LUNDS
UNIVERSITET

Deep Reinforcement Learning Approach to Portfolio Optimization

Lorik Sadriu

Lund University School of Economics and Management
Bachelor's Thesis - Finance
Supervisor: Thomas Fischer

NEKH01
January 2022

Abstract

This paper evaluates whether a deep reinforcement learning (DRL) approach can be implemented, on the Swedish stock market, to optimize a portfolio. The objective is to create and train two DRL algorithms that can construct portfolios that will be benchmarked against the market portfolio, tracking OMXS30, and the two conventional methods, the naive portfolio, and minimum variance portfolio. We evaluate all the portfolios on a five-year period, from the start of 2016 to the end of 2020, in terms of returns and risk-adjusted returns. The two DRL algorithms implemented are Advantage Actor-Critic (A2C) and Deep Deterministic Policy Gradient (DDPG), they are also compared against each other.

The results of this study show that the A2C constructed portfolio significantly outperform the market and all of the other benchmark portfolios, in terms of returns and risk-adjusted returns. The A2C portfolio also outperforms the DDPG constructed portfolio. Even though the DDPG constructed portfolio performs less than the A2C constructed portfolio, it still significantly outperforms all of the other benchmarks on the whole testing period. Thus, concluding that a DRL approach can be implemented, on the Swedish stock market, to optimize a portfolio.

Moreover, the study shows that the two DRL agents can pick up on market trends and profit from them. However, applying the methods in a real-world environment does come with some data-processing caveats. Even though the models may come with caveats linked to them, the results of this study underline the usefulness of machine learning methods in portfolio management.

Contents

List of Tables

List of Figures

List of Abbreviations

A2C Advantage Actor-Critic

ANN Artificial neural network

DDPG Deep Deterministic Policy Gradient

DRL Deep reinforcement learning

EMH Efficient market hypothesis

HHI Herfindahl-Hirschman Index

LSTM Long Short-Term Memory

MPT Modern Portfolio Theory

OMXS30 The Nasdaq OMX Stockholm 30 index

RL Reinforcement learning

RRL Recurrent reinforcement learning

1 Introduction

> AI is the new electricity. It will transform every industry and create huge economic value.
>
> *Andrew Ng (2019)*

Portfolio management is a heavily researched and attractive field in finance, for investors on the market and academics alike. One of the main goals for the actors on the field is to find the optimal portfolio that balances risk and returns while outperforming the market portfolio. Thus, investors on the market aim to maximize the Sharpe ratio, a measure for risk-adjusted return, introduced by Nobel laureate Sharpe (1966). To solve the portfolio management problem and find the optimal portfolio, several methods to construct a portfolio by assigning the right weights to the right assets have been introduced and proposed over the years. The most popular method presented, and still used to this day, introduced by Markowitz (1952), is the modern portfolio theory (or mean-variance analysis). The procedure of the method focuses on optimizing the risk-return trade-off in a diversified portfolio and, in turn, creating a portfolio less volatile than the sum of its compounds. However, the method has received criticism for being founded on assumptions not warranted by the empirical evidence. Thus, investors prefer more heuristic weighting techniques, techniques that are easier to implement, e.g. the naively diversified portfolio, that assigns equal weights to all assets, or the minimum variance portfolio, that aims to minimize the variance of the portfolio.

Other investors and academics are believers of the efficient market hypothesis (EMH), developed by Fama (1970), a theory that states that asset prices reflect all information available. Thus, it is impossible for an investor to purchase or sell assets for other than their fair value, making it impossible to beat the market portfolio through different stock-picking methods. Therefore, believers of the EMH stick to market capitalization-weighted portfolios, e.g. index funds that track a stock index like the Swedish stock market OMXS30, based on a market capitalization-weighting technique.

Other than conventional methods, portfolio managers often make use of more advanced methods by combining engineering methods and advanced statistical methods. The financial industry has been highly computerized during the last couple of years. The digitization of the industry has brought a lot of advantages and profits to the industry, making it way more efficient than it once was. Therefore, it is only a question of when and not if, more advanced models from the fields of artificial intelligence and machine learning will start being the norm in the financial industry.

Machine learning is a scientific field focusing on algorithms to develop models based on sample data. The popularity of implementing machine learning methods in different industries has surged through the years. A current attractive subfield, called deep reinforcement learning (DRL), of machine learning, is a model that combines two other subfields of machine learning. DRL combines artificial neural networks (ANN), used as function approximators, with reinforcement learning (RL), a training method that rewards/penalizes desired/undesired behavior. Recent developments made by Google DeepMind have made the field even more attractive for various industries. These developments made it possible for the first system, Deepmind´s Alphago, to beat the world champion at the board game Go. Thus, questions of whether DRL could be efficient and profitable in industries like finance are starting to arise.

The implementation of RL to the financial field was first initiated by researchers like Neuneier (1996) and Moody et al. (1998), who use two different approaches to implement the models in a financial market, in which both succeeded. More recent research (Xiong et al. (2018), Noguer i Alonso & Srivastava (2020)) uses a DRL approach to solve the portfolio management problem. They show that a portfolio constructed with the help of DRL can outperform the market and other benchmark portfolios. The application of DRL in portfolio management has an advantage compared to other methods, considering that the model bypasses predicting future prices. Instead, the model skips through this phase and directly optimizes the desired portfolio performance, depending on the reward function provided. Thus, the investor can use the model for their own desired objective (Fischer, 2018).

It is interesting to study whether the approach can be applied to the Swedish stock market since previous research have been focused on the American market. Therefore, this study aims to create two DRL agents, constructed using daily data from the Swedish stock market, and evaluate whether the method could be a useful tool within portfolio management. We will be working with 28 stocks from the OMXS30 with data collected from 2000-2020. We will evaluate two portfolios constructed with the help of two DRL algorithms, Advantage Actor-Critic (A2C) and Deep Deterministic Policy Gradient (DDPG) depending on the same neural network called Long Short-Term Memory (LSTM). As stated above, the Sharpe ratio is a common objective for investors to maximize. Therefore, we use a reward function closely linked to the Sharpe ratio, introduced by Moody et al. (1998) called the differential Sharpe ratio. The models are tested on a five-year-long period and benchmarked against commonly used portfolio construction methods, i.e. the market portfolio, minimum variance portfolio, and naive portfolio. Moreover, this study aims to detect whether the two models can pick up on market trends and anomalies since making profits on trends and anomalies is an essential aspect of efficient strategies on the market and an objective for many investors.

The results of this paper show that a DRL approach can be successfully applied to solve the portfolio management problem. The results show that the two DRL models outperform all the proposed benchmarks, in terms of returns and risk -adjusted returns, making them efficient on the Swedish stock market. The efficiency of the two DRL models on the market also provides some evidence that the models can pick up on market trends and make profits on said trends, something that conflicts with even the weakest assumptions of the EMH.

The disposition of this thesis is organized as follows: section 2 presents the conventional weighting techniques used as benchmarks to the portfolios constructed by the use of DRL algorithms. The components of the DRL method and the method itself are presented later on in the section. Moreover, two crucial measures are introduced, along with the efficient market hypothesis, later implemented in the analysis. Section 3 provides an insight into previous research. Section 4 presents the empirical data, followed by section 5 which gives an overview of the methodology of the executed experiment. Section 6 presents the results in conjunction with an analysis. The thesis ends with concluding remarks in section 7.

2 Theoretical prerequisites

In this section, we present all the theoretical prerequisites behind this paper. First, we introduce the conventional methods used as benchmarks to our deep reinforcement learning models. Later on, we present all the background behind the machine learning deployed in this study. Lastly, we introduce the Sharpe ratio, used for evaluating the models, and the efficient market hypothesis, used for further discussion of the performance of the models.

2.1 Conventional Methods

Today, portfolio managers on the market use several conventional methods to construct a portfolio. In this paper, we present and use three popular and commonly used methods in portfolio management as benchmarks to our models: The minimum variance portfolio, which relies on Modern Portfolio Theory, the Naive portfolio, and the Market capitalization-weighted portfolio (OMXS30).

2.1.1 Modern portfolio theory

Modern portfolio theory, or mean-variance analysis, is the most popular model in portfolio optimization and has been since presented by Markowitz (1952). The model approaches the portfolio construction problem by first separating efficient portfolios from inefficient ones and then determining the risk-return opportunities available to the investor. Markowitz (1952) does this by computing the Minimum-Variance Frontier, a set of optimal portfolios that offer the highest expected return for a defined level of risk, or the other way around. The model uses the portfolio variance and mean to compute the Efficient Frontier. The portfolio mean and variance are defined as:

$$E(r_P) = \sum_{i=1}^{N} w_i \mu_i \tag{2.1}$$

$$\sigma_P^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \sigma_{i,j} w_i w_j \tag{2.2}$$

Where $E(r)$ is the expected mean return of the portfolio, w_i is the amount invested in security i, μ_i is the return of asset i. The portfolio variance, σ_P^2, consists of the weights of some asset i and some other asset j, as well as the covariance between the two assets denoted as $\sigma_{i,j}$ and expressed as:

$$\sigma_{i,j} = E([R_i - E(R_i)][R_j - E(R_j)]) \tag{2.3}$$

Or

$$\sigma_{i,j} = \rho_{i,j}\sigma_i\sigma_j \tag{2.4}$$

Where $\rho_{i,j}$ stands for the correlation coefficient and σ_i and σ_j stands for the standard deviation of each asset. With the help of these definitions, the mean-variance efficient portfolio for any targeted expected return can be constructed and plotted in the mean-variance frontier:

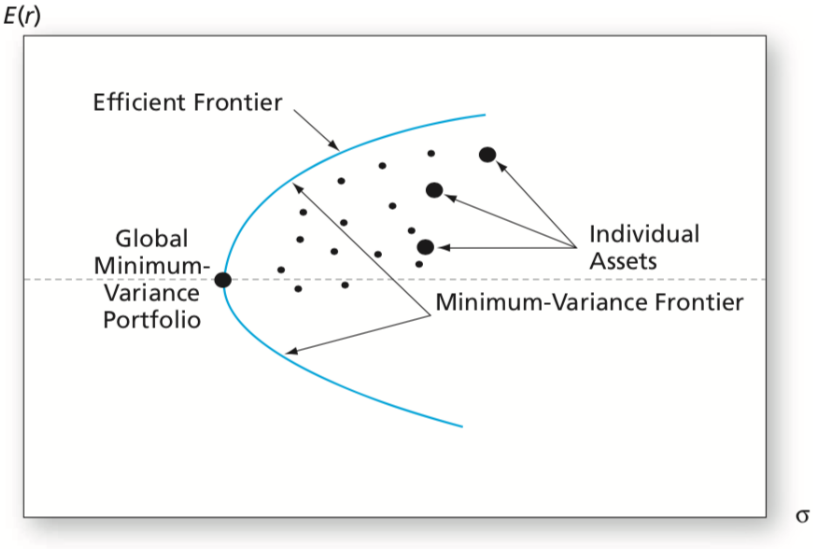


Figure 2.1: Illustration of the efficient frontier when combining risky assets. The market portfolio is located on the frontier. (Bodie et al., 2014, p.220)

The portfolios located on the minimum-variance frontier, at the global minimum-variance portfolio and upward, provide the best risk-return combinations. Therefore, the part of the frontier that lies above the global minimum-variance portfolio is known as the efficient frontier. Figure 2.1 shows that all of the individual assets lie to the right of the efficient frontier. This indicates the existence of a portfolio with the same risk but a higher return. Therefore, a risky portfolio containing only one risky asset is inefficient. For the portfolios located on the lower part of the minimum-variance frontier, there is a portfolio with the same risk associated with it but a greater expected return located directly above it. Therefore, the part of the frontier that lies beneath the Global minimum-variance portfolio is considered inefficient (Bodie et al., 2014).

The model assumes that all investors are rational and risk-averse, prefer certainty over uncertainty. Moreover, the model assumes that investors prefer higher returns over lower returns. Therefore, the optimal portfolio constructed under the modern portfolio theory focuses on minimizing the variance, subject to the constraints that the weights of the portfolio sum up to one and the portfolio's return at least achieves the target return (Markowitz, 1952). Short-selling restrictions are introduced by adding a constraint to the optimization problem, where the weights of each asset in the portfolio must be positive. The construction of the optimal portfolio can be stated as a mathematical optimization problem:

$$\min_{w} \frac{1}{2}\mathbf{w}^T\Omega\mathbf{w} \tag{2.5}$$

Subject to

$$\mathbf{w}^T\mu \geq r_p, \tag{2.6}$$

$$w \geq 0 \tag{2.7}$$

Where $\mathbf{w}$ is the portfolio weight vector and μ the vector of the expected return of each asset in the portfolio, with r_P denoting the target return and 1_N a vector of ones, Ω is the covariance matrix for the returns on the assets in the portfolio.

The MPT helps us understand important aspects of portfolio management, the concept of an efficient portfolio, and the bellow presented mean-variance efficient portfolio used as a benchmark in this study, the minimum variance portfolio.

2.1.2 Minimum Variance Weighting

The minimum variance portfolio is a crucial segment of the MPT, minimizing the portfolio's volatility by assigning weights to the least volatile assets. The minimum variance portfolio is heuristic-based, independent of expected returns, and has the lowest risk of all mean-variance efficient portfolios. Therefore, the portfolio is positioned on the very left tip of the efficient frontier, as seen in figure 2.1.

The optimization setup is as follows,

$$\arg\min_{w}(\mathbf{w}^T\Omega\mathbf{w}) \tag{2.8}$$

2.1.3 Naive Diversification

The naive diversification strategy involves holding equal weights in each risky asset considered. Therefore, the weights of each asset in the "naive portfolio" are defined as:

$$w_i = 1/N$$

Where N is the amount of risky assets. This strategy does not utilize any optimization of the portfolio and completely ignores the available data.

2.1.4 Market capitalization weighting (OMXS30)

A simple technique to implement when constructing a portfolio is the market capitalization weighting method. When stock prices fluctuate, the portfolio automatically re-balances and assigns the largest weights to the largest companies. The portfolio can be interpreted as the market portfolio (Zhang et al., 2009). The market capitalization weights are specified as:

$$w_i = \frac{p_i \cdot n_i}{\sum_{i=1}^{n} p_i \cdot n_i} \tag{2.9}$$

where p_i is the price and n_i the number of outstanding shares of stock i at the time of rebalancing.

2.2 Machine Learning Models

To understand the background and application of deep reinforcement learning, the model used in this paper, one must understand the concept of the models' constituents. The constituents of the model are two other forms of machine learning: artificial neural network and reinforcement learning.

2.2.1 Artificial Neural Network

Artificial neural network (ANN) is a form of machine learning that uses a layered representation of data. The concept behind ANN is loosely based on the human brain. The model is fed with some data, called the input, which is the first layer of the model. The input is then transformed as it goes through the next layer, called a hidden layer. The network can consist of a single hidden layer or several hidden layers. The hidden layers transform the input into an output, known as the last layer and the model target.

The term deep learning refers to the multi-layer ANN models. In the multilayered network, all neurons in a particular layer are connected to all neurons in a subsequent layer. The connections in the network can never skip a layer or form loops backward since the information flow is of feedforward type. A feedforward type works in the way that the output from one layer of neurons feeds forward into the next layer of neurons.

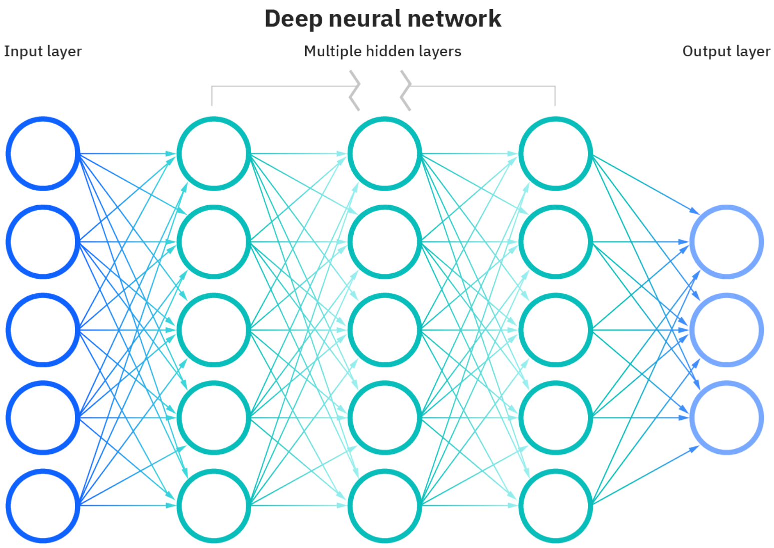


Figure 2.2: Overview of a fully connected multilayer feedforward neural network with multiple hidden layer.(IBM, 2020)

Figure 2.2 shows the architectural layout of a multilayer feedforward neural network for the case of three hidden layers. The multiple neuron units in the ANN connect with the help of weights. The structure of a single neuron, also known as a perceptron, can be mathematically defined as (Heaton, 2011) :

$$y = f\left(\sum_i (w_i * h_i) + b\right) \tag{2.10}$$

Where y represents the output, w represents the weights, h represents the input values, b represents the bias term and f represents the activation function. The activation function helps the model approximate virtually any function. The actual output of the model is defined by the outputs from the previous layers, which are multiplied with the corresponding weights and then summed together with the bias term and passed on to the activation function.

A loss function, L, is introduced to help approximate the accuracy of the feedforward ANN. The loss function is measured as the error between the predicted output and the expected output of the network. As the loss function decreases, the robustness of the network increases (Heaton, 2011). The most commonly used loss functions are the Mean Absolute Error (MAE) and the Mean Squared Error (MSE), defined as:

$$L_{MAE} = \frac{1}{N}\sum_{k=1}^{N} |\hat{y}_k - y_k| \tag{2.11}$$

$$L_{MSE} = \frac{1}{N}\sum_{k=1}^{N} (\hat{y}_K - y_K)^2 \tag{2.12}$$

ANNs learn by processing different examples, known as the training process. They contain a known input and result. The model learns by determining the difference, known as the error, between the output of the model and the target output given. The model does this with the help of a backpropagation algorithm, which aims to optimize the weights by tracking the error term back to the neuron units. The model does this by computing the partial derivative of the loss function for the neuron weights and biases and adjusting them to minimize the loss function. The partial derivatives are given by:

$$[\frac{\partial L}{\partial w_{1,1}}, ..., \frac{\partial L}{\partial w_{n,m}}, \frac{\partial L}{\partial b_1}, ..., \frac{\partial L}{\partial b_n}] \tag{2.13}$$

The definitions above help us understand how the ANN employed in our DRL models approximate the policy and value functions presented below. In this study, we employ an ANN called Long Short-Term Memory (LSTM), an ANN that can process not only single data points but also entire sequences of data. The network has three constituents: a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the gates regulate the flow of information into and out of the cell.

2.2.2 Reinforcement Learning

Reinforcement learning (RL) is an area of machine learning that trains models to make a sequence of decisions. RL allows an agent to learn to achieve a goal in an uncertain, potentially complex environment. The agent faces a game-like situation. The model implements a trial-and-error method to solve the problem. To get the machine to reach a preferred solution, the agent receives either rewards or penalties for the actions it performs.

The main objective of RL is that it follows a Markov decision process, used to model the environment in reinforcement learning (Littman & Szepesvári, 1996). One can define the process as a sequence of states that is Markov if and only if the probability of moving to the next state depends only on the present state and not on the previous state. The formal definition of the process is:

Definition 2.2.1 *A Markov desicion proccess is a tupple (S,A,P,γ,R),where:*

- *S is a finite set of states*
- *A is a finite set of actions*
- *P is the state transition probability matrix:* $P[S_{t+1} = s`|S_t = s, A_t = a]$
- $\gamma \in [0, 1]$ *is called the discount factor.*
- $R : S \times A \rightarrow \mathbb{R}$ *is a reward function*

The goal of RL is to maximize the expected value of the return by choosing the optimal policy and value function. A policy, μ, is the thought process behind picking an action. It is a function that maps the states to the actions (Littman & Szepesvári, 1996). If the policy is deterministic, then we have:

$$\mu : S \rightarrow A \tag{2.14}$$

$$a_t = \mu(s_t) \tag{2.15}$$

The goal is to maximize the expected return where the return, G_t, is a summation of all the discounted rewards received by the agent from time step t onwards:

$$G_t = R_{t+1} + \gamma R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.16}$$

The value function, $V_\mu(s)$, estimates the expected return starting from a specific state:

$$V_\mu(s) = E_\mu[G_t|S_t = s] = E_\mu[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \tag{2.17}$$

The action-value function $Q_\mu(s, a)$ is the expected return starting from state s, taking a specific action, a, and then following the policy μ:

$$Q_\mu(s, a) = E_\mu[G_t|S_t = s, A_t = a] = E_\mu[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a] \tag{2.18}$$

With the help of the Bellman equation, the value function can be decomposed into two parts: the immediate reward, R_{t+1}, and the discounted value of the successor state, $\gamma V_\mu(s')$:

$$\begin{aligned} V_\mu(s) = E_\mu[G_t|S_t = s] &= E_\mu[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \\ &= E_\mu[R_{t+1} + \gamma \sum_{k=1}^{\infty} \gamma^k R_{t+k+1}|S_t = s] \end{aligned}$$

$$= R_{t+1} + \gamma E_\mu[\sum_{k=1}^{\infty} \gamma^k R_{t+k+1}|S_t = s] = R_{t+1} + \gamma V_\mu(s')$$

The Bellman Equation for the action-value function can similarly be written as:

$$Q_\mu(s, a) = R_{t+1} + \gamma Q_\mu(s', a') \tag{2.19}$$

The reward can be maximized by finding the optimal Value function, a function that yields maximum value compared to all other value functions. The Bellman Optimality Equation expresses that the value of a state under an optimal policy must be equal to the expected return from the best action in that state:

$$V^*(s) = \max_a \ Q^{\mu^*}(s, a) = \max_a \ E[R_{t+1}V_\mu(s')|S_t = s, A_t = a] \tag{2.20}$$

$$Q^*(a, s) = E[R_{t+1} + \gamma \ \max_{a'} Q^*(s', a')|S_t = s, A_t = a] \tag{2.21}$$

There are three major approaches to reinforcement learning:

- **Critic-only approach:** In the critic-only approach, the agent takes an action based on a value function, Q. With the help of Q, the agent can analyze the state of the environment and from there base its decision on the best outcome.

$$Q^*(a_t, s_t) = E[R_{t+1} + \gamma \ \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})|s_t = s_t, a_t = a_t] \tag{2.22}$$

- **Actor-only approach:** In the actor-only approach, the agent senses the state of the environment and acts directly. The agent acts without the need of a value function to compute and compare expected outcomes of different actions. The model achieves this by specifying the policy as a set of parameters, θ:

$$a_t = \mu_\theta(s_t) \tag{2.23}$$

- **Actor-Critic approach:** The actor-critic approach, the approach implemented in this study, aims to combine the advantages of the critic-only and the actor-only methods. The model does this by simultaneously deploying an actor, determining the agent´s action given the current state of the environment, and a critic, evaluating the selected decision of the agent.

2.2.3 Deep Reinforcement Learning

Deep reinforcement learning (DRL) combines neural networks with reinforcement learning. The neural network helps the agent from the RL algorithm learn how to reach its goal by approximating the policy and the value function. Figure 2.3 depicts an overview of a DRL model. DRL utilizes both function approximation and target optimization. The advantage of DRL is that the algorithms can take in large inputs of data and decide what actions to perform to optimize the objective.

The DRL algorithm is frequently used in fields such as robotics, video games, and natural language processing. The efficiency of DRL made it possible for the first system, Deepmind´s, to beat the world champions of the Go board game. This was done with the help of DeepMind´s Alphago, a single system that taught itself from scratch how to master the game.

There are two commonly used deep actor-critic algorithms, Deep Deterministic Policy Gradient (DDPG) and Advantage Actor-Critic (A2C). The DDPG, introduced by Lillicrap et al. (2015) is an algorithm that learns the Q-function and the policy. DDPG uses off-policy data and the Bellman equation to learn the Q-function used to learn the policy. The A2C algorithm uses multiple agents to avoid using a replay buffer. Thus, each agent works independently, with different data samples, to interact with the same environment (Mnih et al., 2016).

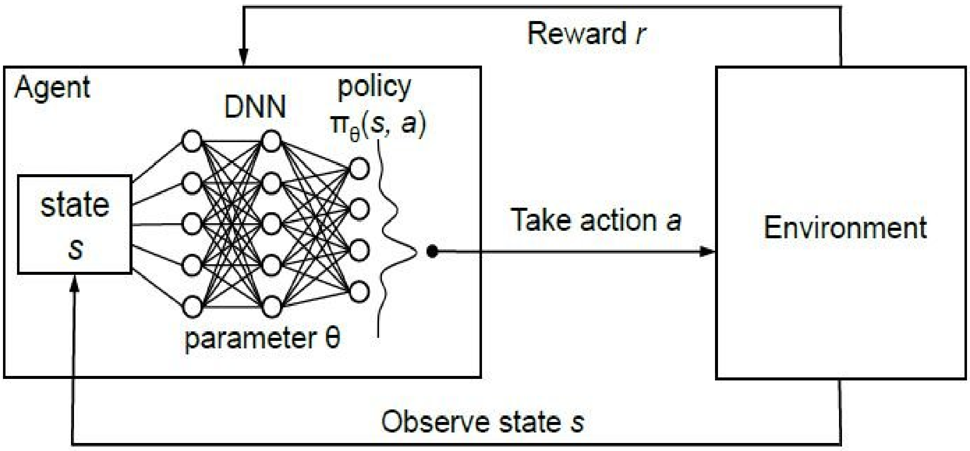


Figure 2.3: Overview of a deep reinforcement learning model relying on a fully connected multilayer feedforward neural network with one hidden layer. (Mao et al., 2016)

2.3 Sharpe Ratio

The Sharpe ratio is a financial measure introduced by Nobel laureate Sharpe (1966). The Sharpe ratio measures the performance of an investment´s, one single security or portfolio of securities, return compared to its risk. Sharpe (1966) defines the ratio as:

$$S_P = \frac{E[r_P - r_f]}{\sigma_P} \tag{2.24}$$

Where r_P is defined as the portfolio return, and r_f is the risk-free return. σ_P is defined as the standard deviation (risk) of the excess return of the portfolio. Therefore, the definition concludes the additional amount of return that the investor receives per unit of increase in risk. (Sharpe, 1966).

2.3.1 Differential Sharpe Ratio

The differential Sharpe ratio was derived by Moody et al. (1998) as a measure for online methods that find approximate solutions to stochastic dynamic programming problems, optimization of trading systems performance. Moody et al. (1998) states that to make reinforcement learning more efficient the influence of the return at time t on the Sharpe ratio needs to be calculated [1]. This result is obtained by deriving the differential Sharpe ratio. The differential Sharpe ratio, D_t, is the derivative of the Sharpe ratio for period t, S_t, for a first-order exponential moving average decay rate η in the first and second moments of the returns:

$$D_t \equiv \frac{\partial S_t}{\partial \eta} = \frac{B_{t-1}\Delta A_{t-1}\Delta B_t}{B_{t-1} - A_{t-1}^2} \tag{2.25}$$

Where A_t and B_t are exponential moving estimates of the first and second moments of the returns for period t, R_t:

$$A_t = A_{t-1} + \eta\Delta A_t = A_{t-1} + \eta(R_t - A_{t-1}) \tag{2.26}$$

$$B_t = B_{t-1} + \eta\Delta B_t = B_{t-1} + \eta(R_t^2 - B_{t-1}) \tag{2.27}$$

2.4 Efficient Market Hypothesis

The Efficient market hypothesis (EMH) had its breakthrough after a paper published by Fama (1970), reviewing empirical and theoretical research on the theory. The EMH states that asset prices reflect all information available on the markets. Therefore, performing excess market returns is impossible for investors on an efficient market since assets always trade at their fair value. According to the EMH, one can only outperform the market by purchasing riskier assets. A market is classified as efficient if the prices always fully reflect all available information. The EMH divides the efficiency of the market into three ”information subsets” Fama (1970):

- **Weak form efficiency**: A market where all historical prices and public information of an asset reflect the current asset price. Thus, depending on historical data as a prediction tool is of no use to an investor.
- **Semi-strong form efficiency**: A market where prices incorporate the information described in the weak form, as well as all new public information. Thus, price adjustments happen rapidly, making all fundamental and technical analyses useless for excess return.
- **Strong form efficiency**: the highest form of efficiency, including and containing both previous subsets. A market where prices fully reflect all public and private information at any given time. Thus, making insider information useless for excess return.

[1] The efficiency of the differential Sharpe ratio, when applied to RL, is discussed more thoroughly in section 3.2.2

3 Previous research

There is vast literature available on the topic of portfolio optimization, both on popular portfolio optimization methods and the application of deep reinforcement learning in the area. Therefore, this section focuses on the conventional models used as benchmarks and on the most relevant works that have applied reinforcement learning in portfolio optimization.

3.1 Conventional Methods

One of the most popular approaches to construct and optimize a portfolio follows the procedure presented by Markowitz (1952) called Mean-variance analysis or Modern Portfolio Theory (MPT). The MPT focuses on optimizing the risk-return tradeoff in a diversified portfolio and creating a portfolio less volatile than the sum of its compounds. Despite its popularity and contribution to Markowitz winning the Nobel prize in Economics, the model has faced criticism. According to the critics, The assumptions underlying the theory are not warranted by the empirical evidence (Maillard et al., 2010). According to Merton (1980), the solution of the mean-variance analysis tend to be overly sensitive to the input parameters, such that small changes can lead to significant variations in the composition of the portfolio. The sensitivity of the solution is most notable in expected returns.

DeMiguel et al. (2009) further issues a more practical criticism towards the modern portfolio theory. DeMiguel et al. (2009) evaluate the out-of-sample performance of the mean-variance portfolios and compare them to the outcome of portfolios constructed using naive diversification, equal share in each asset (1/N). The results show that the estimation errors for the optimized portfolios are so high that none of them consistently outperformed the naively diversified portfolio in risk-adjusted return. DeMiguel et al. (2009) and other studies (Leung et al., 2012) have also introduced improvements to the modern portfolio theory. However, according to Maillard et al. (2010), investors prefer more heuristic-based portfolio construction methods, simple techniques that are easy to implement and do not depend on expected returns. Thus heuristic techniques are more robust. The naive portfolio, the market capitalization portfolio, and the minimum variance portfolio are examples of such heuristic methods. Therefore, we choose to incorporate these heuristic based weighting techniques as benchmarks to our DRL models in this study. However, the minimum variance portfolio suffers from the lack of portfolio concentration, an issue that the naive portfolio solves since it has equal weights in all the shares. However, the drawback of the naive portfolio is that it can suffer from a lack of diversification in risk if the risks assigned to each asset are severely different (Maillard et al., 2010).

As stated before, believers of market efficiency, as described by Fama (1970), are drawn to the market portfolio, which is often based on the market capitalisation technique. The EMH states that different technical asset selection strategies can not outperform the market portfolio without including extra risk to the portfolio (Fama, 1970). Thus, we include the market capitalization weighted portfolio (OMXS30), since it can be interpreted as the market portfolio (Zhang et al., 2009), as a benchmark to our model to evaluate whether our DRL models are a good fit on the Swedish stock market.

All of the conventional methods mentioned above are widely popular and still widely used by investors on the market. However, over the last years, more complex strategies by implementation of machine learning have taken space in the portfolio management literature.

3.2 Application of Deep Reinforcement Learning

Researchers have divided opinions regarding the application of deep reinforcement learning to real-world problems. Several researchers mentioned below have successfully applied a DRL approach to solving real-world financial problems. Other researchers criticize the DRL approach to real-world problems. A common criticism of neural networks and deep reinforcement learning is that they require too much training to be efficiently applied to real-world environments (Oleinik, 2019). This will, be an alluring point in this paper since we take a deep reinforcement learning approach to solve a real-world problem by testing the proposed portfolio optimization strategy on the OMXS30.

There are three main methods used when implementing a DRL method in financial research: critic-only, actor-only, and actor-critic approach (Fischer, 2018). Thus the literature on previous research done in the area is also divided into these three sections. In this section, literature on previous research done with the help of the three mentioned approaches are reviewed and compared.

3.2.1 Critic-Only

The critic-only approach consists of only one agent called the critic. The agent decides on the subsequent action based on the value function, Q. With the help of Q, the agent can analyze the state of the environment and base its decision on the best outcome.

The application of reinforcement learning to portfolio management was first introduced by Neuneier (1996) who used a critic-only approach. Neuneier (1996) formalized the portfolio allocation problem as a Markovian decision problem. The problem was solved using a reinforcement learning framework. In this paper, the framework was used for approximating the value function. The value function, in turn, was used by the agent to decide between two strategies: choosing at each time step between currency pairs (U.S. dollar or Deutsche Mark) and choosing between a risky asset (DAX Index) or holding the risk-free equivalent (German government bonds). Neuneier (1996) concludes that one can successfully apply a reinforcement learning approach to solve the portfolio management problem. More recent research (Lucarelli & Borrotti, 2020), unlike Neuneier (1996), uses deep reinforcement learning with the critic-only method. The difference between the two methods is that the latter uses a neural network to approximate the Q-value function. The use of an ANN shows that one can minimize the mean squared error compared to only using reinforcement learning (Yang et al., 2020). Although, even when applying DRL, the critic-only approach is not practical for a portfolio with several stocks. Since prices are continuous, the method only works with discrete and finite state and action spaces. Therefore, the critic-only approach is not suitable for implementation in this study and one must look to other DRL approaches to construct a portfolio with several assets.

3.2.2 Actor-Only

Moody et al. (1998), contrary to Neuneier (1996), use an actor-only approach (also called a Recurrent reinforcement learning, RRL), an approach where the agent senses the state of the environment and acts directly without the need of a value function to compute and compare expected outcomes of different actions. Moody et al. (1998) used the RRL to optimize the differential Sharpe ratio, based on Sharpe (1966) measure of risk-adjusted returns, to trade a single financial security. Instead of proposing the traditionally used Sharpe ratio as a reward function to the method, Moody et al. (1998) suggests the use of the differential Sharpe ratio as a reward function. The reason behind the use of the differential Sharpe ratio, according to Moody et al. (1998), is its many benefits regarding efficient online optimization:

- **Simplifying the use of recursive updating:** The calculation of A_t and B_t from (2.26) and (2.27) enables the recursive updating of the exponential moving Sharpe ratio forthright. Thus, recomputing the average and standard deviation of returns for the entire trading period is unnecessary for updating the Sharpe ratio, an advantage that is of good use in a study like this one.

- **More efficient out-of-sample performance**: The differential Sharpe ratio outperformed the running and moving average Sharpe ratios in the out-of-sample performance.

- **Straightforward interpretation**: The use of the differential Sharpe ratio enables the interpretation of how risk and reward affect the Sharpe ratio. Since D_t isolates the contribution of the current return to the exponential moving average Sharpe ratio.

The benefits reported by the study of using the differential Sharpe ratio as reward function has led to the choice of implementing the measure as a reward function in this study. Moody & Saffell (2001) extends their previous studies and makes further investigation about their direct policy optimization method by introducing a differential downside ratio to better separate undesirable downside risk from the preferred upside risk. The result obtained in the paper shows that RRL can be successfully applied to optimize a portfolio consisting of only one risky asset and one risk-free asset. Other researchers built on the RRL approach by either expanding the model (Maringer & Ramtohul, 2012) or varying the decision function (Almahdi & Yang, 2017). When it comes to more recent research, a deep reinforcement learning model is used instead of conventional reinforcement learning models. Instead of having the ANN learn the Q-value, as in the critic-only approach, the ANN learns the policy itself. Gold (2003) uses RRL to compare single-layered networks with two-layered networks, finding that the single-layer network outperforms the two-layer network.

An advantage to the actor-only approach is that it solves the problem of handling continuous action space environments. However, the RL algorithms generate an output of discrete trading signals on an asset. Therefore, the algorithms are limited to single-asset trading, in line with the former research (Moody et al. (1998), Moody & Saffell (2001)) implementing the said approach, and therefore not applicable to general portfolio management problems, where trading agents manage multiple assets, which is done in this study.

3.2.3 Actor-Critic

The actor-critic approach, the approach that is used in this study to construct two DRL portfolios, aims to combine the advantages of the critic-only and the actor-only methods.

The combination is done by simultaneously using an actor and a critic. The actor determines the agent´s action, given the current state of the environment, and the critic evaluates the selected decision. The neural network in this approach approximates both the policy and the Q-value function.

Recent algorithmic developments from Google DeepMind, have made it possible to solve the continuous action space problem. This progress was made by developing previous algorithms, created by Silver et al. (2014), called Deterministic Policy Gradients method. A method that targets modeling and optimization of the policy directly by using off-policy data. From this, Google DeepMind has devised a policy-gradient actor-critic algorithm called Deep Deterministic Policy Gradients (DDPG) (Lillicrap et al., 2015). They solve the continuous action space problem, using a neural network to approximate the action policy function while training a second network to estimate the reward function. Therefore, we implement the DDPG model in this study to construct one of our tested portfolios.

Mnih et al. (2016) on the other hand, developed an asynchronous actor-critic framework, called advantage actor-critic (A2C). Mnih et al. (2016) show that the asynchronous version of the actor-critic framework outperforms current state-of-the-art frameworks while needing half the training time. The A2C method also improves the neural network´s attribute of approximating the functions. Thus, we have chosen to implement the A2C method to construct the second portfolio evaluated in this study.

Xiong et al. (2018) and Noguer i Alonso & Srivastava (2020) use deep reinforcement learning for optimizing portfolios and comparing the optimized portfolio to traditional portfolio optimization models, e.g. the minimum variance portfolio. Xiong et al. (2018) do this with the help of the third approach, the actor-critic approach. The algorithm used by Xiong et al. (2018) is the DDPG developed by the researchers mentioned above. The DDPG models large state and action spaces, a target network that stabilizes the training process and experience replay that removes the correlation between samples and increases the use of data. Xiong et al. (2018) found that the DRL network outperformed both the index used as a benchmark (Dow Jones Industrial Average) and the min-variance portfolio allocation method in both accumulated return and Sharpe-ratio. Noguer i Alonso & Srivastava (2020) obtain similar results, where their algorithm outperforms both the minimum variance portfolio and several other conventional strategies, including the naive strategy.

Kang et al. (2018), contrary to Xiong et al. (2018), uses the asynchronous actor-critic algorithm mentioned above to solve the portfolio management problem. Kang et al. (2018) found that the A2C outperforms the S&P 500 while also observing that the framework only needs half the training period to converge much faster than usual DRL frameworks. However, Kang et al. (2018) also found that the model was more effective during the training period than the testing period. In this study, we did not find that the A2C model was more effective during the training period than the testing period.

In this study, we also benchmark our DRL constructed portfolios, using the DDPG like Xiong et al. (2018) and the A2C like Kang et al. (2018), against conventional methods as in the above mentioned papers. However, we do this on the Swedish stock market, which has not been done in previous research.

Yang et al. (2020) implements three DRL algorithms for automated stock trading to compare the three strategies against each other as well as against the Dow Jones Industrial Average (DJIA30). The three algorithms used are the two algorithms mentioned above, A2C and DDPG, and Proximal Policy Optimization. All three DRL algorithms outperformed the market when comparing returns and risk-adjusted returns. The A2C outperformed the DDPG in both returns and risk-adjusted returns, in line with the results of this study.

4 Empirical Data

4.1 Description of Data-set

The stocks from the OMXS30 index were initially selected to comprise the dataset used. The OMXS30 is a market index containing the 30 most traded stocks on the Nasdaq Stockholm stock exchange. The index is market-value-weighted, meaning that the weights assigned to the components depend on the total market value of their outstanding shares. The constituents are revised two times a year, in January and July. However, this research disregards the changes of stocks included in the index. We obtained the data from Yahoo! Finance. Our data ranges from 01/01/2000 to 31/12/2020 and contains daily observations. The stocks selected are from the index composition in 2020 since this will be the end of the period. The restriction imposed is accepted not to impact the outcomes since the purpose of this paper is to evaluate and compare portfolio optimization models, not to create mimicking portfolios of the index. Further, by selecting the constituents of the index included at the end of the period, the issue of survivorship bias is evaded (Sharpe, 1966).

Figure 4.1 shows the closing price for OMXS30 from 2000 to 2019:

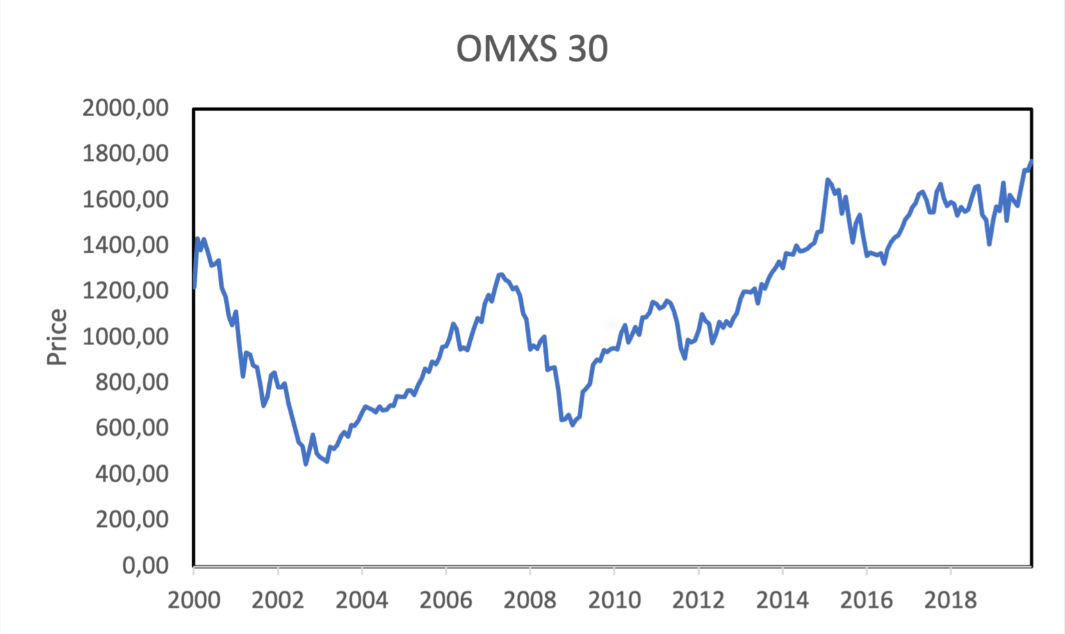


Figure 4.1: Chart over closing price for OMXS30, 2000-2019. Data obtained from Swedish House of Finance

4.2 Data Pre-Processing

This research disregards the changes of stocks included in the index, as explained above. Therefore, we had to do some data pre-processing to fit the model. A few of the chosen companies had been listed on the exchange later than 2000, which constitutes missing data. We cannot train a high-quality model without removing the stocks with missing data points. Therefore, the final data set contains 28 companies in total (see Appendix A.1).

The experiment of this study has three stages, training, validation, and testing. In the training stage, the algorithms generate well-trained trading agents. The key parameters, such as learning rate, number of episodes, and others, are adjusted in the validation stage. In the testing stage, we evaluate the performance of the trading agents.

The data set is split into chronological order to produce reliable results since actual trading actions are always implemented with the current data. If we were to do it in a non-chronological way, a look- ahead-bias to the test set would occur, which would lead to unreliable results. Therefore, we split the data set into these three periods, as shown in figure 4.2: The training period, from 01/01/2000 to 31/12/2014, the validation period, from 01/01/2015 to 31/12/2015, and finally the testing period, from 01/01/2016 to 31/12/2020.

The lengths of the validation and trading periods are in line with previous research (Xiong et al., 2018). However, in this study we choose a longer training period.

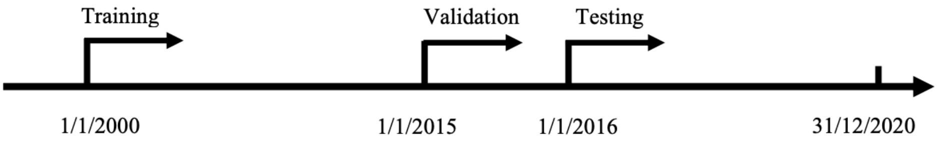


Figure 4.2: Data splitting.

5 Methodology

5.1 Assumptions

Some assumptions were made when constructing the portfolios.

Short-sales constraints are implemented in this study. This is also a reasonable assumption since not all fund managers are allowed to take short positions.

Transaction costs are not taken into consideration in this study. Transaction costs are very much present in the real world and important for systematic trading strategies. However, they are assumed negligible in the construction of each portfolio and assumed to not exceeded the profits of the strategies. Instead of transaction costs, a turnover statistics is deployed to compare the number of transactions between each portfolio.

No margin trading when it comes to the construction of the portfolios. Self-financing of each portfolio is assumed. Therefore, there is no inflow or outflow of equity to the portfolios except for the initial value of the portfolio.

5.2 Rebalancing

The frequency of which the portfolios in this study are rebalanced, the realignment of the portfolio asset weights, is of importance to the discussion regarding the portfolio turnover later on in the study.

All portfolios are rebalanced monthly, except for the market capitalization portfolio, which is in line with previous research on portfolio evaluation (Maillard et al., 2010). The market capitalization portfolio is rebalanced every six months at the same dates as the OMXS30 index, January and July.

5.3 The DRL Models

State

The state space, S, is a finite set of states that defines the observations that the agent receives from the environment. In this study, we acquaint the agent with the environment by choosing several states rather than just a single one. The first state incorporated into the model is the covariance matrix, a frequent feature used within portfolio management to compute the associated standard deviation of a portfolio. The covariance matrix in portfolio construction can help determine which stocks to include in the portfolio. The covariance measures in what direction stocks move in. If the covariance between two stocks is positive, they move in the same direction. However, if it is negative, they move in opposite directions.

One of the most frequently used features, closing price, p_t, is incorporated into the model. Using the closing price as a state to the models is of great importance in this study since it contradicts the assumption of the EMH regarding the use of historical data as an indicator for future returns (Fama, 1970).

The state tensor, s_t, represents the state. A tensor is a mathematical object comparable to but more general than a vector. The input to the model at the end of period t is a tensor, s_t, of rank 3 with shape (d, n, f) where d is a fixed amount defining the length of the observation in days, n is the number of stocks in the environment, and f is the number of features. An array of components that are functions of the coordinates of a space represents the tensor.

Action

As explained previously, the action space, A, is a finite set of actions available to the agent from the state, S. In this paper the action represents the portfolio weight for each stock. Since short-sales constraints is implemented, a is within $[0, 1]$. The action, a_t is represented with the weight vector, w_t, thus $\sum_{i=0}^{n} a_{i,t} = 1$

State transition probability

In this study, the actions taken by the agent assume not to affect the state transition presented in section 2.2.2. This assumption holds since the agent´s policy is updated based on the observed reward in each step. Thus, since market data is very noisy and the environment too complex, the transition probability function remains unknown for the portfolio management problem. Therefore, we define all state transitions in a state transition matrix, P. Each row in P describes the transition probabilities from one state to all possible successor states. The summation of each row is equal to one:

$$\begin{bmatrix} P_{11} & ... & P_{1d} \\ \vdots & \ddots & \vdots \\ P_{d1} & ... & P_{dd} \end{bmatrix} \tag{5.1}$$

Reward Function

We define the reward of the model as the Sharpe ratio for periods $t = [1, ..., T]$. The corresponding reward function, $r(s, a, s‘)$, will be the differential Sharpe ratio, D_t, defined as:

$$D_t \equiv \frac{\partial S_t}{\partial \eta} = \frac{B_{t-1} \Delta A_{t-1} \Delta B_t}{B_{t-1} - A_{t-1}^2} \tag{5.2}$$

Employing the differential Sharpe ratio as a reward function is motivated by Moody et al. (1998) since they found that maximizing the differential Sharpe ratio yields more consistent results than maximizing profits. They also found that agents trained to maximize the differential Sharpe ratio achieve better risk-adjusted returns than those trained to maximize profit.

Algorithms

In this paper, we implement an ANN to help build two deep actor-critic structures by approximating the value function, $Q(s, a)$, and learning the policy, μ, for the portfolio optimization task. The two model-free reinforcement learning algorithms employed in this paper are the Advantage Actor-Critic (A2C) and Deep Deterministic Policy Gradient (DDPG). We use a gradient descent optimization algorithm called ADAM (Kingma & Ba, 2014) to learn and update the ANN weights.

Both algorithms rely on the same deep neural architecture, Long Short Term Memory (LSTM). The LSTM is an ANN that can process not only single data points but also entire sequences of data. The LSTM is practical in this study since we use arrays of data.

5.4 Benchmark Portfolio Strategies

Market Capitalization Weighted Portfolio

We construct the market capitalization-weighted portfolio by collecting data and weight of each constituent from the capitalization-weighted index, OMXS30. The market capitalization data used for each stock at each re-balancing date is supplied by Yahoo Finance.

Minimum Variance Portfolio

The minimum variance portfolio is constructed by minimizing the portfolio´s variance by solving the problem in (2.8). We also include our constraints to the problem,

$$such\, that \begin{cases} 1^T\mathbf{w} & = 1, \\ w & \geq 0 \end{cases} \tag{5.3}$$

This portfolio is referred to as MV in the next section.

Naive Portfolio

The Naive portfolio is constructed by equally weighting each stock from the index. Thus, the weight for each share is $\frac{1}{n}$ where n is the number of stocks in the index.

5.5 Performance Measures

In this study, we implement eight different performance measures to evaluate the five different portfolio construction strategies. Performance statistics, discussed below, correlation, concentration and turnover. The chosen measures are in line with previous research within the area of portfolio evaluation (Maillard et al., 2010). The mentioned measures should provide an in-depth discussion of the differences between the compared portfolio strategies. They should also tell whether DRL is a competitive and valuable strategy in the area.

The Swedish risk-free interest rate for the whole period 2000-2020, obtained from the Swedish national bank, is used whenever a risk-free rate is needed for the performance measures.

Annual return

Annual return is the annualised geometric average return. The Annual return for each portfolio was calculated using the bellow formula (CAGR=compound annual growth rate):

$$CAGR = \left(\left(\frac{Final\,value}{Initial\,value}\right)^{\frac{1}{years}}\right) - 1 \tag{5.4}$$

Cumulative Return

Cumulative returns are cumulative sum of the daily returns. It can also be calculated as a single number, based on the final and initial value:

$$CR = \frac{Final\,value - Initial\,value}{Initial\,value} \tag{5.5}$$

Annual volatility

The annual volatility (standard deviation) is the annualised volatility. The annual volatility of the portfolio was calculated by multiplying the daily volatility by $\sqrt{252}$, since there are usually 252 trading days.

Sharpe ratio

The Sharpe ratio and the calculation of the measure is explained thoroughly in section 2.3.

Maximum drawdown

Maximum drawdown is the largest peak-to-through downturn of the portfolio value. The maximum drawdown of each portfolio was calculated as:

$$MDD(T) = \max\{0, \max_{t\in(0,t)} P(t) - P(T)\} \tag{5.6}$$

Where T is the time at the end of the period and $P(t)$ is the stock price at time t.

Calmar ratio

The Calmar ratio is similar to the Sharpe ratio, it aims to provide risk-adjusted return. The Calmar ratio of the portfolios was calculated using the following formula:

$$CalR = \frac{E(r)}{MDD} \tag{5.7}$$

5.6 Significance

With the help of the performance measures above, the different strategies can be evaluated and compared. However, we must test the significance of the performance measures to compare the methods in a correct and meaningful way. This study aims to conclude whether DRL can be successfully applied to portfolio management. Thus, the results of the two algorithms will be compared to the benchmarks and each other and tested for significant differences. For this, a classical one-sample t-test is conducted. The formula for the t-statistics is:

$$t(\hat{\mu}_k) = \frac{\hat{\mu}_k - \mu_0}{\sqrt{\hat{\sigma}_k^2}}\sqrt{n} \tag{5.8}$$

Where μ_0 is the mean of the benchmarks, n is the number of observations, and k is one of the DRL strategies. If the null hypothesis is true, there exists no significant difference between the measures. Thus, the two hypotheses are specified as follows:

$$H_0 : \hat{\mu}_k = \mu_0$$
$$H_1 : \hat{\mu}_k > \mu_0$$

If the confidence interval includes zero, we can say that there is no significant difference between the means of the two populations at a given level of confidence.

6 Empirical Analysis

In this section, the results of the study are presented and illustrated with the help of graphs and descriptive tables. All of the results are first interpreted to get a better understanding of the outcome so that an analysis can be done later on in the section.

6.1 Performance During a Longer Period

We start by presenting the main results of this study, being the performance of the different portfolios during the whole testing period. In figure 6.1, We illustrate the cumulative returns of all five portfolios. The A2C portfolio has the highest cumulative return of all the portfolios, followed by the DDPG portfolio. The two DRL constructed portfolios follow each other almost identically during the whole period. However, the A2C portfolio outperforms the DDPG at the end of the period. The naive and minimum variance portfolios generate similar cumulative returns, and both exceed the OMXS30, which has the lowest cumulative returns out of all portfolios.

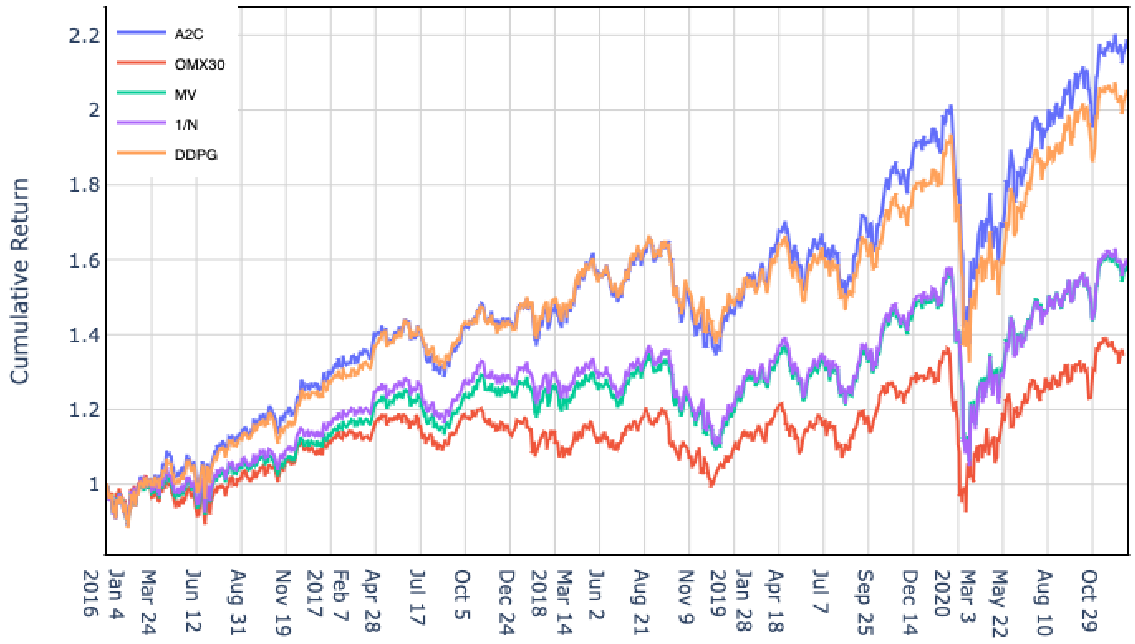


Figure 6.1: Cumulative returns for the five portfolios during 2016-2020.

In table 6.1, we present the statistics for all five portfolios. The annual return shows how much the portfolio has grown or shrunk in one year. The A2C portfolio (16,7%) and the DDPG portfolio (15.3%) had the highest average annual return over the testing period. The OMXS30 (6.35%) had the lowest average annual return. The annual volatility measures how risky the portfolios are and how volatile the returns of the portfolios are. The A2C portfolio (19.9%) and the naive portfolio (19.2%) had the highest annual volatility, and the minimum variance portfolio (18.5%) had the lowest annual volatility. The Sharpe ratio measures risk-adjusted return. Thus, a higher Sharpe ratio is preferred over a lower Sharpe ratio. The A2C (0.879) has the highest Sharpe ratio, and the OMXS30 (0.421) has the lowest. The Calmar ratio is another measure for risk-adjusted returns but as a function of the expected annual rate of return and the maximum drawdown, characterized as the maximum loss from peak to trough over a given period. Thus, the portfolio with the higher ratio performed better on a risk-adjusted basis, which in this case was the A2C (0.531) followed by the DDPG (0.489), the OMXS30 (0.198) had the lowest Calmar ratio. When it comes to the maximum drawdown, a lower value is preferred over a higher one. The minimum variance portfolio (-30.9%) had the lowest maximum drawdown. The Naive portfolio (-33.4%) had the highest maximum drawdown.

Table 6.1: Risk and return statistics over a five year period. From 2016-2020. The highest (absolute) value in each column is in bold.

Portfolio	Annual return (%)	Cumulative return (%)	Annual volatility (%)	Sharpe ratio	Calmar ratio	Maximum drawdown (%)
A2C	**16.7**	**116**	**19.9**	**0.879**	**0.531**	-31.5
DDPG	15.3	104	19.1	0.842	0.489	-31.3
OMXS30	6.35	35.8	18.9	0.421	0.198	-32.0
Minimum Variance	9.47	57.0	18.5	0.582	0.294	-30.9
Naive Diversification	9.69	58.6	19.2	0.579	0.290	**-33.4**

6.2 Performance During a Bullish Period

To get a better understanding of the model and its efficiency on the market, we zoom in on the testing period and present the outcome during a bullish period on the market. As a bullish period, we chose 2019 since the OMXS30 had the highest annual return during 2019 out of all the years included in the testing period. During the bullish period, the A2C portfolio outperformed all the other portfolios in terms of cumulative return, presented in figure 6.2. However, the DDPG portfolio did not exceed the naive and the minimum variance portfolio during this period. It performed just slightly better than the two benchmarks in terms of returns but less in terms of risk-adjusted returns. Once again, all the portfolios outperformed the index, which had the lowest cumulative return. In table 6.2, we can see similar results as in table 6.1, with A2C exceeding all of the other portfolios in terms of annual return, Sharpe ratio, and Calmar ratio. However, this time DDPG (14.9%) had the highest annual volatility.

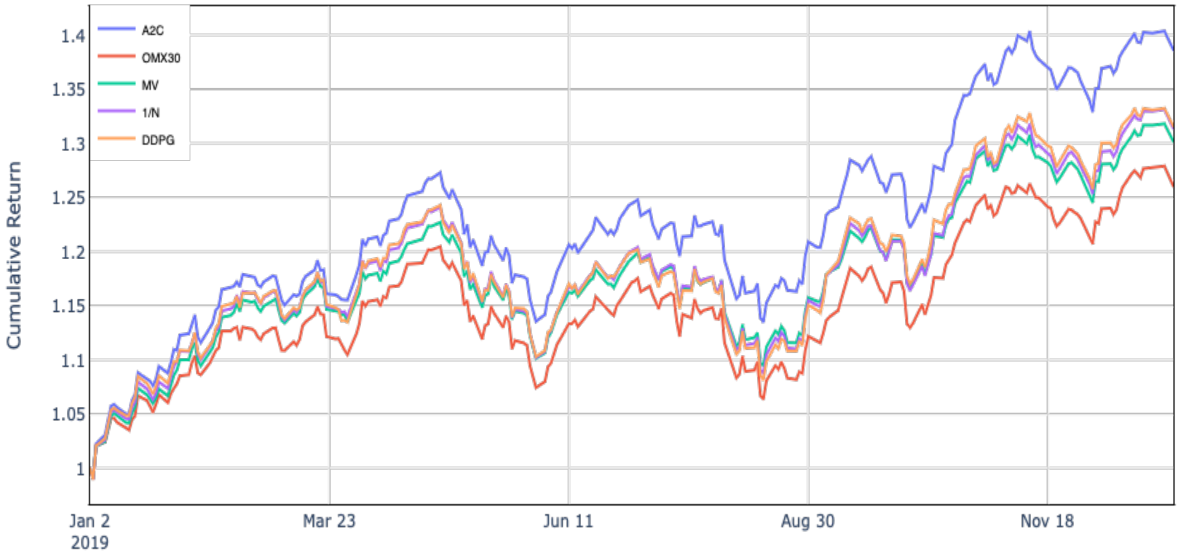


Figure 6.2: Cumulative returns for the five portfolios during 2019.

Table 6.2: Risk and return statistics in 2019. The highest (absolute) value in each column is in bold.

Portfolio	Annual return (%)	Cumulative return (%)	Annual volatility (%)	Sharpe ratio	Calmar ratio	Maximum drawdown (%)
A2C	**39.6**	**38.6**	14.8	**2.33**	**3.66**	-10.8
DDPG	32.3	31.6	**14.9**	1.95	2.49	**-13.0**
OMXS30	28.7	27.8	14.2	1.85	2.47	-11.6
Minimum Variance	30.9	30.1	13.6	2.05	2.86	-10.7
Naive Diversification	32.1	31.3	14.6	1.99	2.60	-12.4

6.3 Performance During a Bearish period

To broaden the understanding of the model and its efficiency on the market, since we evaluated it during a bearish period, we also have to evaluate it during a bearish period. As a consequence of the Covid-19 pandemic, 2020 was a turbulent year on the market. Thus, we chose 2020 as the evaluation year. During the turbulent periods, February to May, the minimum variance portfolio did better than the other portfolios. However, the A2C and the DDPG portfolios recovered and outperformed the rest at the end of the year. During this turbulent period, the OMXS30 performed better than the other two benchmarks. In table 6.3, we see similar results as earlier, the A2C performing better in terms of annual return, Sharpe ratio, and Calmar ratio. The minimum variance portfolio (-28.3%) has once again the lowest drawdown, and the naive portfolio (-32.2%) has the highest drawdown.

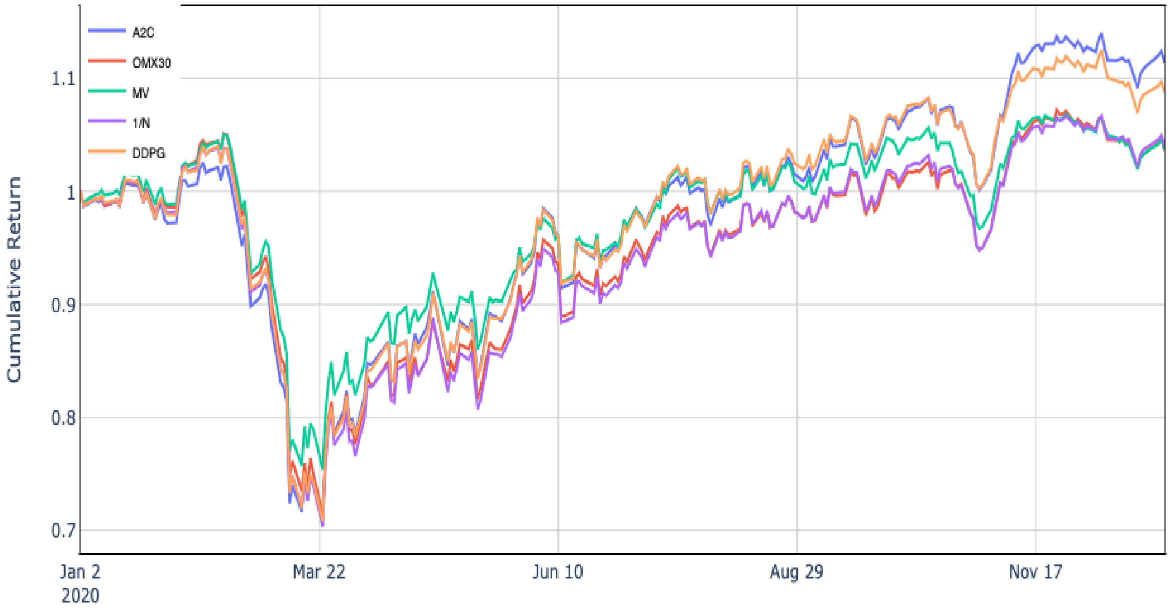


Figure 6.3: Cumulative returns for the five portfolios during 2020.

Table 6.3: Risk and return statistics in 2020. The highest (absolute) value in each column is in bold.

Portfolio	Annual return (%)	Cumulative return (%)	Annual volatility (%)	Sharpe ratio	Calmar ratio	Maximum drawdown (%)
A2C	**11.4**	**11.4**	**29.1**	**0.519**	**0.367**	-31.2
DDPG	8.81	8.81	28.5	0.440	0.276	-31.9
OMXS30	4.69	4.67	28.4	0.305	0.146	-31.9
Minimum Variance	3.77	3.77	25.8	0.274	0.134	-28.2
Naive Diversification	4.04	4.04	28.7	0.283	0.126	**-32.2**

6.4 Hypothesis Testing

Furthermore, it is of interest whether the two DRL strategies have a mean significantly different from the benchmark portfolios. As already described, we use a t-test to test the mean difference. In table 6.4, the p-values from the t-tests comparing the mean of the A2C portfolio to all the other portfolio means are presented. The A2C portfolio is considered to have a significant, at 1%, greater mean for both the annual return and the Sharpe ratio than all other portfolios.

Table 6.4: Hypothesis testing for the Annual return and Sharpe ratio during the five year period against A2C, using the one sample t-test.

Annual Return		Sharpe Ratio	
Portfolio	**p-value (t-Test)**	**Portfolio**	**p-value (t-Test)**
A2C	-	A2C	-
DDPG	0.0064***	DDPG	0.0035***
OMXS30	0.0001***	OMXS30	0.0001***
Minimum Variance	0.0001***	Minimum Variance	0.0001***
Naive Diversification	0.0001***	Naive Diversification	0.0001***

In table 6.5, we present the p-values from the t-tests comparing the mean of the DDPG portfolio to all the other portfolio means. Similar results as in table 6.4 are presented, except for the fact that the mean for annual return and Sharpe ratio of the DDPG is considered to be less than or equal to the means of A2C.

Table 6.5: Hypothesis testing for the Annual return and Sharpe ratio during the five year period against DDPG, using the one sample t-test.

Annual Return		Sharpe Ratio	
Portfolio	**p-value (t-Test)**	**Portfolio**	**p-value (t-Test)**
DDPG	-	DDPG	-
A2C	0.8772	A2C	0.5123
OMXS30	0.0001***	OMXS30	0.0001***
Minimum Variance	0.0001***	Minimum Variance	0.0001***
Naive Diversification	0.0001***	Naive Diversification	0.0001***

6.5 Correlation

In table 6.6, we present the correlation coefficients for the portfolios. The correlation coefficients show how well portfolios correlate to one another. In general, all portfolios correlate highly to one another, with correlation coefficients ranging from 0.91 (A2C and minimum variance) to 0.98 (minimum variance and naive diversification). All the portfolios are highly correlated with the market, but the naive portfolio (0.97) has the highest correlation to the market and the mean-variance (0.94) has the lowest.

Table 6.6: Portfolio correlation over the whole testing period, 2016-2020.

Portfolio	Annual Return (%)	Annual Volatility (%)	Correlations OMXS30	 A2C	 DDPG	 MV	 Naive
OMXS30	6.35	18.9	1.00	0.96	0.95	0.94	0.97
A2C	16.7	19.9		1.00	0.92	0.91	0.93
DDPG	15.3	19.1			1.00	0.92	0.94
MV	9.47	18.5				1.00	0.98
Naive	9.69	19.2					1.00

The correlation coefficients are the Pearson's correlation coefficients calculated as the covariance of the two portfolios divided by the product of their standard deviations.

6.6 Concentration

The portfolio concentration was evaluated using the average Herfindahl-Hirschman Index (HHI) value. The HHI is usually used to measure the size of firms in relation to the industry they are in. In this thesis and the stock portfolio context, it is used as a concentration measure, measuring how many different types there are in a data-set, and how evenly the weights are distributed among these. A lower HHI value is preferred.

In table 6.7 the average HHI values are presented. The portfolios with the lowest average HHI values are the Naive portfolio (0.00) since it has uniform weights after each rebalancing day, and the A2C portfolio (0.04). The minimum variance portfolio (0.20) had the highest HHI value.

Table 6.7: Portfolio concentration over the whole testing period, 2016-2020.

Portfolio	Average HHI Value
A2C	0.04
DDPG	0.06
OMXS30	0.08
Minimum Variance	0.20
Naive Diversification	0.00

Average HHI is the average of the daily Herfindahl-Hirschman Index (HHI) values, calculated as described in A.2

6.7 Portfolio Turnover

As mentioned before, transaction costs are not implemented, in absolute value, in this study. Instead, we measure the effect of the transaction costs in terms of portfolio turnover. The portfolio turnover shows how frequently the assets are bought and sold in relation to the average portfolio value. The portfolio turnover can therefore be used as a proxy for transaction costs since the two of them are closely correlated (Dow, 2007). Thus, the portfolio turnover can describe how a portfolio performs in comparison to the market portfolio when adjusted for transaction costs. As noted before the market portfolio is known for having a low portfolio turnover, which can be seen in table 6.7, due to the way it rebalances infrequently. Thus, the market portfolio is a good benchmark in terms of a portfolio with low transaction costs/turnover.

In table 6.7 the average annual portfolio turnover is presented. The portfolio with the lowest average annual turnover is the OMXS30 (0.06) followed by the Naive portfolio (0.21) and the A2C portfolio (0.27). The portfolios with the highest average annual turnover were the minimum variance portfolio (0.64) and the DDPG portfolio (0.32).

Table 6.8: Portfolio turnover over the whole testing period, 2016-2020.

Portfolio	Average annual turnover
A2C	0.27
DDPG	0.32
OMXS30	0.06
Minimum Variance	0.64
Naive Diversification	0.21

The annual portfolio turnover is defined as: $\frac{Minimum\ of\ securities\ bought\ or\ sold}{Average\ portfolio\ value}$. The average annual turnover was calculated as the average of these annual turnovers.

6.8 Discussion

In this study, different performance measures have been implemented to evaluate the two portfolios constructed with the help of the two DRL algorithms, A2C and DDPG, on the Swedish stock market. The DRL constructed portfolios were then benchmarked against the market capitalization portfolio, i.e., the portfolio tracking the OMXS30 index, the mean-variance portfolio, and the naive portfolio. The outcomes of the two DRL algorithms are also compared and discussed.

Advantage Actor-Critic (A2C)

The rationale behind the A2C method is to implement several deterministic agents working simultaneously, in the same environment but independently, to update gradients. The A2C has an actor-critic architecture and uses a neural network as a function approximator. The actor outputs the policy for a state, a vector of probabilities for each action. The critic outputs the value of a state. Therefore, the A2C is stable and efficient for stock trading and portfolio construction. We can detect the efficiency and stability of the A2C in this study. The A2C outperforms all of the benchmarks, and the DDPG, in both returns and risk-adjusted returns. The results also show that the A2C is more robust in balancing risk and return, considering that it has the highest Sharpe ratio and Calmar ratio in all of the tests performed. After performing the t-test, we can conclude that the A2C method generates significantly higher returns and balances risk and return better than the other portfolios. However, the model generates higher annual volatility than the benchmark portfolios, indicating that it chooses to include more high-risk assets in the portfolio. It also has a higher maximum drawdown than the market and the minimum variance portfolio but a lower drawdown than the naive portfolio.

The A2C portfolio highly correlates with the index, the second-highest correlation out of all the other portfolios. It has the lowest average HHI value, except for the naive portfolio. Finally, the portfolio keeps a low portfolio turnover, slightly higher than the market.

In conclusion, the A2C method succeeded constructing a portfolio that outperforms all of the proposed benchmarks, including the DDPG constructed portfolio, in terms of return and risk-adjusted performance. The results of this study are in line with previous research (Yang et al., 2020).

In addition the A2C algorithm converges (2 min 44s) much faster than the DDPG (18 min 50 s) when training the models, in line with Mnih et al. (2016).

Deep Deterministic Policy Gradient (DDPG)

The rationale behind the DDPG algorithm is the combination of frameworks for Q-learning and policy gradient to deterministically map states to actions for a more efficient fit of the continuous action space environment. The DDPG constructed portfolio outperformed all of the benchmarks during the five years and the bearish period, in terms of returns and risk-adjusted returns. However, during the bullish period, it only did slightly better than the other benchmarks. The results in terms of robustness for balancing risk and returns show similar results for the DDPG portfolio as it has a significantly higher Sharpe and Calmar ratio during the two above-mentioned periods. However, during the bullish period, the portfolio has a lower Sharpe ratio than the minimum variance portfolio and the naive portfolio, it also has the lowest Calmar ratio out of the three. The model has the lowest

Calmar ratio since it also has the highest maximum drawdown during the bullish period. The poor performance of the DDPG during this period may result from the fact that it is a short period, only one year. Therefore, the model might need a prolonged period to perform better, which it does when testing it on the five years.

The DDPG portfolio had a high correlation to the market, but compared to the other portfolios, it only had a higher correlation than the minimum variance portfolio. However, 0.95 is still considered a high correlation. The portfolio has a slightly lower HHI value than the market but slightly higher than the A2C. Finally, the portfolio keeps a low portfolio turnover. However, it has a higher turnover than the market, A2C, and naive portfolio turnover.

In conclusion, the DDPG constructed portfolio outperforms the proposed benchmarks, but not the A2C portfolio, in terms of return and risk-adjusted returns during the whole period. However, the method produces some turbulent results during shorter periods of testing.

Caveats and Market Efficiency

The proposed approach implemented in this study does have some caveats linked to it. These caveats can affect the overall performance and efficiency of the models in the real world.

First, the testing period could be extended past five years to assess the long-term value of the portfolio built by the algorithms in a more robust and significant way. The market capitalization-weighted tracking portfolios have a long history of performance. Their holdings are also infrequently rebalanced. The infrequent rebalancing leads to low transaction costs, a factor not implemented in the presented models, which is not realistic in the real world where transaction costs are very much present. However, we include portfolio turnover as a proxy measure for the transaction costs. This study does not implement margin trading either, which is present in reality as a fund tracking an index will have inflow and outflow of capital. However, margin trading is often not implemented in studies. The management of the algorithms in a real-world environment comes with its implications. Keeping up with data engineering could become difficult since data must be fed in time and must be correct and up to date for the model to make efficient decisions. The approach in this study only proposes monthly portfolio updates, compared to the high-frequency trading conducted by some portfolio managers in the real world, where decisions are made every second. Therefore, it is critical to note that the models do not consider price changes occurring when the market is closed. The models also do not consider a company going into bankruptcy and stopping trading on the market.

More than evaluating the performance of the models on the Swedish stock market, this study aims to detect whether the models can pick up market trends and make profits on these. Considering the portfolio performance observed for the A2C and DDPG algorithms, it seems that the models can pick up on market trends. This conflicts with the efficient market hypothesis presented in section (2.4). The fact that the models use historical closing prices and other appropriate factors to learn shows that it even conflicts with the weakest form of the above-mentioned hypothesis. It is also important to note that the proposed approach is not frequently used in the real world. Thus, the above-average performance of the models could depend on the fact that the models are narrowly used by investors on the market. Therefore, if more investors on the market start using the models frequently, the excess return over the market portfolio may be arbitraged away, which would be in line with the EMH.

7 Conclusion

The objective of this study was to evaluate whether a deep reinforcement learning approach can be applied to optimize a stock portfolio on the Swedish stock market. Therefore, two DRL algorithms, A2C and DDPG, relying on the same artificial neural network called LSTM, were implemented to construct two portfolios. The portfolios consisted of 28 stocks from the OMXS30 index and were benchmarked against commonly used weighting techniques in portfolio management. The different strategies were evaluated with the help of different performance measures.

This study concludes that a DRL approach can be applied to the Swedish stock market for constructing an optimized portfolio based on the portfolio performance observed for the A2C and DDPG algorithms. Both of the DRL constructed portfolios significantly outperformed all of the conventional benchmarks in terms of risk and returns statistics over a five-year period. The results of this study are in line with similar previous research conducted in the area (Xiong et al. (2018), Yang et al. (2020), Noguer i Alonso & Srivastava (2020)) where the DRL portfolios could be implemented as a portfolio management tool and outperform conventional methods.

When compared to each other, the A2C outperforms the DDPG in all the tests. The A2C achieves higher cumulative returns (+12%) and higher risk-return ratios (+0.037 in Sharpe ratio, +0.042 in Calmar ratio).

The two DRL methods also seem to pick up market trends and profit from them. A result which conflicts with the efficient market hypothesis. However, it is critical to note that the methods come with some caveats when implementing them in the real world. There exist data engineering that needs to be tweaked for the models to be efficient and profitable in a real-world environment.

Even though the models may come with implications, the results of this study underline the usefulness of machine learning methods in portfolio management.

To further investigate the application of DRL methods in portfolio management, future research could incorporate a richer source of data, which could lead to more robust and complete results. A richer source of data could be an extended testing period, a broader index, or including more financial factors, e.g., financial news, as inputs to train the models. In this study, we implemented the differential Sharpe ratio as a reward function to the models, future research could implement different reward functions and evaluate whether another reward function could generate better results.

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*, *vol.87*, pp.267–279.

Bodie, Z., Kane, A., & Marcus, A. (2014). *"investments-global edition"*. McGraw Hill. (pp.205-230)

Chollet, F., et al. (2015). *"Deep Deterministic Policy Gradient (DDPG)"*. GitHub. (Available online: https://github.com/fchollet/keras [Accessed 2 January 2022])

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). "How Inefficient are Simple Asset Allocation Strategies". *Review of Financial Studies*, *vol.22*(no.5), pp.1915–1953.

Dow, C. G. (2007). "Portfolio Turnover and Common Stock Holding Periods". *Dow Publishing Company, Inc.* (pp.1–20)

Fama, E. (1970). "Efficient Capital Markets: A Review of Theory and Empirical Work". *Journal of Finance*, *vol.25*, pp.383–417.

Fischer, T. G. (2018). "Reinforcement Learning in Financial Markets-A survey". *FAU Discussion Papers in Economics*.

Gold, C. (2003). "FX Trading via Recurrent Reinforcement Learning". In *Ieee international conference on computational intelligence for financial engineering, 2003. proceedings.* (pp. 363–370).

Heaton, J. (2011). "Introduction to the Math of Neural Networks (Beta-1)". *Heaton Research Inc.*

IBM. (2020). *"What are Neural Networks?"*. (Available online: https://www.ibm.com/se-en/cloud/learn/neural-networks [Accessed 20 December 2021])

Kang, Q., Zhou, H., & Kang, Y. (2018). "an asynchronous advantage actor-critic reinforcement learning method for stock selection and portfolio management". In *Proceedings of the 2nd international conference on big data research* (pp. 141–145).

Kingma, D. P., & Ba, J. (2014). *"Adam: A method for stochastic optimization"*. ([Preprint] Available online: https://arxiv.org/abs/1412.6980 [Accessed 26 November 2021])

Leung, P.-L., Ng, H.-Y., & Wong, W.-K. (2012). "An Improved Estimation to make Markowitz's Portfolio Optimization Theory Users Friendly and Estimation Accurate with Application on the US Stock Market Investment". *European Journal of Operational Research*, *vol.222*(no.1), pp.85–95.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., … Wierstra, D. (2015). *"Continuous Control with Deep Reinforcement Learning".* ([Preprint] Available online: https://arxiv.org/abs/1509.02971 [Accessed 25 November 2021])

Littman, M. L., & Szepesvári, C. (1996). "A Generalized Reinforcement-Learning Model: Convergence and applications". In *Icml* (Vol. 96, pp. 310–318).

Lucarelli, G., & Borrotti, M. (2020). "A Deep Q-learning Portfolio Management Framework for the Cryptocurrency Market". *Neural Computing and Applications*, *vol.32*(no.23), pp.17229–17244.

Maillard, S., Roncalli, T., & Teïletche, J. (2010). "The Properties of Equally Weighted Risk Contribution Portfolios". *The Journal of Portfolio Management*, *vol.36*(no.4), pp.60–70.

Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). "Resource Management with Deep Reinforcement Learning". In *Proceedings of the 15th acm workshop on hot topics in networks* (pp. 50–56).

Maringer, D., & Ramtohul, T. (2012). "Regime-Switching Recurrent Reinforcement Learning for Investment Decision Making". *Computational Management Science*, *vol.9*(no.1), pp.89–107.

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

Merton, R. C. (1980). "On Estimating the Expected Return on the Market: An exploratory investigation". *Journal of financial economics*, *vol.8*(no.4), pp.323–361.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., … Kavukcuoglu, K. (2016). "Asynchronous Methods for Deep Reinforcement Learning". In *International conference on machine learning* (pp. 1928–1937).

Moody, J., & Saffell, M. (2001). "Learning to Trade via Direct Reinforcement". *IEEE transactions on neural Networks*, *vol.12,*(no.4), pp.875–889.

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.

Neuneier, R. (1996). "Optimal Asset Allocation using Adaptive Dynamic Programming". *Advances in Neural Information Processing Systems*, 952–958.

Noguer i Alonso, M., & Srivastava, S. (2020). *"Deep Reinforcement Learning for Asset Allocation in US Equities".* ([Preprint] Available online: https://arxiv.org/abs/2010.04404 [Accessed 20 November 2021])

Oleinik, A. (2019). "What are Neural Networks Not Good at? On artificial creativity". *Big Data & Society*, *vol.6*(no.1).

Sharpe, W. F. (1966). "Mutual Fund Performance". *The Journal of business*, *vol.39*(no.1), pp.119–138.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). "Deterministic Policy Gradient Algorithms". In *International conference on machine learning* (pp. 387–395).

Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., ... Botvinick, M. (2018). "Prefrontal Cortex as a Meta-Reinforcement Learning System". *Nature neuroscience*, *vol.21*(no.6), pp.860–868.

Xiong, Z., Liu, X.-Y., Zhong, S., Yang, H., & Walid, A. (2018). *"Practical Deep Reinforcement Learning Approach for Stock Trading".* ([Preprint] Available online: https://arxiv.org/abs/1811.07522 [Accessed 10 December 2021])

Yang, H., Liu, X.-Y., Zhong, S., & Walid, A. (2020). "Deep Reinforcement Learning for Automated Stock Trading: An ensemble strategy". *ACM International Conference on AI in Finance.*

Zhang, J., Shan, R., & Su, W. (2009). "Applying Time Series Analysis Builds Stock Price Forecast Model". *Modern Applied Science*, *vol.3*(no.5), pp.152–157.

Appendix A

A.1 Data-set

Table A.1: Complete list of stocks, data acquired from Yahoo Finance

Company	**Ticker**	**Sector**
ABB Ltd	ABB	Industrials
Assa Abloy B	ASSA B	Industrials
Astra Zeneca	AZN	Healthcare
Atlas Copco A	ATCO A	Industrials
Atlas Copco B	ATCO B	Industrials
Autoliv Inc. SDB	ALIV SDB	Consumer Cyclical
Boliden	BOL	Basic Materials
Electrolux B	ELUX B	Consumer Cyclical
Ericsson B	ERIC B	Technology
Getinge B	GETI B	Healthcare
Hennes & Mauritz B	HM B	Consumer Cyclical
Hexagon AB B	HEXA B	Technology
Investor B	INVE B	Financial Services
Kinnevik B	KINV B	Financial Services
Nordea Bank	NDA SE	Financial Services
Sandvik	SAND	Industrials
Securitas B	SECU B	Security & Protection Services
SEB A	SEB A	Financial Services
Skanska B	SKA B	Industrials
SKF B	SKF B	Industrials
SSAB A	SSAB A	Basic Materials
Svenska Cellulosa AB	SCA B	Basic Materials
Svenska Handelsbanken A	SHB A	Financial Services
Swedbank A	SWED A	Financial Services
Swedish Match	SWMA	Consumer Defensive
Tele2 B	TEL2B	Communication Services
Telia Company	TELIA	Communication Services
Volvo B	VOLV B	Industrials

A.2 Herfindahl-Hirschman Index

The concentration of the portfolio is computed using the Herfindahl-Hirschman Index. It is defined as follows. Let $(w_1, w_2, \dots, w_n)$ be a sequence of n weights, where $w_i \in [0, 1]$. The definition of the Herfindahl-Hirschman Index is:

$$H = \sum_{i=1}^{n} w_i^2, \tag{A.1}$$

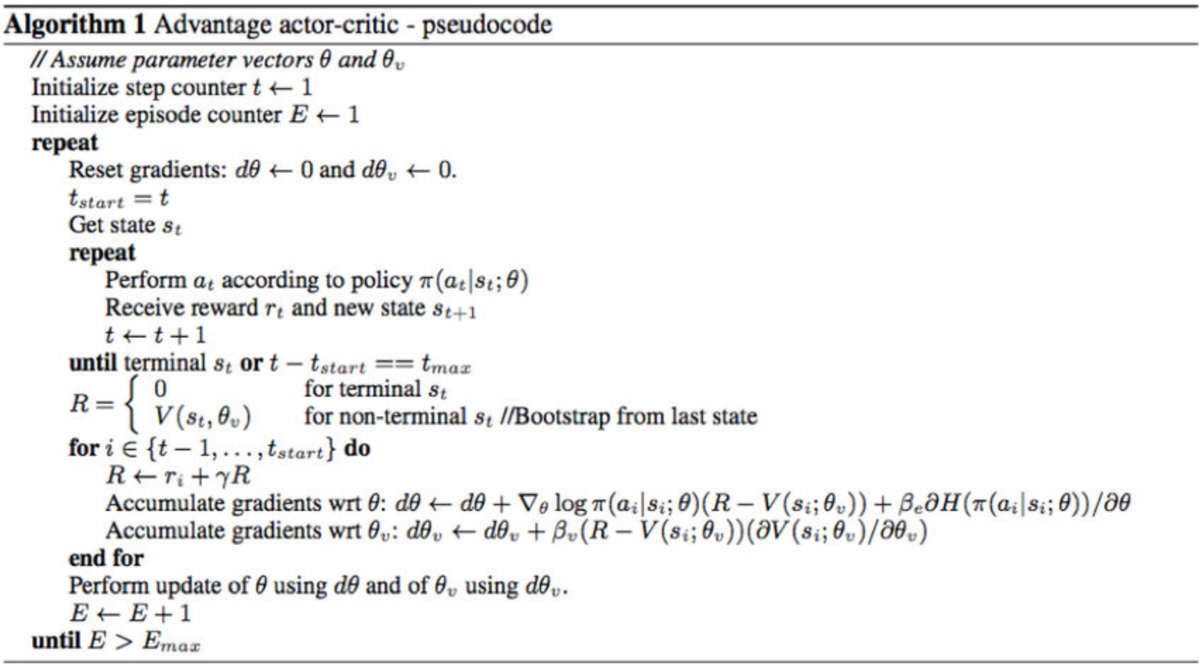
with $H \in [\frac{1}{n}, 1]$ and $\sum_{i=1}^{n} w_i = 1$.

In this study, we use a modified HHI, to scale the statistics onto $[0, 1]$:

$$H^* = \frac{H - \frac{1}{n}}{1 - \frac{1}{n}} \tag{A.2}$$

the modified HHI, takes the value 1 for a perfectly concentrated portfolio and 0 for the Naive portfolio with uniform weights.

A.3 Pseudo Code A2C

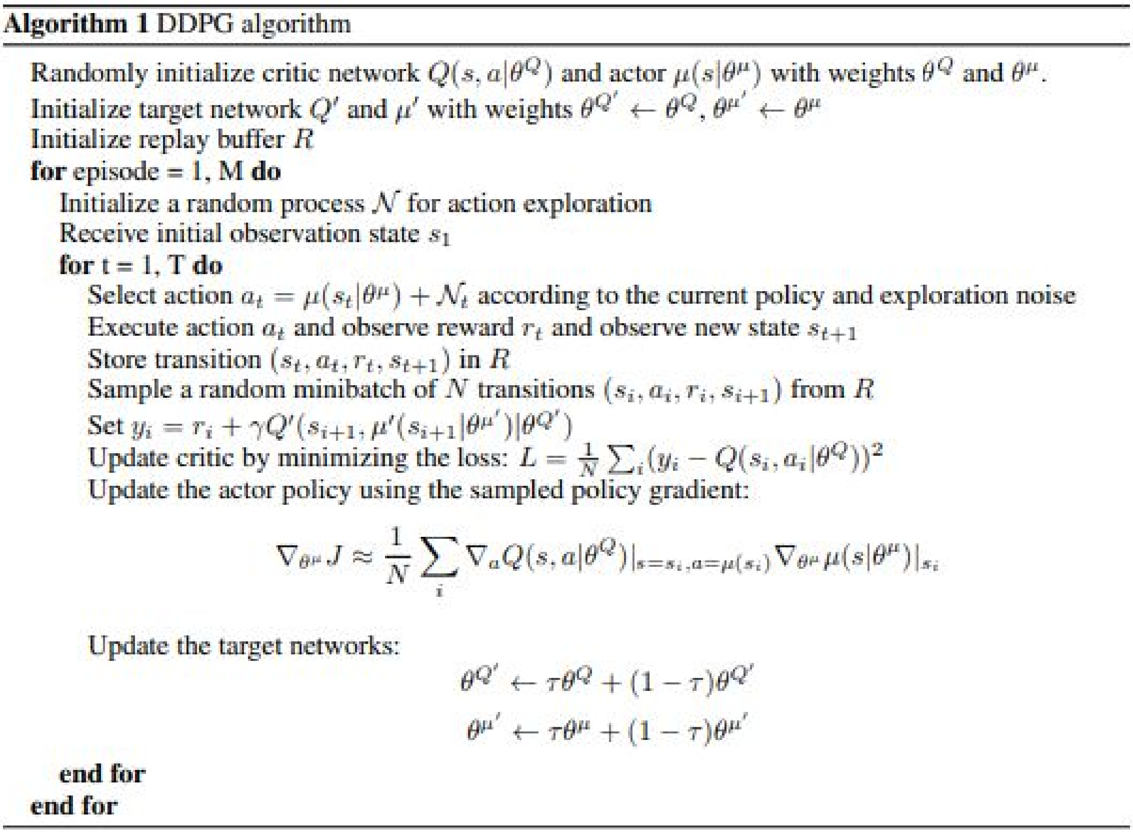


Algorithm 1 Advantage actor-critic - pseudocode

```
// Assume parameter vectors $\theta$ and $\theta_v$
Initialize step counter $t \leftarrow 1$
Initialize episode counter $E \leftarrow 1$
repeat
    Reset gradients: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
    $t_{start} = t$
    Get state $s_t$
    repeat
        Perform $a_t$ according to policy $\pi(a_t|s_t;\theta)$
        Receive reward $r_t$ and new state $s_{t+1}$
        $t \leftarrow t + 1$
    until terminal $s_t$ or $t - t_{start} == t_{max}$
    $R = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta_v) & \text{for non-terminal } s_t \text{ //Bootstrap from last state} \end{cases}$
    for $i \in \{t-1, \ldots, t_{start}\}$ do
        $R \leftarrow r_i + \gamma R$
        Accumulate gradients wrt $\theta$: $d\theta \leftarrow d\theta + \nabla_\theta \log \pi(a_i|s_i;\theta)(R - V(s_i;\theta_v)) + \beta_e \partial H(\pi(a_i|s_i;\theta))/\partial\theta$
        Accumulate gradients wrt $\theta_v$: $d\theta_v \leftarrow d\theta_v + \beta_v(R - V(s_i;\theta_v))(\partial V(s_i;\theta_v)/\partial\theta_v)$
    end for
    Perform update of $\theta$ using $d\theta$ and of $\theta_v$ using $d\theta_v$.
    $E \leftarrow E + 1$
until $E > E_{max}$
```

Figure A.1: Pseudo code for the A2C algorithm, Wang et al. (2018)

A.4 Pseudo Code DDPG



```
Algorithm 1 DDPG algorithm
  Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights $\theta^Q$ and $\theta^\mu$.
  Initialize target network $Q'$ and $\mu'$ with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
  Initialize replay buffer $R$
  for episode = 1, M do
    Initialize a random process $\mathcal{N}$ for action exploration
    Receive initial observation state $s_1$
    for t = 1, T do
      Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
      Execute action $a_t$ and observe reward $r_t$ and observe new state $s_{t+1}$
      Store transition $(s_t, a_t, r_t, s_{t+1})$ in $R$
      Sample a random minibatch of $N$ transitions $(s_i, a_i, r_i, s_{i+1})$ from $R$
      Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
      Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
      Update the actor policy using the sampled policy gradient:

        $$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

      Update the target networks:
        $$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
        $$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

    end for
  end for
```

Figure A.2: Pseudo code for the DDPG algorithm, Chollet et al. (2015)