Benchmarking Strategies for Asset Allocation

Ricard Durall López

rduralll@uoc.edu

28 de 06 2022

Tutor/a: Carolina Natividad Morales Moreno

Treball Final de Màster

Curs 22, semestre 2

INDEX

ABSTRACT

Asset allocation is an investment strategy that aims to balance risk and reward by constantly redistributing the portfolio's assets according to certain goals, risk tolerance, and investment horizon. Unfortunately, there is no simple formula that can find the right allocation for every individual. As a result, investors may use different asset allocations' strategy to try to fulfil their financial objectives. In this work, we conduct an extensive benchmark study to determine the efficacy and reliability of a number of optimization techniques. In particular, we focus on traditional approaches based on Modern Portfolio Theory, and on machine-learning approaches based on deep reinforcement learning. We assess the model's performance under different market tendency, i.e., both bullish and bearish markets, as well as different time frequencies of reallocation., i.e., on a daily and weekly basis.

Keywords:

Asset Allocation, Portfolio Selection, Markowitz Portfolio, Deep Reinforcement Learning, Benchmarking

ABSTRACT

L'assignació d'actius és una estratègia d'inversió que permet equilibrar el risc i el retorn de la cartera mitjançant una redistribució constant dels actius en funció de determinats objectius, tolerància al risc i horitzó d'inversió. Malauradament, no hi ha una fórmula senzilla que permeti trobar l'assignació adient per a cada persona. És per això que existeixen diferents estratègies d'assignació d'actius que els inversors poden utilitzar per intentar assolir les metes financeres. En aquest treball, realitzem un ampli estudi per determinar l'eficàcia i la fiabilitat d'una sèrie de tècniques d'optimització de carteres. En concret, ens centrem en els models tradicionals basats en la teoria moderna de la cartera, i en els models d'aprenentatge automàtic basats en l'aprenentatge profund per reforç. Un cop tenim els models implementats, avaluem el rendiment de cadascun en diferents tendències de mercat, i.e., tant mercats alcistes com a baixistes, així com en diferents freqüències temporals de reassignació, i.e., diàries i setmanals.

Paraules clau:

Assignació d'Actius, Selecció de Cartera, Cartera de Markowitz, Aprenentatge de Reforç Profund, Anàlisi Comparativa

LISTS OF CONTENTS

List of Tables

List of Figures

1. INTRODUCTION

Asset allocation or portfolio selection is the process of finding optimal weights for the different component assets in a portfolio, and it is one of the most significant areas in modern finance. For example, during the first quarter of 2019, the largest 20 companies in the world had a combined Assets Under Management (AUM) of 44.9 trillion of dollars. For this reason, the portfolios' managers need to have a profound a deep understanding of the investment vehicles, the way how these are valued and traded in the financial market, and which strategies can be used to select the investment that should be included in a portfolio to accomplish investors' objectives.

1.1. Objectives

This thesis investigates several approaches to optimize the portfolio's asset allocation. Each of these techniques seeks the weights of the assets so that the portfolio's returns are maximized. To achieve that, the models take into consideration the market's evolution, risk, transaction costs, among other constraints. Traditionally, the researchers have mostly focused on techniques based on financial and risk planning, such as Markowitz model. Nonetheless, in the last years, due to the rise of artificial intelligence, machine-learning-based solutions, especially reinforcement-learning ones, have started to solve decision-making problems, including the asset management. As a result, nowadays, deep-reinforcement-learning algorithms actively contribute to take many finance decisions.

We conduct a benchmark study to determine the relevance and impact of various techniques, where we assess both traditional and deep-learning-based methods, under different day-trading market scenarios. In particular, we study the effect when the market has a bullish tendency, and when the market has a bearish tendency. We hypothesize that market's behaviour can dramatically affect the model's performance, invalidating some approaches. For example, approaches based on Markowitz model tend to perform poorly in markets with high volatility. Furthermore, we provide several finance metrics, such as annual return, annual volatility, Sharpe ratio among others, to better evaluate each algorithm. Finally, we analyse the impact when we reallocate assets on a weekly basis, instead of on daily ones.

1.2. Thesis Organization

This thesis consists of three main parts, not counting the introduction and conclusion. A brief overview of the chapters is given hereunder.

Chapter 2: We start presenting the asset allocation cornerstones; we explain the types of financial assets that currently exist, as well as the strategies and measures to evaluate the portfolio's performance. Then, we provide a detailed literature review, where we revise the evolution of allocation algorithms, from Markowitz (traditional approaches) to deep-learning solutions (state-of-the-art approaches).

Chapter 3: In this chapter, we derive mathematically the Modern Portfolio Theory to gain insight into the underlaying optimization problem, i.e., the optimal asset allocation. Furthermore, we analyse some variations of Markowitz's work, such as the tangent portfolio or the risk parity portfolio. We will refer to such methods as traditional approaches. Next, we introduce the deep-learning framework, namely deep reinforcement learning, which offers an alternative optimization strategy for the asset allocation scenario. Similar to the traditional cases, we also analyse different variations and flavours based on reinforcement algorithms.

Chapter 4: Last but not least, we benchmark a set of optimization methods presented in the latter chapter, including both traditional and reinforcement-based ones. We conduct our experimental

setup on different market's conditions and different reallocation's frequencies. In this fashion, we can better assess the methods' behaviours allowing us to draw a few heuristic conclusions.

2. OVERVIEW OF ASSET ALLOCATION

In this chapter, we define the fundamental pillars of asset allocation. We describe the main classification of financial assets as well as the criteria and evaluation that investors should follow according to their goals, risk tolerances, and investment horizons. Moreover, we revise the literature in this domain, starting from the seminal work from Markowitz up to the present. In this manner, we can provide a grounded evolution of the asset allocation's optimization.

2.1. Types of Financial Assets

An important task in the asset management/allocation process is the selection of the investment vehicles (financial assets) that will make up the portfolio. Often, these financial assets can easily be converted into cash in a short amount of time, and their values are derived from a contractual right or ownership claim of what these assets represent, e.g., a part of a business. Cash, stocks, bond, and mutual funds are all are examples of financial assets. Unlike land, property (real state), commodities, or other physical assets, financial assets are intangible and hence, they do not necessarily have inherent physical worth. Instead, their value reflects factors of supply and demand in the marketplace in which they are traded, and the degree of risk that they intrinsically carry.

Each kind of financial asset has a different goal for the entity that issues it. The major classes of financial assets are cash, equities (stocks) and bonds.

1. Cash and cash equivalents are a type of financial asset with high liquidity. Therefore, they are short-term and easily convertible into cash with higher credit quality. Examples of these assets includes cash money, cheques, and money available in bank accounts.

2. An equity shareholder (or investor) is a fractional owner who decides to undertake the risk associated with the business venture that he invested in. Equity shares are a type of financial assets that give the owners the right to receive the dividends, the right to vote, the right to the capital appreciation of the stock being held, among other rights. However, during liquidation, equity shareholders have the last claim on assets.

3. Bonds (or debentures) are a type of financial asset issued by a company that gives the holders the right to receive regular interest payments on a fixed date along with the principal repayment on maturity. Unlike, dividend on stocks, interest payments on bonds are compulsory even if the company makes a loss. In the event of liquidation, these instrument holders get preference over equity shareholders.

2.2. Selecting a Portfolio Strategy

Although the choice of type of assets plays an important role in the management process, establishing the strategy guidelines (policy) may be even more relevant to satisfy the investors' objectives. Note that for strategy, it is understood the process that decides how it should focus the available resources within a portfolio. However, the selection of the portfolio strategy is not trivial nor unique; usually it would need to be personalized. In general, portfolio strategies can be classified as active or passive. On the one hand, an active portfolio strategy uses available

information and forecasting techniques to seek a better performance than a portfolio that is simply broadly diversified. In other words, all active strategies rely on expectations about the factors that have been found to influence the performance of an asset class. On the other hand, a passive portfolio strategy involves minimal expectational input and relies on diversification to match the performance of some market indexes. In fact, a passive strategy starts from the assumption that market prices impound all available information. Finally, between these two extreme strategies, several possibilities have sprung up combining elements from both.

Factors such as time horizon, current financial situation, and tolerance for market swings, will determine the final choice of the portfolio's strategy (see Figure 1). For example, the more risk an investor can bear, the more aggressive his portfolio should be. As a result, the investor can devote a large portion of his portfolio to equities, and a small portion to bonds and other fixed-income securities. Conversely, if the investor cannot deal with such levels of risk, a more conservative portfolio would be a better match.

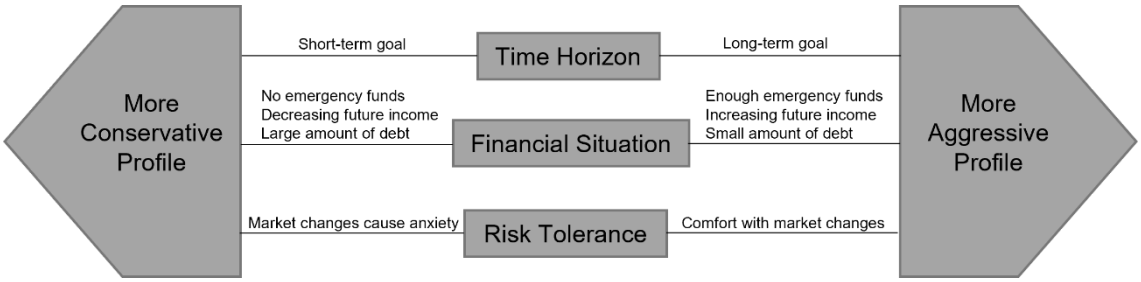

Figure 1: This figure describes the extreme investors' profile based on three key elements: time horizon, financial situation, and risk tolerance. Source prepared by the author.

2.3. Measuring and Evaluating Performance

After selecting a portfolio's strategy, the actual management process begins. This includes monitoring the investments by measuring the portfolio's performance and comparing them to some standardized benchmarks. To that end, it is necessary to report the portfolio's results at regular intervals, typically quarterly, and to review the portfolio's strategy at least once a year. In that revision, it is not only the assets' evolution that are subjected to scrutiny, but also the investor's current situation. In this manner, it can be determined if there is any change, either on the private level or in the market itself, throughout the last twelve months that can potentially affect the portfolio's stability. If this were the case, a new strategy would need to be proposed to suit the new investor's profile.

For those investors that plan for lifelong goals, the portfolio reallocation process will always be required. As investors move through their life stages, in all probability, changes, such as job promotions and dismissals, births and deaths, among others, will occur. Consequently, their portfolio will need to be constantly readjusted to match the new financial situation, risk-reward profile, or horizons. External factors like market or economic conditions would also dictate the reallocation process, making the whole process begins anew.

2.4. Literature Review

Asset allocation is an investment strategy that aims to balance risk and reward by apportioning a portfolio's assets according to an individual's financial situation, risk tolerance, investment horizon, and goals. Fixed asset allocation refers to the portfolio that remains the same until the investor, or the portfolio's manager on behalf of the investor, decides to change the portfolio. The "60/40 portfolio" is a well-known fixed allocation strategy that has been employed as a trusty guidepost for moderate risk investors. It essentially consists of allocating 60% of the portfolio to equities, and the other 40% to bonds and other fixed-income instruments. Another popular fixed asset allocation is the equal weight method. This strategy gives the same importance to each asset in the portfolio, independent of external events. Although fixed allocation approaches have been revered for their simplicity and reliability, the field of asset allocation still has a lot of room for improvement.

The mean-variance optimization model, proposed by Markowitz (Markowitz, 1968) serves as the keystone to Modern Portfolio Theory (MPT). It is a practical method for selecting investments to maximize their overall returns within an acceptable level of risk; the Sharpe ratio (Sharpe, 1998) is one of its most employed add-ons to measure the risk-adjusted return with respect to the risk-free asset. However, Markowitz model applies historical return and volatility as a proxy for future expectations in the allocation model. Consequently, the model can make inaccurate assumptions since the returns and variances will likely not be the same in the future (Merton, 1980). (Black & Litterman, 1992) solved this issue by applying an equilibrium return, based partially on Capital Asset Pricing Model (CAPM) (Sharpe, 1964) (Lintner, 1975), as a baseline for defining the expected return vector. Further alternatives are the equal volatility portfolio, that uses the same amount of volatility in every asset; minimum variance portfolio, (Haugen & Baker, 1991) (Chopra & Ziemba, 2013), that provides the lowest variance among all possible portfolios of risky assets; maximum diversification portfolio (Choueifaty & Coignard, 2008) (Choueifaty, et al., 2013), that maximizes the ratio of weighted-average asset volatilities to portfolio volatility; maximum decorrelation portfolio (Christoffersen, et al., 2012), that maximizes the diversification ratio based on the correlation matrix; and risk parity (Maillard, et al., 2010) (Roncalli & Weisang, 2016), that uses the concept of the Security Market Line (SML) as part of its approach; being SML a graphical representation of the CAPM. Besides, the debatable assumption about the stationarity in time of the market, another drawback from Markowitz's approach appears when the amount of different assets is large enough. MPT uses the covariance to determine which assets are included in the portfolio. This statistical measure has quadratic growth with the number of assets and thus, those portfolios with rich diversity of assets will inevitably suffer from computational problems. To circumvent this issue, (Bai, et al., 2009) proposed to employ the theory of the large-dimensional random matrix. Finally, another reason for poor performance of the mean-variance portfolio might be caused by the symmetry of asset returns. (Low, et al., 2016) showed that it is possible to enhance Markowitz's portfolio selection by allowing distributional asymmetries.

With the increasing use of artificial intelligence, deep-learning approaches have achieved remarkable breakthroughs leading to state-of-the-art results in various domains such as computer vision (Krizhevsky, et al., 2012) (Goodfellow, et al., 2014), natural language processing (Vaswani, et al., 2017) (Devlin, et al., 2018), and speech recognition (Deng, et al., 2013) (Chiu, et al., 2018). Such remarkable success has also sparked the interest of the finance research community. As a result, in the past years, the number of deep-learning applications for portfolio asset allocation has dramatically increased. (Lin, et al., 2006) designed a dynamic portfolio selection model by incorporating the Recurrent Neural Network (RNN) (Rumelhart, et al., 1985). (Freitas, et al., 2009) (Niaki & Hoseinzade, 2013) (Nguyen, et al., 2015) (Heaton, et al., 2017) relied on deep neural networks to model the market's behaviour so that the final solution found the optimal asset allocation. A similar approach by (Chakravorty, et al., 2018) dealt with macroeconomic data in conjunction with price-volume data in a walk-forward setting. (Obeidat, et al., 2018) proposed to predict future portfolio's returns using Long Short-Term Memory (LSTM) (Schmidhuber, et al., 1997) neural networks, as they are a more suitable than RNNs for processing time-series (sequential data). Nonetheless, all the previous models lack interaction with the market. In other

words, they cannot adapt and consequently, they underperform in non-stationary scenarios. To address this limitation, reinforcement-learning-based systems are suitable candidates. (Almahdi & Yang, 2017) introduced a recurrent-reinforcement-learning method, with a coherent risk-adjusted performance objective function, named the Calmar ratio, to obtain both buy and sell signals that updated the asset allocation weights. (Jiang & Liang, 2017) presented a Convolutional Neural Network (CNN) (LeCun, et al., 1998) to dynamically optimize cryptocurrency portfolios. A follow-up work by (Jiang, et al., 2017) (i Alonso & Srivastava, 2020) assessed the impact of different types of layers, including CNN, LSTM and RNN. Similarly, (Liang, et al., 2018) applied deep-reinforcement-learning algorithms with continuous action space to asset allocation. In particular, they investigated the model-free Deep Deterministic Policy Gradient (DDPG) (Silver, et al., 2014) as well as the Proximal Policy Optimization (PPO) (Schulman, et al., 2017). (Buehler, et al., 2019) presented a framework to hedge a portfolio of derivatives, where the system did not depend on specific market dynamics, such as transaction costs, market impact, liquidity constraints or risk limits. (Kolm & Ritter, 2020) investigated the link between portfolio allocation and reinforcement learning, namely, showing how the latter could be used to solve intertemporal financial problems. (Yang, et al., 2020) introduced an ensemble strategy that employed various deep-reinforcement schemes to learn a unified strategy that maximized the investment return. (Ye, et al., 2020) proposed a framework, coined state augmented reinforcement learning, that aimed to leverage additional diverse information from alternative sources other than classical structured financial data like asset prices. Finally, driven by the Covid-19 financial crisis, (Benhamou, et al., 2021) focused on models that detected extreme negative patterns, and consequently dis-investing the assets.

3. METHODOLOGY

In this chapter, we introduce the methods that we use in the experimental evaluation. First, we present the traditional approaches, where we discuss different strategies based on the Markowitz model. Then, we describe and derive the fundamentals of deep reinforcement learning, presenting some state-of-the-art optimization methodologies.

3.1. Markowitz Mean-Variance Portfolio Theory

Modern Portfolio Theory (Markowitz, 1968) introduces a financial method, for risk-averse investors, to construct diversified portfolios that optimize their returns. Namely, the mean-variance optimization approach. This technique aims at assembling a portfolio given some pre-defined constraints. Risk and return trade-off is at the heart of such a method, and its main components are the standard deviation and expected return of the assets. While variance (derived from the standard deviation) expresses the degree of spread in the data set, by showing how spread out the returns of a specific asset are, the expected return represents the probability of the estimated return of the investments.

MPT makes several key assumptions that the practitioners should be aware of before using the mean-variance optimization. The main ones are the following:

- The risk of the portfolio is based on its volatility of returns, i.e., price fluctuations.
- The analysis is conducted on a single-period model of investment.
- Investors are rational, averse to risk and eager to increase consumption. As a result, the utility function is concave and increasing.
- Investors seek either to maximize their portfolio return for a given level of risk or to minimize their risk for a given return.

From a mathematical perspective, given a portfolio p with n assets, we can calculate the standard deviation as:

$$\sigma_p = \sqrt{\sigma_p^2},$$

the variance as:

$$\sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_i Conv(r_i, r_j) = \mathbf{w^T \Omega w},$$

and the expected return as:

$$\mathbb{E}(r_p) = \sum_{i=1}^{n} w_i \, \mathbb{E}(r_i).$$

The variable $\mathbf{w}$ denotes the weights of the individual assets, and the variable r_p the return of the portfolio.

As we mentioned before, the trade-off between the risk and the expected return will be critical when building a portfolio. Using the Markowitz model for analysing portfolios helps to discover the efficient frontier, which is the combination of assets that offers the highest expected return for a defined level of risk, or the lowest risk for a given level of expected return (see Figure 2). Portfolios that lie below the efficient frontier are suboptimal, as they do not provide enough return for the level of risk. Portfolios that cluster to the right of the efficient frontier are also suboptimal because they have a higher level of risk for the defined rate of return. In general, when working with the efficient frontier, we are interested in three important points along the efficient frontier: minimum variance portfolio, maximum return portfolio and tangency portfolio.

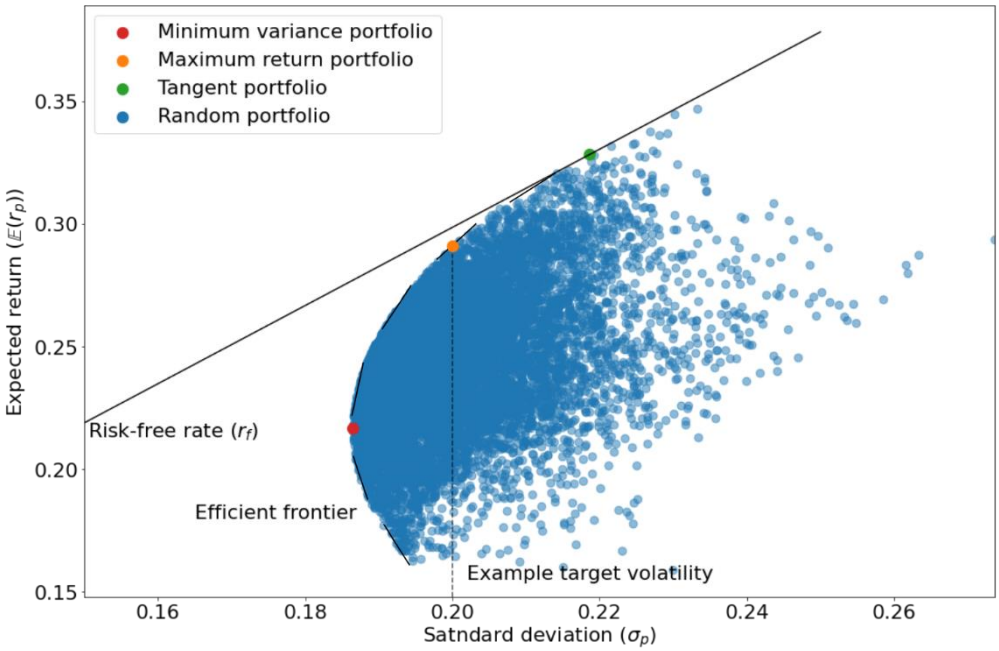

Figure 2: This graph shows the different types of portfolios based on Markowitz model. Each portfolio is plotted according to its expected return and standard deviation. If a portfolio is plotted on the right side of the chart, it indicates that the level of risk is too high for its return. If it is plotted low on the graph, the portfolio offers low returns for the risk that it carries. Source prepared by the author.

3.1.1. The Minimum Risk Mean-Variance Portfolio

The point where the hyperbola (efficient frontier) changes from convex to concave is where the minimum variance portfolio lies. This portfolio allocation has a unique solution that can be found by solving a simple quadratic optimization problem via standard Lagrange multiplier methods. The optimization problem can be formulated as:

$$\min_{\mathbf{w}} \ \frac{1}{2}\mathbf{w}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{w}$$

$$\text{subject to} \quad \mathbf{w}^{\mathrm{T}}\mathbf{r} = \mathbb{E}(r_p)$$

$$\text{and} \quad \mathbf{w}^{\mathrm{T}}1 = 1.$$

The vector $\mathbf{w}$ denotes the individual investments (weights of the assets) subject to the condition that the available capital is fully invested, i.e., $\mathbf{w}^{\mathrm{T}}1 = 1$. The lower bound on the target return $\mathbb{E}(r_p)$ is expressed by the condition $\mathbf{w}^{\mathrm{T}}\mathbf{r} = \mathbb{E}(r_p)$, where the vector $\mathbf{r}$ estimates the expected mean of the assets ($\mathbb{E}(r_i)$).

3.1.2. The Maximum Return Mean-Variance Portfolio

In contrast to minimum variance portfolios, where the optimizer minimizes the risk; in maximum return portfolio, the optimizer maximizes the return given a target risk σ_p^2. The optimization problem can be formulated as:

$$\max_{\mathbf{w}} \ \mathbf{w}^{\mathrm{T}}\mathbf{r}$$

$$\text{subject to} \quad \mathbf{w}^{\mathbf{T}}\boldsymbol{\Omega}\mathbf{w} = \sigma_p^2$$

$$\text{and} \quad \mathbf{w}^{\mathbf{T}}1 = 1.$$

If an investor decides to employ such a method, first he would need to define the risk that can tolerate.

3.1.3. The Tangency Portfolio

The tangency portfolio is the asset allocation that maximizes the Sharpe ratio (Sharpe, 1998). This ratio measures the excess return earned over the risk-free rate per unit of volatility or total risk, which helps investors to better understand the return of their investment. It can be formulated as:

$$\text{Sharpe ratio} \ = \ \frac{\mathbb{E}(r_p) \ - \ r_f}{\sigma_p},$$

where r_f stands for risk-free rate, i.e., the theoretical rate of return of an investment with zero risk like U.S. treasury rate.
The tangency portfolio optimization can be formulated as:

$$\max_{w} \ \frac{\mathbf{w}^{\mathrm{T}}\mathbf{r} \ - \ r_f}{\mathbf{w}^{\mathbf{T}}\boldsymbol{\Omega}\mathbf{w}}$$

$$\text{subject to} \quad \mathbf{w}^{\mathbf{T}}1 = 1.$$

Graphically, it is the point where a straight line through the r_f is tangent to the efficient frontier, in the Markowitz model space.

3.1.4. The Risk Parity Portfolio

Risk parity (Maillard, et al., 2010) (Roncalli & Weisang, 2016) is an alternative approach to the Markowitz model that focuses on the allocation of the risk instead of the capital. This method asserts that when asset allocations are adjusted to the same risk level, the portfolio can achieve a higher Sharpe ratio and thus, it can be more resistant to market downturns. To achieve that, the risk parity portfolio tries to constrain each asset to contribute equally to the portfolio overall volatility. The optimization problem can be formulated as:

$$\min_{\mathbf{w}} \ \frac{1}{2}\mathbf{w}^{\mathbf{T}}\boldsymbol{\Omega}\mathbf{w} \ - \ \frac{1}{n}\ln(\mathbf{w})$$

$$\text{subject to} \quad \mathbf{w}^{\mathrm{T}}1 = 1.$$

3.2. Deep Reinforcement Learning

Machine learning is a branch of artificial intelligence that allows machines to learn from data, identify patterns and make decisions without being explicitly programmed for it. Reinforcement learning is an area of machine learning that focuses on training an algorithm following the cut-and-try approach. More specifically, the algorithm needs to learn to take actions that maximize the final reward in a particular situation. To that end, this algorithm (agent) evaluates a current situation (state), takes an action, and receives feedback (reward) from the environment. Positive feedback is given when the action is correct, and negative feedback otherwise (see Figure 3).

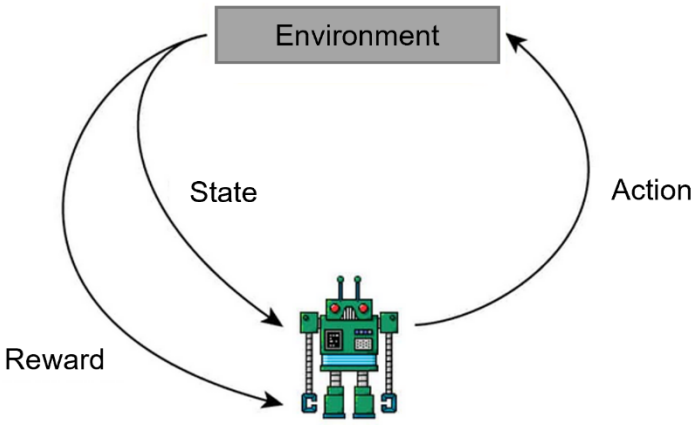

Figure 3: This figure illustrates the typical reinforcement cycle. Source prepared by the author.

Similar to other mathematical methods, machine-learning algorithms have different flavours, each of them with its own advantages and disadvantages. Commonly, these algorithms can be divided into supervised, unsupervised and reinforcement learning. Supervised learning works with labelled data and mainly deals with regression and classification tasks. The unsupervised learning, however, employs unlabelled data and tries to determine patterns and associations within the data. This technique tackles clustering and associative rule mining problems. Finally, reinforcement learning uses a learning agent to interact with the environment based on an action-reward system through a trade-off between exploitation and exploration. The main goal of this type of learning is to find the best sequence of decisions that maximizes the long-term reward. Due to the absence of training data, reinforcement-learning algorithms are bound to learn from their experience. In particular, they learn how to act best through many attempts and failures.

For the sake of completeness, let us now illustrate a reinforcement-learning-toy example (Figure 4): We have an agent (robot), a reward (gold), and many hurdles (traps) in between. The objective in this setup is to train an agent that finds the optimal path to reach the reward, while avoiding the hurdles. To achieve that, the robot tries many possible paths until it learns the one with higher reward, i.e., the path with fewer hurdles. Note that each correct step gives a positive reward to the robot, and each wrong step subtracts from it the reward; the total reward is calculated when it reaches the gold.

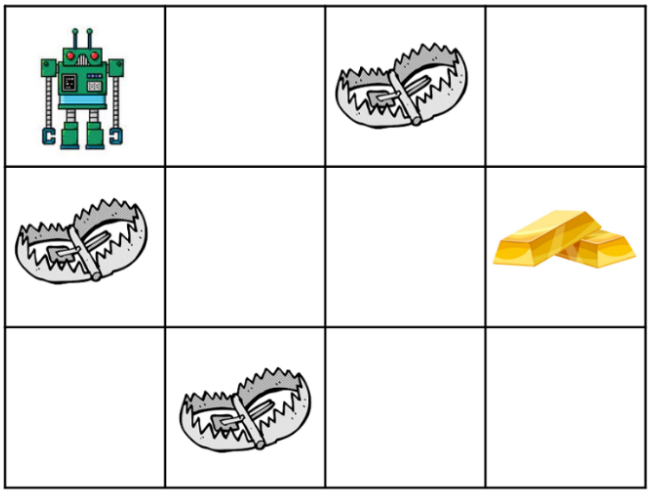

Figure 4: The environment is the world where the agent (robot) interacts. This illustration models the environment in a 3x4 matrix, where the agent navigates to find the reward (gold) while avoiding the traps. Source prepared by the author.

3.2.1. Fundamentals of Deep Learning

Most of the state-of-the-art approaches of reinforcement learning are based on deep-learning models (at least for training the agent). Deep learning is a subfield of machine learning that uses artificial neural networks to approximate a mathematical function that maps the input to the output. These neural networks can be viewed as an attempt to mimic the human brain through a combination of learnable parameters (weights $\mathbf{w}$ and bias $\mathbf{b}$) trained on large amounts of data. From a topology point of view, deep neural networks consist of several layers, made of multiple neurons, where each layer is built upon the previous one. Depending on the position of these layers, they are classified as input layer, where the data is fed for being processed; hidden layers, these layers help to optimize and refine for accuracy by adding more computational power to the system; and output layer, where the final prediction or classification is made. Figure 5 shows the schematics of a neural network, and the mathematical description of a neuron.

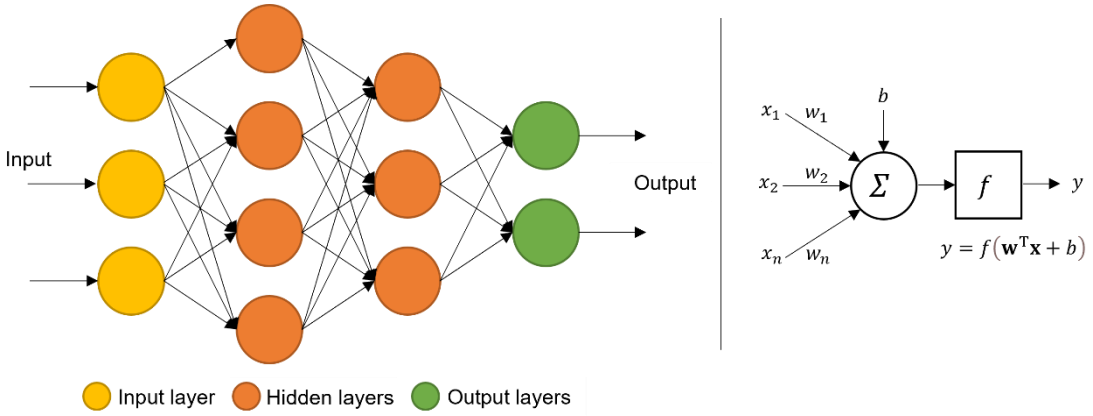

Figure 5: (Left) Schematics of an artificial neural network with its different layer types. (Right) Mathematical function of one artificial neuron (f). The concatenation of thousands of neurons creates the neural network function ($\mathbf{F}$). Source prepared by the author.

In order to have a system that can solve our target task, first we need to train our deep neural network $\mathbf{F}$. The process of training involves an optimization algorithm that searches through a space of possible values for the model parameters that results in good performance on the training dataset. Stochastic Gradient Descent (SGD) is a simple yet very efficient algorithm to find the parameters of the model that match the predicted values (output from the model) with the actual values (labels in a supervised setup). The SGD algorithm has two main parts:

1. Forward propagation is how neural networks make predictions. The input data $\mathbf{x}$ is fed into the first layer (input layer), then the data is propagated through the intermediate layers (hidden layers) and finally, the last layer (output layer) outputs a prediction $\mathbf{y}$ (see top of Figure 6).
2. Back-propagation refers to the optimization step, where the algorithm modifies the model's parameters to match the ground-truth solutions (labels) $\mathbf{t}$. To achieve that, partial derivatives of the loss function with respect to the parameters of the network are calculated and used (see bottom of Figure 6).

By repeating these two steps many times, the optimizer usually converges into a local minimum, i.e., optimization solution. Ideally, this point in the parameter space guarantees that the model accuracy also holds when testing in unseen future scenarios.

Finally, it is important to notice that the impact of deep learning goes beyond the academic domain. In fact, nowadays, deep-learning approaches drive many artificial intelligence applications and services that improve automation in various application areas, including smart healthcare, business intelligence, smart cities, cybersecurity intelligences, as well as physical tasks without human intervention. As a result, this technology is at the forefront of the Industry 4.0 revolution.

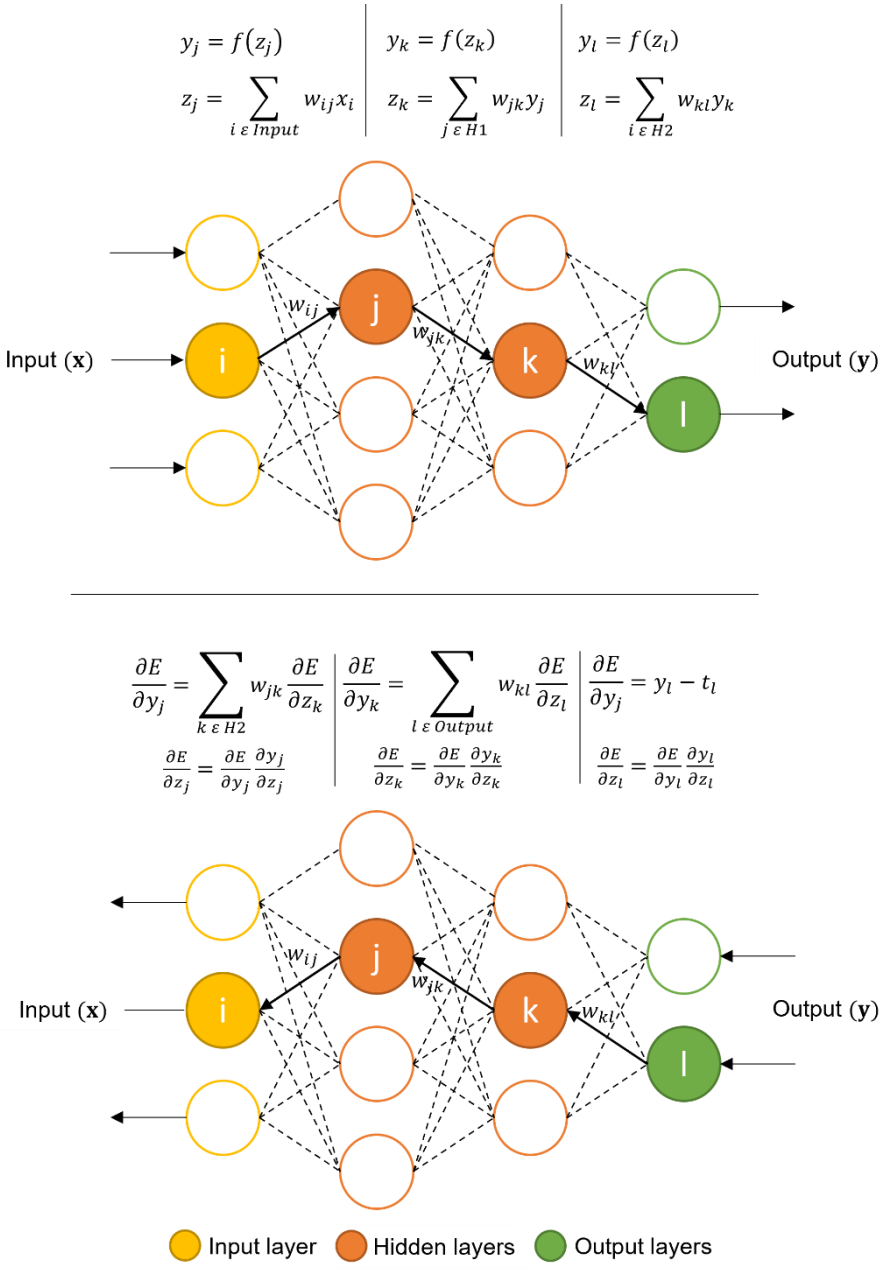

Figure 6: Back-propagation is a widely used algorithm for training deep neural networks. (Top) Schematics of the forward propagation from a single path. (Bottom) Schematics of the back-propagation from the forward propagation. Notice that these two steps are applied for all the paths' combination. Therefore, adding a single neuron may have a perceptible computational effect. Source prepared by the author.

3.2.2. Fundamentals of Reinforcement Learning

We start with the definition of a Markov process: A sequence of states is Markov if and only if the probability of moving to the next state s_{t+1} depends only on the present state s_t and not on the previous states $s_1, s_2, s_3 \dots s_{t-1}$. In other words, the future is independent of the past given the present. Mathematically, we can express this statement as:

$$\mathbb{P}[s_{t+1}|s_t] = \mathbb{P}[s_{t+1}| s_1, s_2, s_3 \dots s_{t-1}].$$

The probabilities of transitioning from s_t to s_{t+1} is given by the state transition probability matrix, defined as:

$$p_{ss'} = \mathbb{P}[s_{t+1} = s'|s_t = s].$$

If we write these two definitions together, we obtain that the dynamics of Markov process starts in some state s_0, and moves to some successor state s_1, drawn from $p_{s_0 s_1}$. Then, the process moves to s_2 drawn from $p_{s_1 s_2}$, and keep on in the same fashion.

Although, the dynamics of the Markov are relevant, they are not enough for reinforcement-learning- based approaches since they do not aim at maximizing the reward. To deal with that, we add the reward r, the action a and the discount factor $\gamma \in [0,1]$ into the Markov process, originating the Markov decision process. Now, in the Markov decision process, the transition to the next state s_{t+1} not only depends on the current state s_t, but also on the action a_t make at the current state. Furthermore, each state-action pair is attached with a reward function r defined as:

$$r_s^a = \mathbb{E}[r_{t+1}|s_t = s, a_t = a].$$

If now we join these terms together, we have that the dynamics of Markov decision process starts in some state s_0, chooses an action a_0 to take and moves randomly to some successor state s_1, drawn from $p_{s_0 s_1}^{a_0}$ and attached to an $r_{s_1}^{a_0}$. Then it moves to s_2, chooses another action a_1 and moves to state s_2, drawn from $p_{s_1 s_2}^{a_1}$ and attached to another $r_{s_2}^{a_1}$, and keep on in the same fashion.

As we mentioned, in reinforcement learning the goal is to optimize the cumulative reward, i.e., all the rewards that the agent receives from the environment. The total sum of rewards is called returns, and it can be written as:

$$g_t = r_{t+1} + \gamma r_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}.$$

Herein g_t, we can see that the discount factor γ determines how much importance is to be given to the immediate reward, and how much to the future rewards. Values close to 0 lead to "myopic" evaluation, i.e., more importance is given to the immediate reward; values close to 1 lead to "far-sighted" evaluation, i.e., more importance is given to future rewards.

To choose those actions that maximize the expected value of the return over time, the algorithm needs to follow a policy π that guides the choice of action at a given state. We can formally define the policy as:

$$\pi(a|s) = \mathbb{P}[a_t = a|s_t = s].$$

To find a policy, we can implement two value functions: state-value function and action-value function.

The state-value function v_π measures how good it is to be in a particular state s according to the return g when following policy π. It can be written as:

$$v_\pi(s) = \mathbb{E}_\pi[g_t|s_t = s].$$

The action-value function q_π is the expected return g starting from state s, taking action a, and then following policy π. It can be written as:

$$q_\pi(s,a) = \mathbb{E}_\pi[g_t|s_t = s, a_t = a].$$

Now that we have defined both v_π and q_π, we can formalize their relationship. We can assert that the state-value function is equivalent to the sum of the action-value functions of all outgoing (from s) actions a, multiplied by the policy probability of selecting each action:

$$v_\pi(s) = \sum_{a \in A} \pi(a|s)\, q_\pi(s,a).$$

At this point, we only need to deal with these value functions to solve our optimization task. To do that, we use Bellman equation. Bellman equation decomposes the value function into two parts: the immediate reward and the discounted future values. In this manner, it simplifies the computation of the value function, such that rather than summing over multiple time steps, we can find the solution of a complex problem by breaking it down into simpler, recursive sub-problems and finding their solutions. In practice, we can solve the Bellman equation using dynamic programming.

Bellman equation for state-value function v_π:

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \left(r_s^a + \gamma \sum_{s' \in S} p_{ss'}^a\, v_\pi(s') \right).$$

Bellman equation for action-value function q_π:

$$q_\pi(s,a) = r_s^a + \gamma \sum_{s' \in S} p_{ss'}^a \sum_{a' \in A} \pi(a'|s')\, q_\pi(s',a').$$

In a Markov decision process environment, there are different value functions obtained from different policies. Nonetheless, we are interested in the optimal value function as it yields the maximum value compared to all other value functions. Therefore, when we say we are solving a Markov decision process, it actually means that we are finding the optimal value function. Mathematically, we can define optimal state-value function $v_*(s)$ and action-value function $q_*(s,a)$ as follows:

$$v_*(s) = \max_\pi v_\pi(s) \quad \text{and} \quad q_*(s,a) = \max_\pi q_\pi(s,a).$$

The same principle holds for policies, i.e., not all policies are equally good. As a consequence, if the agent's goal is to maximize the total cumulative reward in the long run, it will need to find the optimal policy. The policy π can be considered better than or same as the policy π' if the expected return g of policy π is greater than or equal to the expected return of policy π' for all states s. We can define the optimal policy as:

$$\pi \geq \pi' \text{ if only if } v_\pi(s) \geq v_{\pi'}(s), \qquad \forall s.$$

The optimal policy offers optimal value functions:

$$\pi_* = \arg\max_\pi v_\pi(s) \quad \text{and} \quad \pi_* = \arg\max_\pi q_\pi(s,a).$$

In short, when optimizing there will be two main approaches to representing and training agents: value-based, i.e., optimal value function, and policy-based, i.e., optimal policy function. Note that this classification assumes a model-free setup. In other words, the agent tries to maximize the expected reward only from real experience, without a model/prior experience. The agent does not know which state he will be in after taking an action; he only cares about the reward associate with the state/state-action.

3.2.3. Actor-Critic Optimization

Actor-Critic (AC) (Konda & Tsitsiklis, 1999) is a temporal difference method that has two separated memory structures to explicitly represent the policy and the value function. On the one hand, the policy structure is known as the actor because it decides which action should be taken. It essentially controls how the agent behaves by learning the optimal policy π (policy-based). On the other hand, the estimated value function is known as the critic. It evaluates the action made by the actor by computing the value function (value-based), which can be the action-value q_π or state-value v_π. These two structures participate in an optimization game, where they both get better in their own role as the time passes. The outcome is that the overall method (system) achieves superior results than systems based on solely one structure. See Figure 7 for a graphical description.

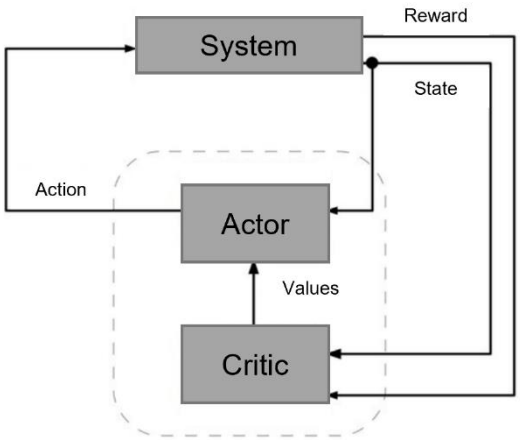

Figure 7: This figure displays the actor-critic optimization cycle. Source prepared by the author.

The Advantage Actor-Critic (A2C) (Mnih, et al., 2016) algorithm is a variation of AC, where the algorithm specifically uses estimates of the advantage function for its bootstrapping, i.e., to update a value based on some estimates and not on some exact values. The function of the advantage function is to determine how good an action compared to average action for a specific state is. By doing so, the variance between the old and the new policies is reduced, and consequently the stability of the reinforcement-learning algorithm improves. The advantage function can be defined as:

$$\mathrm{a}_\pi(s,a) \;=\; q_\pi(s,a) \;-\; v_\pi(s).$$

Deep Deterministic Policy Gradient (DDPG) (Lillicrap, et al., 2015) is another AC method. It combines ideas from Deterministic Policy Gradient (DPG) (Silver, et al., 2014) and Deep Q-Network (DQN) (Mnih, et al., 2013). Namely, DDPG uses a critic that learns from a temporal loss and an actor that learns using policy gradient. However, DDPG is an off-policy method. This means that it can sample batches from large experience buffers, making this approach more sample-efficient, at least at training.

Although DDPG can provide excellent results, it is frequently brittle with respect to hyperparameters and other kinds of tiresome fine-tuning dependencies. Furthermore, a common failure of DDPG is that this algorithm continuously overestimates the q_π values of the critic network, and it can eventually lead to the agent falling into a local optimum or to a catastrophic forgetting. Twin Delayed DDPG (TD3) (Fujimoto, et al., 2018) is an algorithm that addresses this issue by introducing three novel critical tricks: (1) employing two critic networks, (2) delaying the updates of the actor and (3) adding noise to regularize the target action.

Soft Actor-Critic (SAC) (Haarnoja, et al., 2018) is an algorithm that optimizes a stochastic policy in an off-policy fashion, forming a bridge between stochastic policy optimization and DDPG-based approaches. The biggest feature of SAC is its modified objective function, where instead of only seeking to maximize the lifetime rewards, the algorithm also tries to maximize the entropy of the

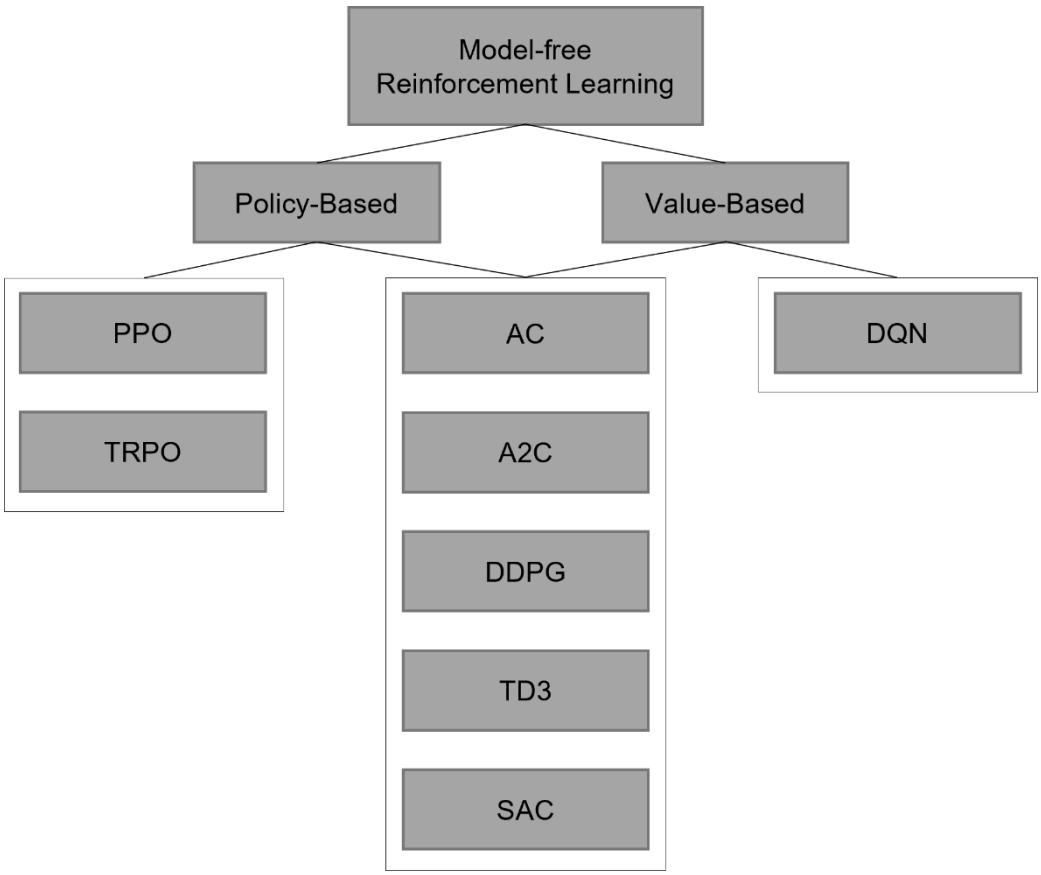

Figure 8: Taxonomy of the model-free reinforcement learning algorithms related to this work. Source prepared by the author.

policy. We can think of entropy as how unpredictable a random variable is. For instance, if a random variable always takes a single value, then it has zero entropy. As a rule of thumb, we want a high entropy in our policy to explicitly encourage exploration, by assigning equal probabilities to actions that have same or nearly equal q_π values, and to ensure that it does not collapse.

Another alternative to modify policy gradient methods using experiences from old versions of the policy is through the importance sampling technique. This technique weights samples based on the difference between the action probability distributions, given by the current and old policies. Trust region policy optimization (TRPO) (Schulman, et al., 2015) is one policy gradient method that guarantees that the new update's policy is not far away from the old policy, or at least, that the new policy is within the trust region of the old policy. However, in practice, TRPO is a relatively complicated algorithm to implement and not always a suitable candidate. Proximal Policy Optimization (PPO) (Schulman, et al., 2017) is a follow-up work that simplifies the algorithm. More specifically, PPO is a first-order optimization that defines the probability ratio between the new and old policies. Instead of adding complex constraints, e.g., Kullback-Leibler, PPO imposes a policy ratio to stay within a small interval around 1. It can be viewed as a combination of A2C (having multiple workers) and TRPO (using trust region to improve the actor).

3.2.4. Deep Reinforcement Learning for Asset Allocation

Portfolio management can be modelled as a deep-reinforcement-learning (DRL) problem, where deep neural networks are used to optimize the portfolio risk-adjusted returns for a given set of assets. To guarantee convergence, these neural networks need to understand the market's behaviour when reallocating the assets. The DRL framework for portfolio management can be built as follows:

- Agent: A deep neural network is proposed as the agent, whose goal is to find an optimal function (either policy- or value-based) that learns actions that maximize the reward.
- Actions: They are the outputs of the network for a given period, and they represent the final portfolio weights, i.e., the percentage of each asset within the portfolio.
- State: It is the input used to feed the network and describes the current situation of the stock market via financial indicators. These indicators are made of a tensor of features, which includes the current balance of our portfolio, the stock prices, and the owned assets.
- Environment: The stock market acts as the environment. It receives the actions taken by the agent and sends back the reward to the agent.
- Reward: The difference between the previous and the current portfolio values is our reward (differential returns).

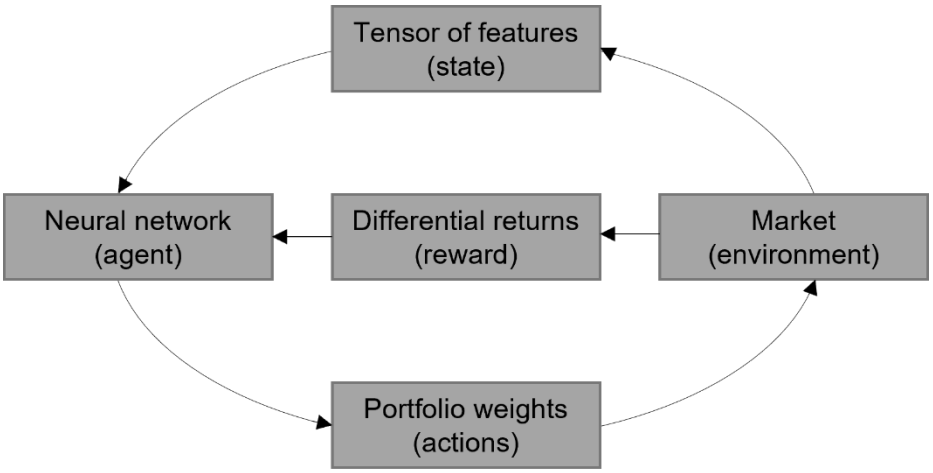

Figure 9: Representation of our deep-reinforcement-learning framework. Source prepared by the author.

In a DRL approach, usually the agent uses the SGD, based on a market snapshot (state) and the reward (cost-adjusted returns), to learn the action (weights) that leads to the optimal portfolio allocation for a given point in time. In Figure 9, we can see the connection among the different elements of a DRL framework. Finally, we need to make a few assumptions for a correct interpretability of our results:

- It is possible to trade at the market any time.
- Our transactions do not to affect the market price of the assets.

4. EXPERIMENTS

In this work, we address the task of asset allocation. Given a portfolio with a set of n assets, we aim to benchmark several allocation optimization methods, including traditional and DRL ones. To that end, we conduct a comprehensive evaluation, where we investigate the robustness of such approaches for different market's conditions as well as different time frequencies of reallocation. In particular, we study bull and bear market scenarios, and day- and week-trading cases.

4.1. Experimental Settings

When the exchanges close, the last trading price of the stock is recorded as the closing price of the share. Nonetheless, it is not always reliable since it might not provide an accurate picture of the true value. Therefore, the closing price is adjusted considering factors such as dividends, stock splits, and new stock issues, originating the adjusted close price. In all our experiments, we use adjusted close price as input.

For each market scenario, we select a set of 8 stocks, from well-known companies, that follow the market tendency that we target to study. We use data from a period of seven years, where 80% of the dataset goes into the training set, and the remaining 20% of the dataset goes into the testing set. Notice that for the traditional methods, where no learning is involved, we do not have such a split. Furthermore, for these methods, we set a windows size of 50 and no transaction costs are involved. On the contrary, for the DRL-based counterpart, we use a windows size of 1 and their costs are set to 0.1% for each trade total value.

We evaluate the performance of 9 different algorithms: tangency portfolio, minimum variance portfolio, risk parity, equal weight, A2C, PPO, DDPG, SAC and TD3. While the first four approaches are deterministic, the last five approaches, based on the Actor-Critic algorithm, are not due to the (stochastic) weight initialization process. Thus, we report results from 10 independent runs to obtain uncertainty boundaries.

4.2. Bull Market Scenario

Bull market occurs when investment prices rise for a sustained period of time. Propelled by the thriving economies and low unemployment rate, investors are eager to buy or hold onto securities. The result is a buyer's market. Bull markets tend to last for months or even years. Famous bull markets were the 1970s economic recovery as well as the pre-global financial crisis bull market.

Our first case of study focuses on 8 stocks that follow a bullish trend. These securities belong to the following companies: Apple (APP), General Electric (GE), JPMorgan Chase (JPM), Microsoft (MSFT), Vodafone Group (VOD), Nike (NKE), Nvidia (NVDA) and 3M (MMM). We use adjusted close prices from the 1st of January 2010 to the 1st of January 2017 (see Figure 10).

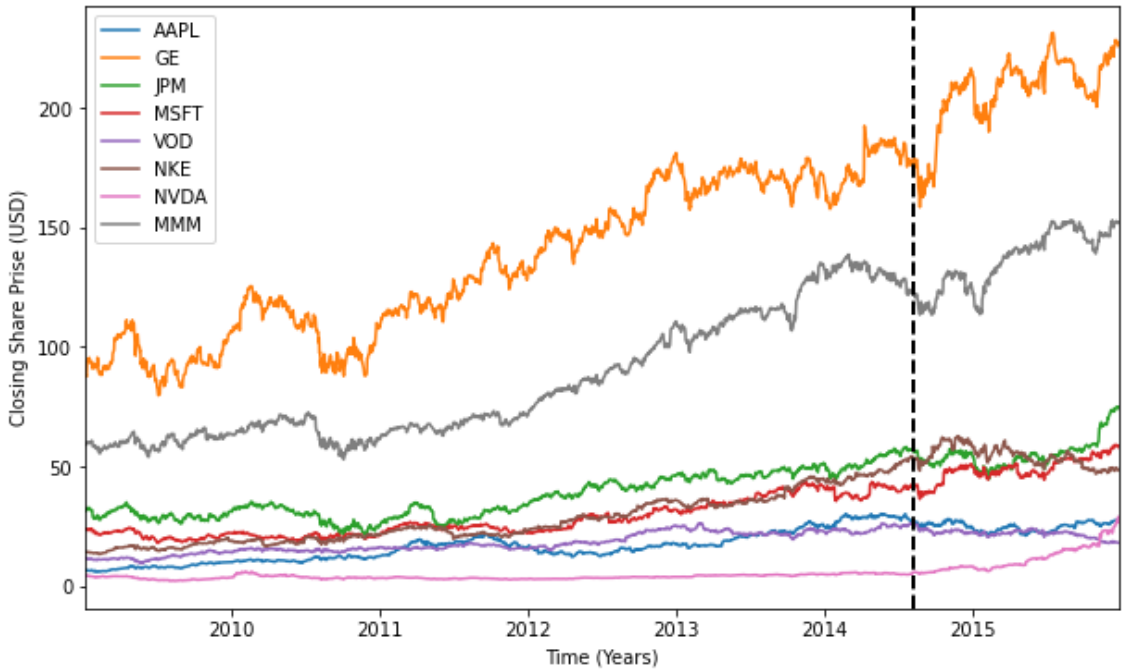

Figure 10: Evolution of the closing share price of our set of assets during bull market; from the 1st of January 2010 to the 1st of January 2017. The vertical dashed line separates the training data (left) from the testing data (right).

Figure 11 and Figure 12 plot the cumulative returns of the different asset allocation methodologies proposed in the current work. While Figure 11 shows the run that achieves the best performances, Figure 12 shows the run that obtains the worst. Furthermore, Table 1 and Table 2 display other metrics that help to dissect the methods, to better understand their outcomes. From all these results, we can derive two main observations: (1) DRL-based models outperform all the traditional approaches, except for PPO, and (2) DRL-based models seem to be unstable, i.e., weight initialization-dependent, and therefore, not as reliable as traditional approaches. As a result, a hybrid combination between traditional and DRL-based could be a suitable solution since it could find an optimal trade-off: high returns plus stability in the long run.

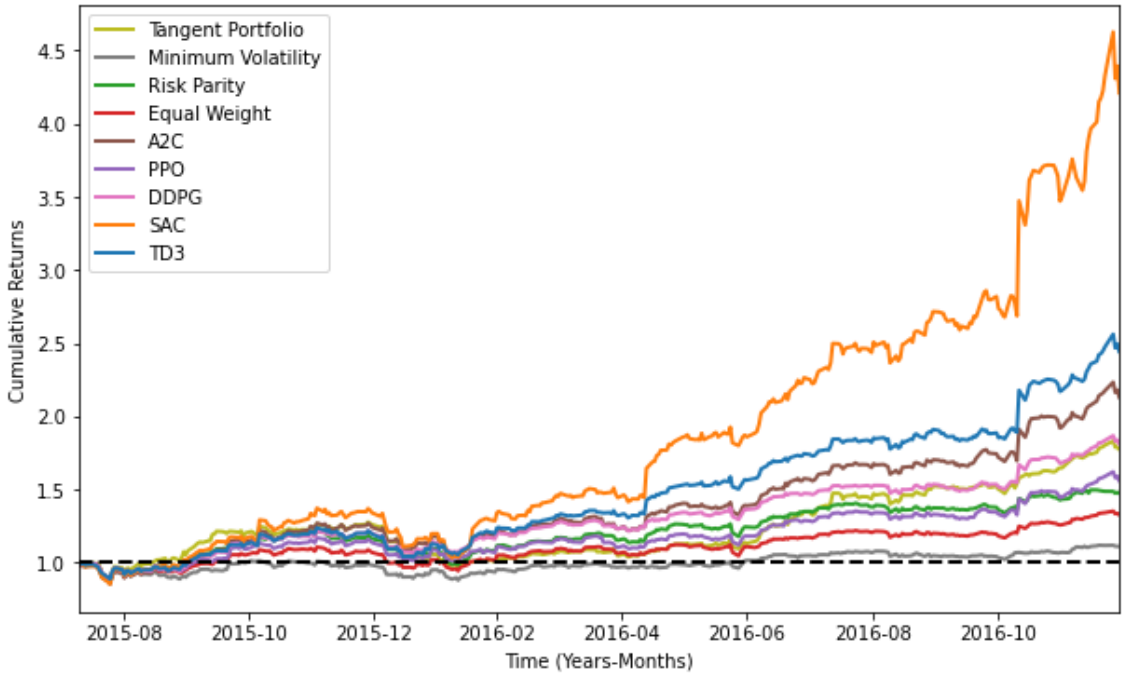

Figure 11: Evolution of the cumulative returns for the different asset allocation strategies during bull market. The results from the DRL-based models are the best from 10 runs.

	Annual return (%)	Cumulative returns (%)	Annual volatility (%)	Sharpe ratio	Calmar ratio	Stability	Max drawdown (%)
Tangent Portfolio	50.6	77.4	22.2	1.95	2.10	0.63	-24.1
Minimum Volatility	7.7	11.0	**16.4**	0.53	0.59	0.65	-13.2
Risk Parity	31.8	47.2	20.8	1.43	1.67	0.84	-19.1
Equal Weight	22.3	32.5	18.1	1.20	1.51	0.82	-14.7
A2C	71.6	113.0	26.5	2.17	3.93	0.92	-18.2
PPO	37.4	56.0	21.7	1.57	2.64	0.86	-14.2
DDPG	52.8	81.1	20.3	2.19	4.14	0.93	**-12.7**
SAC	**179.0**	**321.0**	43.1	**2.58**	**7.23**	0.93	-24.8
TD3	89.2	144.2	26.0	**2.58**	5.59	**0.95**	-16.0

Table 1: Financial metrics for the different asset allocation strategies during bull market. The results from the DRL-based models are the best from 10 runs.

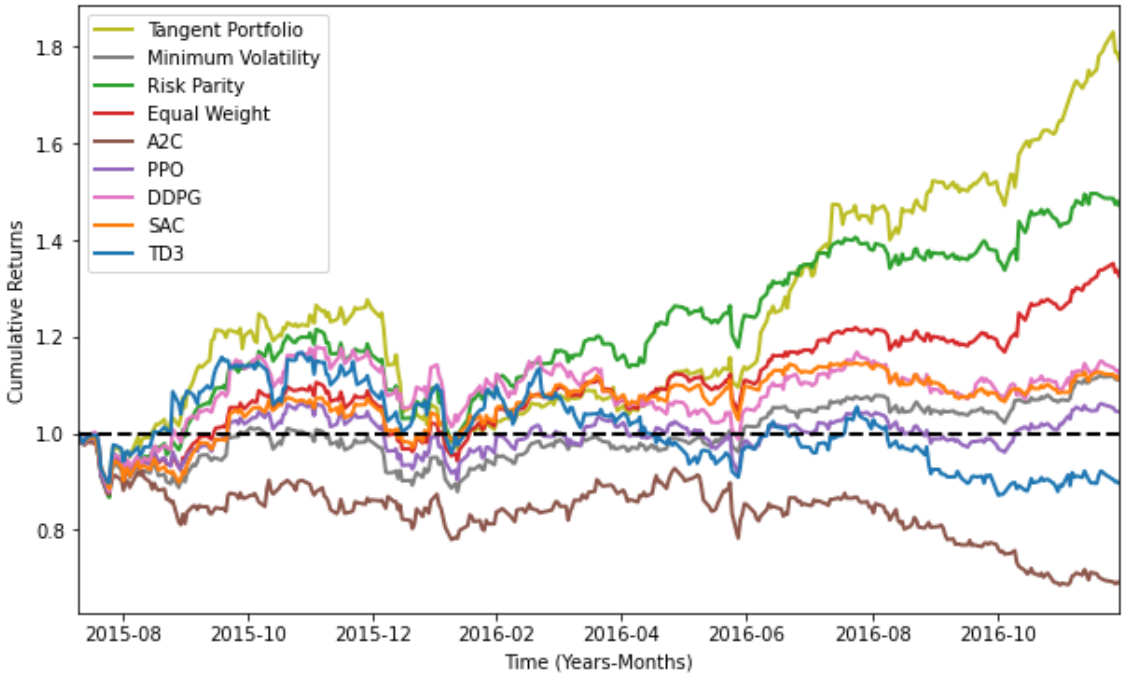

Figure 12: Evolution of the cumulative returns for the different asset allocation strategies during bull market. The results from the DRL-based models are the worst from 10 runs.

	Annual return (%)	Cumulative returns (%)	Annual volatility (%)	Sharpe ratio	Calmar ratio	Stability	Max drawdown (%)
Tangent Portfolio	**50.6**	**77.4**	22.2	**1.95**	**2.10**	0.63	-24.1
Minimum Volatility	7.7	11.0	**16.4**	0.53	0.59	0.65	-13.2
Risk Parity	31.8	47.2	20.8	1.43	1.67	**0.84**	-19.1
Equal Weight	22.3	32.5	18.1	1.20	1.51	0.82	-14.7
A2C	-23.3	-31.1	23.7	-1.00	-0.74	0.48	-31.4
PPO	3.2	4.4	19.2	0.26	0.21	0.14	-15.1
DDPG	8.5	12.1	21.1	0.49	0.52	0.15	-16.2
SAC	7.9	11.2	17.3	0.53	0.69	0.61	**-11.5**
TD3	-7.5	-10.4	24.5	-0.20	-0.30	0.47	-25.3

Table 2: Financial metrics for the different asset allocation strategies during bull market. The results from the DRL-based models are the worst from 10 runs

4.3. Bear Market Scenario

Bear market occurs when stock prices fall 20% or more for a sustained period of time. Triggered by periods of economic slowdown and higher unemployment rate, investors are reluctant to buy, often fleeing for the safety of cash or fixed-income securities. The result is a seller's market. Bear markets can last from a few weeks to several years. The first and most famous bear market was the great depression. The dot com bubble in 2000 and the housing crisis of 2007–2008 are other examples.

Our second case of study focuses on 8 stocks that follow a bearish trend. These securities belong to the following companies: PepsiCo (PEP), AAR Corp (AIR), British Petroleum (BP), BASF (BAS), Bayer AG (BAYN), Lufthansa (LHA), The Walt Disney Company (DIS) and The Coca-Cola Company (KO). We use adjusted close prices from the 1st of January 2003 to the 1st of January 2010 (see Figure 13).

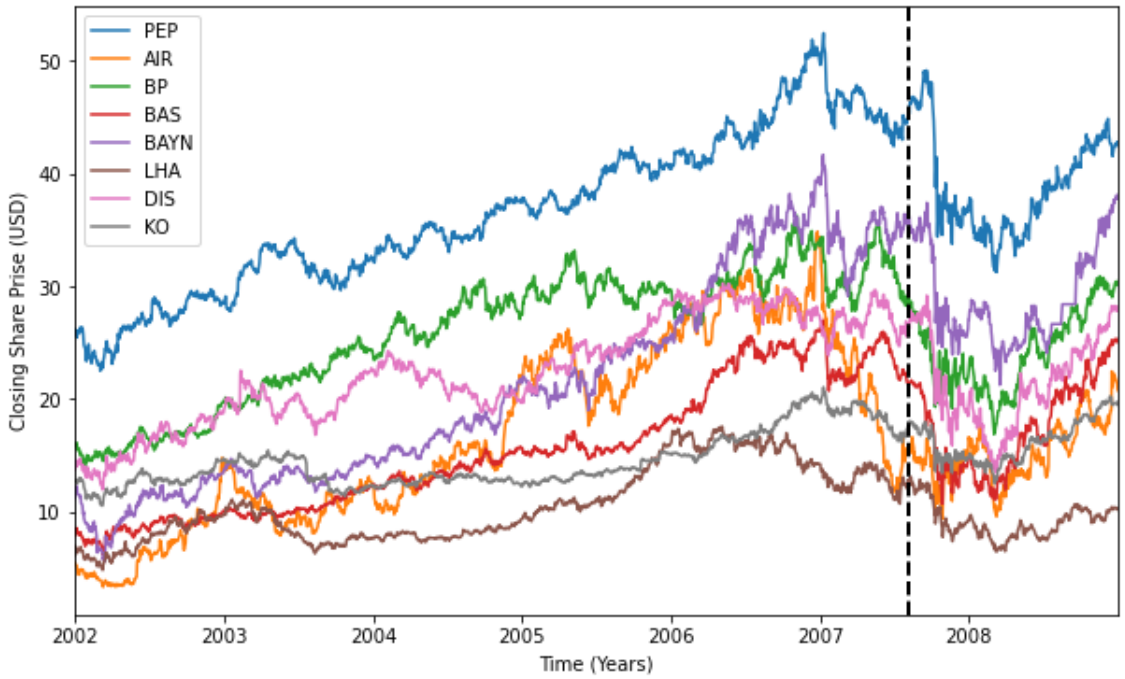

Figure 13: Evolution of the closing share price of our set of assets during bear market; from the 1st of January 2003 to the 1st of January 2010. The vertical dashed line separates the training data (left) from the testing data (right).

Figure 14 and Figure 15 plot the cumulative returns of the different asset allocation methodologies proposed in the current work. While Figure 14 shows the run that achieves the best performances, Figure 15 shows the run that obtains the worst. Furthermore, Table 3 and Table 4 display other metrics that help to dissect the methods, to better understand their outcomes. In Figure 14, we can observe that although all methods, except for minimum volatility, end up with positive returns, only tangent portfolio is able to deliver positive returns during (almost) the whole testing period. Moreover, in this bear setup, the gap between traditional and DRL-based approaches is much reduced, raising concerns of the utility of DRL in declining markets, with the exception of the PPO algorithm.

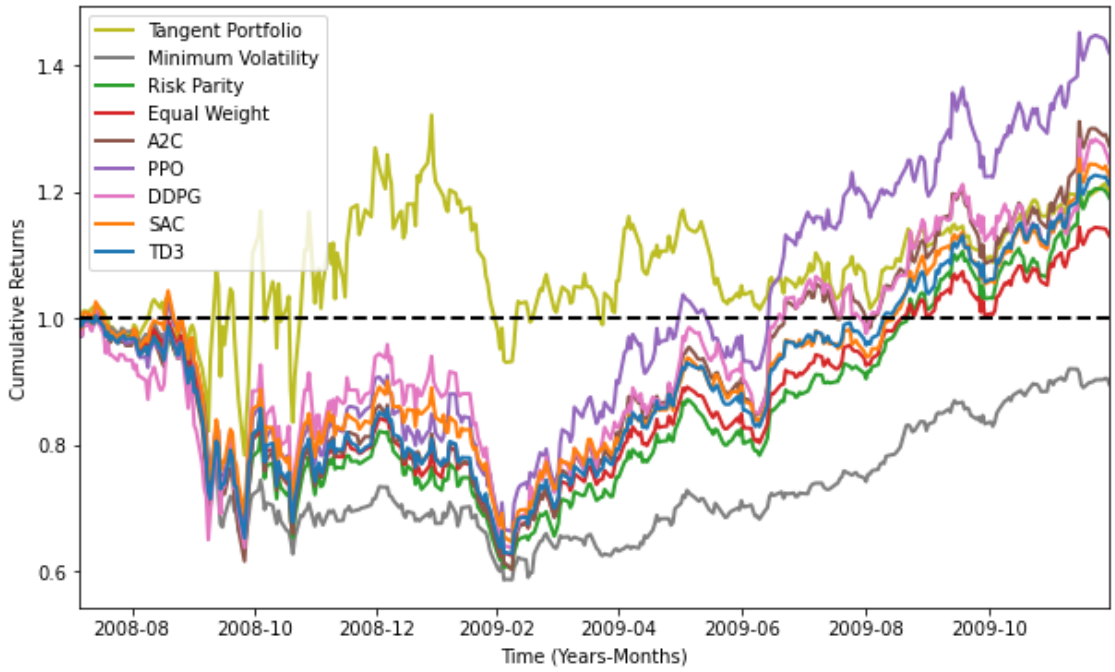

Figure 14: Evolution of the cumulative returns for the different asset allocation strategies during bear market. The results from the DRL-based models are the best from 10 runs.

	Annual return (%)	Cumulative returns (%)	Annual volatility (%)	Sharpe ratio	Calmar ratio	Stability	Max drawdown (%)
Tangent Portfolio	13.6	20.2	51.3	0.50	0.46	0.23	**-29.6**
Minimum Volatility	-7.4	-10.4	**27.2**	-0.14	-01.8	0.00	-41.6
Risk Parity	12.8	18.9	33.1	0.53	0.32	0.27	-39.6
Equal Weight	8.9	13.0	33.5	0.42	0.23	0.27	-37.8
A2C	18.2	27.3	42.1	0.61	0.45	0.43	-40.5
PPO	**27.6**	**42.0**	41.4	**0.79**	**0.75**	**0.59**	-36.8
DDPG	16.4	24.5	52.2	0.55	0.44	0.44	-37.4
SAC	15.1	22.4	37.1	0.56	0.40	0.32	-38.1
TD3	14.2	21.1	36.9	0.54	0.38	0.37	-37.4

Table 3: Financial metrics for the different asset allocation strategies during bear market. The results from the DRL-based models are the best from 10 runs.

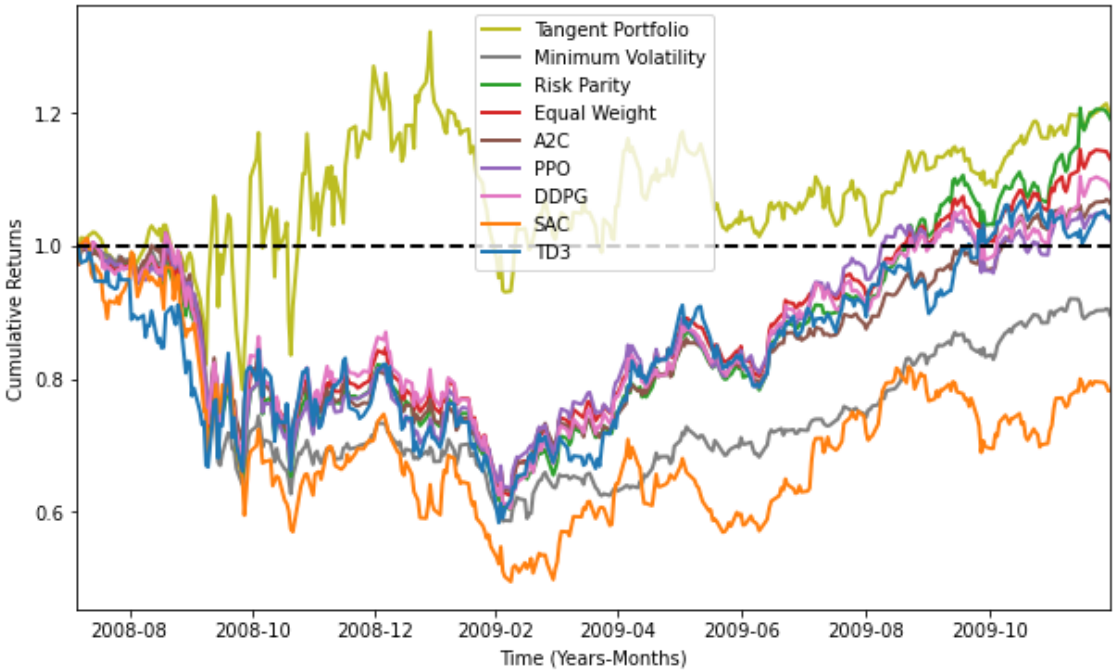

Figure 15: Evolution of the cumulative returns for the different asset allocation strategies during bear market. The results from the DRL-based models show the worst from 10 runs.

	Annual return (%)	Cumulative returns (%)	Annual volatility (%)	Sharpe ratio	Calmar ratio	Stability	Max drawdown (%)
Tangent Portfolio	**13.6**	**20.2**	51.3	0.50	**0.46**	0.23	**-29.6**
Minimum Volatility	-7.4	-10.4	**27.2**	-0.14	-01.8	0.00	-41.6
Risk Parity	12.8	18.9	33.1	**0.53**	0.32	**0.27**	-39.6
Equal Weight	8.9	13.0	33.5	0.42	0.23	**0.27**	-37.8
A2C	4.4	6.3	31.2	0.29	0.11	0.16	-38.2
PPO	2.6	3.8	31.8	0.24	0.07	0.24	-37.0
DDPG	5.9	8.6	38.2	0.34	0.14	0.20	-40.6
SAC	-15.7	-21.8	42.4	-0.19	-0.31	0.02	-51.0
TD3	2.8	4.1	48.2	0.30	0.07	0.26	-41.6

Table 4: Financial metrics for the different asset allocation strategies during bear market. The results from the DRL-based models show the worst from 10 runs.

4.4. In-Depth Allocation Comparison

In this subsection, we visualize the evolution of the weight allocation for each of the previous scenarios. This helps us to gain insight of the different running. Additionally, we plot the statistics (mean and standard deviation) of the DRL-based models for 10 independent runs.

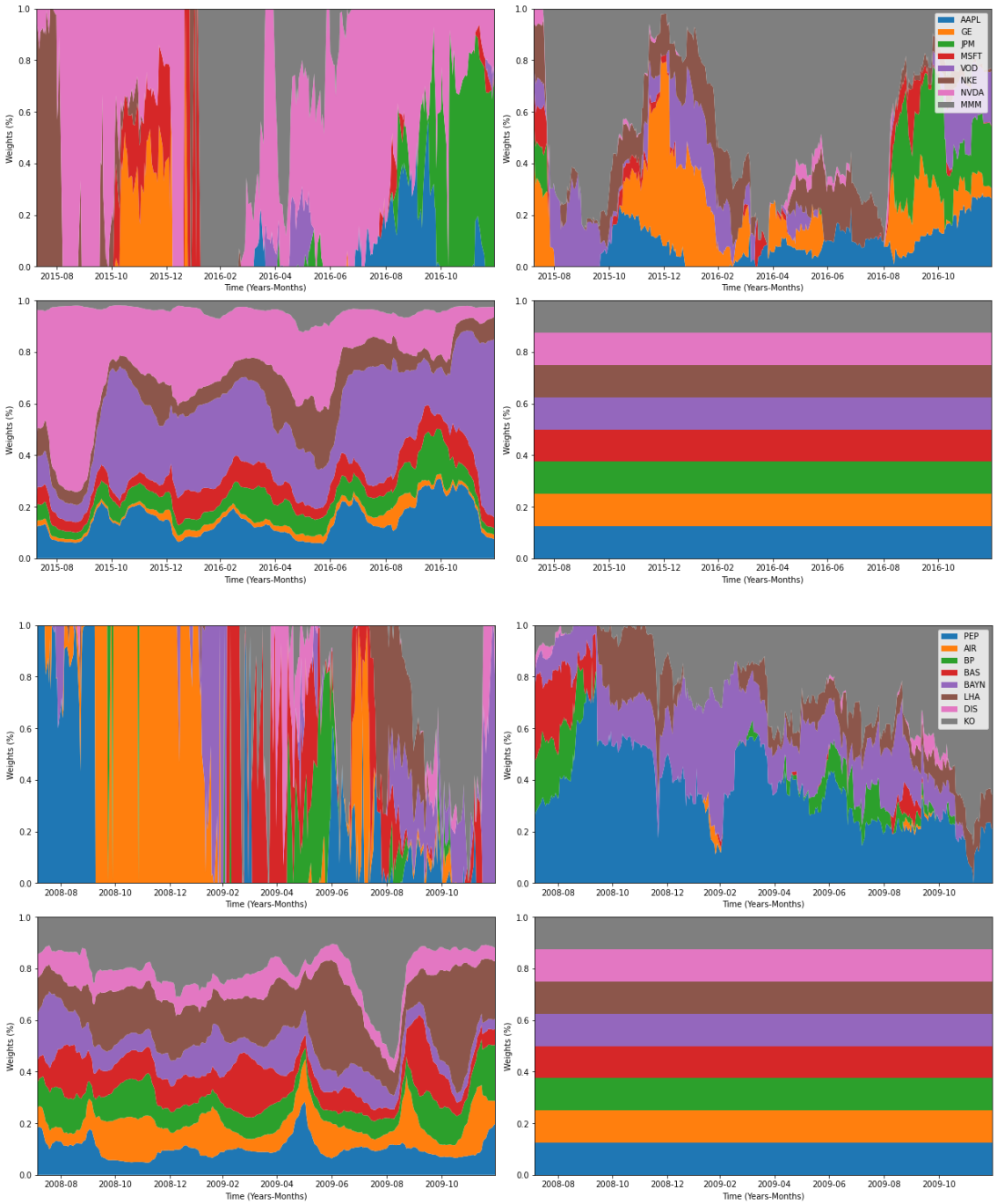

Figure 16: Weights for assets for the traditional methods. Top: bull market. Bottom: bear market. Clockwise from top left: tangent portfolio, minimum volatility, equal weight, and risk parity.

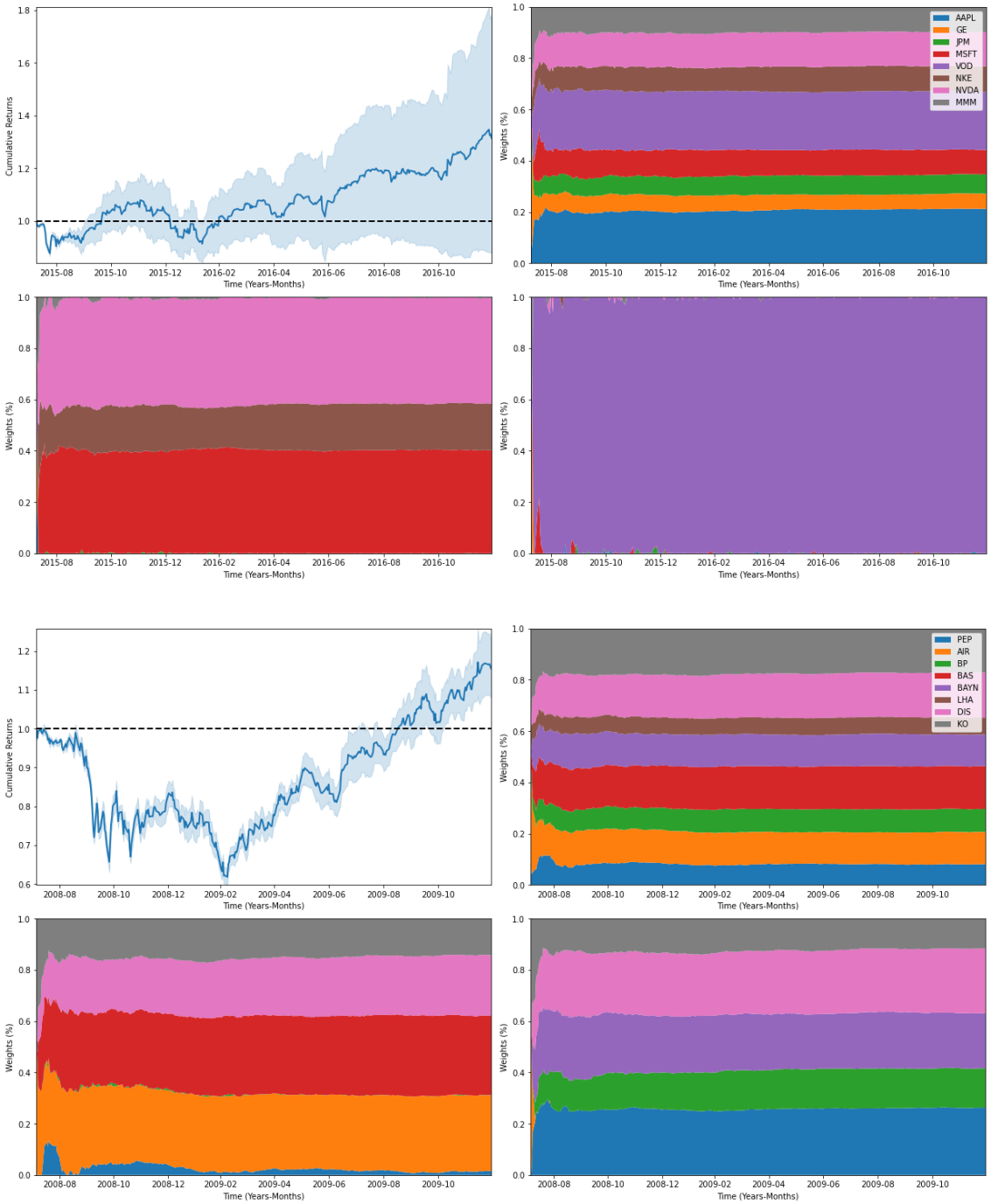

Figure 17: Evaluation of A2C method. Top: bull market. Bottom: bear market. Clockwise from top left: statistics of the cumulative returns, average weights allocation, worst weight allocation, and best weight allocation.

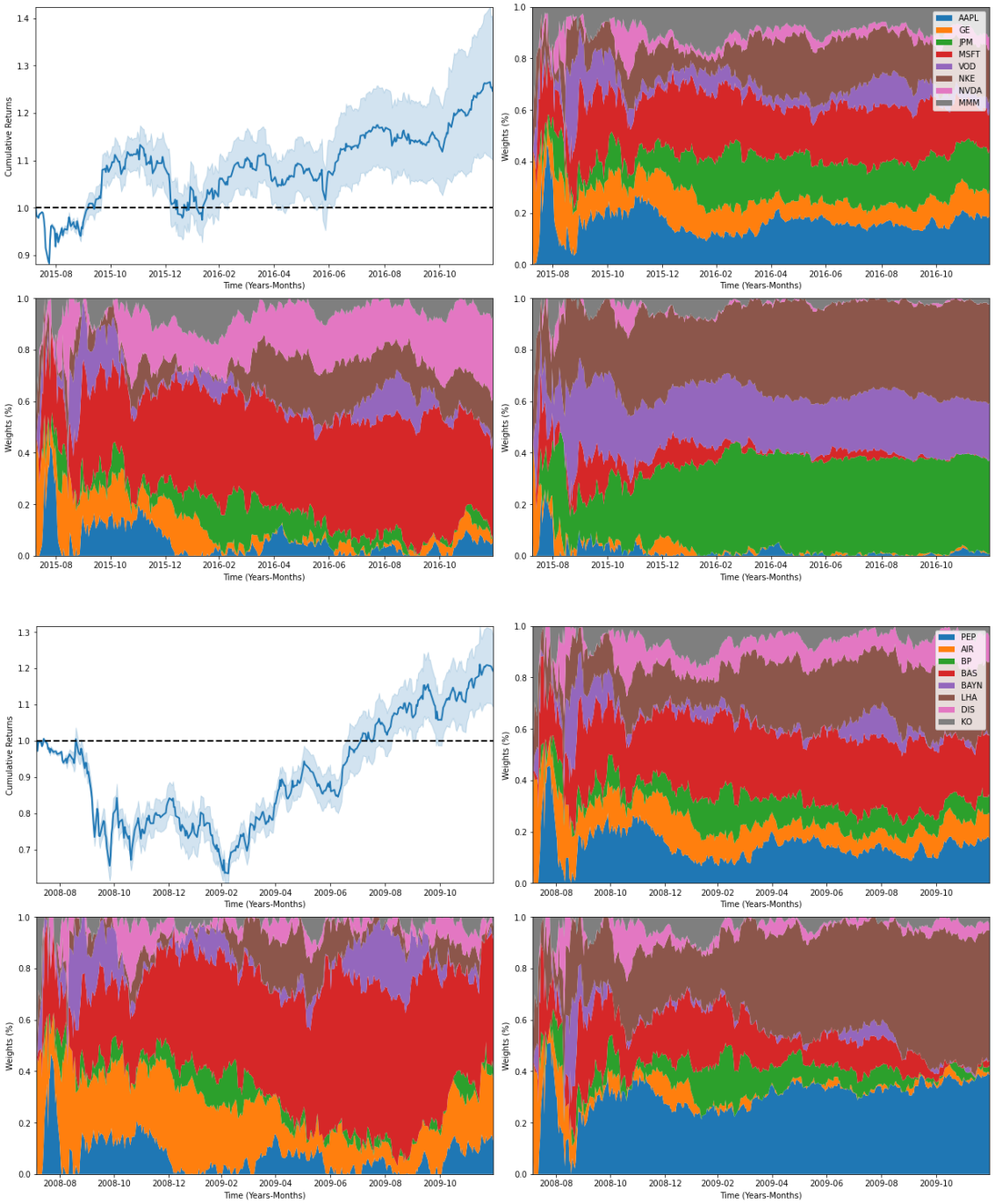

Figure 18: Evaluation of PPO method. Top: bull market. Bottom: bear market. Clockwise from top left: statistics of the cumulative returns, average weights allocation, worst weight allocation, and best weight allocation.

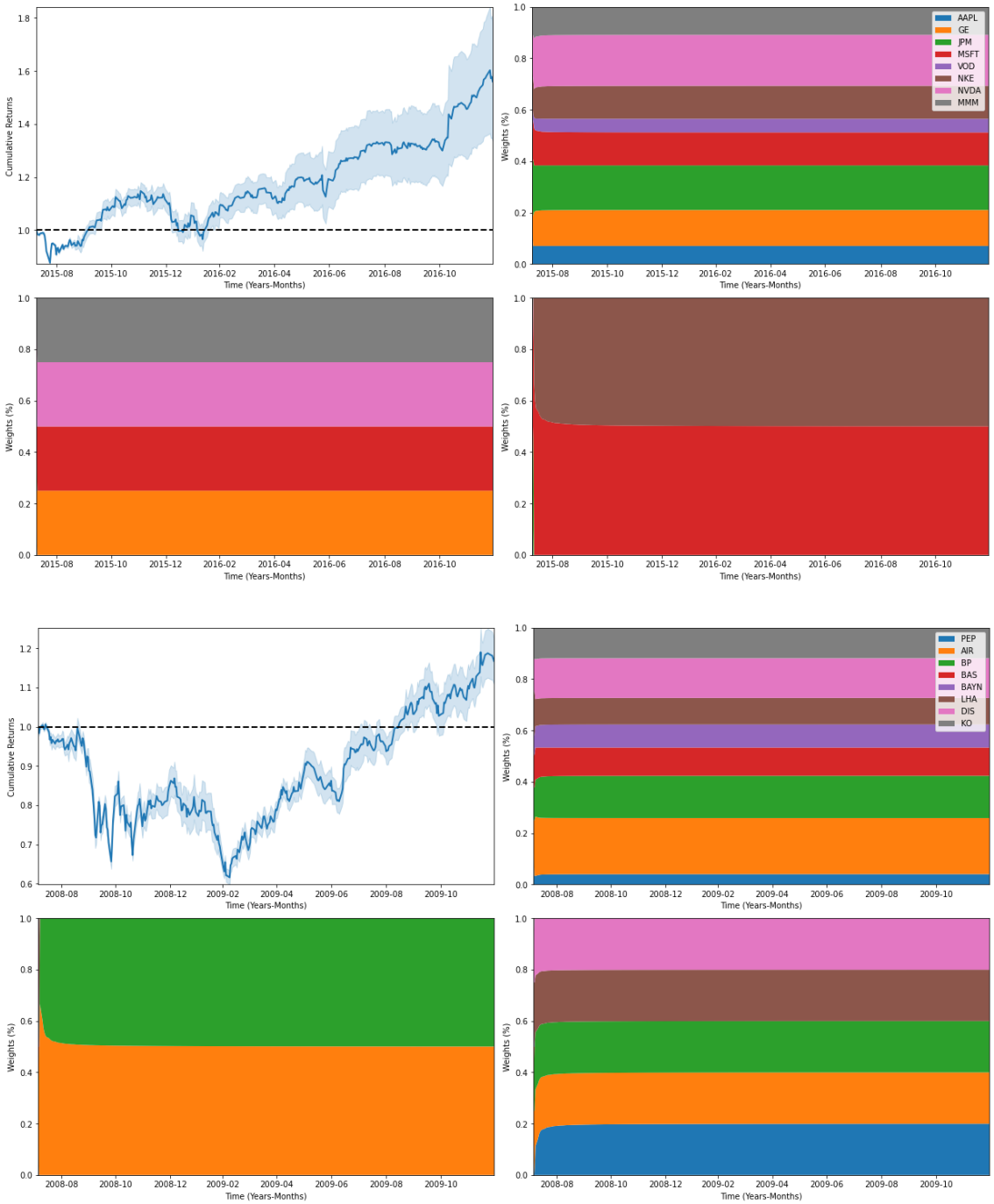

Figure 19: Evaluation of DDPG method. Top: bull market. Bottom: bear market. Clockwise from top left: statistics of the cumulative returns, average weights allocation, worst weight allocation, and best weight allocation.

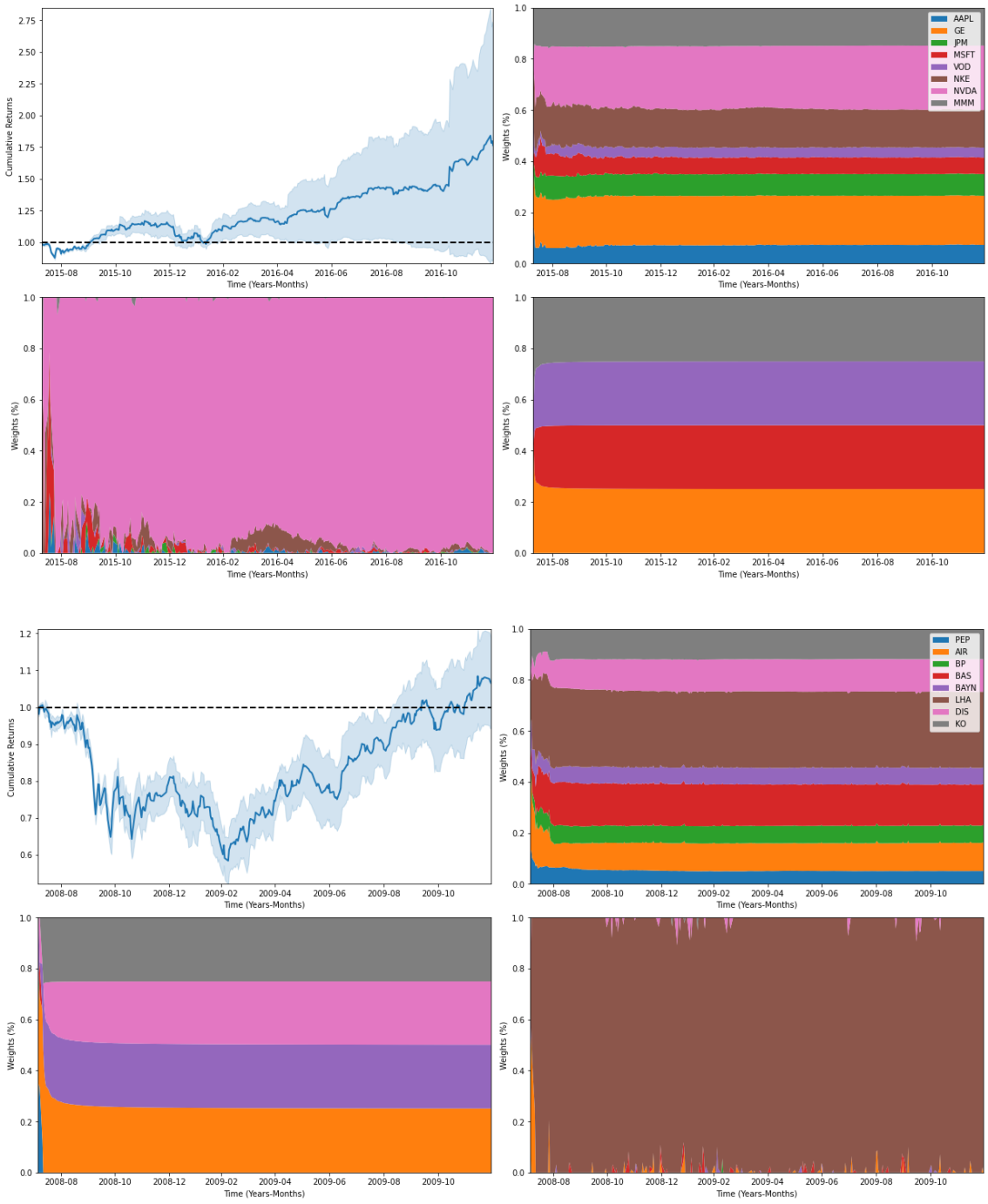

Figure 20: Evaluation of SAC method. Top: bull market. Bottom: bear market. Clockwise from top left: statistics of the cumulative returns, average weights allocation, worst weight allocation, and best weight allocation.

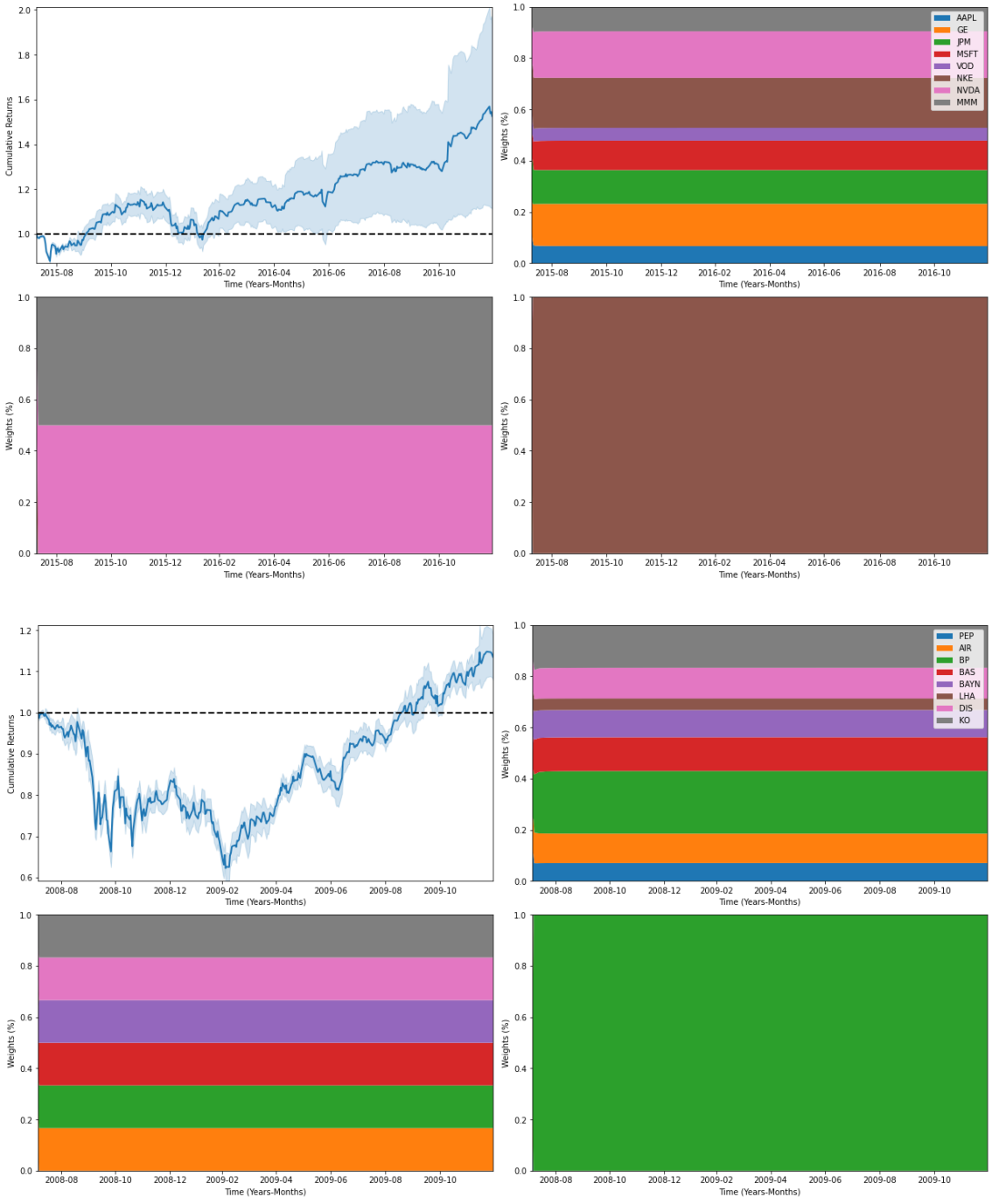

Figure 21: Evaluation of TD3 method. Top: bull market. Bottom: bear market. Clockwise from top left: statistics of the cumulative returns, average weights allocation, worst weight allocation, and best weight allocation.

By analysing the evolution of the different weight allocations, we can draw several conclusions. First, in general, all the DRL-based models avoid dramatic reallocation, changing the portfolio's configuration in a stepwise fashion. In other words, there are no big differences in the assets' allocation between two consecutive portfolios. This is, however, an expected behaviour since these models are trained with a transaction cost that penalizes reallocation. On the other hand, traditional approaches, except for the equal weight, display much dynamic reallocation as no transaction costs are included in their algorithms. Second, in the bull market, all DRL-based proposals have substantial variations among different runs, which only some of them result into optimal performance. Nonetheless, we notice that this is not the case for the bear scenario, since the results show a smaller variation, and thus, a more reliable and stable running. Finally, it is interesting to observe that often the worst runs are associated with "mono" asset portfolios (see Figure 17, Figure 20 and Figure 21), i.e., allocation with only one asset.

4.5. Allocation on a Weekly Basis

In this last subsection, we evaluate the impact of conducting asset reallocation on a weekly basis. Our goal is to verify whether the allocation methods are sensitive to scenarios with sparser reallocation (in time), i.e., less frequent updates. To that end, we move from the previous setups, where we follow a day-trading strategy to exploit short-term price movements, to a new setup with weekly reallocation to profit from stabler tendencies.

Similar to previous evaluations, we assess the performance of different runs under bull and bear markets. Figure 22 and Figure 23 plot the best and worst runs for each method in both market scenarios. We can observe, that unlike day-trading results, herein the tangent portfolio clearly dominates and outperforms all the other approaches in every single configuration. We hypothesize that the reason for it is that DRL-based solutions do not have enough data to be successfully trained, and therefore, they cannot match the previous results (based on daily inputs). Notice that deep-learning technologies are data greedy systems, and our weekly allocation setup uses seven times less data.

Besides the dramatic drop in performance, we observe among the solutions similar tendencies to the day-trading results. DRL-based models keep having significant variations among runs, making the traditional proposals more reliable candidates. However, in this case a hybrid solution would not be our recommendation as tangent portfolio is simply superior.

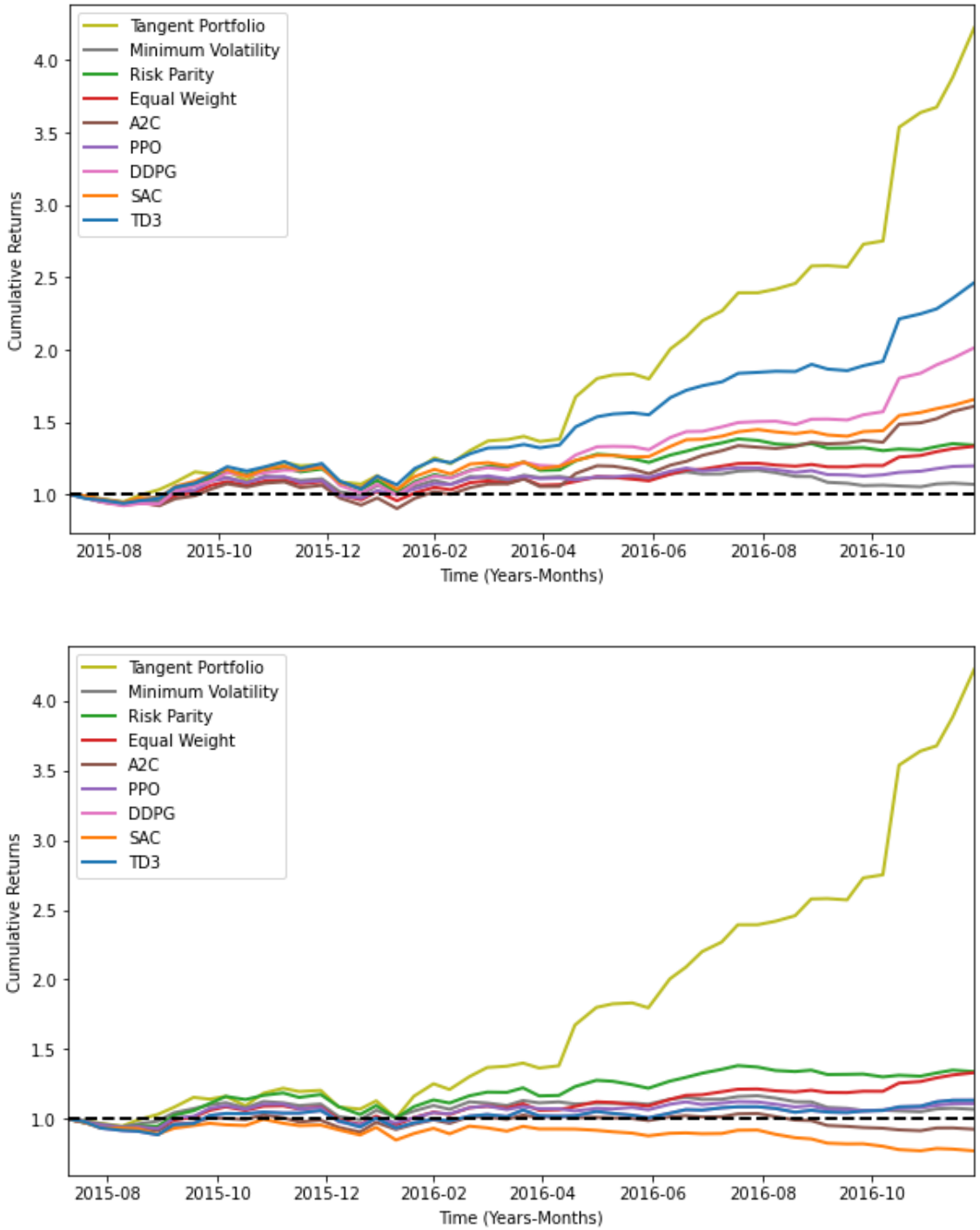

Figure 22: Evolution of the cumulative returns for the different asset allocation strategies during bull market. Results from the DRL-based models show the best (top) and the worst (bottom) from 10 runs.

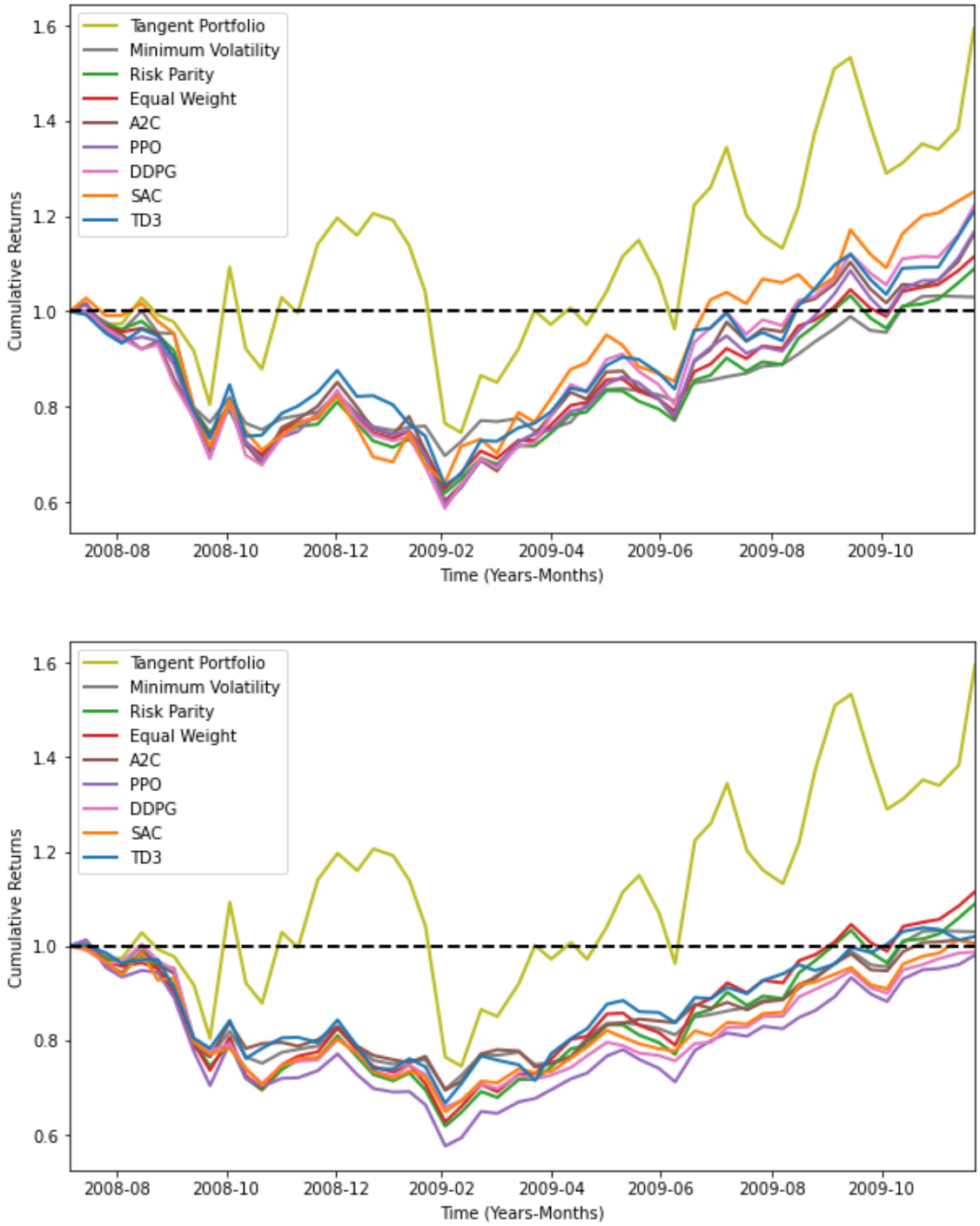

Figure 23: Evolution of the cumulative returns for the different asset allocation strategies during bear market. Results from the DRL-based models show the best (top) and the worst (bottom) from 10 runs.

5. DISCUSSION AND OUTLOOK

In this work, we explore the potential of using optimization algorithms for the asset allocation task. To that end, we conduct an extensive benchmark study on 9 different algorithms: tangency portfolio, minimum variance portfolio, risk parity, equal weight, A2C, PPO, DDPG, SAC and TD3. We evaluate their efficacy and reliability on different market conditions, bullish and bearish tendencies, as well as on different time frequencies of reallocation., on a daily and weekly basis.

Traditional approaches, based on Markowitz portfolio, do not require any fitting optimization process (training) since they do not employ learnable parameters. In our experiments, these models show stable results, achieving competitive performance on both market scenarios. Among them, tangency portfolio stands out as this method almost always provides the highest annual and cumulative returns as well as the best Sharpe and Calmar ratios. Risk parity obtains slightly inferior results, except for a few specific cases, where it outperforms the rest. As for minimum variance portfolio, it excels at keeping a low annual volatility (fulfilling this extra requirement), but in return the other financial metrics are negatively affected. Finally, equal weight offers surprisingly decent outcomes, taking into consideration that no optimization is involved. Although all these traditional proposals can successfully deal with stable market environment, they all are sensitive to outliers and abrupt market's changes. Therefore, in high volatile markets, traditional approaches are not well suited. A second drawback arises from their specificity. These algorithms were conceived for specific financial scenarios, and thus, they are rigid tools based solely on asset returns. In case we wanted to consider other relevant technical indicators such as moving average convergence/divergence, these methods would not be our best candidate.

On the other hand, DRL results are more difficult to interpret. While it is true that deep-learning approaches tend to have runs (random seed settings) that surpass their traditional counterparts (at least in the day-trading scenarios), they also provide runs with weaker performance. For example, PPO and SAC achieve overall the best results in both bullish and bearish markets, respectively, nonetheless, none of them can even beat the equal weight strategy when having poor runs. The reason for such fluctuations is the training process of these models. In other words, at training time when the algorithms optimize their agents to learn to make a sequence of decisions through the "trial and error" process, there are involved stochastic events. As a result, independent runs might eventually lead to different optimal or suboptimal solutions. To cope with this flaw, one could use more data that would indirectly help with convergence and stabilize the training process. In fact, we partially verify this hypothesis in the weekly reallocation, where we observe a clear drop in performance due to the limited amount of training data. Another complementary solution would be to feed the models with more technical indicators or larger input-data windows, leveraging in this manner the intrinsic flexibility of these neural network architectures.

We expect in the upcoming years to see a lot of exciting new research connecting even more to the fields of finance (asset allocation) and deep reinforcement learning. Namely, we believe that novel self-attention implementations, such as transformers, could further boost the model's performance. We would also expect that solutions, trained on synthetic data as a proxy, could help learning better features to lead to more reliable and stable results. Finally, future work that includes ethics and social aspects, like sustainability, needs to be developed. While it is true that some assets are already following certain ethical codes, to the best of our knowledge, there is no DRL-based approach that incorporates such behaviour into its internal running.

REFERENCES

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, Volume 87, p. 267–279.

Bai, Z., Liu, H. & Wong, W.-K., 2009. Enhancement of the applicability of Markowitz's portfolio optimization by utilizing random matrix theory. Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics, Volume 19, p. 639–667.

Benhamou, E., Saltiel, D., Ohana, J.-J. & Atif, J., 2021. Detecting and adapting to crisis pattern with context based Deep Reinforcement Learning. s.l., s.n., p. 10050–10057.

Black, F. & Litterman, R., 1992. Global portfolio optimization. Financial analysts journal, Volume 48, p. 28–43.

Buehler, H., Gonon, L., Teichmann, J. & Wood, B., 2019. Deep hedging. Quantitative Finance, Volume 19, p. 1271–1291.

Chakravorty, G., Awasthi, A. & Da Silva, B., 2018. Deep learning for global tactical asset allocation. Available at SSRN 3242432.

Chiu, C.-C.et al., 2018. State-of-the-art speech recognition with sequence-to-sequence models. s.l., s.n., p. 4774–4778.

Chopra, V. K. & Ziemba, W. T., 2013. The effect of errors in means, variances, and covariances on optimal portfolio choice. In: Handbook of the fundamentals of financial decision making: Part I. s.l.:World Scientific, p. 365–373.

Choueifaty, Y. & Coignard, Y., 2008. Toward maximum diversification. The Journal of Portfolio Management, Volume 35, p. 40–51.

Choueifaty, Y., Froidure, T. & Reynier, J., 2013. Properties of the most diversified portfolio. Journal of investment strategies, Volume 2, p. 49–70.

Christoffersen, P., Errunza, V., Jacobs, K. & Langlois, H., 2012. Is the potential for international diversification disappearing? A dynamic copula approach. The Review of financial studies, Volume 25, p. 3711–3751.

Deng, L., Hinton, G. & Kingsbury, B., 2013. New types of deep neural network learning for speech recognition and related applications: An overview. s.l., s.n., p. 8599–8603.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Freitas, F. D., De Souza, A. F. & De Almeida, A. R., 2009. Prediction-based portfolio optimization model using neural networks. Neurocomputing, Volume 72, p. 2155–2170.

Fujimoto, S., Hoof, H. & Meger, D., 2018. Addressing function approximation error in actor-critic methods. s.l., s.n., p. 1587–1596.

Goodfellow, I. et al., 2014. Generative adversarial nets. Advances in neural information processing systems, Volume 27.

Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S., 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. s.l., s.n., p. 1861–1870.

Haugen, R. A. & Baker, N. L., 1991. The efficient market inefficiency of capitalization–weighted stock portfolios. The journal of portfolio management, Volume 17, p. 35–40.

Heaton, J. B., Polson, N. G. & Witte, J. H., 2017. Deep learning for finance: deep portfolios. Applied Stochastic Models in Business and Industry, Volume 33, p. 3–12.

i Alonso, M. N. & Srivastava, S., 2020. Deep Reinforcement Learning for Asset Allocation in US Equities, s.l.: s.n.

Jiang, Z. & Liang, J., 2017. Cryptocurrency portfolio management with deep reinforcement learning. s.l., s.n., p. 905–913.

Jiang, Z., Xu, D. & Liang, J., 2017. A deep reinforcement learning framework for the financial portfolio management problem. arXiv preprint arXiv:1706.10059.

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

Konda, V. & Tsitsiklis, J., 1999. Actor-critic algorithms. Advances in neural information processing systems, Volume 12.

Krizhevsky, A., Sutskever, I. & Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, Volume 25.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, Volume 86, p. 2278–2324.

Liang, Z. et al., 2018. Adversarial deep reinforcement learning in portfolio management. arXiv preprint arXiv:1808.09940.

Lillicrap, T. P. et al., 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Lin, C.-M., Huang, J.-J., Gen, M. & Tzeng, G.-H., 2006. Recurrent neural network for dynamic portfolio selection. Applied Mathematics and Computation, Volume 175, p. 1139–1146.

Lintner, J., 1975. The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. In: Stochastic optimization models in finance. s.l.:Elsevier, p. 131–155.

Low, R. K. Y., Faff, R. & Aas, K., 2016. Enhancing mean–variance portfolio selection by modeling distributional asymmetries. Journal of Economics and Business, Volume 85, p. 49–72.

Maillard, S., Roncalli, T. & Teïletche, J., 2010. The properties of equally weighted risk contribution portfolios. The Journal of Portfolio Management, Volume 36, p. 60–70.

Markowitz, H. M., 1968. Portfolio selection. In: Portfolio selection. s.l.:Yale university press.

Merton, R. C., 1980. On estimating the expected return on the market: An exploratory investigation. Journal of financial economics, Volume 8, p. 323–361.

Mnih, V. et al., 2016. Asynchronous methods for deep reinforcement learning. s.l., s.n., p. 1928–1937.

Mnih, V. et al., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Nguyen, T. H., Shirai, K. & Velcin, J., 2015. Sentiment analysis on social media for stock movement prediction. Expert Systems with Applications, Volume 42, p. 9603–9611.

Niaki, S. T. A. & Hoseinzade, S., 2013. Forecasting S&P 500 index using artificial neural networks and design of experiments. Journal of Industrial Engineering International, Volume 9, p. 1–9.

Obeidat, S. et al., 2018. Adaptive portfolio asset allocation optimization with deep learning. International Journal on Advances in Intelligent Systems, Volume 11, p. 25–34.

Roncalli, T. & Weisang, G., 2016. Risk parity portfolios with risk factors. Quantitative Finance, Volume 16, p. 377–388.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J., 1985. Learning internal representations by error propagation, s.l.: s.n.

Schmidhuber, J., Hochreiter, S. & others, 1997. Long short-term memory. Neural Comput, Volume 9, p. 1735–1780.

Schulman, J. et al., 2015. Trust region policy optimization. s.l., s.n., p. 1889–1897.

Schulman, J. et al., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Sharpe, W. F., 1964. Capital asset prices: A theory of market equilibrium under conditions of risk. The journal of finance, Volume 19, p. 425–442.

Sharpe, W. F., 1998. The sharpe ratio. Streetwise–the Best of the Journal of Portfolio Management, p. 169–185.

Silver, D. et al., 2014. Deterministic policy gradient algorithms. s.l., s.n., p. 387–395.

Vaswani, A. et al., 2017. Attention is all you need. Advances in neural information processing systems, Volume 30.

Yang, H., Liu, X.-Y., Zhong, S. & Walid, A., 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. s.l., s.n., p. 1–8.

Ye, Y. et al., 2020. Reinforcement-learning based portfolio management with augmented asset movement prediction states. s.l., s.n., p. 1112–1119.