:** The number of steps the agent takes in the environment before updating its policy, controlling how many experiences are collected prior to a learning update.
- **ent_coef (entropy coefficient):** Controls the exploration-exploitation trade-off by influencing the randomness of the agent's actions—higher values promote more exploration by increasing policy entropy.
- **buffer_size:** Defines the maximum number of past experiences (state-action-reward-next_state transitions) stored in the replay buffer for training the agent through random sampling.
- **batch_size:** Specifies the number of experiences sampled from the replay buffer during each training step to update the agent's networks.
- **clip_range:** Limits the magnitude of policy updates by clipping the ratio between new and old policies, preventing destabilizing changes.
- **Tau (Polyak coefficient):** Controls the update speed of the target network in SAC through soft updates—a smaller tau results in slower, more stable updates by adjusting the mixing of the current network's weights with the target network's weights.

5.4 Reward Function Implementations

For each of the constructed portfolio, the Maximum Sharpe portfolio in the main test and the Minimum Variance and ETF portfolio in the robustness test, two different reward functions were used. One without DSR which optimized on portfolio value and one with DSR which optimized on risk-adjusted returns. By doing so, we seek to get insights into how DRL models' performance compare to the benchmark, and if performance can be improved when optimizing for risk-adjusted returns directly.

5.5 Tools Used for Programming and Data Processing

5.5.1 Python

Python served as the primary programming language for implementing and evaluating DRL models. Libraries like NumPy and Pandas facilitated data manipulation and numerical operations, while TensorFlow/PyTorch enabled the construction and training of neural networks. These libraries allowed for efficient handling of large datasets and the rapid iteration of model architectures tailored to portfolio optimization tasks.

5.5.2 FinRL

The Financial Reinforcement Learning (FinRL) framework provided prebuilt financial environments for developing and testing DRL algorithms specifically within financial

contexts (Liu, Yang, Chen, Zhang, Yang, Xiao & Wang, 2020). In this thesis, FinRL was adapted for a customized trading environment based on historical stock price data from the S&P 500. It also included the data collection process, as historical data was retrieved directly from Yahoo Finance, and supported various DRL algorithms. Additionally, FinRL's examples with Optuna facilitated hyperparameter tuning, optimizing model parameters. Extensions were made to include the DSR as the reward function. FinRL also supplied examples in Jupyter notebooks, which helped developing additional resources for this study.

5.6 Trade Assumptions

Transactions costs were assumed to be 0,1% for both buying and selling stocks. Furthermore, it was assumed that the trades have zero market impact and zero slippage. The risk-free rate was assumed to be 2%.

6 Empirical Results

In this section the results from the evaluation period tests will be presented, comparing the performance of the DRL models against the MVO in five performance metrics, with the main focus on Sharpe Ratio (SR).

Figure 6 and Figure 7 shows the portfolio value over time during the test period for the value-based and the DSR-based portfolio, respectively. The main takeaway is that the curves for the DRL models are diverting on the positive side compared to the MVO curve, showing that the DRL models have performed better than MVO in terms of annual returns.

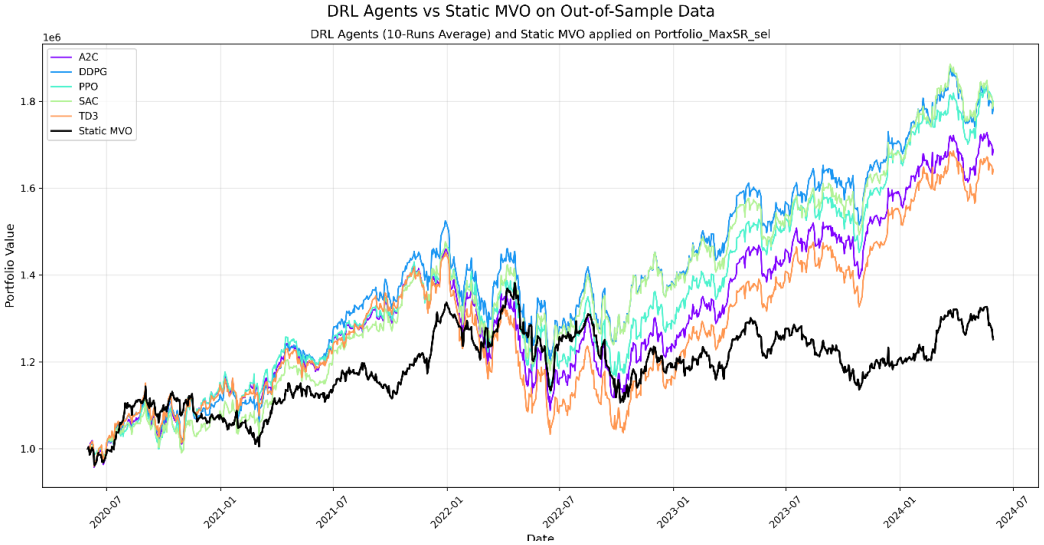

Figure 6. Graph showing portfolio values for Static MVO and the DRL models using portfolio value as reward function, applied on the Maximum Sharpe portfolio stocks. Averages over 10 runs for the DRL models

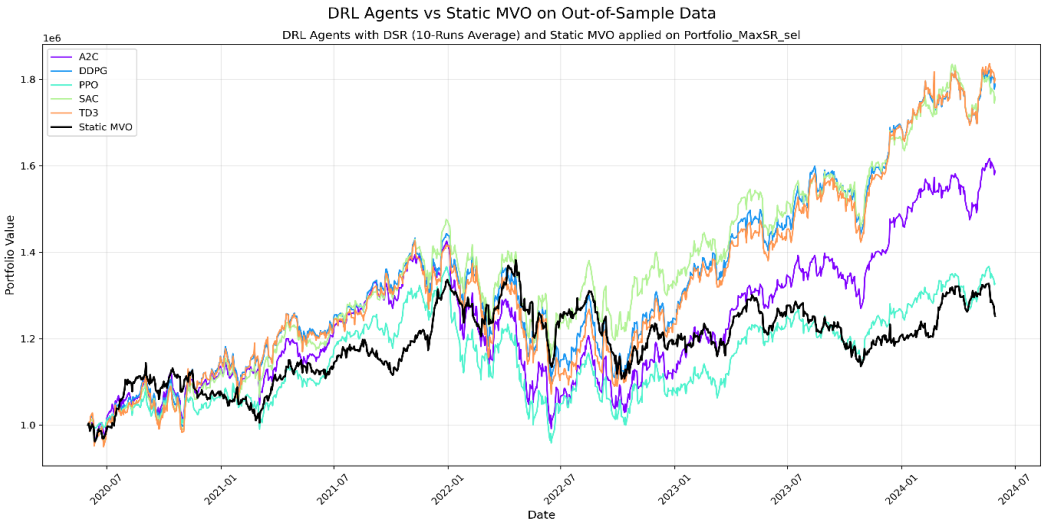

Figure 7. Graph showing portfolio values for Static MVO and the DRL models using DSR as reward function, applied on the Maximum Sharpe portfolio stocks. Averages over 10 runs for the DRL models

Although portfolio value development is always of interest when analysing performance, the main comparison in this study relates to performance in risk-adjusted returns. Figure 8 and Figure 9 presents the out-of-sample performance of the various DRL models, both with and without the inclusion of the DSR as the reward function. The primary performance metric used for comparison is the SR, with results visualized through boxplots that illustrate the mean, variance, and outliers across 10 runs for each model. The results show that all DRL models without DSR outperformed the MVO-constructed portfolio in terms of SR. When DSR was employed as the reward function, models like TD3, SAC, and DDPG continued to outperform the MVO portfolio, although DDPG exhibited some negative outliers. A2C, while achieving a higher average SR, showed greater variability across runs. PPO, contrary to expectations, underperformed compared to the MVO portfolio.

Notably, DRL models using DSR did not show clear superiority over those optimized on portfolio value. Four of the models had lower SR, although the largest and most complex model, TD3, showed a higher SR and lower variability.

Further, Table 5 and Table 6 shows some additional performance metrics. The DRL models, both without and with DSR, except for PPO with DSR, performed better than the MVO constructed portfolio in terms of Sortino ratio, but the volatility and maximum drawdown were higher for all.

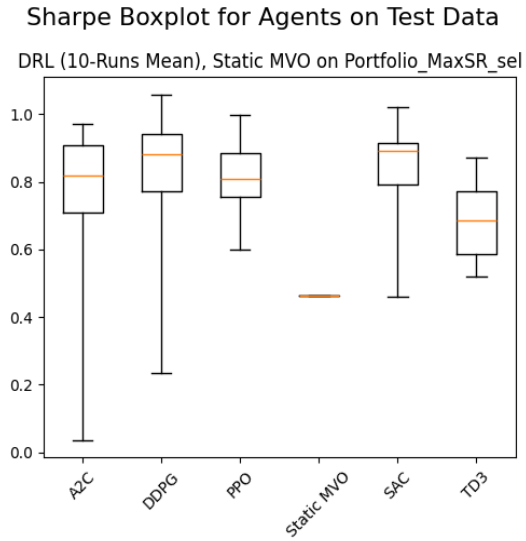

Figure 8. Maximum Sharpe portfolio boxplots for the 10-run average, interquartiles and outliers for the portfolios created by the 5 DRL models, using portfolio value as reward function

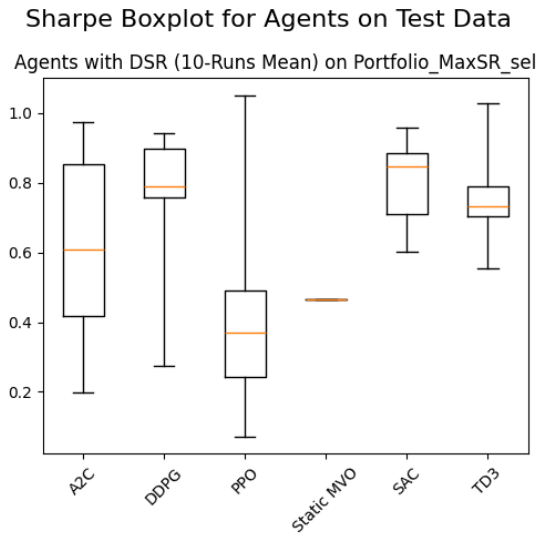

Figure 9. Maximum Sharpe portfolio boxplots for the 10-run average, interquartiles and outliers for the portfolios created by the 5 DRL models using DSR as reward function

Table 5. Maximum Sharpe portfolio performance metrics for the 5 DRL models and the MVO, using portfolio value as reward function

	A2C	DDPG	PPO	SAC	TD3	Static_MVO
Annual return	0.1342	0.1531	0.1576	0.1567	0.1318	0.0578
Annual volatility	0.2139	0.1994	0.2109	0.195	0.2208	0.1435
Max drawdown	-0.2882	-0.2437	-0.2684	-0.2335	-0.3142	-0.1992
Sharpe ratio	0.704	0.8086	0.811	0.8474	0.6815	0.4641
Sortino ratio	1.0002	1.1539	1.169	1.2068	0.9618	0.6451

Table 6. Maximum Sharpe portfolio performance metrics for the 5 DRL models and the MVO, using DSR as reward function

	A2C	DDPG	PPO	SAC	TD3	Static_MVO
Annual return	0.1184	0.1546	0.0689	0.151	0.1578	0.0578
Annual volatility	0.2272	0.217	0.2098	0.1996	0.2312	0.1435
Max drawdown	-0.3426	-0.2725	-0.3431	-0.2299	-0.2844	-0.1992
Sharpe ratio	0.6174	0.7769	0.4262	0.8081	0.7639	0.4641
Sortino ratio	0.8759	1.1099	0.6017	1.1489	1.0968	0.6451

6.1 Robustness Test

While comparing portfolios using MVO as a benchmark with those created by DRL models provide valuable insights, the results cannot be considered a complete truth or generalized across all markets and conditions. To assess the reliability and broader applicability of these findings, a robustness test is conducted by applying the models to both an ETF portfolio and a Minimum Variance portfolio.

The results showed some variation, not fully confirming the main results. A2C stands out with strong performance across most metrics, whereas PPO underperforms in all metrics compared to the MVO portfolio.

For the ETF portfolio, the performance metrics for the DRL models without DSR indicate that these models outperform the MVO portfolio in terms of annual return, SR, and Sortino Ratio, albeit with higher volatility and maximum drawdown, as shown in Table 7. When using DSR as the reward function on the ETF portfolio, the DRL models showed similar performance compared to those without DSR, except for PPO, which demonstrated lower annual returns and SR. This can be seen in Table 8. The large TD3 model still shows higher SR and lower variability, which can be seen in Figure 15 and Figure 16 in Appendix B: Supplementary results from the robustness test.

On the Minimum Variance portfolio in Table 9 and Table 10, the results were less definitive. DDPG, SAC, and TD3 produced results comparable to the MVO portfolio, particularly in

terms of annual return and SR. Overall, DRL models using DSR as the reward function seem to perform similarly to those without DSR on the Minimum Variance portfolio.

Table 7. ETF portfolio performance metrics for the 5 DRL models and the MVO, using portfolio value as reward function

	A2C	DDPG	PPO	SAC	TD3	Static_MVO
Annual return	0.1685	0.1453	0.1187	0.1366	0.1368	0.0927
Annual volatility	0.188	0.1716	0.1632	0.1745	0.1759	0.1293
Max drawdown	-0.2475	-0.2214	-0.2187	-0.2324	-0.2444	-0.1558
Sharpe ratio	0.9254	0.8768	0.7786	0.8416	0.8254	0.7513
Sortino ratio	1.3384	1.2643	1.1168	1.2055	1.1871	1.0503

Table 8. ETF portfolio performance metrics for the 5 DRL models and the MVO, using DSR as reward function

	A2C	DDPG	PPO	SAC	TD3	Static_MVO
Annual return	0.1374	0.1395	0.1069	0.1314	0.1563	0.0927
Annual volatility	0.1663	0.1646	0.158	0.1654	0.185	0.1293
Max drawdown	-0.2149	-0.2166	-0.2197	-0.2145	-0.2592	-0.1558
Sharpe ratio	0.8596	0.8746	0.7203	0.839	0.8772	0.7513
Sortino ratio	1.233	1.258	1.0312	1.2012	1.26	1.0503

Table 9. Minimum Variance portfolio performance metrics for the 5 DRL models and the MVO, using DSR as reward function

	A2C	DDPG	PPO	SAC	TD3	Static_MVO
Annual return	0.0851	0.0699	0.0222	0.0636	0.0625	0.0611
Annual volatility	0.134	0.1391	0.1562	0.1376	0.1412	0.1242
Max drawdown	-0.1505	-0.1638	-0.2718	-0.1701	-0.1889	-0.1512
Sharpe ratio	0.6723	0.5584	0.2232	0.518	0.502	0.5404
Sortino ratio	0.9601	0.7906	0.3117	0.7351	0.7086	0.7539

Table 10. Minimum Variance portfolio performance metrics for the 5 DRL models and the MVO, using portfolio value as reward function

	A2C	DDPG	PPO	SAC	TD3	Static_MVO
Annual return	0.0857	0.0689	0.04	0.0676	0.0577	0.0611
Annual volatility	0.1432	0.1387	0.1472	0.1406	0.1451	0.1242
Max drawdown	-0.1808	-0.1671	-0.206	-0.1902	-0.2008	-0.1512
Sharpe ratio	0.6459	0.5465	0.344	0.5292	0.4677	0.5404
Sortino ratio	0.9195	0.7727	0.4902	0.7518	0.6628	0.7539

6.2 Discussion of Results

In the results analysis, we observed that DRL models generally outperformed the MVO. This outcome aligns with previous research by Yang et al. (2020) and Gu et al. (2023), who also found that DRL models surpassed benchmark models in terms of risk-adjusted returns. The results indicated that the DRL models achieved both higher Sharpe and Sortino ratios compared to traditional methods, although with a higher maximum drawdown. This pattern mirrors the findings of Benhamou et al. (2020), suggesting that while DRL models can enhance risk-adjusted performance, they may also introduce increased downside risk.

The higher maximum drawdowns observed could imply that DRL models adopt more aggressive investment strategies, potentially yielding higher returns, but also exposing the portfolio to greater volatility and potential losses. This trade-off between improved risk-adjusted returns and increased downside risk warrants careful consideration. It indicates that DRL models, despite their potential for superior performance, may require robust risk management techniques to mitigate potential losses.

The robustness tests generally supported these observations, with DRL models outperforming the MVO-constructed portfolio in metrics such as the SR, annualized return, and Sortino ratio, particularly in the ETF portfolio. However, the performance was mixed in the Minimum Variance portfolio. This variation suggests that the effectiveness of DRL models may depend on the portfolio strategy employed. While DRL models tend to achieve higher returns and better risk-adjusted performance, they also exhibit higher volatility and maximum drawdowns, exposing the portfolio to greater market fluctuations and potential losses.

These findings indicate that Maximum Sharpe portfolios might offer a more conducive learning environment for DRL models. The ability of DRL models to dynamically adjust to changing market conditions may be less impactful in the context of Minimum Variance portfolios, where low-risk assets dominate. This suggests that the potential of DRL models could be best realized when applied to portfolio strategies that balance risk and reward, providing robust signals for adaptive strategy development.

The observation that DRL models do not clearly outperform the MVO approach for the Minimum Variance portfolio is noteworthy and may indicate limitations in the models' learning effectiveness in this context. One possible explanation relates to the fundamental assumptions of the MVO benchmark. The Maximum Sharpe portfolio relies on historical

returns, volatility, and correlations to estimate future performance. If historical returns are unreliable predictors of future returns, this reliance could lead to suboptimal asset selection during the testing period.

In contrast, the Minimum Variance portfolio focuses solely on minimizing portfolio variance without considering expected returns. This approach relies on historical volatility and correlations, which might be more stable and reliable predictors of future volatility than historical returns are of future returns. If we assume that past volatility is indeed a more consistent indicator, the Minimum Variance portfolio may perform more consistently during the testing period compared to the Maximum Sharpe portfolio. This could explain why the MVO benchmark might produce better results with the Minimum Variance selected stocks and why the DRL models do not clearly outperform it in this context.

Another factor to consider is that the composition of the Minimum Variance portfolio, predominantly featuring low-volatility stocks, may limit the DRL models' ability to detect and exploit market anomalies. Low-volatility environments reduce the magnitude of price movements, making it more challenging for the DRL models, particularly those utilizing Actor-Critic networks, to identify mispriced assets or patterns that could be leveraged for higher returns. This challenge is compounded by the inclusion of transaction costs in our analysis. With reduced potential for significant gains in low-volatility stocks, the costs associated with frequent trading become more prohibitive, rendering the strategy less effective.

This limitation might explain why our results differ from those reported by Sood et al. (2023). In their study, the absence of transaction costs and possibly a different asset selection could have provided a more conducive environment for the DRL models to outperform traditional methods. Therefore, the underperformance of the Minimum Variance portfolio relative to the DRL models in our study suggests that factors such as asset volatility, transaction costs, and the inherent assumptions of the optimization methods play crucial roles in determining the effectiveness of DRL models in portfolio management.

An interesting observation is that DRL models without DSR generally achieved higher Sharpe ratios compared to those with DSR, despite DSR being specifically designed to optimize risk-adjusted returns. Similar results were found by Sadighian (2020), who also reported inconsistent performance for the DRL models using DSR. This outcome suggests that models may have struggled to identify the necessary signals for optimizing the SR when DSR was

used as the reward function. One possible explanation is the complexity introduced by the DSR reward function. Balancing both return and volatility increases the dimensionality and interdependencies between assets, which might hinder the models' ability to generalize effectively during the learning process, especially in the presence of noisy financial data. The added complexity can make it challenging for DRL models to learn optimal strategies, as they must navigate a more intricate reward landscape.

This hypothesis is further supported by the performance of the TD3 model. TD3, being the largest and most complex DRL model in our study, was the only one that performed better with DSR than without it in the Maximum Sharpe and ETF portfolios. TD3's extensive Actor-Critic network, with a higher number of neurons, might be better equipped to capture the complex patterns required when using DSR as the reward function. However, even TD3 did not perform as well with DSR in the Minimum Variance portfolio, suggesting that the advantages of using DSR may be context-dependent and reliant on the model's capacity to handle increased complexity.

In contrast, simpler models like A2C did not exhibit improved performance with DSR, indicating that the benefits of DSR may only be realized when paired with models capable of managing the added complexity. The stochastic nature of the training process and the inherent noise in financial data could further contribute to the observed variance in outcomes, as these factors make it more difficult for less complex models to converge on an effective strategy.

These observations suggest that while DSR is theoretically advantageous for optimizing risk-adjusted returns, its practical implementation may require sufficiently complex models to harness its potential. This underscores the importance of aligning the choice of DRL model and reward function with the specific characteristics of the portfolio and the computational resources available.

The performance of individual Actor-Critic networks showed some consistency across different datasets but also revealed variations. A2C and DDPG demonstrated strong, consistent performance across different portfolios, while SAC and TD3 generally performed well but with more variability. PPO, on the other hand, exhibited inconsistency, performing well in some contexts but poorly in others, particularly in the Minimum Variance portfolio. PPO may have been disfavoured because we used constant hyperparameter values for all tests, and as Henderson, Islam, Bachman, Pineau, Precup and Meger (2018) showed, it often requires careful tuning to adapt to varying environments.

7 Conclusions

This study compared advanced DRL models with the traditional MVO approach in portfolio optimization, focusing on risk-adjusted returns. Our findings indicate that DRL models generally outperformed the MVO benchmark in terms of SR, particularly in the Maximum Sharpe and ETF portfolios. The DRL models achieved higher Sharpe and Sortino ratios compared to traditional methods, although they exhibited higher maximum drawdowns. This suggests that while DRL models can enhance risk-adjusted performance, they may also introduce increased downside risk, necessitating robust risk management techniques.

However, the outperformance of DRL models was not consistent across all portfolio strategies. In the Minimum Variance portfolio, DRL models did not clearly surpass the MVO approach. This inconsistency may be attributed to limitations in the models' learning effectiveness when dealing with low-volatility assets and the impact of transaction costs. The composition of the Minimum Variance portfolio, predominantly featuring low-volatility stocks, may have limited the DRL models' ability to detect and exploit market anomalies, reducing their effectiveness compared to MVO in this context.

Regarding the reward functions, our study found that DRL models optimized using portfolio value generally performed better than those using the goal-oriented DSR. Only the most complex DRL model, TD3, showed improved performance with DSR in certain portfolios. This outcome suggests that the benefits of DSR may require more sophisticated models to be fully realized and that the added complexity does not consistently lead to better performance across all models and portfolio strategies.

The inclusion of transaction costs in our analysis underscored the importance of factoring in real-world constraints when evaluating model performance. Transaction costs significantly affected both DRL and MVO strategies, particularly in low-volatility environments where potential gains are modest. This suggests that frequent rebalancing, a characteristic of DRL models, may be penalized in practical settings, and strategies need to account for these costs to remain effective.

Overall, our study indicates that while DRL models hold significant potential for enhancing portfolio optimization, their effectiveness is contingent upon factors such as portfolio strategy, model complexity, reward function, and transaction costs. DRL models appear to be more

effective in portfolios that balance risk and reward, providing robust signals for adaptive strategy development. The practical implementation of DRL models may require careful alignment of the model architecture and reward function with the specific characteristics of the portfolio and consideration of real-world constraints like transaction costs.

7.1 Future Research

In a broader perspective, these findings contribute to the evolving understanding of how DRL can be harnessed for portfolio optimization. While DRL holds significant potential as a powerful tool in dynamic financial environments, its success depends on careful model selection and reward design tailored to specific investment goals and market conditions.

Future research could integrate a wider range of data sources that influence stock prices. Incorporating macroeconomic indicators, fundamental financial data such as earnings and financial ratios, and sentiment analysis from news and social media could enhance the models' predictive power and responsiveness to market dynamics, enabling DRL models to navigate complexities more effectively.

Another avenue is to explore the incorporation of hedging strategies to achieve more stable growth by balancing assets that respond differently to various market conditions. Expanding the number of assets in the portfolio could offer diversification benefits but also introduces challenges, such as increased estimation errors and a more complex covariance matrix. To address these, techniques like shrinkage estimators or Bayesian methods could be employed to improve the robustness of expected return and covariance estimates, potentially enhancing the effectiveness of both DRL models and traditional approaches like MVO in larger asset pools.

Finally, investigating how DRL models perform over different investment horizons could provide insights into their suitability for various investor profiles. Analysing the effectiveness of DRL strategies in long-term investment scenarios versus short-term trading may necessitate adjustments to the reward function or state representation to align the models with specific investment horizons. By exploring these avenues, future research can contribute to developing more robust and adaptable DRL models, thereby advancing the field of portfolio optimization.

References

Aboussalah, A. M. & Lee, C. G. (2020). Continuous Control with Stacked Deep Dynamic Recurrent Reinforcement Learning for Portfolio Optimization, *Expert Systems with Applications*, vol. 140, p.112891

Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. (2019). Optuna: A Next-Generation Hyperparameter Optimization Framework, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/1907.10902 [Accessed 22 July 2024]

Almahdi, S. & Yang, S. Y. (2017). An Adaptive Portfolio Trading System: A Risk-Return Portfolio Optimization Using Recurrent Reinforcement Learning with Expected Maximum Drawdown, *Expert Systems with Applications*, vol. 87, pp.267–279

Bellman, R. E. (1957). Dynamic Programming, Princeton, NJ: Princeton University Press

Benhamou, E., Saltiel, D., Ungari, S. & Mukhopadhyay, A. (2020). Bridging the Gap between Markowitz Planning and Deep Reinforcement Learning, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/2010.09108 [Accessed 16 April 2024]

Cornuejols, G. & Tütüncü, R. (2006). Optimization Methods in Finance, Vol. 5, Cambridge University Press

Fujimoto, S., van Hoof, H. & Meger, D. (2018). Addressing Function Approximation Error in Actor-Critic Methods, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/1802.09477 [Accessed 12 August 2024]

Goodfellow, I., Bengio, J. & Courville, A. (2016). Deep Learning, Cambridge, MA: MIT Press

Gu, J., Du, W., Muntasir Rahman, A. M. & Wang, G. (2023). Margin Trader: A Reinforcement Learning Framework for Portfolio Management with Margin and Constraints, in *ICAIF 2023 - 4th ACM International Conference on AI in Finance*, 2023, Brooklyn, NY, pp.610–618

Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, in *35th International Conference on Machine Learning, ICML 2018*, Vol. 5, 2018, Stockholm, Sweden, Available Online: https://proceedings.mlr.press/v80/haarnoja18b/haarnoja18b.pdf [Accessed 27 April 2024]

Harvey, C. R. & Siddique, A. (2000). Conditional Skewness in Asset Pricing Tests, *The Journal of Finance*, vol. 55, no. 3, pp.1263–1295

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D. & Meger, D. (2018). Deep Reinforcement Learning That Matters, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1

Jang, J. & Seong, N. Y. (2023). Deep Reinforcement Learning for Stock Portfolio Optimization by Connecting with Modern Portfolio Theory, *Expert Systems with Applications*, vol. 218, 119556

Jiang, Z., Xu, D. & Liang, J. (2017). A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/1706.10059 [Accessed 6 May 2024]

Konda, V. R. & Tsitsiklis, J. N. (1999). Actor-Critic Algorithms, in *Proceedings of the 12th Annual Conference on Neural Information Processing Systems (NeurIPS 1999)*, November 1999, Denver, CO, USA, Available Online: https://papers.nips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf [Accessed 11 May 2024]

Ledoit, O. & Wolf, M. (2004). Honey, I Shrunk the Sample Covariance Matrix, *Journal of Portfolio Management*, vol. 30, no. 4, pp.110–119

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. & Wierstra, D. (2015). Continuous Control with Deep Reinforcement Learning, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/1509.02971 [Accessed 27 April 2024]

Liu, X.-Y., Xia, Z., Yang, H., Gao, J., Zha, D., Zhu, M., Wang, C. D., Wang, Z. & Guo, J. (2023). Dynamic Datasets and Market Environments for Financial Reinforcement Learning, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/2304.13174 [Accessed 1 May 2024]

Liu, X.-Y., Yang, H., Chen, Q., Zhang, R., Yang, L., Xiao, B. & Wang, C. D. (2020). FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/2011.09607 [Accessed 18 April 2024]

López de Prado, M. (2018). Advances in Financial Machine Learning, Hoboken,NJ: Wiley

Ma, Y., Han, R. & Wang, W. (2021). Portfolio Optimization with Return Prediction Using Deep Learning and Machine Learning, *Expert Systems with Applications*, vol. 165, 113973

Markowitz, H. (1952). Portfolio Selection, *The Journal of Finance*, vol. 7, no. 1, pp.77–91

Michaud, R. O. (1989). The Markowitz Optimization Enigma: Is 'Optimized' Optimal?, *Financial Analysts Journal*, vol. 45, pp.31–42

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D. & Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning, in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 19 June 2016, New York, NY, USA, pp.1928–1937

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/1312.5602 [Accessed 22 September 2024]

Moody, J., Wu, L., Liao, Y. & Saffell, M. (1998). Performance Functions and Reinforcement Learning for Trading Systems and Portfolios, *Journal of Forecasting*, vol. 17, no. 5–6, pp.441–470

Pástor, Ľ. & Stambaugh, R. F. (1999). Costs of Equity Capital and Model Mispricing, *Journal of Finance*, vol. 54, no. 1, pp.67–121

Sadighian, J. (2020). Extending Deep Reinforcement Learning Frameworks in Cryptocurrency Market Making, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/2004.06985 [Accessed 22 September 2024]

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. (2017). Proximal Policy Optimization Algorithms, *ArXiv Preprint*, Available Online: https://arxiv.org/pdf/1707.06347 [Accessed 30 April 2024]

Sharpe, W. F. (1966). Mutual Fund Performance, *The Journal of Business*, vol. 39, no. S1, pp.119–138

Sood, S., Papasotiriou, K., Vaiciulis, M., Balch, T., Morgan, J. P. & Research, A. I. (2023). Deep Reinforcement Learning for Optimal Portfolio Allocation: A Comparative Study with Mean-Variance Optimization, in *ICAPS 2023 Workshop on Financial Planning*, July 2023, Prague, Czech Republic, Available Online: https://icaps23.icaps-conference.org/papers/finplan/FinPlan23_paper_4.pdf [Accessed 24 April 2024]

Sutton, R. S. & Barto, A. G. (2018). Reinforcement Learning: An Introduction, *The MIT Press*, 2nd edn, Cambridge, MA: MIT Press

Théate, T. & Ernst, D. (2021). An Application of Deep Reinforcement Learning to Algorithmic Trading, *Expert Systems with Applications*, vol. 173, 114632

Wilder Jr., J. W. (1978). New Concepts in Technical Trading Systems, Greensboro, NC, USA: Trend Research

Yang, H., Liu, X. Y., Zhong, S. & Walid, A. (2020). Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy, in *ICAIF 2020 - 1st ACM International Conference on AI in Finance*, 2020, Article No.31, pp.1–8

Young, M. T., Hinkle, J. D., Kannan, R. & Ramanathan, A. (2020). Distributed Bayesian Optimization of Deep Reinforcement Learning Algorithms, *Journal of Parallel and Distributed Computing*, vol. 139, pp.43–52

고민수고대화. (2024). Deep Learning Bible, *WikiDocs*, Available Online: https://wikidocs.net/175903 [Accessed 2 August 2024]

Appendix A: ETF and Minimum Variance portfolios

Table 11. Full name of the ETFs in the ETF portfolio

Ticker	Full Name
VGT	Vanguard Information Technology Index Fund ETF Shares
XLF	The Financial Select Sector SPDR Fund
XLB	The Materials Select Sector SPDR Fund
XLY	The Consumer Discretionary Select Sector SPDR Fund
XLI	The Industrial Select Sector SPDR Fund
VOX	Vanguard Communication Services Index Fund ETF Shares
XLP	The Consumer Staples Select Sector SPDR Fund
XLE	The Energy Select Sector SPDR Fund
VNQ	Vanguard Real Estate Index Fund ETF Shares
XLU	The Utilities Select Sector SPDR Fund
XLV	The Health Care Select Sector SPDR Fund

Table 12. Stocks selected in the Minimum Variance portfolio

Ticker	Full Name
JNJ	Johnson & Johnson
CLX	The Clorox Company
GIS	General Mills, Inc.
ED	Consolidated Edison, Inc.
WMT	Walmart Inc.
MCD	McDonald's Corporation
TECH	Bio-Techne Corporation
HRL	Hormel Foods Corporation
AJG	Arthur J. Gallagher & Co.
NEM	Newmont Corporation
CAG	Conagra Brands, Inc.
BDX	Becton, Dickinson and Company
CPB	Campbell Soup Company
KR	The Kroger Co.
CHD	Church & Dwight Co., Inc.
MO	Altria Group, Inc.

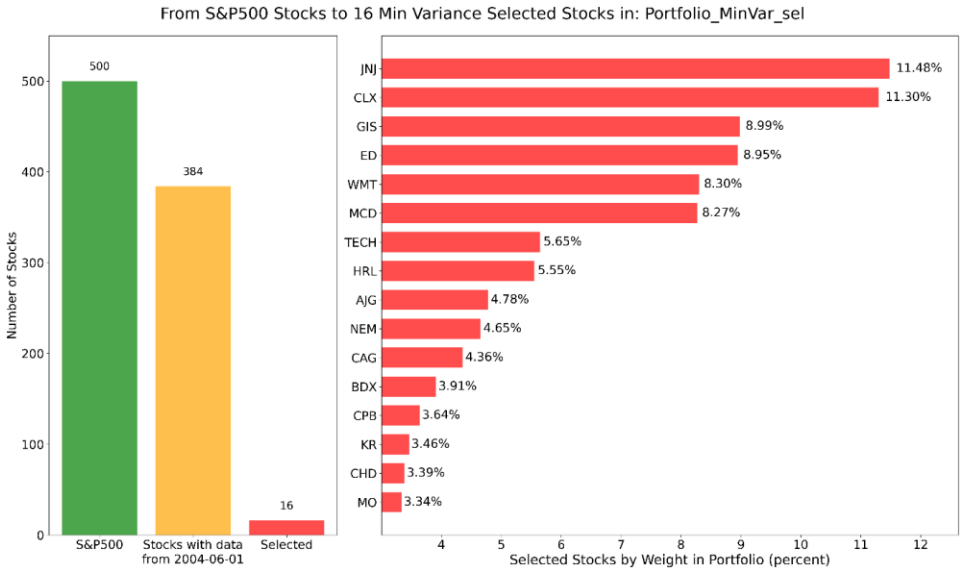

Figure 10. The process of selecting 16 stocks for the Minimum Variance portfolio

Appendix B: Supplementary results from the robustness test

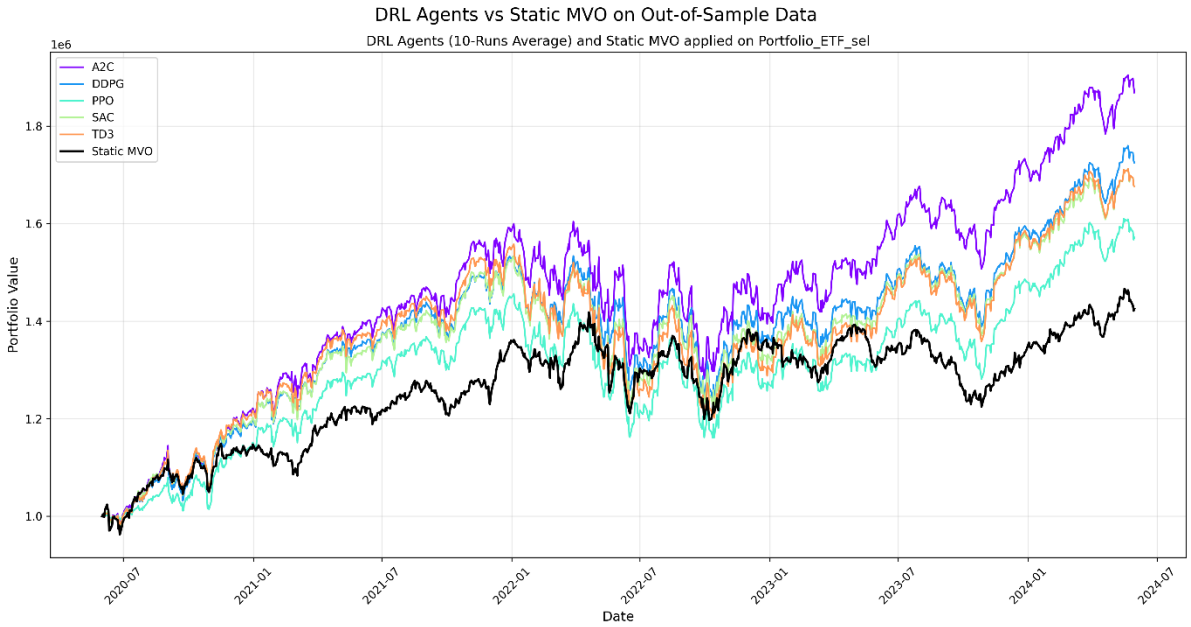

Figure 11. Portfolio values for Static MVO and the DRL models using portfolio value as reward function, applied on the ETF portfolio. Averages over 10 runs for the DRL models.

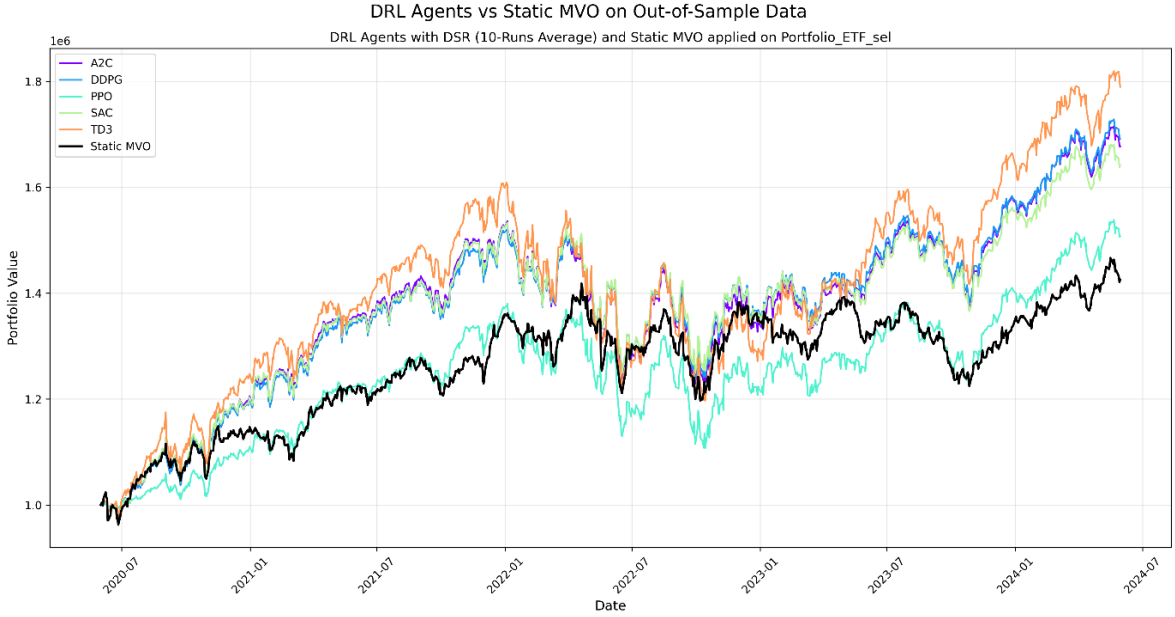

Figure 12. Portfolio values for Static MVO and the DRL models using DSR as reward function, applied on the ETF portfolio. Averages over 10 runs for the DRL models.

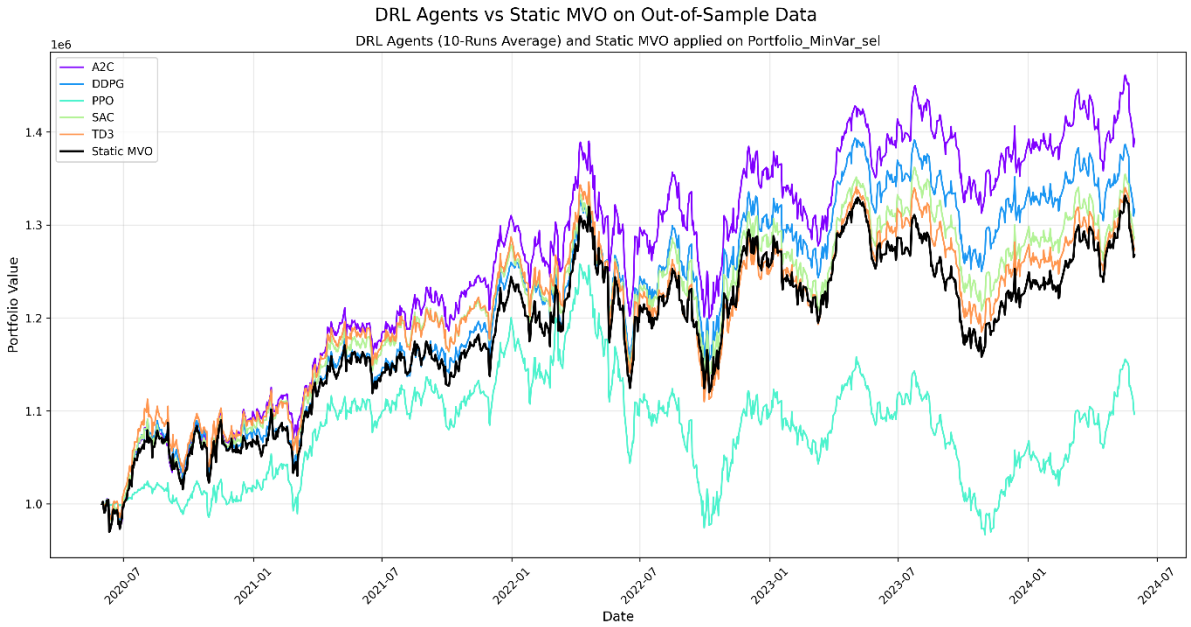

Figure 13. Portfolio values for Static MVO and the DRL models using portfolio value as reward function, applied on the Minimum Variance portfolio stocks. Averages over 10 runs for the DRL models.

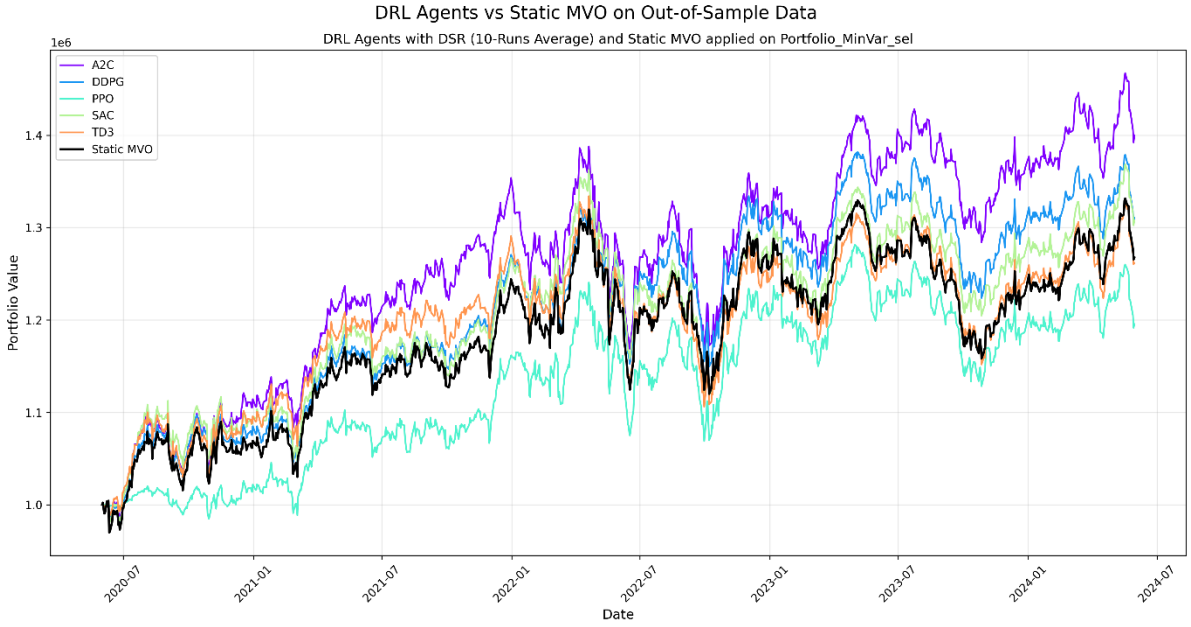

Figure 14. Portfolio values for Static MVO and the DRL models using DSR value as reward function, applied on the Minimum Variance portfolio stocks. Averages over 10 runs for the DRL models.

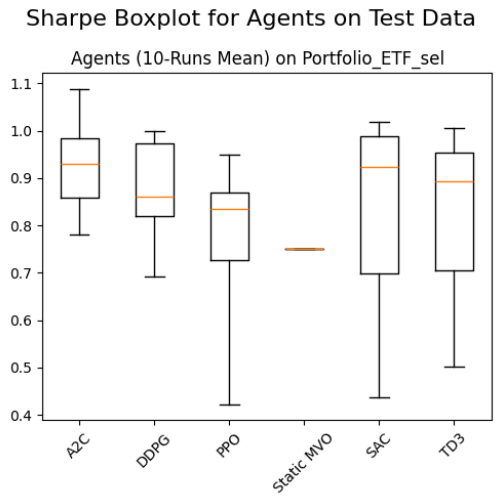

Figure 15. ETF portfolio boxplots for the 10-run average, interquartiles and outliers for the portfolios created by the 5 DRL models, using portfolio value as reward function

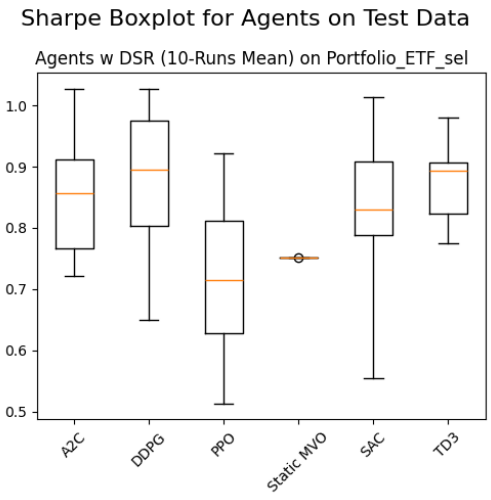

Figure 16. ETF portfolio boxplots for the 10-run average, interquartiles and outliers for the portfolios created by the 5 DRL models, using DSR as reward function

Figure 17. Minimum Variance portfolio boxplots for the 10-run average, interquartiles and outliers for the portfolios created by the 5 DRL models, using portfolio value as reward function,

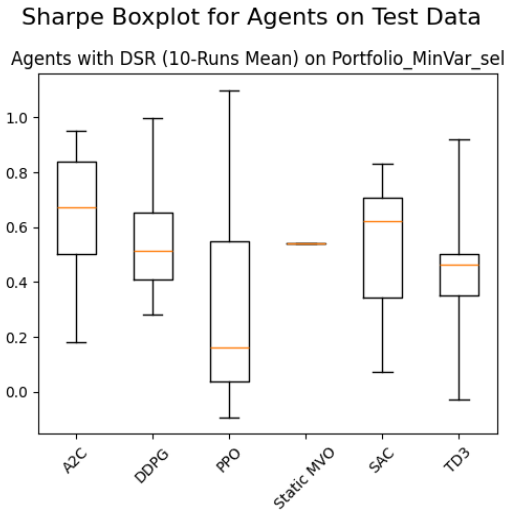

Figure 18. Minimum Variance portfolio boxplots for the 10-run average, interquartiles and outliers for the portfolios created by the 5 DRL models, using DSR as reward function

Appendix C: Technical Indicators

Moving average

The moving average is one of the most well-known technical indicators. It calculates the average price over a certain period, which can carry information on trends and potential trading signals.

$$SMA_n = \frac{P_1 + P_2 + \cdots + P_n}{n}$$

$$EMA_{today} = \left(P_{today} \times multiplier\right) + \left(EMA_{yesterday} \times (1 - multiplier)\right)$$

$$Multiplier = \frac{2}{1 + n}$$

Moving Average Convergence Divergence (MACD)

The MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price. It consists of two lines: the MACD line, which is the difference between the 26-day and 12-day exponential moving averages, and the signal line, which is the 9-day exponential moving average of the MACD line. A third component often included is the histogram, which plots the difference between the MACD line and the signal line. Traders use MACD to identify potential buy and sell signals, which are typically indicated by the MACD line crossing above or below the signal line.

$$MACD = EMA_{12} - EMA_{26}$$

$$Signal\ line = EMA_9(MACD)$$

Relative Strength Index (RSI)

Wilder Jr. (1978) introduced RSI as a momentum oscillator. It typically uses a 14-day timeframe and oscillates between 0 and 100. Lower values, generally below 30, indicate that an asset may be oversold, while higher values above 70 suggest it may be overbought. These extremes suggest a potential trend reversal, particularly when coupled with divergences between the asset's price movement and RSI direction. This highlights the importance of confirming signals with other technical analysis tools.

The RSI is calculated using the formula:

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

$$RS = \frac{Average\ Gain}{Average\ Loss}$$

Average Directional Index (ADX)

Wilder Jr. (1978) introduced the ADX as a tool for assessing the strength of a trend. Higher ADX values indicates stronger trends but does not specify the direction. The ADX is derived from two directional movement indicators: DI+, the Positive Directional Indicator, and DI-, the Negative Directional Indicator. A crossover of DI+ above DI- suggests an upward trend, typically interpreted as a buy signal. Conversely, when DI- crosses above DI+, it suggests a downward trend, indicating a potential sell signal.

$$ADX = 100 * \frac{\sum_{t=1}^{n} |DI^{+} - DI^{-}|}{\sum_{t=1}^{n} (DI^{+} + DI^{-})}$$

Commodity channel index (CCI)

The CCI is an indicator or recent strength and deviations relative to a longer average. It can work either in a trend strategy by identifying upward trend or in a mean-reverting strategy. Values above +100 indicates overbought and values below -100 indicates oversold and a potential imminent price rally. With a mean-reverting strategy low prices combined with a not so low CCI could indicate reversal and high prices but not high CCI could indicate a bearish divergence.

$$CCI = \frac{Typical\ Price - SMA\ of\ Typical\ Price}{0.015 \times Mean\ Deviation}$$

$Typical\ Price$ is the average of low, high and close prices.

Turbulence Index (TI)

The Turbulence Index gives an indication of the current state of the market in terms of stress or instability. Identifying periods of increased volatility could give warnings of higher risk in the market and thereby signalling to investors about caution.

$$TI_t = (R_t - \mu)^T \Sigma^{-1} (R_t - \mu)$$

R_t is the vector of asset returns at time t, μ is the vector of historical mean returns of the assets and Σ^{-1} is the inverse of the covariance matrix of the historical returns.

Appendix D: Hyperparameter tuning

Figure 19 shows the optimization process over multiple trials. The red line represents the best objective value achieved, stabilizing after about 15 trials. The scattered blue dots highlight the variability in individual trial performances, underscoring the need for efficient hyperparameter tuning.

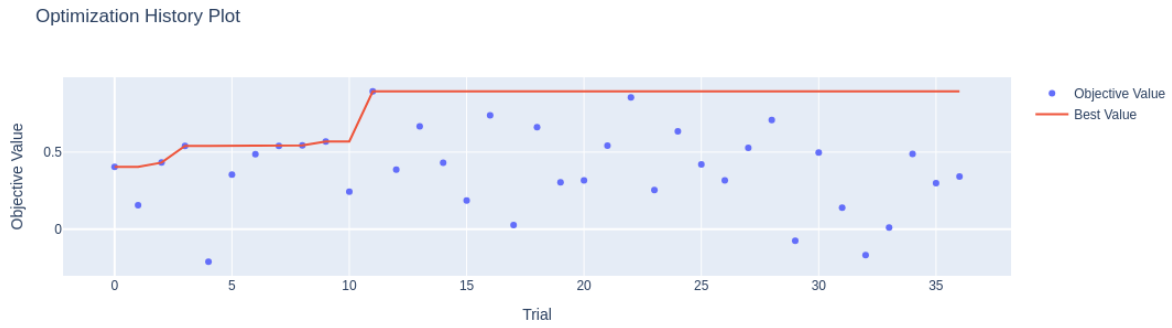

Figure 19. Hyperparameter tuning for the A2C model

Figure 20 illustrates the relative importance of each hyperparameter for the A2C model. The Number of Steps emerged as the most critical parameter, significantly influencing the model's learning process. Learning Rate and Gamma also played important roles in determining performance.

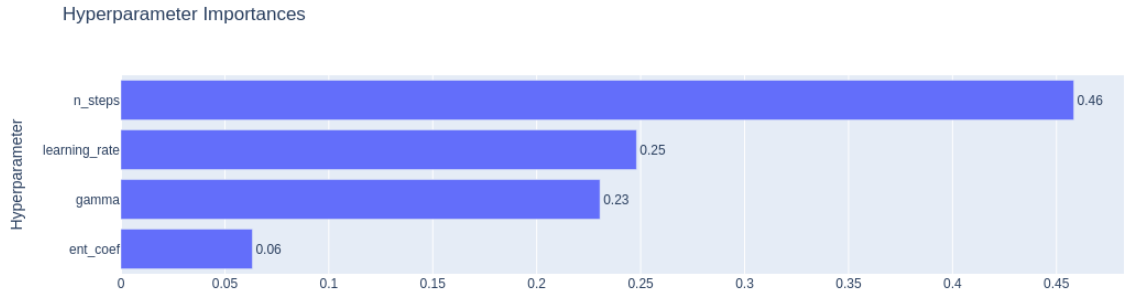

Figure 20. The hyperparameters for the A2C and their relative performance

The optimization process yielded significant improvements as shown in Table 13. This demonstrates the impact of hyperparameter optimization on DRL model performance in stock trading applications. Similar optimization processes were applied to the other DRL models.

Table 13. Metrics before and after hyperparameter optimization for the A2C model

Metric	Before	After	Improvement
Annual Return	7.11%	15.95%	+124%
Cumulative Returns	31.58%	80.65%	+155%
Sharpe Ratio	0.607	0.895	+47%